IBM

IBM TotalStorage Enterprise Storage Server Model 800
Performance Monitoring and Tuning Guide

- Efficiently using ESS Model 800 capabilities
- Optimizing performance in the ESS
- Monitoring I/O processing in the ESS

Gustavo Castets
Sofia Cristina Baquero
Paul (Pablo) Clifton
Donald (Chuck) Laing
Jukka Myyryläinen

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM TotalStorage Enterprise Storage Server Model 800 Performance Monitoring and Tuning Guide

July 2003

SG24-6422-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xix.

First Edition (July 2003)

This edition applies to the IBM TotalStorage Enterprise Storage Server Model 800 (ESS Model 800 — 2105-800).

Contents

Figures

Tables

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
AIX 5L™
AS/400®
AS/400e™
CICS®
CUA®
DB2®
DB2 Universal Database™
DFSMS/MVS®
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSORT™
DYNIX®
ECKD™
Enterprise Storage Server™
Enterprise Systems Connection® Architecture®
ESCON®
eServer™
FICON™
FlashCopy™
GDPS®
IBM®
ibm.com®
IBM eServer™
IMS™
iSeries™
Lotus®
Lotus Notes®
MVS™
Notes®
NUMA-Q®
OS/2®
OS/390®
OS/400®
PR/SM™
Parallel Sysplex®
pSeries™
Redbooks™
Redbooks Logo™ 
RMF™
RS/6000®
S/390®
Seascape®
SP™
Tivoli®
TotalStorage™
VSE/ESA™
xSeries™
z/Architecture™
z/OS™
z/VM™
zSeries™

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook provides guidance on the best way to configure, monitor, and manage your IBM TotalStorage Enterprise Storage Server (ESS) to achieve optimum performance. The information presented in this publication applies mainly to the ESS Model 800, but many of the discussions and recommendations can also be considered with previous F models.

We describe the ESS Model 800 performance features and characteristics and how they can be exploited with the different server platforms that can attach to the ESS. Then in consecutive chapters we go over the specific performance recommendations and discussions that apply for each server environment, as well as for database and ESS Copy Services environments.

We also outline the various tools available for monitoring and measuring I/O performance for the different server environments, as well as how to monitor performance of the entire ESS subsystem.

The team that wrote this redbook

This publication was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Gustavo Castets is a Project Leader at the International Technical Support Organization, San Jose Center. He has co-authored six previous IBM Redbooks™ and teaches IBM classes worldwide on areas of Disk Storage Systems. Before joining the ITSO, Gustavo worked as Field Technical Sales Specialist in Buenos Aires. Gustavo has worked for more than 22 years in many IT areas, for IBM Argentina.

Sofia Cristina Baquero is a Storage Advanced Technical Support Specialist in Colombia. She holds a degree in Computer Science from Universidad de los Andes in Colombia. She has worked more than 20 years for IBM Colombia as a performance and capacity planning Systems Programmer for VM systems, IBM @server zSeries™ and Storage Sales Specialist, and Storage IT Specialist. Her areas of expertise include the IBM TotalStorage Enterprise Storage Server when attached to the various server platforms, IBM 3494 Virtual Tape Server (VTS), IBM 3494 Tape Library, and IBM 3590 tapes, plus midrange storage solutions.

Paul (Pablo) Clifton is a Sr. UNIX Admin/Storage Specialist for Technology Service Partners Inc. (tspi.com) in the USA. He has 14 years of experience, mostly with AIX®, and has achieved the certifications: IBM Certified Advanced Technical Expert - RS/6000® AIX, McData Storage Network Implementer, and Brocade Certified Fabric Professional. He holds a degree in Electrical Engineering from the University of Texas at Austin. His interests include maximizing IBM TotalStorage Enterprise Storage Server sequential performance, improving backup speeds, and teaching his daughter how to play soccer.

Donald (Chuck) Laing is a Sr. Systems Management Integration Professional, specializing in open systems UNIX disk administration in the IBM South Delivery Center (SDC). He has co-authored one previous IBM Redbook on the IBM TotalStorage Enterprise Storage Server. He holds a degree in Computer Science. Chuck's responsibilities include planning and implementation of midrange storage products. His responsibilities also include department wide education and cross training on various storage products such as the ESS and FAStT. He has worked at IBM for five years. Before joining IBM, Chuck was a hardware CE on UNIX

systems for 10 years and taught basic UNIX at Midland College for six years in Midland Texas.

Jukka Myyryläinen is an Advisory IT Specialist with IBM Global Services, Finland. He has 17 years of experience in storage product implementations. He has contributed to several storage related redbooks in the past.

The authors of previous ESS performance and monitoring books are:

Alison Pate	Matthias Effmert	John Newman
Manabu Okano	Jacqueline Tout	Paulus Usong

Thanks to the following people from Valero Energy Corp. for their contributions to this project:

James Brents	Greg Jordan	Gary Mays

Thanks to the following people from IBM for their contributions to this project:

Stephen Atkins	Joseph F Bacco	Dan Braden
Helen Burton	Gene Cullum	James R. Chandler
Mark Chitti	Niggel Griffiths	Kenneth Hallam
Frank Krueger	Lee La Frese	Phil Lee
Ian Mac Quarrie	Bruce McNutt	John Ponder
Russ Porter	Scott Susi	Constance Rea
Nancy Roper	Robert Vaupel	Glenn P. Williamson
Rainer Wolafka	Nathan Zingg	Julie Czubik

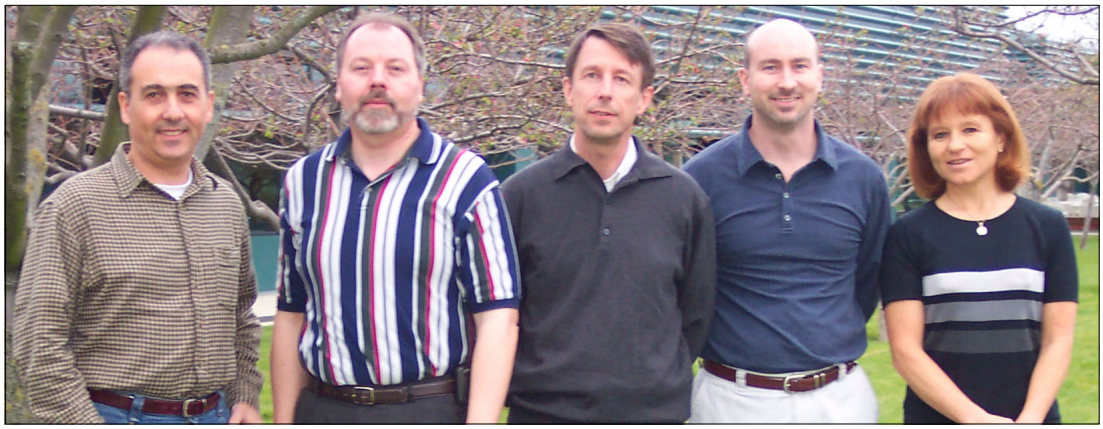


Figure 1 The team: Gustavo, Chuck, Jukka, Pablo, Sofia

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

1

ESS Model 800 characteristics

In this chapter we give an overview of the IBM TotalStorage Enterprise Storage Server Model 800. This is an introduction to the design and performance characteristics of the ESS Model 800 that are discussed in detail in succeeding chapters of this publication.

1.1 The IBM TotalStorage Enterprise Storage Server

The IBM TotalStorage™ Enterprise Storage Server™ (ESS) is IBM's most powerful disk storage server, developed using IBM Seascape® architecture. The ESS provides un-matchable functions for all the @server family of e-business servers, and also for the non-IBM (that is, Intel-based and UNIX-based) families of servers. Across all of these environments, the ESS features unique capabilities that allow it to meet the most demanding requirements of performance, capacity, and data availability that the computing business may require.

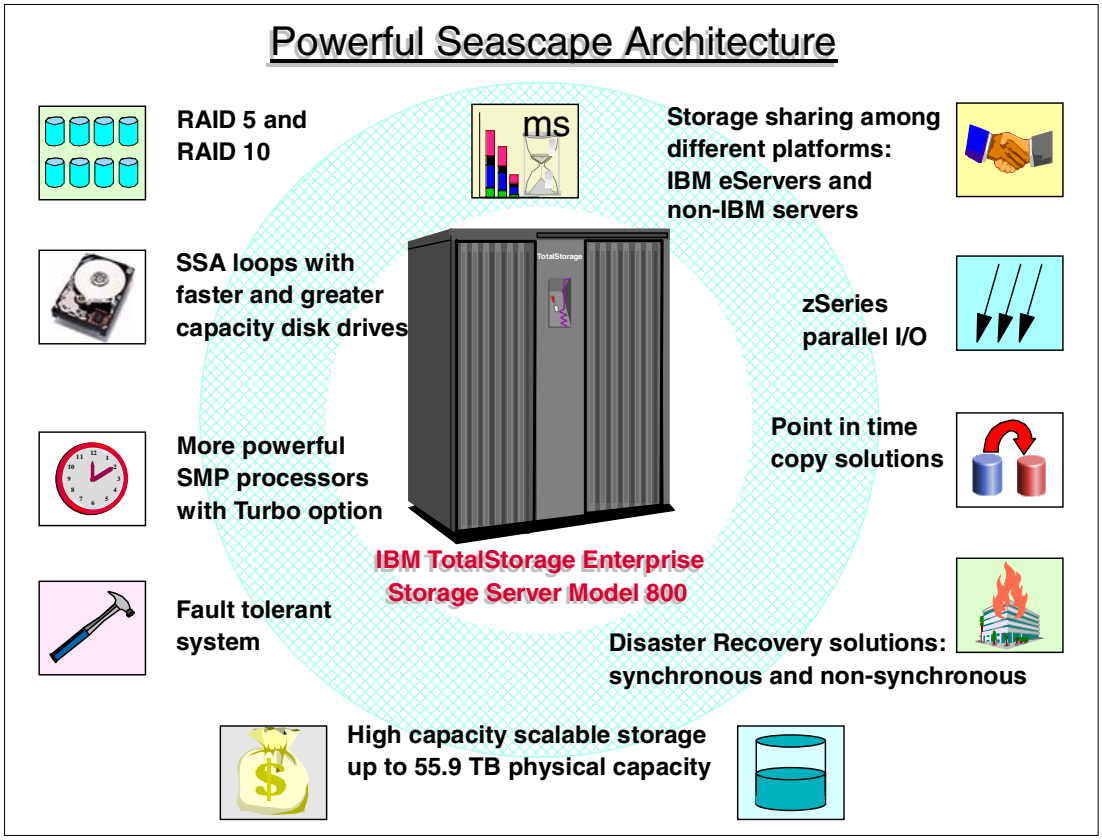


Figure 1-1 IBM's Seascape architecture - ESS Model 800

The Seascape architecture is the key to the development of IBM's storage products. Seascape allows IBM to take the best of the technologies developed by the many IBM laboratories and integrate them, producing flexible and upgradeable storage solutions. This Seascape architecture design has allowed the IBM TotalStorage Enterprise Storage Server to evolve from the initial E models to the succeeding F models, and to the most recent 800 models, each featuring new, more powerful hardware and functional enhancements, and always integrated under the same successful architecture with which the ESS was originally conceived.

The move to e-business presents companies with both extraordinary opportunities and significant challenges. Consequently, companies also face an increase of critical requirements for more information that is universally available online, around the clock, every day of the year.

To meet the unique requirements of e-business, where massive swings in the demands placed on your systems are common and continuous operation is imperative, you will need very high-performance and intelligent storage technologies, and systems that can support

any server application in your business, today and into the future. The IBM TotalStorage Enterprise Storage Server has set new standards in function, performance, and scalability in these most challenging environments.

1.2 IBM TotalStorage Enterprise Storage Server Model 800

Since its initial availability with the ESS Models E10 and E20, and then with the succeeding F10 and F20 models, the ESS has been the storage server solution offering exceptional performance, extraordinary capacity, scalability, heterogeneous server connectivity, and an extensive suite of advanced functions to support users' mission-critical, high-availability, multi-platform environments. The ESS set a new standard for storage servers back in 1999 when it was first available, and since then it has evolved into the F models and the more recently announced third-generation ESS Model 800, keeping up with the pace of users' needs by adding more sophisticated functions to the initial set, enhancing the connectivity options, and powering its performance features.

The IBM TotalStorage Enterprise Storage Server Model 800 provides significantly improved levels of performance, throughput, and scalability while continuing to exploit the innovative features introduced with its preceding E and F models such as Parallel Access Volumes, Multiple Allegiance, I/O Priority Queuing, the remote copy functions (synchronous, non-synchronous, and asynchronous), and the FlashCopy™ point-in-time copy function. Also, the heterogeneous server support characteristics—for connectivity and remote copy functions—of previous models are continued with the ESS Model 800.

With the IBM TotalStorage Enterprise Storage Server Model 800 important changes have been introduced, that dramatically improve the overall value of the ESS in the marketplace and provide a strong base for strategic Storage Area Network (SAN) initiatives.

Among the enhancements found in the Model 800 are:

- 2 Gb Fibre Channel/FICON™ host adapter
- 64 bit ESCON® host adapter
- Up to 64 GB of cache
- New, more powerful SMP cluster processors with a Turbo feature option
- 2 GB non-volatile storage (NVS) with double the bandwidth
- A doubling of the bandwidth of the Common Platform Interconnect (CPI)
- RAID-10 array configuration capability

The hardware enhancements that the IBM TotalStorage Enterprise Storage Server Model 800 provides all combined to provide a balanced two-fold performance boost as compared to the predecessor F models—and up to two-and-a-half times boost with the Turbo processor option.

1.2.1 Benefits

The new IBM TotalStorage Enterprise Storage Server Model 800 can help you achieve your business objectives in many areas. It provides a high-performance, high-availability storage subsystem with flexible characteristics that can be configured according to your requirements.

1.3 Storage consolidation

The ESS attachment versatility—and large capacity—enable the data from different platforms to be consolidated onto a single high-performance, high-availability box. Storage

consolidation can be the first step towards server consolidation, reducing the number of boxes you have to manage and allowing you to flexibly add or assign capacity when it is needed. The IBM TotalStorage Enterprise Storage Server supports all the major operating systems platforms, from the complete set of IBM @server series of e-business servers and IBM NUMA-Q®, to the non-IBM Intel-based servers and the different variations of UNIX-based servers, as shown in Figure 1-2.

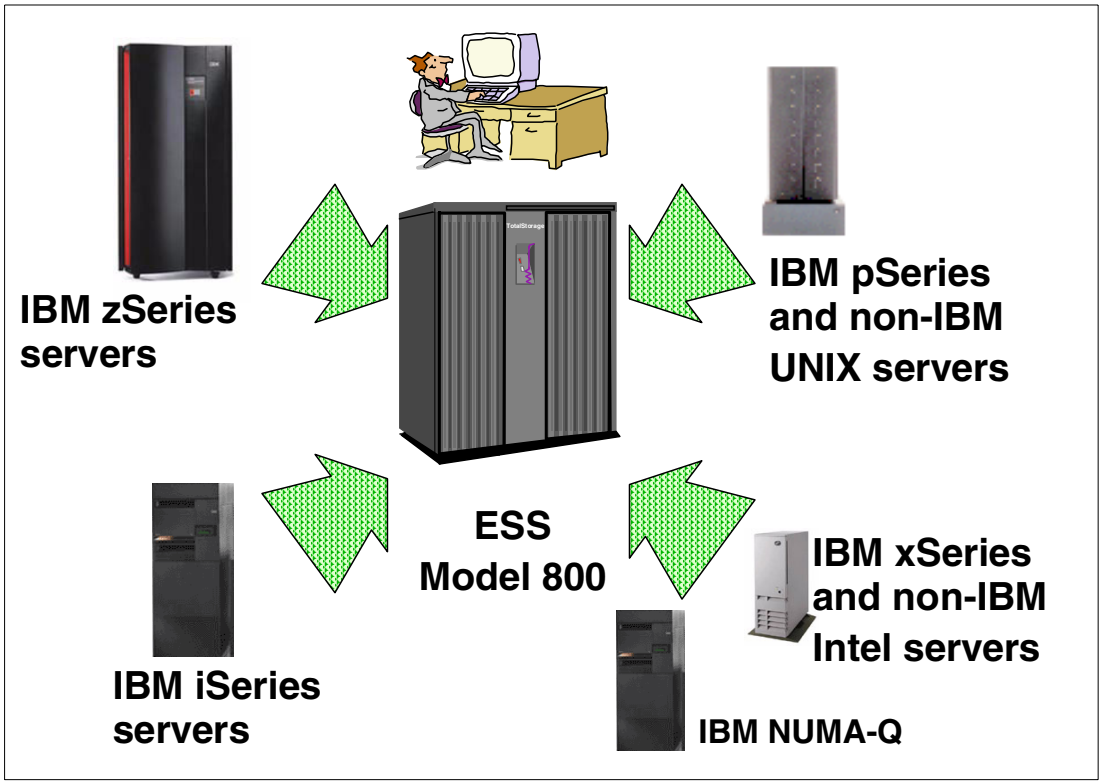


Figure 1-2 IBM TotalStorage Enterprise Storage Server for storage consolidation

With a total capacity of 55.9 TB, and a diversified host attachment capability—SCSI, ESCON, and Fibre Channel/FICON—the IBM TotalStorage Enterprise Storage Server Model 800 provides outstanding performance while consolidating the storage demands of the heterogeneous set of server platforms that must be dealt with nowadays.

1.4 Performance

The IBM TotalStorage Enterprise Storage Server is a storage solution with a design for high performance that takes advantage of IBM's leading technologies.

In today's world, you need business solutions that can deliver high levels of performance continuously every day, day after day. You also need a solution that can handle different workloads simultaneously, so you can run your business intelligence models, your large databases for enterprise resource planning (ERP), and your online and Internet transactions alongside each other. Some of the unique features that contribute to the overall high-performance design of the IBM TotalStorage Enterprise Storage Server Model 800 are shown in Figure 1-3 on page 5.

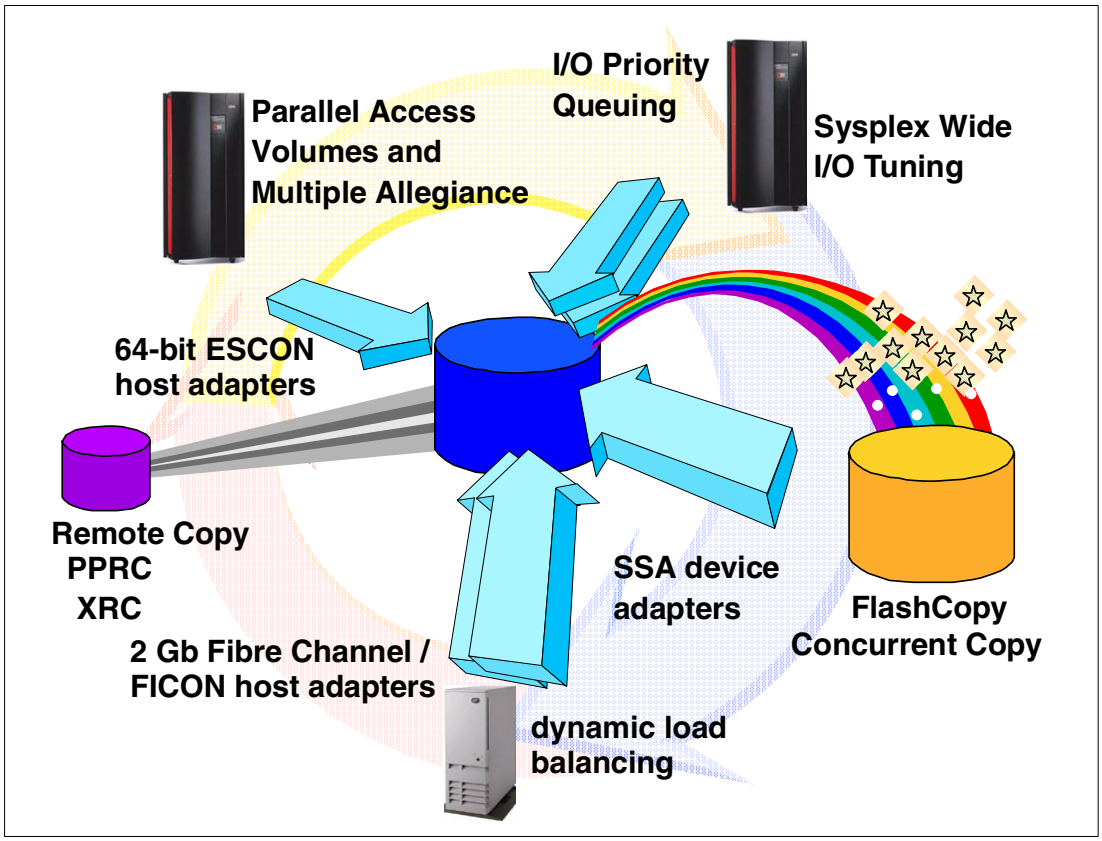


Figure 1-3 IBM TotalStorage Enterprise Storage Server capabilities

1.4.1 Third-generation hardware - ESS Model 800

The IBM TotalStorage Enterprise Storage Server Model 800 integrates a new generation of hardware from top to bottom, allowing it to deliver unprecedented levels of performance and throughput. Key features that characterize the performance enhancements of the ESS Model 800 are:

- The ESS Model 800 generally is capable of delivering twice the throughput of its predecessor Model F20.
- With the optional Turbo feature, it is capable of providing 2.5 times the throughput of its predecessor Model F20, for increased scalability and response times.
- 64 GB cache supports much larger system configurations and increases cache hit ratios, driving down response times.
- Double the internal bandwidth provides high sequential throughput for digital media, business intelligence, data warehousing, and life science application.
- Larger 2 GB NVS with twice the bandwidth allows greater scalability for write-intensive applications.
- Third-generation hardware provides response time improvements of up to 40 percent for important database applications.
- 2 Gb Fibre Channel/FICON host adapters provide doubled performance sustained and instantaneous throughput for both open systems and zSeries environments.
- 64-bit ESCON host adapters, enhanced with a faster microprocessor providing increased channel throughput and sequential read bandwidth.

- RAID-10 can provide up to 75 percent greater throughput for selected database workloads compared to equal physical capacity configured as RAID-5. While most typical workloads will experience excellent response times with RAID-5, some cache-unfriendly applications and some applications with high random write content can benefit from the performance offered by RAID-10.
- 15,000 rpm drives provide up to 80 percent greater throughput per RAID rank and 40 percent improved response time as compared to 10,000 rpm drives. This allows driving the workloads to significantly higher access densities, while also experiencing improved response times.

All this performance boost is built upon the reliable and proven ESS hardware architecture design and unique advanced features.

1.4.2 Efficient cache management and powerful back-end

The ESS is designed to provide the highest performance for the different types of workloads, even when mixing dissimilar workload demands. For example, zSeries servers and open systems put very different workload demands on the storage subsystem. A server like the zSeries typically has an I/O profile that is very cache-friendly, and takes advantage of the cache efficiency. On the other hand, an open system server does an I/O that can be very cache-unfriendly, because most of the hits are solved in the host server buffers. For the zSeries type of workload, the ESS has the option of a large cache (up to 64 GB) and—most importantly—it has efficient cache algorithms. For the cache unfriendly workloads, the ESS has a powerful back-end, with the SSA high-performance disk adapters providing high I/O parallelism and throughput for the ever-evolving high-performance hard disk drives.

1.4.3 Sysplex I/O management

In the zSeries Parallel Sysplex® environments, the z/OS™ Workload Manager (WLM) controls where work is run and optimizes the throughput and performance of the total system. The ESS provides the WLM with more sophisticated ways to control the I/O across the sysplex. These functions, described in detail later in this publication, include parallel access to both single-system and shared volumes and the ability to prioritize the I/O based upon WLM goals. The combination of these features significantly improves performance in a wide variety of workload environments.

1.4.4 Parallel Access Volume (PAV) and Multiple Allegiance

Parallel Access Volume and Multiple Allegiance are two distinctive performance features of the IBM TotalStorage Enterprise Storage Server for the zSeries users, allowing them to reduce device queue delays, which means improving throughput and response time.

1.4.5 I/O load balancing

For selected open system servers, the ESS in conjunction with the Subsystem Device Driver (SDD), a pseudo device driver designed to support multi-path configurations, provides dynamic load balancing. Dynamic load balancing helps eliminate data-flow bottlenecks by distributing the I/O workload over multiple active paths, thus contributing to improve I/O throughput and response time of the open system server.

1.4.6 2 Gb Fibre Channel/FICON host adapters

As the amount of data and transactions grow, so does the traffic over the Storage Area Networks (SAN). As SANs migrate to 2 Gb technologies to cope with this increased amount

of data transit, so does the IBM TotalStorage Enterprise Storage Server Model 800 with its 2 Gb host adapters. These host adapters double the bandwidth of the previous adapters, thus providing more throughput and performance for retrieving and storing users' data.

1.4.7 64-bit ESCON host adapters

The ESCON adapters have been enhanced with a faster microprocessor that offers up to a 45 percent improvement in full box sequential read bandwidth and up to 10 percent increase in channel throughput for random operation workloads as compared to the previous 32-bit ESCON adapters. When used in a PPRC configuration on both the primary and secondary ESS, up to a 10 percent increase in PPRC link throughput for random write operations and sequential bandwidth may be achieved as compared to the previous 32-bit ESCON host adapters.

1.5 Data protection and availability

Many design characteristics and advanced functions of the IBM TotalStorage Enterprise Storage Server Model 800 contribute to protect the data in an effective manner.

1.5.1 Fault-tolerant design

The IBM TotalStorage Enterprise Storage Server is designed with no single point of failure. It is a fault-tolerant storage subsystem, which can be maintained and upgraded concurrently with user operation. Some of the functions that contribute to these attributes of the ESS are shown in Figure 1-4 on page 8.

1.5.2 RAID-5 or RAID-10 data protection

With the IBM TotalStorage Enterprise Storage Server Model 800 the disk arrays can be configured in a RAID-10 (mirroring plus striping) or a RAID-5 (striping with distributed parity) arrangement, thus giving more options when selecting the redundancy technique for protecting data.

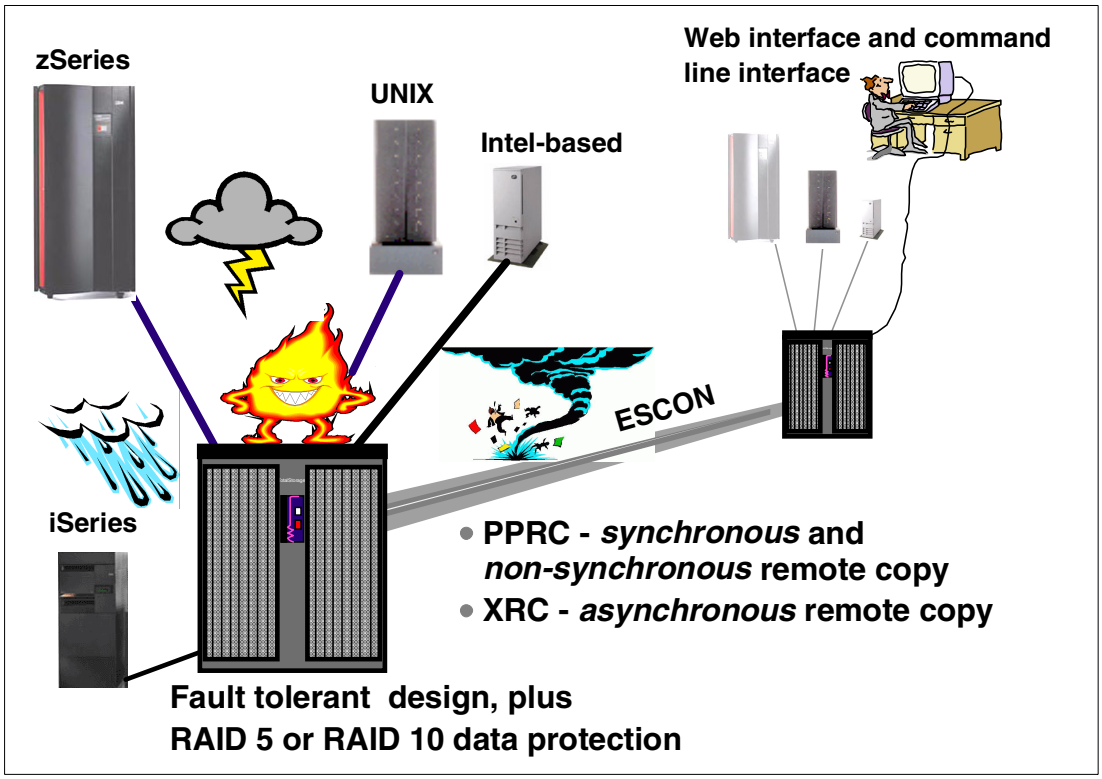


Figure 1-4 Data integrity and availability

1.5.3 Remote copy functions

The IBM TotalStorage Enterprise Storage Server Model 800 provides a set of remote copy functions (illustrated in Figure 1-3 on page 5 and Figure 1-4) that allow you to more flexibly plan your business continuance solution.

Peer-to-Peer Remote Copy (PPRC)

The Peer-to-Peer Remote Copy (PPRC) function is a hardware-based solution for mirroring logical volumes from a primary site (the application site) onto the volumes of a secondary site (the recovery site). PPRC is a remote copy solution for the open systems servers and for the zSeries servers.

Two modes of PPRC are available with the IBM TotalStorage Enterprise Storage Server Model 800:

- PPRC synchronous mode, for real-time mirroring between ESSs located up to 103 km apart
- PPRC Extended Distance (PPRC-XD) mode, for non-synchronous data copy over continental distances

For all ESS-supported server platforms, PPRC can be managed using a Web browser to interface with the ESS Copy Services Web user interface (WUI). PPRC can also be operated using commands for selected open systems servers that are supported by the ESS Copy Services command-line interface (CLI). For the z/OS and OS/390® environments, the TSO commands can be used to manage PPRC, as well as the ICKDSF utility—that can also be used in the rest of the zSeries environments including the stand-alone.

PPRC channel extension, DWDM, and connectivity options

PPRC flexibility is further enhanced with support for additional network connectivity options when using channel extenders. PPRC is supported over all the network technologies that are currently supported by the CNT UltraNet Storage Director or the INRANGE 9801 Storage Networking System, including Fibre Channel, Ethernet/IP, ATM-OC3, and T1/T3.

Also the Cisco ONS 15540 Dense Wave Division Multiplexer (DWDM) and the Nortel Networks OPTera Metro 5300 DWDM are supported for PPRC connectivity.

Please visit http://www.storage.ibm.com/hardsoft/products/ess/pdf/1012-01.pdf for the most current list of supported connectivity options.

These new options support the exploitation of existing or new communication infrastructures and technologies within metropolitan networks and the WAN to help optimize cost, performance, and bandwidth.

Extended Remote Copy (XRC)

Extended Remote Copy (XRC) is a combined hardware and software remote copy solution for the z/OS and OS/390 environments. The asynchronous characteristics of XRC make it suitable for continental distance implementations.

1.5.4 Point-in-Time Copy function

Users still need to take backups to protect data from logical errors and disasters. For all environments, taking backups of user data traditionally took a considerable amount of time. Usually backups are taken outside prime shift because of their duration and the consequent impact to normal operations. Databases must be closed to create consistency and data integrity, and online systems are normally shut down.

With the IBM TotalStorage Enterprise Storage Server, the backup time has been reduced to a minimal amount of time when using the FlashCopy function. FlashCopy creates an instant point-in-time copy of data, and makes it possible to access both the source and target copies immediately, thus allowing the applications to resume with minimal disruption.

For all ESS-supported server platforms, FlashCopy can be controlled using a Web browser by means of the ESS Copy Services Web user interface of the IBM TotalStorage Enterprise Storage Server. Under z/OS, FlashCopy can also be invoked using DFSMSdss™ utility, and TSO commands.

For selected open systems servers the IBM TotalStorage Enterprise Storage Server also provides the ESS Copy Services command-line interface (CLI) for invocation and management of FlashCopy functions through batch processes and scripts.

1.6 Storage Area Network (SAN)

The third-generation IBM TotalStorage Enterprise Storage Server Model 800 continues to deliver on its SAN strategy, initiated with its predecessor E and F models. As SANs migrate to 2 Gb technology, then storage subsystems must exploit this more powerful bandwidth. Keeping pace with the evolution of SAN technology, the IBM TotalStorage Enterprise Storage Server Model 800 introduced new 2 Gb Fibre Channel/FICON host adapters for native server connectivity and SAN integration.

The 2 Gb Fibre Channel/FICON host adapters, which double the bandwidth and instantaneous data rate of the previous adapters available with the F Model, have one port

with an LC connector for full-duplex data transfer over long-wave or short-wave fiber links. These adapters support the SCSI-FCP (Fibre Channel Protocol) and the FICON upper-level protocols.

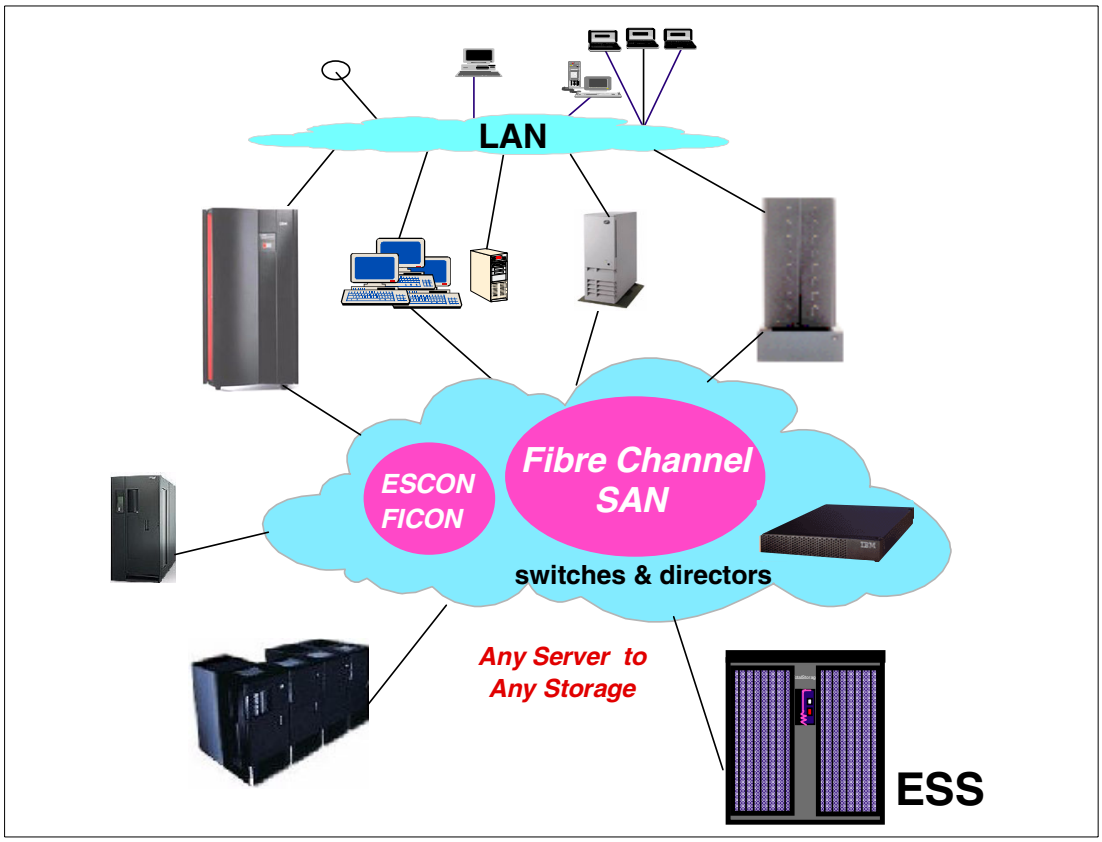


Figure 1-5 Storage Area Network (SAN)

Fabric support currently includes the following equipment:

- IBM TotalStorage SAN switches IBM 3534 and IBM 2109
- McDATA Intrepid Enterprise Fibre Channel Directors—IBM 2032—for 2 Gb FICON and FCP attachment
- McDATA Spheron switches for FCP attachment—IBM 2031
- INRANGE FC/9000 Director for FCP attachment and FICON attachment—IBM 2042
- Cisco MDS 9216 and MDS 9509—IBM 2062—switches for 2 Gbps FCP attachment

The ESS supports the Fibre Channel/FICON intermix on the INRANGE FC/9000 Fibre Channel Director and the McDATA Intrepid Enterprise Fibre Channel Directors. With Fibre Channel/FICON intermix, both FCP and FICON upper-level protocols can be supported within the same director on a port-by-port basis. This new operational flexibility can help users to reduce costs with simplified asset management and improved asset utilization.

The extensive connectivity capabilities make the ESS the unquestionable choice when planning the SAN solution. For the complete list of the ESS fabric support, please refer to:

`http://www.storage.ibm.com/disk/ess/supserver.htm`

For a description of the IBM TotalStorage SAN products, please refer to:

`http://www.storage.ibm.com/ibmsan/products/sanfabric.html`

2

Hardware configuration planning

This chapter reviews the performance features and characteristics of the ESS Model 800 that must be considered when planning your ESS hardware configuration.

There are many choices when planning your ESS Model 800 hardware configuration, and their relevance is related to the workload characteristics and performance expectations. This chapter discusses:

- The hardware options available on the ESS Model 800
- How these options improve response time and throughput
- Recommendations on when and how to use these options to enhance I/O performance

2.1 Rules of thumb and benchmarks

In the past, IT technical personnel have used *rules of thumbs* (ROTs), as a fast and easy way to plan the disk subsystem configuration. Some ROTs based on capacity—like example, you should configure one 1 Gb Fibre Channel host adapter per 400 GB of capacity—were easy to use because a good understanding of the user workload was not required. Because *rules of thumb* were used to cover most of the users' workloads, disregarding their I/O and unique processing requirements, they tended to be too conservative. With the numerous options available in the ESS Model 800, with the more stringent and unique application requirements, and the lower cost objectives that organizations face nowadays, ROTs inherent lack of accuracy makes them an impractical approach. Benchmarks also look like an easy way for planning a disk configuration. But if you intend to make a configuration decision based on benchmark performance results, you need to ensure that the *workload* that is used for the benchmark resembles, as closely as possible, the workload that you intend to run on your ESS. You should also know both the *physical* and the *logical configuration* of the ESSs used during the benchmark, so your workload gets the results you are expecting. Sometimes you will not be able to replicate the configuration documented for the benchmark, and some other times you may not find documented a benchmark that resembles your requirements. So benchmarks cannot always be used.

For these reasons, the recommended approach for correctly estimating the disk subsystem configuration that is needed is to use the set of tools and techniques that we discuss in this publication. These tools allow easy and accurate monitoring, analyzing, sizing, and modelling for the required ESS Model 800 configuration. Among these tools, we have:

- Disk Magic
- Sequential Sizer for S390
- Capacity Magic
- ESSUTIL for UNIX
- ESS Expert

These tools will be discussed in detail in their respective chapters later in this publication.

2.2 Understanding your workload characteristics

To properly configure your ESS, it is important to have an understanding of the I/O workload you intend to run. As you read the following sections when discussing the ESS hardware components, you will be able to realize how the *I/O workload characteristics* significantly affect what will be the optimum ESS hardware configuration.

The *I/O workload characteristics* that we are speaking about are:

- Cache friendliness: Friendly, unfriendly, standard. This becomes visible in the read hit ratios and write hit ratios of the I/O workload.
- Read/write ratio: This is the number of reads per writes.
- Block size of the I/O operations.
- Random versus sequential I/O processing.
- The I/O rate: I/O operations per second, and the associated *access density*. This is the number of I/O operations per second per GB of data (IO/sec/GB).

Another important characteristic of an I/O workload is whether the data is being *remote copied* or not.

If you are moving existing workloads to a new ESS, then you have information that can be used to model and estimate this new ESS configuration. You will also be able to model any activity growth that you are planning in advance.

On the other hand, if the workload that you plan to run on the ESS is a new workload or one that you do not have a good understanding of, then we recommend that you be conservative when planning the disk storage subsystem hardware configuration.

If you will be running multiple heterogeneous servers, each server with different workload characteristics, you will have the most complex case. You must ensure that your final hardware configuration has enough capacity to cope with the maximum data rate, while *aggregating* the whole set of applications.

When the workload demands of all the servers being consolidated are well understood, and assuming they are predictable and consistent, it could be possible to manage the peaks and thus get some resource savings.

On the other hand, if you are combining workloads that are not well understood or whose requirements fluctuate in an unpredictable manner, then a more conservative approach must be taken when considering the peaks in your hardware capacity planning.

2.3 ESS Model 800 major hardware components

We will discuss now the features of the ESS Model 800 and their effects on the resulting performance of the I/O processing. When reading these sections, you must remember that any of these performance characteristics will also be determined by the type of workload that is run.

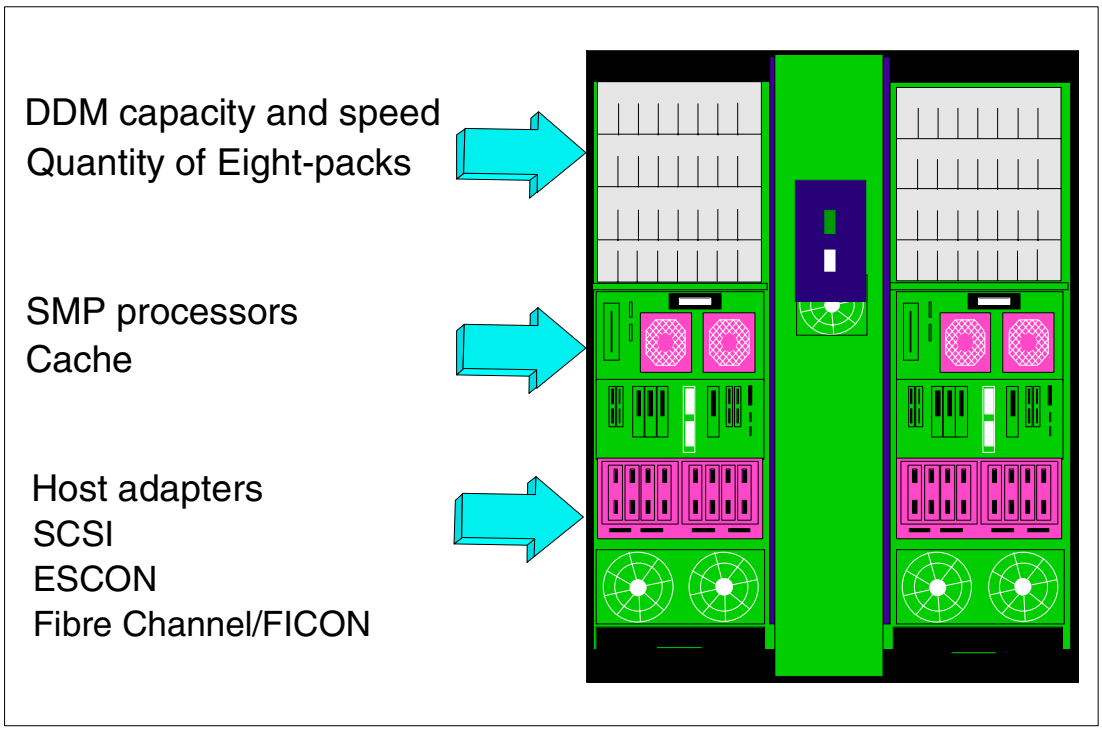


Figure 2-1 Planning the ESS hardware configuration

From the performance perspective, the major components you need to consider when planning the ESS Model 800 hardware configuration are (refer to Figure 2-1 on page 13):

- Processor, Standard, or Turbo
- Cache
- DDM capacity and speed (RPM)
- Number of arrays and RAID type
- Number and type of host adapters

The hardware components presented in Figure 2-1 on page 13 and their implications in the overall ESS performance behavior is discussed in the following sections.

2.4 ESS Processors

In this section we describe the processor options available with the ESS Model 800 and their implications on the ESS performance. We also present guidelines that can help you when choosing among these options.

2.4.1 Standard and Turbo options

The IBM TotalStorage Enterprise Storage Server Model 800 uses high-performance IBM *reduced instruction-set computer* (RISC) processors to manage its operations. Each of the two active clusters within the ESS contains one of these fast *symmetrical multiprocessors* (SMP). There are two processor options available:

- Standard processor feature
- Turbo processor feature

The ESS processors utilize innovative copper and *silicon-on-insulator* (SOI) technology, with built-in fault and error-correction functions to maximize system availability.

2.4.2 Choosing the processor option

Compared to the previous F models, the ESS 800 with the Standard processor provides up to two times the throughput and performance. And with the optional Turbo processor feature the ESS Model 800 provides up to two and a half times the throughput and performance of the previous generation F models, for heavy-duty workload environments.

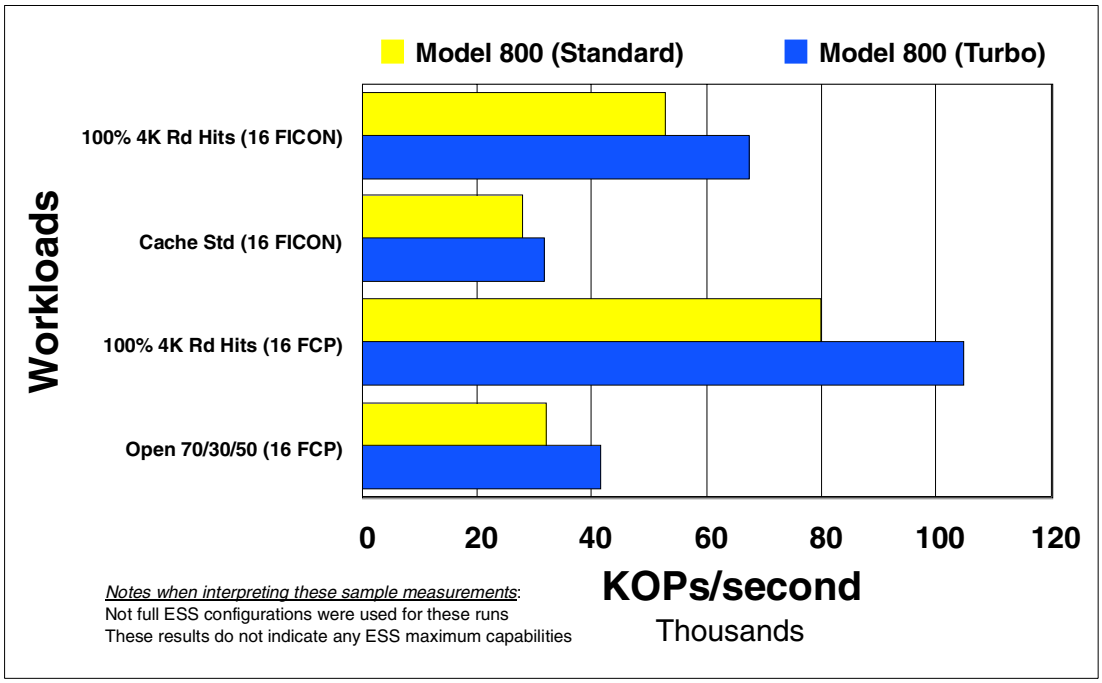


Figure 2-2 Standard vs. Turbo processor options - KOPs/sec with different workloads

Figure 2-2 illustrates examples of throughput results that compare the Standard vs. the Turbo processor options, when different workloads are executed in a certain ESS Model 800 configuration. You can see that benefits of the Turbo processor option vary when different workloads are run.

In Figure 2-2 Cache Standard is a z/OS workload where read/write ratio = 3, read hit ratio = 0.735, destage rate = 11.6 percent, and transfer size = 4 K. For a discussion of the I/O characteristics of the different types of workload, please refer to Chapter 11, “Understanding your workload” on page 367.

The sample results shown in Figure 2-2 should not be interpreted as indicative of any ESS maximum capabilities, as the runs were done with restricted ESS configurations with the sole purpose of comparing the processors’ capabilities under similar workloads and configurations.

The Turbo processor option can be beneficial in situations where there is:

- Very high random throughput requirements with the following characteristics:
  - Online Transaction Processing (OLTP), Transaction Processing Facility (TPF), or some database workloads.
  - Cache friendly or cache unfriendly environments.
  - Workloads greater than 30000 IO/sec.
  - Larger capacities—over 25 TB, because high throughput demands usually follow from larger capacities. For larger capacities, depending on the *access density*, the Turbo processor feature will be beneficial. *Access density* is defined as the number of I/O requests per second per gigabyte (IO/sec/GB) of storage.
- Heavy copy services workloads—large number of volumes with high I/O rates.

  In Figure 2-3 on page 16 you can see examples in which the ESS with the Turbo processor feature presents similar performance with or without PPRC, even at highly intensive levels of I/O activity. On the other hand, for the Standard processor example,

when the throughput activity gets highly intensive, the ESS performance is better when running without PPRC.

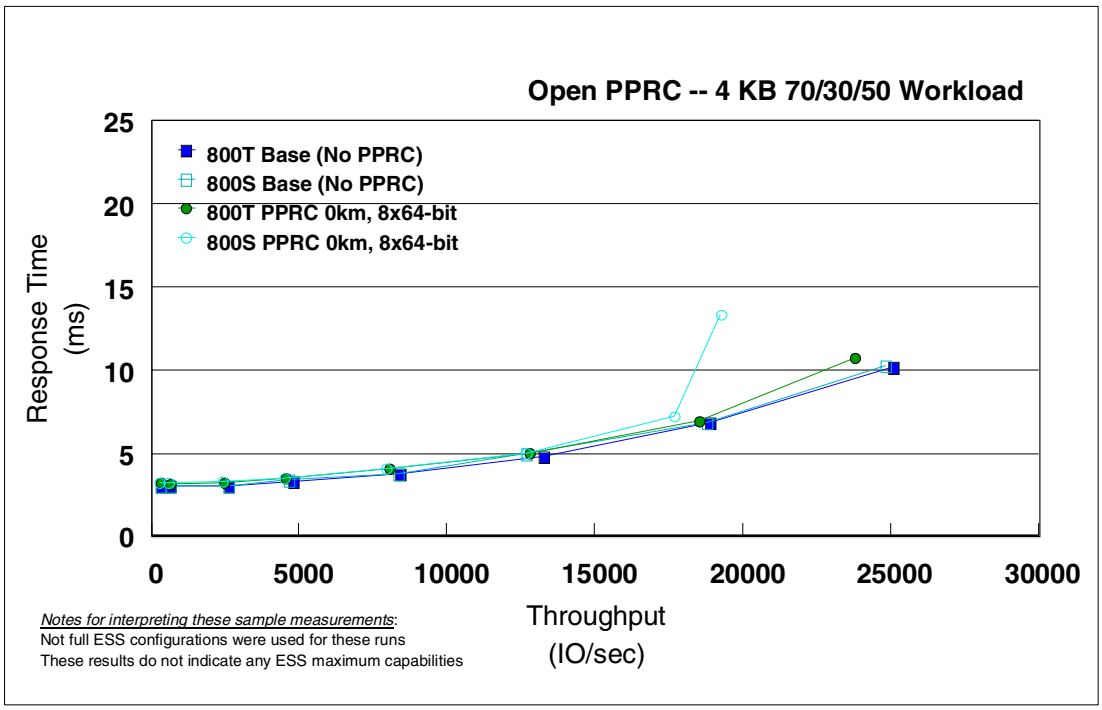


Figure 2-3 ESS Standard vs. Turbo processor option - with and without PPRC

Note: 70/30/50 is an open workload where read/write ratio = 2.33, read hits = 50 percent, destage rate = 17.2, percent and transfer size = 4 K.

For certain conditions under *heavy sequential* workloads, the Turbo processor feature capabilities could become overshadowed if the I/O bandwidth is limited by the internal buses, and not by the processor speed.

Figure 2-4 on page 17 is an example of the Turbo and the Standard processors throughput results when a certain heavy sequential workload is running in a specific ESS hardware configuration, and at the point when the internal bandwidth capacity limits the Turbo processor capabilities. The internal bandwidth will be determined by factors such as the number of SSA loops upon which the data resides. The more you spread the data across loops, the more internal bandwidth you will get. The characteristics of the sequential workload also determine the point at which the internal bandwidth limit is reached.

For each configuration and workload, a specific estimation should be done using the recommended tools.

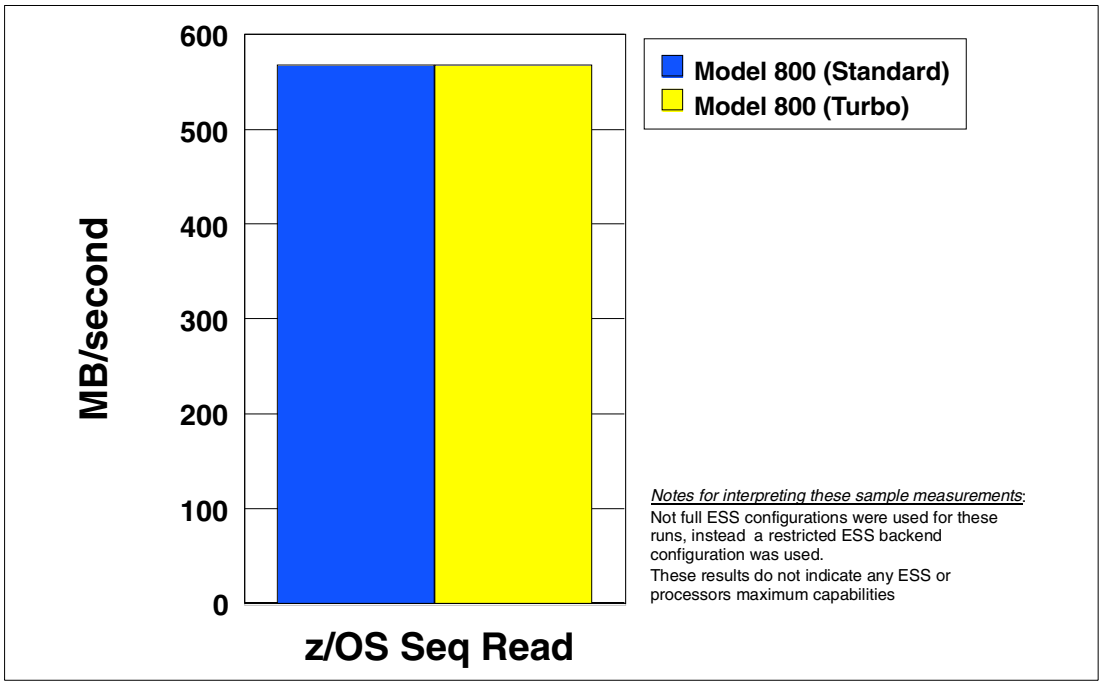


Figure 2-4 Standard versus Turbo - Intensive sequential

In general, the Standard processor feature can be sufficient. For heavy random workload and for heavy copy services workloads consider the Turbo processor feature. For heavy sequential workloads the Turbo processor feature benefits can be of no importance if the limiting factor is the internal bandwidth capabilities of your configuration.

Consider that the ESS *processor* (Standard or Turbo) is not an isolated factor when estimating the overall ESS performance, but must be considered together with other important factors like the I/O workload characteristics; the cache size; the disk drives' capacity and speed; the number and type of ESS host adapters; and the backend data layout and SSA loops.

Our recommendation is to use the Disk Magic modelling tool to estimate the benefits of the Turbo processor feature on your configuration.

2.5 Cache and NVS

In this section we describe the cache and NVS operations and characteristics in the ESS Model 800. We believe that if you have a good understanding of how cache works, you will better be able to understand why workloads can be differentiated as cache friendly or cache hostile. This understanding helps in making better decisions when planning for the performance of your environment.

2.5.1 Cache

Cache is used to keep both the read and write data that the host server needs to process. Having the cache as an intermediate repository, the host has no need to wait for the hard disk drive to either obtain or store the data that is needed. Instead, the operations of reading from the hard disk drive (*stage*), as well as the operation of writing into the hard disk drive (*destage*), are done by the ESS asynchronously from the host I/O processing. These allow the

completion of the host I/O operations at the electronic cache speeds without waiting for the much slower hard disk drives' operations.

Cache processing significantly improves the performance of the I/O operations done by the host systems that attach to the ESS. Cache size, together with the efficient internal structure and algorithms that the ESS uses, are factors that improve the I/O performance. The significance of this benefit will mostly be determined by the type of workload that is run.

In the ESS Model 800 there is the choice of 8, 16, 24, 32, or 64 GB of cache. This cache is divided between the two clusters of the ESS, giving the clusters their own non-shared cache.

To protect the data that is written during the I/O operations, the ESS stores two copies of the data: One in the cache and another in its non-volatile storage (NVS).

2.5.2 Non-volatile storage (NVS)

The non-volatile storage is used to store a second copy of the written data to ensure data integrity should there be a power failure or a cluster failure and the cache copy would be lost. The NVS of cluster 1 is located in cluster 2 of the ESS, and the NVS of cluster 2 is located in cluster 1. In this way, in the event of a cluster failure, the updated data of the failed cluster will be in the NVS of the surviving cluster. This write data can then be destaged to the disk arrays. At the same time, the surviving cluster will start to use its own NVS for write data, ensuring that two copies of write data are still maintained. This ensures that no data is lost even in the event of a component failure.

The ESS Model 800 has a 2 GB NVS. Each cluster has 1 GB of NVS, made up of four cards. Each pair of NVS cards has its own battery-powered charger system that protects data even if power is lost on the entire ESS for up to 72 hours.

2.5.3 Cache algorithms

With its effective caching algorithms, the IBM TotalStorage Enterprise Storage Server Model 800 is able to minimize wasted cache space, reduce disk drive utilization, and consequently reduce its back-end traffic.

The ESS manages its cache in 4 KB segments, so for small data blocks (4 KB and 8 KB are common database block sizes) minimum cache is wasted. In contrast, large cache segments could exhaust cache capacity while filling up with small random reads. Thus the ESS, having smaller cache segments, is able to avoid wasting cache space for situations of small record sizes that are common in the interactive applications.

This efficient cache management, together with the ESS Model 800 powerful back-end implementation that integrates new (optional) 15,000 rpm drives, enhanced SSA device adapters, and twice the bandwidth (as compared to previous models) to access the larger NVS (2 GB) and the larger cache option (64 GB), all integrate to give greater throughput while sustaining cache speed response times.

Let us see how the ESS cache algorithms operate when doing read and write I/O operations.

2.5.4 Read operations

The cache in the ESS is split between the clusters and is not shared. Each cluster has up to 32 GB of cache. The cache is managed in 4 KB segments (for FB operations a track is up to 9 segments, for CKD operations a full track of data in 3380 track format takes 12 segments, and a full track in 3390 track format takes 14 segments). As already discussed, the small size

allows efficient utilization of the cache, even with small records, and blocks operating in record mode.

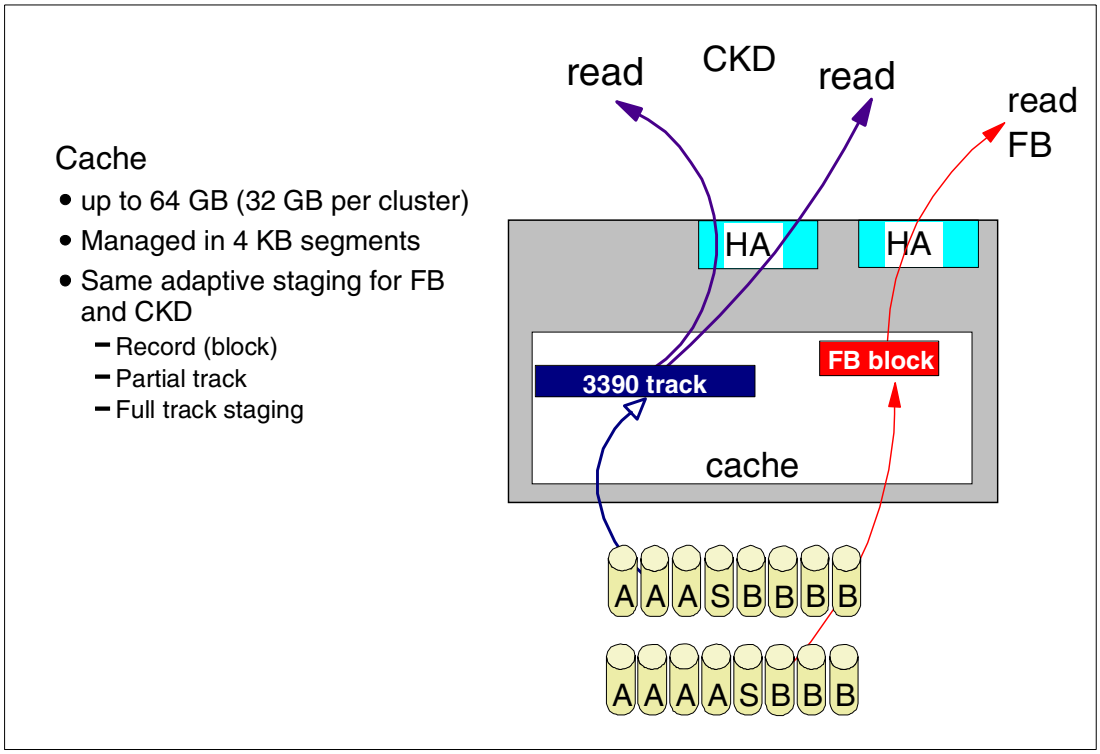


Figure 2-5 Read operation

A read operation sent to the cluster (illustrated in Figure 2-5) results in:

- A cache hit if the requested data resides in the cache. In this case the I/O operation will not disconnect from the channel/bus until the read is complete. Highest performance is achieved from read hits.
- A cache miss occurs if the data is not in the cache. The I/O is logically disconnected from the host, allowing other I/Os to take place over the same interface, and a stage operation from the RAID rank takes place. The stage operation can be one of three types:
  - Record or block staging.
  - Partial track staging.
  - The entire track is staged into the cache.

The method selected by the ESS to stage data is determined by the data access patterns. Statistics are held in the ESS on each zone. The statistics gathered on each zone determine which of the three cache operations is used for a specific track. This characteristic of operation is called *adaptive caching mode*, and is one of the ESS's intelligent cache algorithms. This is how it works:

- Data accessed randomly will tend to use the record access or block mode of staging.
- Data that is accessed normally with some locality of reference will use partial track mode staging. This is the default mode.
- Data that is not a regular format, or where the history of access indicates that a full stage is required, will set the full track mode.

The *adaptive caching mode* metadata is stored on disk and is reloaded at IML.

2.5.5 Write operations

As Figure 2-6 illustrates, at any moment there are always two secured copies of any update written into the ESS—one in cache and one in the NVS.

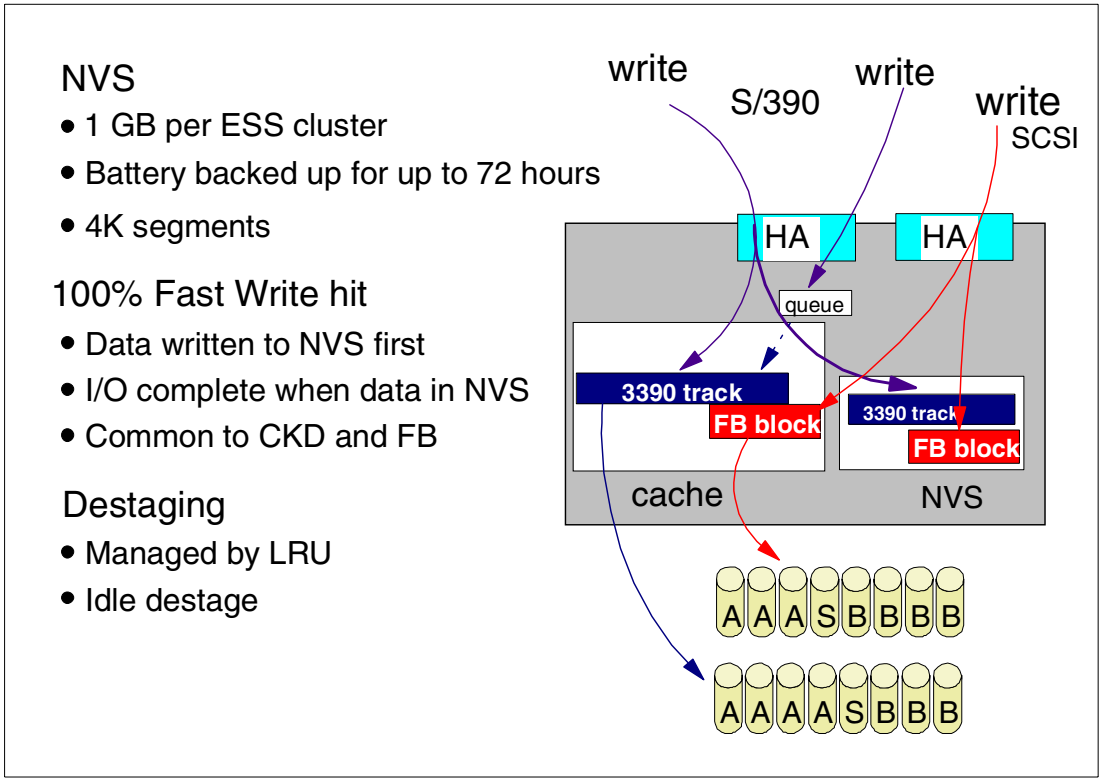


Figure 2-6 Write operation

Write operations - Fast writes

Data written to an ESS is almost 100 percent fast write hits. A fast write hit occurs when the write I/O operation completes as soon as the data is in the ESS cache and non-volatile storage (NVS). The benefit of this operation is very fast write operations.

Data received by the host adapter is transferred first to the NVS, and a copy is held in the host adapter buffer. The host is notified that the I/O operation is complete as soon as the data is in NVS. The host adapter, once the NVS transfer is complete, then transfers the data to the cache.

The data remains in the cache and NVS until it is destaged. Destage is triggered by the ESS caching algorithms based on cache and NVS usage thresholds.

NVS LRU

NVS is basically managed by a *least recently used* (LRU) algorithm. The ESS attempts to keep free space in the NVS by anticipatory destaging of tracks when the space used in NVS exceeds a threshold. In addition, if the ESS is idle for a period of time, an idle destage function will also destage tracks.

Both cache and NVS operate on LRU lists. Typically space in the cache occupied by sequential data is released earlier than space occupied by data that is likely to be re-referenced. Sequential data in the NVS is destaged ahead of random data.

When destaging tracks, the ESS attempts to destage all the tracks that would make up a RAID stripe, minimizing the RAID-related activities in the backend SSA adapters.

NVS location

NVS for cluster 1 is located physically in an I/O drawer (where the device adapters reside) of cluster 2, and vice versa. This ensures that we always have one good copy of data, should a failure in one cluster occur.

2.5.6 Sequential read operations

For sequential reading, either for RAID-10 or for RAID-5 ranks, the ESS implements unique highly efficient algorithms (summarized in Figure 2-7). There are two ways to trigger the ESS sequential processing: One is automatically initiated by the ESS when it detects that sequential operations are occurring; the other is requested by the host application when it is going to process sequential I/Os.

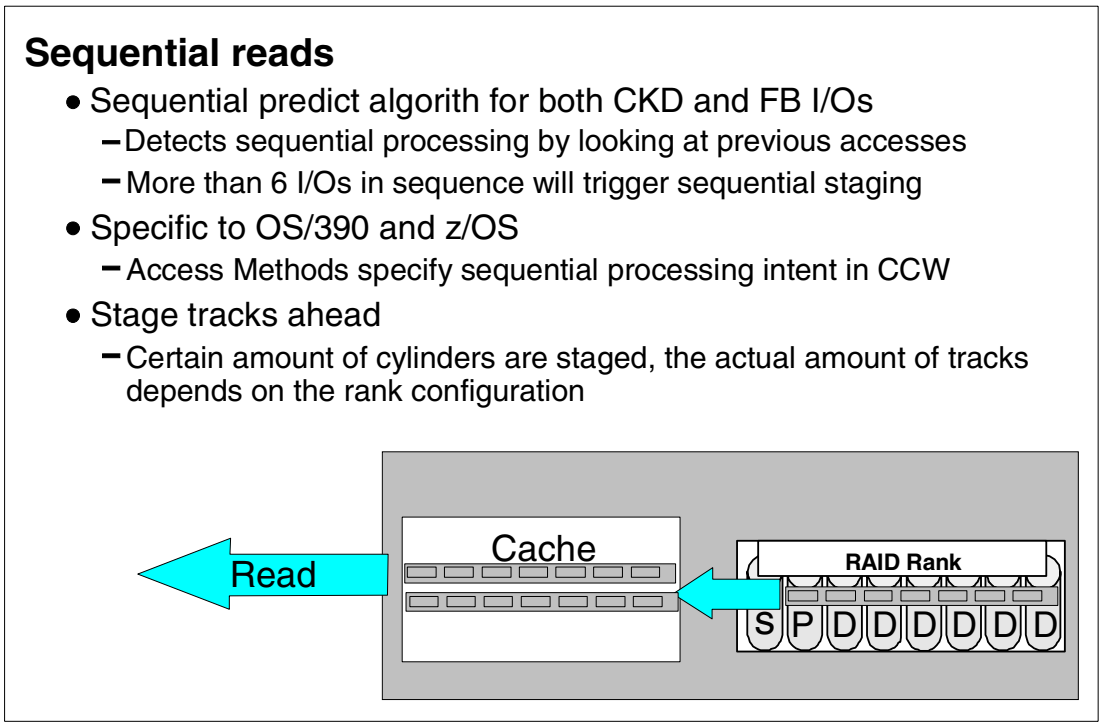


Figure 2-7 Sequential read

The sequential staging reads ahead a certain amount of cylinders; the actual amount depends on the array configuration. For example:

- On 18.2 GB disks arrays for 6+P it is 30 tracks, and for 7+P it is 28 tracks.
- On 36.4 GB, 72.8 GB, and 145.6 GB disks arrays for 6+P it is 24 tracks, and for 7+P it is 28 tracks.
- For RAID-10 (both 3+3' and 4+4' rank configurations, all disk capacities) it is 24 tracks.

As the tracks are read, when about the middle of a staging group is read then the next group starts to be staged. This delivers maximum sequential throughput with no delays waiting for data to be read from disk.

Stage requests for sequential operations can be performed in parallel on the RAID array, giving the ESS its high sequential throughput characteristic. Parallel operations can take

place because the logical data tracks are striped across the physical data disks in the RAID array.

Cache space used by tracks that have been read sequentially is eligible to be freed quickly to release the used cache space. This is because sequential data is rarely re-read within a short period.

Sequential detection

The ESS *sequential detection* algorithm analyzes sequences of I/Os to determine if data is being accessed sequentially. As soon as the algorithm detects that six or more tracks have been read in succession, the algorithm triggers a sequential staging process.

This algorithm applies equally when accessing CKD data or FB data.

Software setting

The second method of triggering sequential staging—implemented by zSeries operating systems—consists of specifying the sequential access through the software in the channel program.

It is common that z/OS sets a bit into the channel program notifying the disk subsystem that all subsequent I/O operations will be sequential read requests. ESS supports these bits in the channel program and helps to optimize its pre-fetch process.

2.5.7 Sequential write operations

Sequential write operations on the RAID-5 ranks are done in a RAID-3 mode (parallel transfer of all stripes of the set). This is beneficial when operating upon the RAID-5 ranks because it avoids the read and recalculation overheads, thus neutralizing the RAID-5 write penalty. An entire stripe of data is written across all the disks in the RAID array, and the parity is generated once for all the data simultaneously and written to the parity disk (the rotating parity disk). This technique does not apply for the RAID-10 ranks, because there is no write penalty involved when writing upon RAID-10 ranks. Figure 2-8 summarizes the sequential write process.

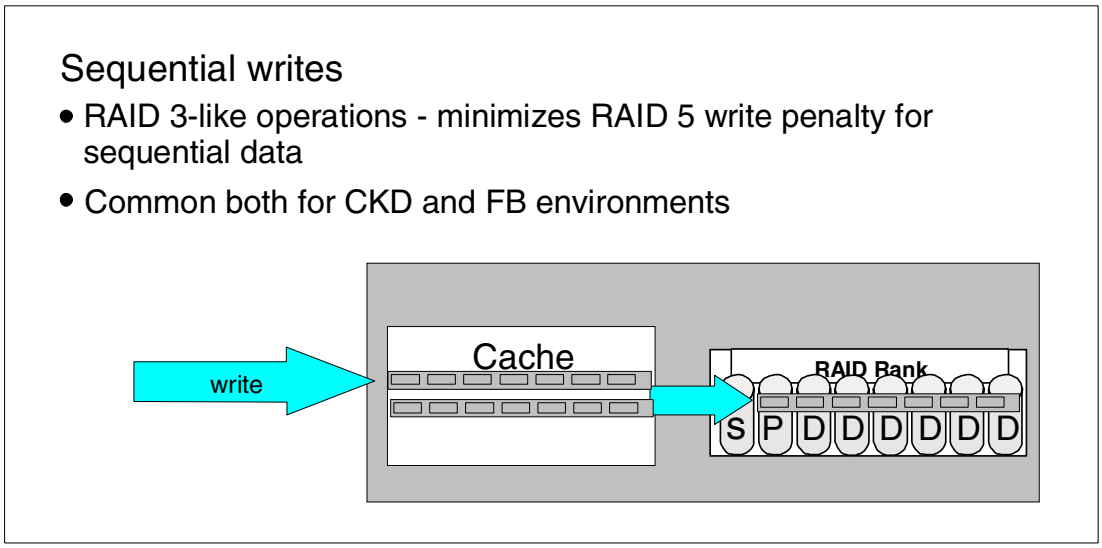


Figure 2-8 Sequential write

2.5.8 Choosing the cache size

Cache sizes in the ESS can be either 8, 16, 24, 32, or 64 GB. With different cache sizes available, the ESS can be configured for optimal performance. The factors that will have to be considered to determine the proper cache size will be:

- The total amount of disk capacity that the ESS backend will hold
- The characteristic access density (I/Os per GB) for the stored data
- The characteristics of the I/O workload (cache friendly, unfriendly, standard; block size; random or sequential; read/write ratio; I/O rate)

Basically, the larger the cache size the better the I/O performance characteristics. An approximate and conservative rule of thumb (ROT), disregarding the access density and the I/O operations specific characteristics, based solely on the backend total capacity, is to estimate between 2 GB to 4 GB of cache per 1 TB of storage. This ROT can work for many workloads, but it can also be very inaccurate for many other workloads.

Alternatively, if you do not increase disk capacity (access density remaining the same) but double the cache size, then the cache hit ratio may improve resulting in better I/O response times to the application. How significant the hit ratio improvement is will depend on the caching attributes of the I/O workload.

Consider that the *cache size* is not an isolated factor when estimating the overall ESS performance, but must be considered together with other important factors like the I/O workload characteristics; the disk drives capacity and speed; the ESS processors (Turbo or Standard); the number and type of ESS host adapters; and the backend data layout and SSA loops.

Our recommendation is to use Disk Magic for properly determining the more convenient cache size to include in your ESS hardware configuration.

2.6 ESS disks

The increased number of options of disk drive capacities available with the ESS Model 800, including intermix support, provide more flexibility to choose your configuration, but it also adds more complexity when doing the selection. In this section we summarize disk drive options available in the ESS and then we discuss guidelines that will help you define your optimal ESS disk configuration.

2.6.1 ESS disk capacity

The maximum number of hard disk drives that a fully configured IBM TotalStorage Enterprise Storage Server Model 800 holds is 384. For this configuration the base enclosure will hold 128 disk drives, and the expansion rack will hold 256 disk drives. When configured with 145.6 GB capacity disk drives, this gives a total physical disk capacity of approximately 55.9 TB.

The minimum ESS Model 800 available configuration is 582 GB. This capacity can be configured with 32 disk drives of 18.2 GB contained in four *eight-packs*. All increments of capacity are installed in pairs of eight-packs; thus the minimum capacity increment is a pair of eight-packs of either 18.2 GB, 36.4 GB, 72.8 GB, or 145.6 capacity.

2.6.2 Disk eight-packs

The ESS *eight-pack* is the basic unit of disk capacity within the ESS base and expansion rack. Each eight-pack consists of eight hard disk drives of similar capacity and speed. As mentioned before, these eight-packs are ordered and installed in pairs. Each eight-pack can be configured as a RAID-5 rank (6+P+S or 7+P) or as a RAID-10 rank (3+3+2S or 4+4).

The ESS Specialist will configure the eight-packs on a loop with spare DDMs as required. When the configuration includes an intermix of different capacity drives, this may result in the creation of additional DDM spares on a loop as compared to non-intermixed configurations. Spare configurations and considerations are explained in detail in Chapter 3, "Logical configuration planning" on page 49.

Currently with the ESS Model 800 there is the choice of different disk drive capacities and speeds:

- 18.2 GB 10,000/15,000 rpm disks
- 36.4 GB 10,000/15,000 rpm disks
- 72.8 GB 10,000/15,000 rpm disks
- 145.6 GB 10,000 rpm disks

The eight disk drives assembled in each eight-pack unit are all of the same capacity and speed. But it is possible to mix eight-packs of different capacity and speed (rpm) within an ESS, within the guidelines described in 2.6.4, "Disk eight-pack intermixing" on page 25.

2.6.3 Disk eight-pack capacity

Eight-packs in the ESS can be of different capacities and speed. Additionally, once installed in the ESS, they can be configured either as RAID-5 or RAID-10 arrays. Thus we can speak of a raw *physical capacity* that will depend on the number and type of eight-packs that the ESS holds, and we can also speak of the resulting *effective capacity* of the ESS that will depend on the RAID configuration being done.

RAID-5 as implemented on the ESS offers the most cost-effective performance/capacity trade-off options for the ESS internal disk configurations, because it optimizes the disk storage capacity utilization. RAID-10 can offer higher performance for selected application workloads, but requires considerably more disk space. Refer to Figure 2-9 on page 25.

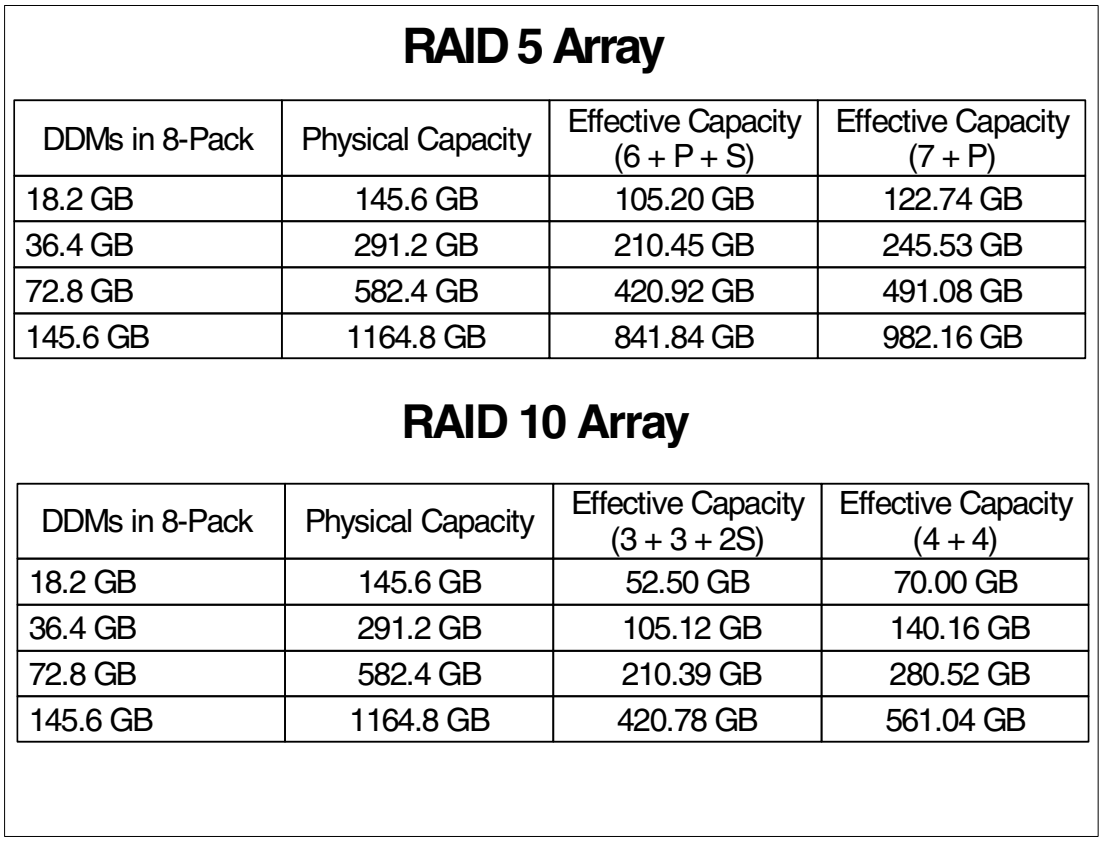
Physical capacity

The physical capacity (or raw capacity) of the ESS is the result of adding the physical capacities of each of all the installed disk eight-packs in the ESS. The physical capacity of the eight-packs will be determined by the disk drives capacity that it holds (refer to Figure 2-9 on page 25).

Effective capacity

The effective capacity of the ESS is the capacity available for user data. The combination and sequence in which eight-packs are added to the ESS, and then how they are logically configured, will determine the effective capacity of the ESS (refer to Figure 2-9 on page 25).

The logical configuration alternatives and the resulting effective capacities are discussed later in Chapter 3, "Logical configuration planning" on page 49.



RAID 5 Array

DDMs in 8-Pack	Physical Capacity	Effective Capacity (6 + P + S)	Effective Capacity (7 + P)
18.2 GB	145.6 GB	105.20 GB	122.74 GB
36.4 GB	291.2 GB	210.45 GB	245.53 GB
72.8 GB	582.4 GB	420.92 GB	491.08 GB
145.6 GB	1164.8 GB	841.84 GB	982.16 GB

RAID 10 Array

DDMs in 8-Pack	Physical Capacity	Effective Capacity (3 + 3 + 2S)	Effective Capacity (4 + 4)
18.2 GB	145.6 GB	52.50 GB	70.00 GB
36.4 GB	291.2 GB	105.12 GB	140.16 GB
72.8 GB	582.4 GB	210.39 GB	280.52 GB
145.6 GB	1164.8 GB	420.78 GB	561.04 GB

Figure 2-9 ESS Model 800 arrays - Physical and effective capacities

2.6.4 Disk eight-pack intermixing

It is possible for an ESS to have an intermix of eight-packs of different capacity and speed characteristics. Some guidelines apply for these intermixed configurations.

Capacity intermix

Disk eight-packs of different capacities can be installed within the same ESS, in the same or in different device adapter loops (SSA loops). In the SSA loops of the ESS, it is possible to intermix:

- 18.2 GB, 36.4 GB, 72.8 GB, and 145.6 GB capacity disk eight-packs
- RAID-5 and RAID-10 array configurations

For a newly installed ESS with an intermixed disk drive capacity configuration, the eight-packs will be installed in sequence from highest capacity to lowest capacity.

In the device adapter loop, a *spare pool* consisting of two disk drives is created for each different drive capacity installed on the SSA loop. The spares are reserved from either two 6+P arrays (RAID-5) or one 3+3 array (RAID-10). Sparing is discussed in detail in 2.8, “RAID implementation” on page 32.

Speed (RPM) intermix

An ESS can have eight-packs that differ in their drive speed, but these eight-packs cannot be of the same drive capacity. For example, 36.4 GB 15,000 rpm can be intermixed with 72.8 GB 10,000 rpm eight-packs in the same ESS, but 36.4 GB 15,000 rpm cannot be intermixed with 36.4 GB 10,000 rpm within the same ESS currently.

Preview

IBM is previewing plans for the ESS to support the intermix of 15 K rpm and 10 K rpm disks of the same capacity within ESS Model 800.

Note: Previews are intended to provide information concerning IBM's future plans and directions. Such plans and directions are subject to change.

2.6.5 Disk conversions

There is the ability to exchange an installed eight-pack with an eight-pack of greater capacity, or higher rpm, or both. As well as enabling you to best exploit the intermix function, the capacity conversions are particularly useful for increasing storage capacity at sites with floor-space constraints that prohibit the addition of the Expansion Enclosures.

The eight-pack conversions must be ordered in pairs and are subject to the disk intermix guidelines discussed previously.

2.6.6 Step Ahead option

The Step Ahead capacity-on-demand program enables the installation of capacity ahead of the user requiring it (the capacity is purchased at a later time). Step Ahead provides two additional eight-packs of the same capacity, pre-installed in the ESS. The same disk intermix rules, as explained previously, apply with the Step Ahead eight-packs.

2.7 Choosing the ESS disks

In this section we discuss the considerations to analyze when deciding the disks capacities and speeds that will be included in the ESS hardware configuration.

2.7.1 Disk capacity

Analysis of workload trends over the recent years shows that *access densities* continue to decline. Access density is defined as the number of I/O requests per second per gigabyte (IO/sec/GB) of storage. This trend results in that workloads can often be migrated to higher capacity disk drives without any significant impact in their I/O performance.

Also the evolution in technology results are that as new larger capacity disk drives become available, they usually come with improved characteristics of performance. Thus we are seeing that now many installations feel more confident when moving to the larger capacity disk drives configurations.

When choosing the capacity of your ESS disk drives, these considerations should be regarded:

- The characteristics of the I/O workload (cache friendly, unfriendly, standard; block size; random vs. sequential; read/write ratio; I/O rate) is a key factor when deciding the capacity and number of the disk drives that will be included in the ESS configuration. For example, if the workload is cache friendly, more I/Os are completed in cache and less activity is performed on the backend disk drives. This type of workload is a good candidate for storing its data in the larger capacity disk drives.
- Some I/O workloads may be very cache unfriendly or have a very high random write content. These workloads, where a larger part of the I/Os are completed in the backend disks, may perform better when using more disk drives.

A disk drive by itself can do a number of operations per second, so using more disks drives can result in better performance.

- Additional cache to the ESS can result in a reduction of the access density to the backend disk drives. This is especially true if the I/O workload is cache friendly.
- When making decisions on the capacity of the ESS disk drives remember that you have the option of different speeds for that same capacity. Faster disks (rpm) perform better, and the relevance of this will be determined by type of the workload, whether an important part of the I/O needs to complete on the backend disks as opposed to the cache.
- For remote copy implementations when the I/O response time at the secondary ESS is not a critical issue, the larger capacity disk drives can be a good and less expensive choice. This can be especially true for PPRC Extended Distance (PPRC-XD) secondaries and XRC secondaries. For these implementations the secondary ESS can be configured with the larger 145.6 GB capacity disk drives.

Consider that the disk drive *capacity* is not an isolated factor when estimating the overall ESS performance, but must be considered together with other important factors like the I/O workload characteristics, the cache size, the ESS processors (Turbo or Standard), the disk drives speed, the number and type of ESS host adapters, and the backend data layout and SSA loops.

Our recommendation is to use Disk Magic for properly determining the more convenient disk drives capacity mix to include in your ESS hardware configuration.

2.7.2 Examples using 145.6 GB disk drives

In this section we present examples of ESS measurement results when using the larger capacity 145.6 GB disk drives. The discussions for these examples will help you better understand what the performance implications are when using the larger disks.

Cache-hostile workload

Figure 2-10 on page 28 illustrates an example of a z/OS *cache-hostile* workload that is run in a 145.6 GB/10 krpm disk drive configuration and in a 72.8 GB/10 Krpm disk drive configuration.

For this example of cache-hostile workload, and where the total capacity was the same on both configurations, the 145.6 GB disk drive configuration shows a higher response time compared to the 72.8 GB disk drive configuration.

Note: For *cache-hostile* workload read/write ratio = 2; read hit ratio = 0.34; destage rate = 18.3 percent; and transfer size = 4 K.

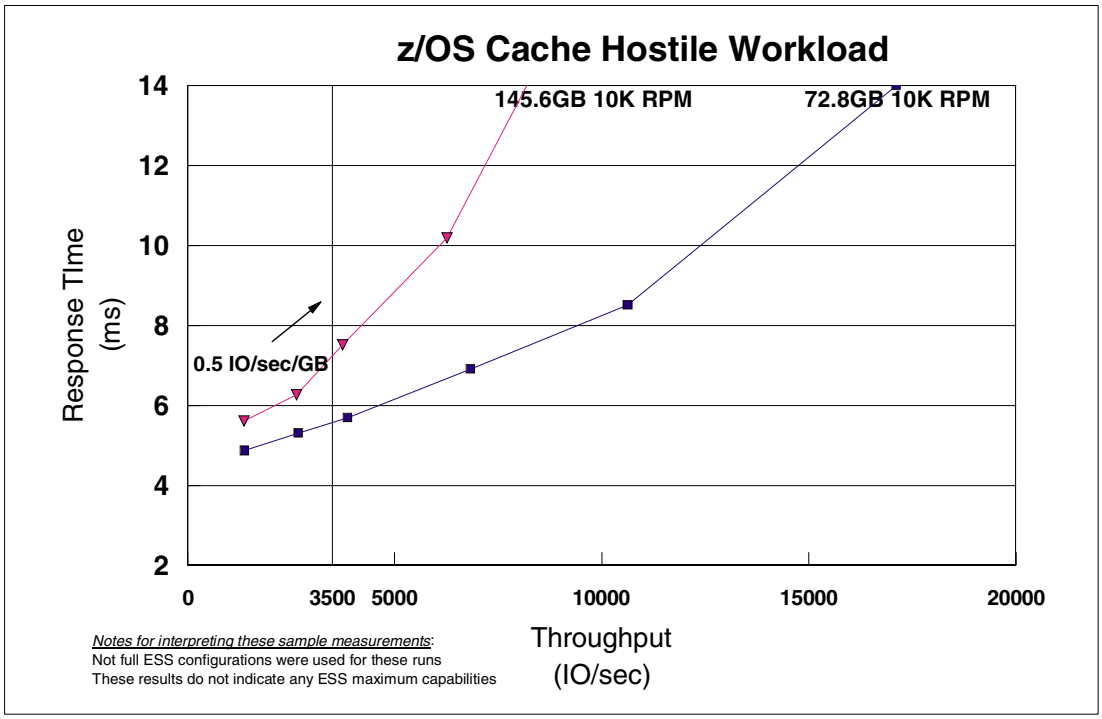


Figure 2-10 Cache-hostile workload on a 145.6 GB disk drive configuration

You can see in our example in Figure 2-10 that for the lower access density values—less than 0.5 IO/sec/GB—the response time of the larger capacity 145.5 GB disk drive configuration can be acceptable for many applications.

Cache-standard workload

Figure 2-11 on page 29 illustrates an example of a z/OS *cache-standard* workload that is run in a 145.6 GB/10 Krpm disk drive configuration and in a 72.8 GB/10 Krpm disk drive configuration.

For this example of cache-standard workload, and where the total capacity was the same on both configurations, the 145.6 GB disk drive configuration performs quite similar to the 72.8 GB configuration when the access density is 1 IO/sec/GB or less. This can be acceptable for many production applications.

Note: For *cache-standard* workload read/write ratio = 3; read hit = 0.735; destage rate = 11.6 percent; and transfer size = 4 K.

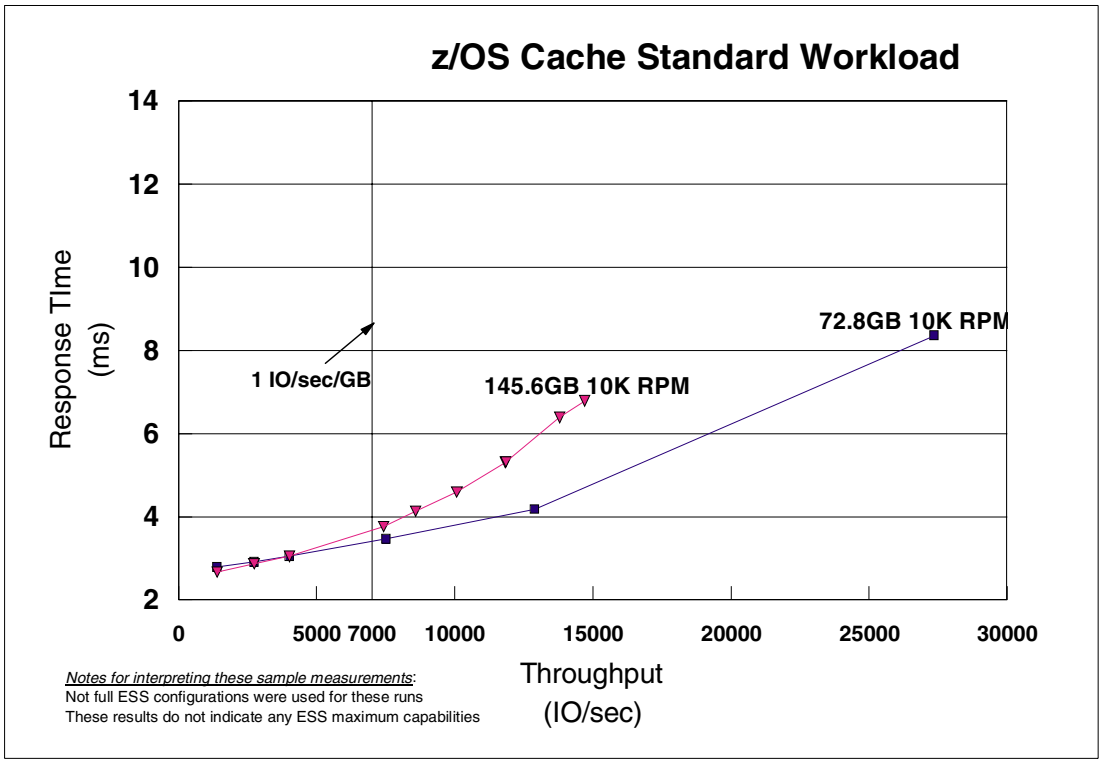


Figure 2-11 Cache-standard workload on a 145.6 GB disk drive configuration

Cache-friendly workload

Figure 2-12 on page 30 illustrates an example of a z/OS *cache-friendly* workload that is run in a 145.6 GB/10 Krpm disk drive configuration and in a 72.8 GB/10 Krpm disk drive configuration.

For this example of cache-friendly workload, and where the total capacity was the same on both configurations, the 145.6 GB disk drive configuration shows very good response time for access densities of 2 IO/sec/GB or less. This can be applicable to many of your applications.

Note: For *cache-friendly* workload read/write ratio = 4.9; read hit = 0.82; destage rate = 7.5 percent; and transfer size = 4 K.

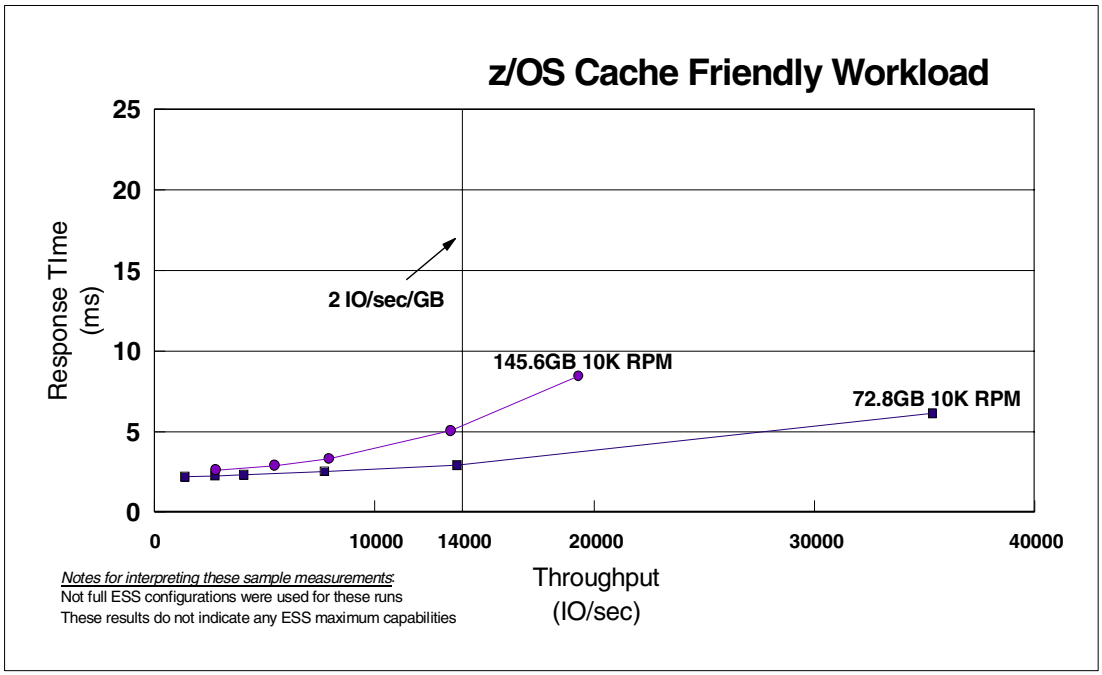


Figure 2-12 145.6 GB - Cache friendly workload

2.7.3 Disk speed (RPM)

Rotational speed of the hard disk drives is measured in *revolutions per minute* (RPM) and, with the other physical and mechanical characteristics remaining the same, the faster the disk the better the performance. Because of this, disk drive vendors have continued to increase rotational speed in order to improve both disk *transfer rates* and *rotational latency* (the time a read/write head must wait for a sector on a disk to pass under it).

Being so, one of the simplest ways of improving the overall performance of a disk subsystem is to install the highest speed (RPM) disk drives. This is especially relevant for workloads that are *cache-unfriendly* or hostile, who benefit more than others from the ESS configurations that include the faster ESS 15 Krpm drives.

Consider that the disk drive *speed* is not an isolated factor when estimating the overall ESS performance, but must be considered together with other important factors like the I/O workload characteristics, the cache size, the ESS processors (Turbo or Standard), the disk drives capacity, the number and type of ESS host adapters, and the backend data layout and SSA loops.

Our recommendation is to use Disk Magic for properly assessing how significant it is for your I/O processing the performance benefits of having faster disk drives in your ESS hardware configuration.

2.7.4 Examples using 15 Krpm and 10 Krpm disk drives

In this section we present examples of ESS measurement results when using 15 Krpm and 10 Krpm disk drives. The discussions for these examples will help you better understand what the performance implications of the disk drive speed factor are.

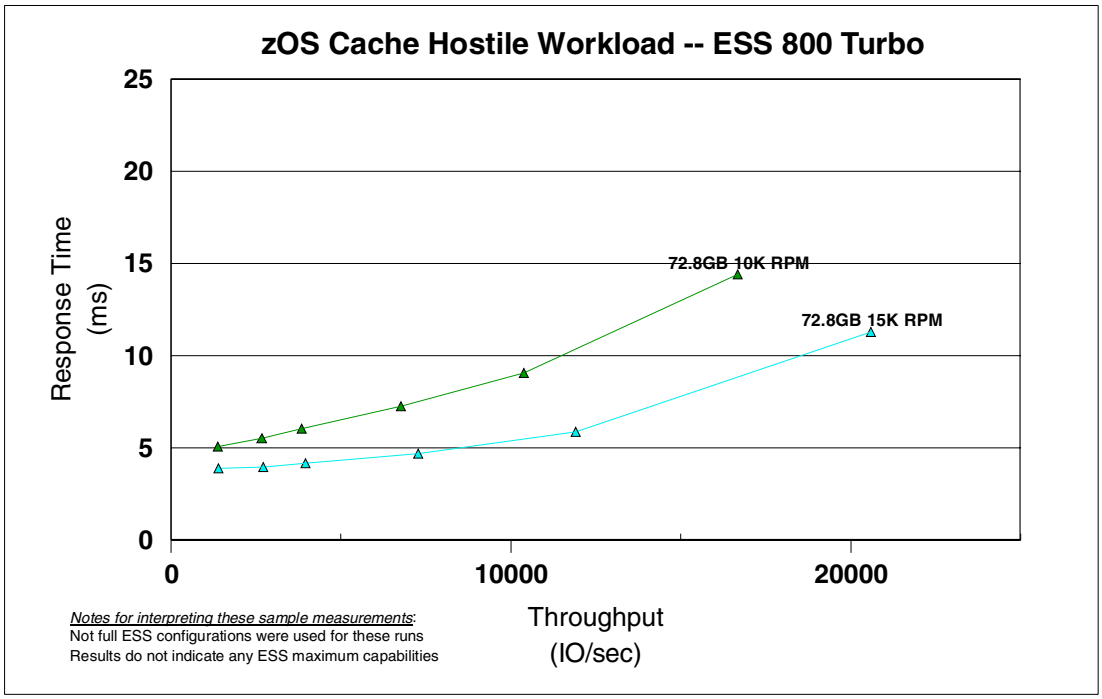


Figure 2-13 Cache-hostile workload - 15 Krpm vs. 10 Krpm disk drives

Figure 2-13 illustrates an example of a cache-hostile workload that is run on two 72.8 GB disk drives configurations: One of the configurations has 10 Krpm disk drives and the other one has 15 Krpm disk drives. You can see that the 15 Krpm configuration performs better than the 10 Krpm configuration, always delivering better response times.

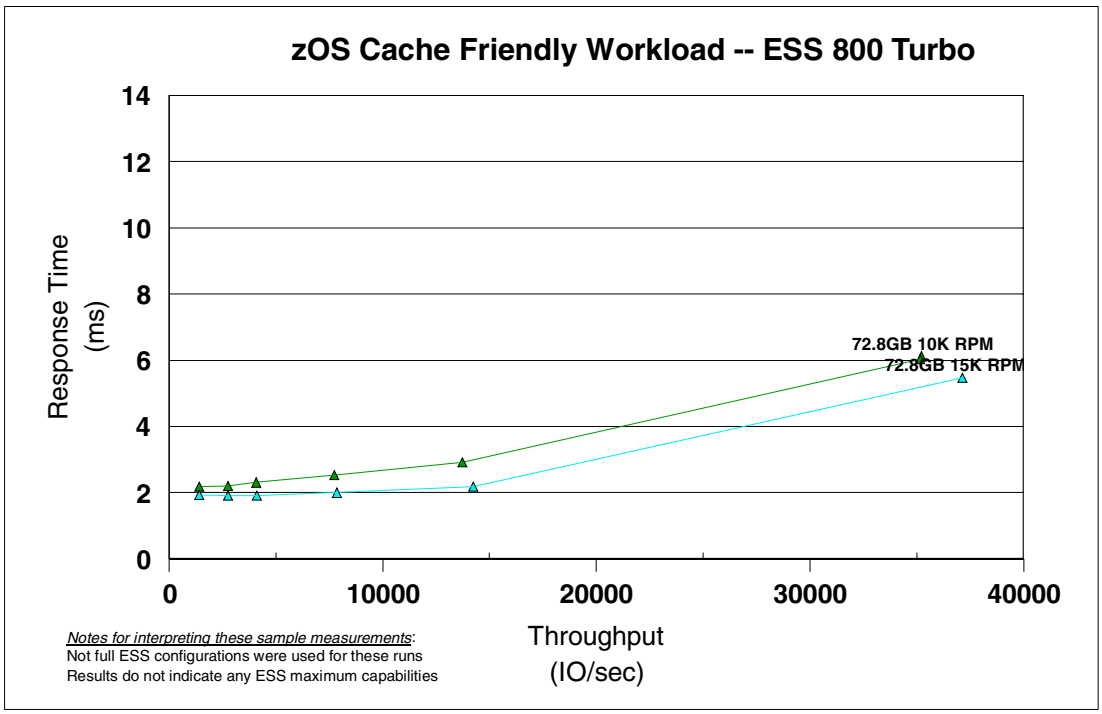


Figure 2-14 Cache friendly workload - 15 Krpm vs. 10 Krpm drives

Figure 2-14 on page 31 illustrates the example when a cache-friendly workload was run on both the 10 Krpm and the 15 Krpm configurations.The 15 Krpm configuration performs better, always delivering lower response times for all throughputs as compared to the 10 Krpm configuration.

You can appreciate comparing the results in Figure 2-13 on page 31 and Figure 2-14 on page 31, that when running a cache friendly workload the performance gain is less significant than when running the cache-hostile workload.

2.8 RAID implementation

In this section we describe the characteristics of the RAID implementation in the ESS Model 800. The performance considerations regarding the possible RAID configurations are discussed in detail later in Chapter 3, “Logical configuration planning” on page 49.

2.8.1 RAID ranks

The basic unit where data is stored in the ESS is the *hard disk drive* (also referred as DDM). Disk drives for the ESS Model 800 are available in capacities of 18.2 GB, 36.4 GB, 72.8 GB, or 145.6 GB. Physically, eight disk drives (of the same capacity and speed) are grouped together in an *eight-pack*, and these eight-packs are installed in pairs on the SSA loops. One SSA loop can hold up to six eight-packs (three pairs), which means that the maximum number of 48 DDMs can be found in a loop. The *raw* or *physical capacity* installed on a loop will depend on the eight-packs (their capacity) that are installed on that loop. Physical capacity is discussed in 2.6.3, “Disk eight-pack capacity” on page 24.

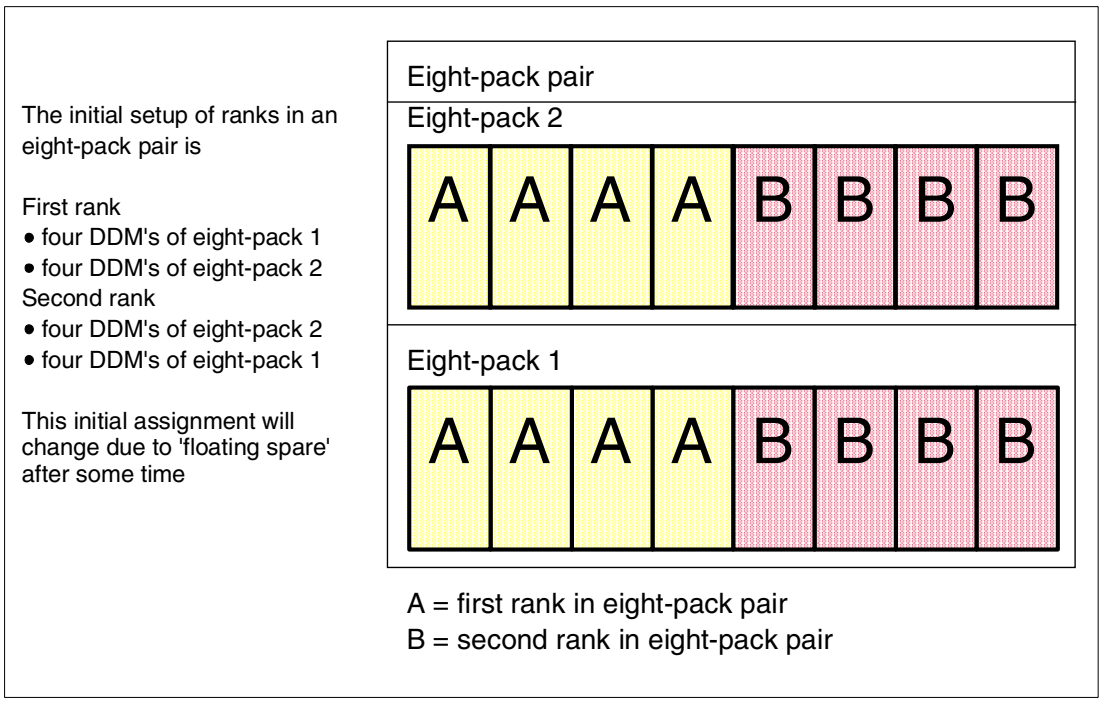


Figure 2-15 Initial rank setup

Logically, eight DDMs (out of an eight-pack pair) are grouped in an *array*, also referred to as an ESS *rank*. As illustrated in Figure 2-15, initially four DDMs of the first eight-pack and four DDMs of the second eight-pack make up the *rank*. This initial correspondence will change with time after initial configuration, due to the floating spare characteristic of the ESS.

The IBM TotalStorage Enterprise Storage Server Model 800 disk arrays are configured in Redundant Array of Independent Disks (RAID) implementations. A RAID *rank* (or RAID *array*) is owned by one ESS *logical subsystem* (LSS) only, either an FB LSS or a CKD LSS. During the logical configuration process, the decision is made on which type of RAID rank the array will be. With the ESS Model 800, the ranks can be configured as either RAID-5 or RAID-10.

Then each rank is formatted as a set of *logical volumes* (LVs). The number of logical volumes in a rank depends on the amount of capacity by the disk drives in the array, and the size of the LUNs (for FB attachment) or the emulated 3390 DASDs (for CKD attachment). The logical volumes are also configured during the logical configuration procedure. When configured, the logical volumes are striped across all the data disks and then mirrored—if it is a RAID-10 rank; or striped across all data disks in the array along with the parity disk (floating)—if it is a RAID-5 rank.

2.8.2 RAID-5 rank

One of the two possible RAID implementations in an ESS Model 800 rank is RAID-5. The ESS RAID-5 implementation consists of eight disk drives: A set of 6 or 7 disks for *user data*, plus a *parity* disk for reconstruction of any of the user data disks should one become unusable. In fact there is no dedicated physical parity disk, but a *floating parity disk* striped across the rest of the data disks in the array. This prevents any possibility of the parity disk becoming an I/O hot spot.

Because the ESS architecture for maximum availability is based on two *spare* drives per SSA loop (and per capacity), if the first two ranks that are configured in a loop are defined as RAID-5 then they will be defined by the ESS with six data disks plus one parity disk plus one spare disk—this is a 6+P+S rank configuration. This will happen for the first two ranks of each capacity installed in the loop if configured as RAID-5.

Once the *two spares per capacity* rule is fulfilled, then further RAID-5 ranks in the loop will be configured by the ESS as seven data disks and one parity disk—this is a 7+P rank configuration. Figure 2-16 on page 34 illustrates the two arrangements of disks possible in the ESS when configuring RAID-5 ranks.

RAID-5 configurations:

- Ranks in first eight-pack pair configured in the loop will be:
  6 + P + S
- Ranks in second and third pairs configured in the loop will be:
  7 + P
- For a loop with an intermixed capacity, the ESS will assign two spares for each capacity. This means there will be two 6+P+S arrays per capacity

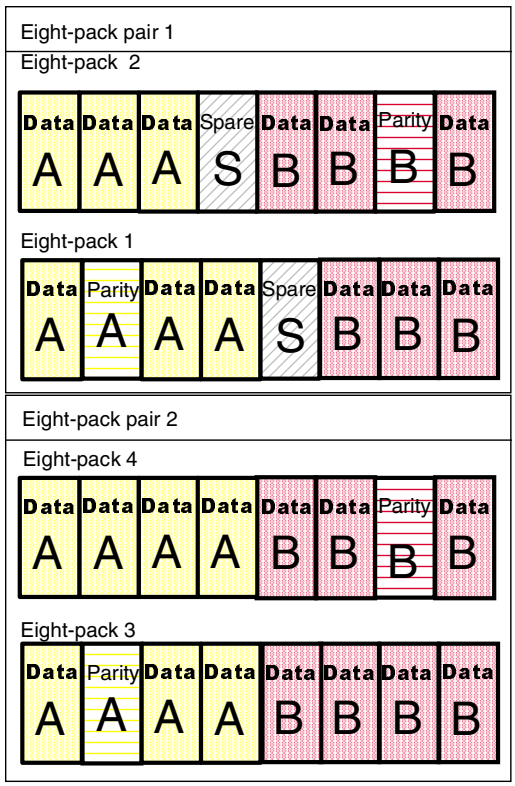


Figure 2-16 RAID-5 rank implementation

In a RAID-5 implementation, the disk access arms can move independently for each disk, thus enabling multiple concurrent accesses to the array. This results in multiple concurrent I/O requests being satisfied, thus providing a higher random-transactions throughput.

RAID-5 is well suited for random access to data in small blocks. Most data transfers, reads, or writes, involve only one disk and hence operations can be performed in parallel and provide a higher throughput. In addition to this efficiency for random transaction operations, the RAID-5 implementation of the ESS is able to work in a RAID-3 style when processing sequential operations for maximum sequential throughput.

2.8.3 RAID-10 rank

The other possible RAID implementation in an ESS Model 800 is RAID-10 (also known as RAID 1+0). A RAID-10 rank consists of a set of disks for *user data* and their *mirrors*. There is no parity disk to rebuild a failed disk. In case one disk becomes unusable, then its *mirror* will be used to access the data and also to build the *spare*.

Because the ESS architecture for maximum availability is based on *two spare drives* per SSA loop (and per capacity), if the first rank that is configured in a loop is defined as a RAID-10 rank, then it will be defined by the ESS with three *data* disks, plus the three *mirrors*, plus two spares—this is a 3+3+2S rank configuration. This will happen for the first rank of each capacity installed in the loop if configured as RAID-10.

Once the *two spare per capacity* rule is fulfilled, then further RAID-10 ranks in the loop will be configured by the ESS as four data disks plus four mirrors—this is a 4+4 rank configuration. Figure 2-17 on page 35 illustrates the two arrangements of disks that can be found in the ESS when there are RAID-10 ranks.

RAID-10 configurations:

- First RAID-10 rank configured in the loop will be: **3 + 3 + 2S**
- Additional RAID-10 ranks configured in the loop will be **4 + 4**
- For a loop with an intermixed capacity, the ESS will assign two spares for each capacity. This means there will be one 3+3+2S array per capacity

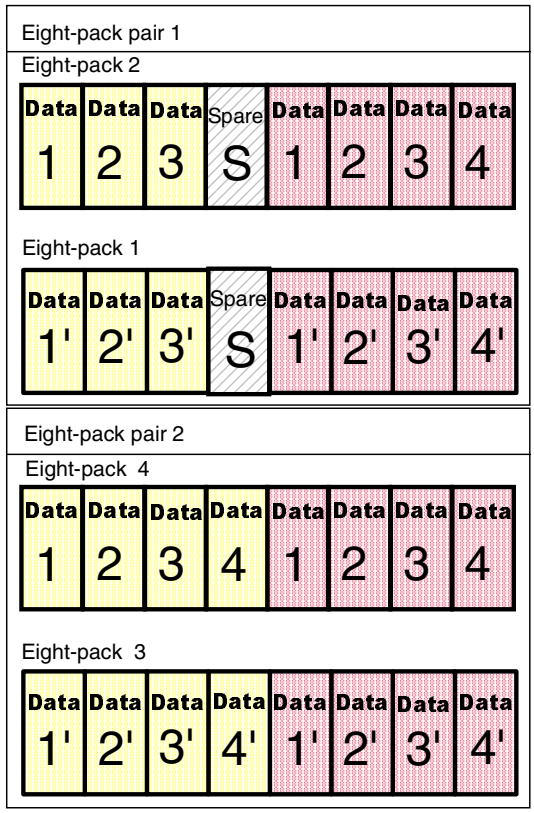


Figure 2-17 RAID-10 rank implementation

RAID-10 is also known as RAID 1+0, because it is a combination of RAID-1 (mirroring) and RAID-0 (striping). The striping optimizes the performance by striping volumes across several disk drives (in the ESS Model 800 implementation, three or four DDMs). RAID-1 is the protection against a disk failure by having a mirror copy of each disk. By combining the two, RAID-10 provides data protection and good I/O performance.

2.8.4 Combination of RAID-5 and RAID-10 ranks

It is possible to have RAID-10 and RAID-5 ranks configured within the same loop, as illustrated in Figure 2-18 on page 36.

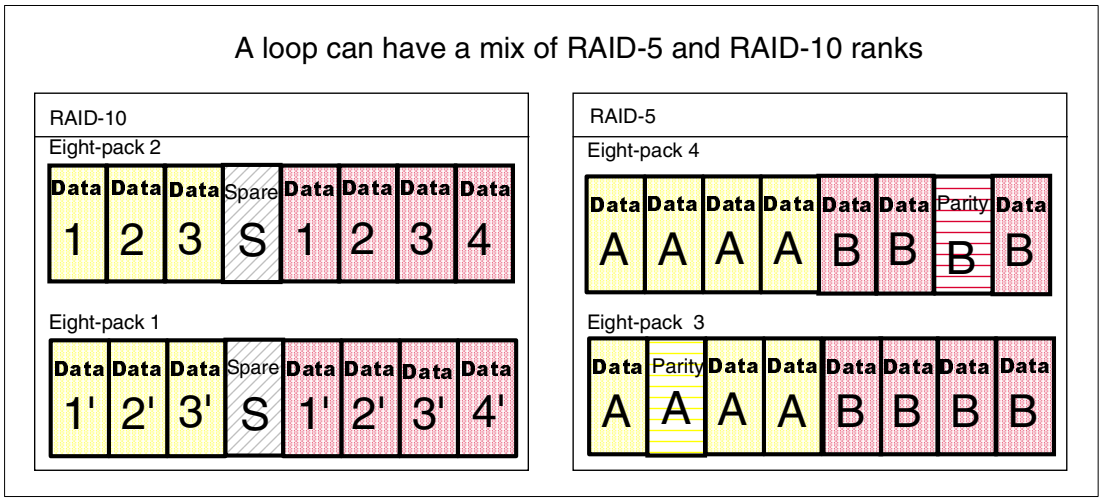


Figure 2-18 RAID-5 and RAID-10 in the same loop

Chapter 3, “Logical configuration planning” on page 49, discusses the considerations for getting balanced configurations when combining RAID-5 and RAID-10 ranks.

2.9 Host adapters

The ESS connects to the application servers by means of its *host adapters* (HAs). The ESS host adapters are mounted in four *bays* and each of the four bays is able to hold up to four host adapters, making a maximum of 16 host adapter cards for one ESS. Each host adapter can communicate with either cluster of the ESS.

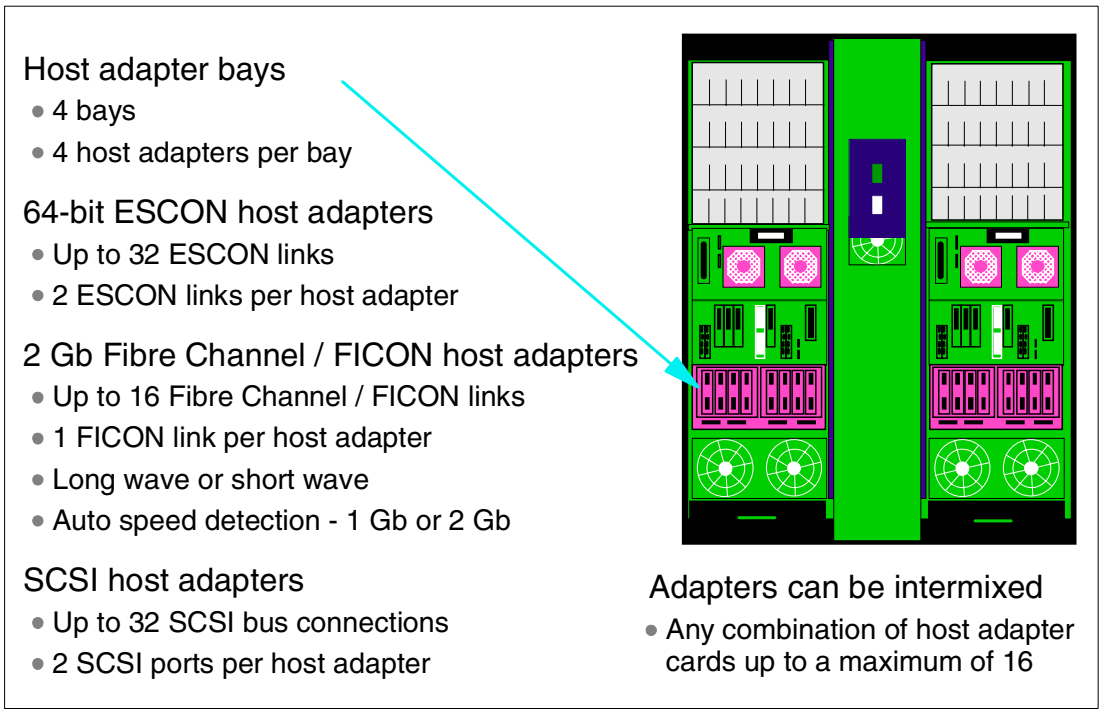


Figure 2-19 ESS Model 800 host adapters

The host adapter cards can either be ESCON, SCSI, or Fibre Channel/FICON (long wave or short wave) as presented in Figure 2-19 on page 36. The Fibre Channel/FICON host adapters can be configured as either FCP or FICON (one or the other but not simultaneously) on an adapter-by-adapter basis. The Fibre Channel/FICON card is a single port host adapter, whereas SCSI and ESCON have two ports for connection.

The host severs supported by the ESS for each host adapter interface can be found at:

http://www.storage.ibm.com/hardsoft/products/ess/supserver.htm

2.9.1 ESCON attachment

The ESS can connect up to 32 ESCON links, two per ESCON host adapter. Each ESCON host adapter is connected to both clusters. The ESS has 16 CKD logical subsystems (LSSs), each seen by the zSeries operating system as a 3990 logical control unit (LCU). Half of the LSSs (even numbered) are in the ESS cluster 1, and the other half (odd-numbered) are in the ESS cluster 2. Because the ESCON host adapters are connected to both clusters, each adapter can address all 16 LCUs.

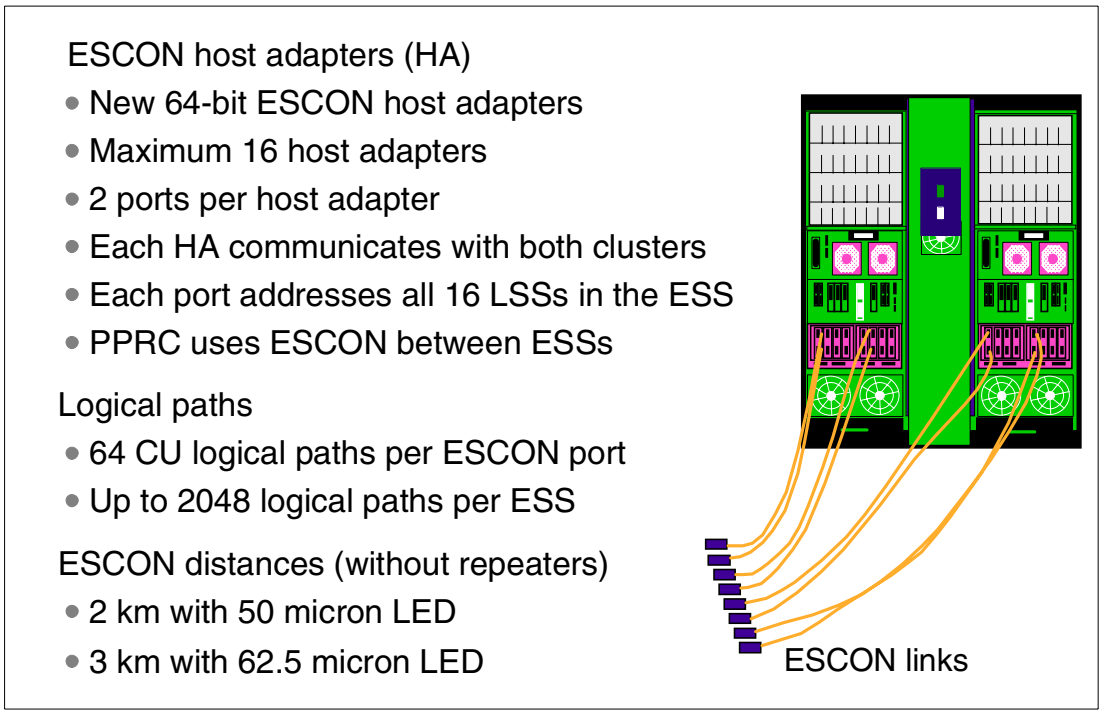


Figure 2-20 ESCON host adapters

64-bit ESCON host adapters

The ESS Model 800 can be configured with new 64-bit ESCON host adapters. The adapter has been enhanced with a faster microprocessor and offers up to a 45 percent improvement in full box sequential read bandwidth, and up to 10 percent increase in channel throughput for random operation workloads as compared to the 32-bit ESCON host adapter.

Logical paths

An ESCON link consists of two fibers, one for each direction, connected at each end by an ESCON connector to an ESCON port. Each ESCON adapter card supports two ESCON ports or links, and each link supports 64 logical paths. With the maximum of 32 ESCON ports, the number of logical paths per ESS is 2048.

PPRC and ESCON

The PPRC remote copy function uses ESCON connections between the two participating ESSs. For synchronous PPRC implementations, you can extend the distance at which you can operate the ESS up to 103 km. This distance requires at least two pairs of Dense Wavelength Division Multiplexers (DWDM) that can transport multiple protocols over the same fiber optic link. The link can be done using dark fiber.

Even greater distances can be achieved when using non-synchronous PPRC Extended Distance (PPRC-XD) and channel extender technology. The actual distance achieved is typically limited only by the capabilities of your network and the channel extension technologies.

When using the 64-bit ESCON host adapters in a PPRC configuration on both the primary and secondary ESS, up to a 10 percent increase in PPRC link throughput for random write operations and sequential bandwidth may be achieved as compared to the 32-bit ESCON host adapters.

Figure 2-20 on page 37 summarizes the characteristics of the ESCON host adapters.

2.9.2 SCSI attachment

The IBM TotalStorage Enterprise Storage Server Model 800 provides Ultra SCSI interface with SCSI-3 protocol and command set for attachment to open systems (refer to Figure 2-21). This interface also supports SCSI-2.

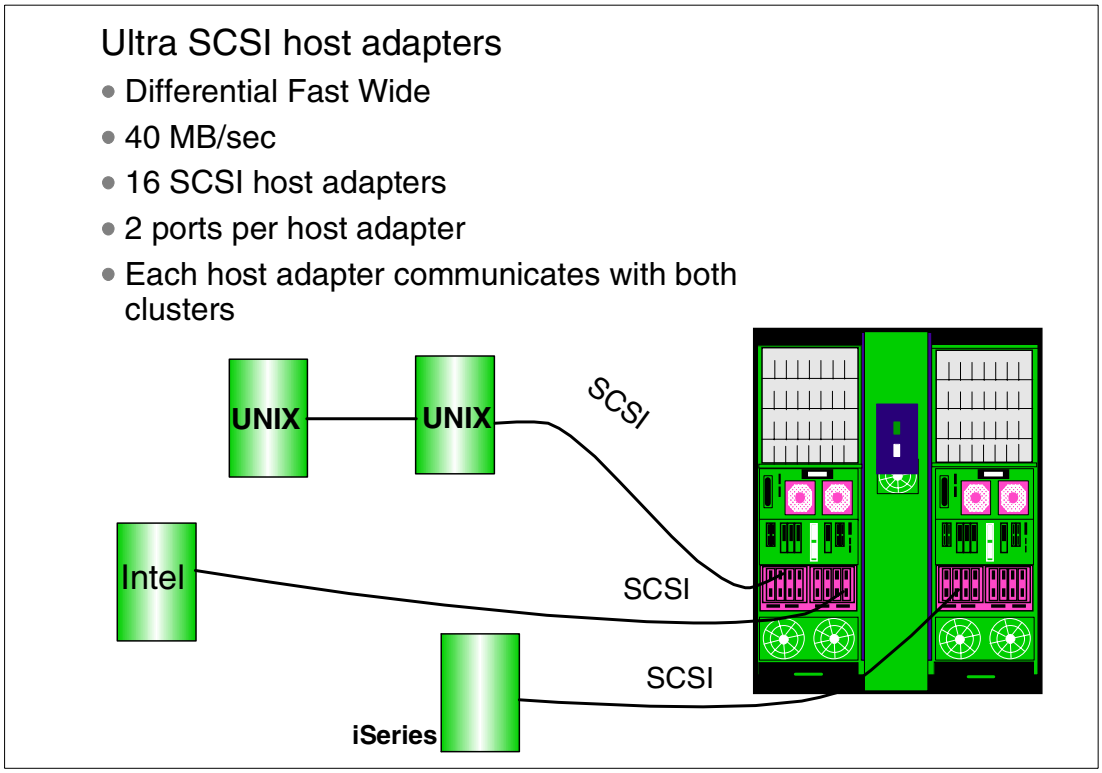


Figure 2-21 SCSI host adapters

Each SCSI host adapter supports two SCSI port interfaces. These interfaces are Wide Differential and use the Very High Density Connection Interface (VHDCI).

SCSI targets and LUNs

The ESS SCSI interface supports 16 target SCSI IDs (the host requires one initiator ID for itself, so this leaves 15 targets for the ESS definitions) with up to 64 logical unit numbers (LUNs) per target (the SCSI-3 standard). The number of LUNs actually supported by the host systems varies from 8 to 32, and it is a characteristic of the server. Check with your host server supplier on the number supported by any specific level of driver or machine.

2.9.3 FCP attachment

Fibre Channel is a technology standard that allows data to be transferred from one node to another at high speeds (up to 200 MB/s) and greater distances (up to 10 km).

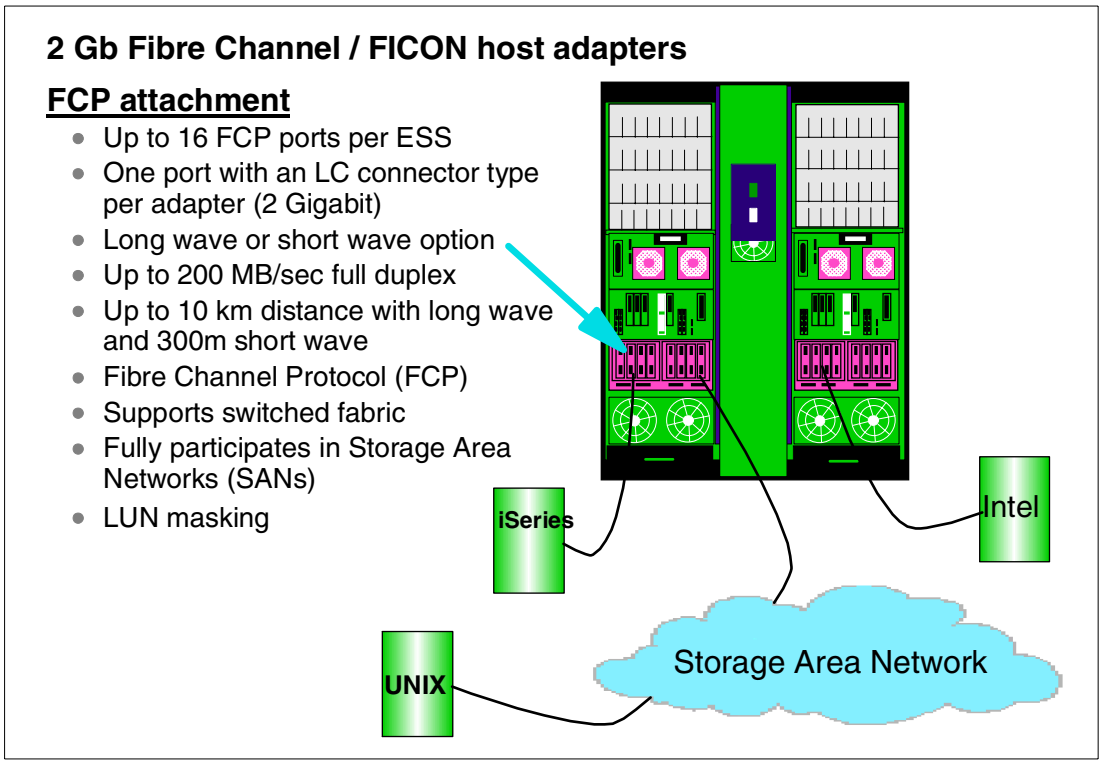


Figure 2-22 Fibre Channel/FICON host adapters - FCP attachment

It is the very rich connectivity options of the Fibre Channel technology that has resulted in the Storage Area Network (SAN) implementations. The limitations seen on SCSI in terms of distance, performance, addressability, and connectivity are overcome with Fibre Channel and SAN.

The ESS with its Fibre Channel/FICON host adapters provides Fibre Channel Protocol (FCP, which is SCSI traffic on a serial fiber implementation) interface, for attachment to open systems that use Fibre Channel adapters for their connectivity.

Note: The Fibre Channel/FICON host adapter is an adapter card that supports both FICON or FCP, but not simultaneously; the protocol to be used is configurable on an adapter-by-adapter basis.

The ESS supports up to 16 host adapters, which allows for a maximum of 16 FCP ports per ESS. Each Fibre Channel/FICON host adapter provides one port with an LC connector type.

There are cable options that can be ordered with the ESS to enable connection of the adapter port to an existing cable infrastructure.

As SANs migrate to 2 Gb technology, your storage should be able to exploit this bandwidth. The ESS Fibre Channel/FICON adapters operate at up to 2 Gb. The adapter auto-negotiates to either 2 Gb or 1 Gb link speed, and will operate at 1 Gb unless both ends of the link support 2 Gb operation.

There are two types of host adapter cards you can select: *Long wave* or *short wave*. With long-wave laser, you can connect nodes at distances of up to 10 km (non-repeated). With short-wave laser, you can connect at distances of up to 300 m. The distances can be extended if using a SAN fabric.

When equipped with the Fibre Channel/FICON host adapters, configured for FCP, the ESS can participate in all three topology implementations of Fibre Channel:

- Point-to-point
- Switched fabric
- Arbitrated loop

Figure 2-22 on page 39 summarizes the characteristics of the ESCON host adapters.

Fibre Channel distances

The type of ESS Fibre Channel/FICON host adapter, short wave or long wave, and the physical characteristics of the fiber used for the link will determine the maximum distances for connecting nodes to the ESS.

2.9.4 FICON attachment

Fiber Connection (FICON) is based on the standard Fibre Channel architecture, and therefore shares the attributes associated with Fibre Channel. This includes the common FC-0, FC-1, and FC-2 architectural layers; the 100 MB/s bidirectional (full-duplex) data transfer rate; and the point-to-point distance capability of 10 kilometers. The ESCON protocols have been mapped to the FC-4 layer, the Upper Level Protocol (ULP) layer, of the Fibre Channel architecture. All this provides a full-compatibility interface with previous S/390® software and puts the zSeries servers in the Fibre Channel industry standard.

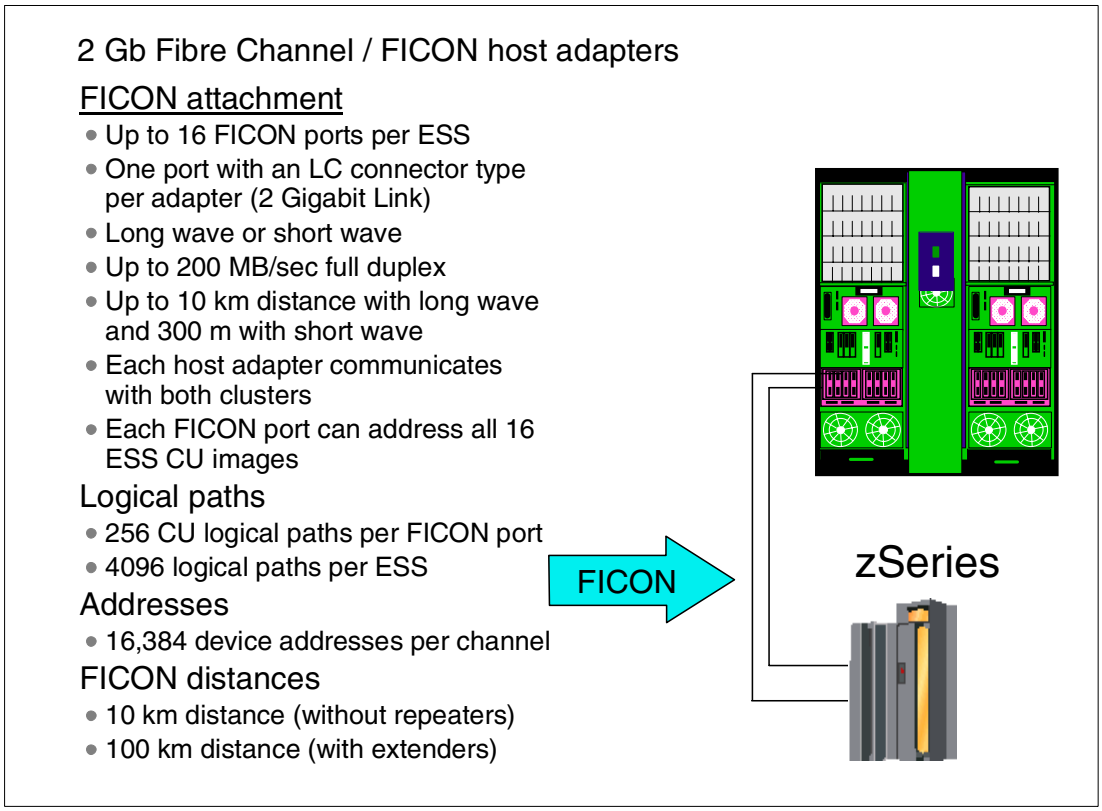


Figure 2-23 Fibre Channel/FICON host adapters - FICON attachment

FICON goes beyond ESCON limits:

- Addressing limit, from 1024 device addresses per channel to up to 16,384 (maximum of 4096 devices supported within one ESS).
- Up to 256 control unit logical paths per port.
- FICON channels allow multiple concurrent I/O connections (the ESCON channel supports only one I/O connection at one time).
- Greater channel and link bandwidth: FICON has up to 10 times the link bandwidth of ESCON and up to more than four times the effective channel bandwidth.
- FICON path consolidation using switched point-to-point topology.
- Greater non-repeated fiber link distances (from 3 km for ESCON to up to 10 km, or 20 km with an RPQ, for FICON).

These characteristics allow more powerful and simpler configurations. The ESS supports up to 16 Fibre Channel/FICON host adapters, which allows for a maximum of 16 FICON ports per machine.

Note: The Fibre Channel/FICON host adapter is an adapter card that supports both FICON or FCP, but not simultaneously; the protocol to be used is configurable on an adapter-by-adapter basis.

Each Fibre Channel/FICON host adapter provides one port with an LC connector type. The adapter is a 2 Gb card and provides a nominal 200 MB/s full-duplex data rate. The adapter will auto-negotiate between 1 Gb and 2 Gb, depending upon the speed of the connection at the other end of the link. For example, from the ESS to a switch/director, the FICON adapter

can negotiate up to 2 Gb if the switch/director also has 2 Gb support. The switch/director to host link can then negotiate at 1 Gb.

There are two types of host adapter cards you can select: *Long wave* and *short wave*. With long-wave laser, you can connect nodes at distances of up to 10 km (without repeaters). With short-wave laser, you can connect distances of up to 300 m.

Each Fibre Channel/FICON host adapter provides one port with an LC connector type. There are cable options that can be ordered with the ESS to enable connection of the adapter port to an existing cable infrastructure.

Figure 2-23 on page 41 summarizes the characteristics of the ESCON host adapters.

Topologies

When configured with the FICON attachment, the ESS can participate in point-to-point and switched topologies. The supported switch/directors for FICON connectivity can be found at:

`http://www.storage.ibm.com/disk/ess/supserver.htm`

2.9.5 Host adapters-server attachment

You must specify the type and number of host adapters when doing the ESS hardware configuration. The minimum is two host adapters of the same type. Later, once the ESS is installed, you can also add or replace host adapters. Note that the four bays are a standard part of the ESS independently of the number of host adapters included in the configuration.

The order in which the ESS host adapter cards are installed in the machine during the manufacturing process is:

1. Cluster 1 - Bay 1 - Adapter 1
2. Cluster 2 - Bay 4 - Adapter 1
3. Cluster 1 - Bay 2 - Adapter 1
4. Cluster 2 - Bay 3 - Adapter 1
5. Cluster 1 - Bay 1 - Adapter 2
6. Cluster 2 - Bay 4 - Adapter 2
7. Cluster 1 - Bay 2 - Adapter 2
8. Cluster 2 - Bay 3 - Adapter 2
9. Cluster 1 - Bay 1 - Adapter 3
10. And so on, until filling the 16 host adapter card positions across the four bays

In addition, the ESCON host adapters are installed first, then the long-wave Fibre Channel/FICON adapters, then the short-wave Fibre Channel/FICON adapters, and finally the SCSI adapters.

When planning the servers' attachments always spread the host connections across all the host adapter bays. Distribute the server connections to the host adapters across the bays in the following sequence: Bay 1 - Bay 4 - Bay 2 - Bay 3. This recommendation is for the following reasons:

- In normal operation the bays operate with only one of the clusters, and by spreading the connections across the bays, you also spread the workload among the clusters.
- If you need to replace or upgrade a host adapter in a bay, then you have to quiesce all the adapters in that bay. If you spread them evenly, then you will only have to quiesce a quarter of your adapters. For example, for an ESCON configuration with eight ESCON links spread across the four bays, then the loss of two ESCON links out of eight may have only a small impact, compared with four out of eight if they were all installed in one bay.

2.10 Host adapters configuration

This section presents results of workload runs using different host adapters configurations, as they can illustrate its implication on the overall ESS performance. Also discussed are performance recommendations and considerations helpful when planning the ESS host adapters hardware configuration.

When planning the configuration of the ESS host adapters that you will need, consider the following:

1. Make a list of all the servers that will be connected to the ESS, documenting the amount of storage capacity they will be using, and the type of I/O workload they will execute.
2. For the zSeries servers document the type and number of channels you have available for ESS connection. Channels can be ESCON or FICON. FICON surpasses ESCON performance and connectivity capabilities, so it is the recommended option.
3. For the open systems servers you should also document the type and number of host bus adapters (HBA) you have available for ESS connection. Host bus adapters can be SCSI or Fibre Channel. Fibre Channel surpasses SCSI performance and connectivity capabilities, so it is the recommended option.
4. For FICON and Fibre Channel connectivity you have the additional consideration of the speed of the connecting ports. 2 Gb should be the option for the servers' HBAs and SAN switches that connect to the 2 Gb Fibre Channel/FICON host adapter ports of the ESS. For ESCON, the new 64-bit ESCON host adapters should be the option for better performance.
5. For all types of servers, the more ESS host adapters they connect the more I/O bandwidth they will have available for their I/O processing. You must evaluate which is the optimum number for each application/server, beyond which the significance of adding more host attachment connections is negligible from a performance point of view.

Once the necessary information is ready, then Disk Magic can be run to evaluate the alternatives for the ESS hardware configuration. As with other components of the ESS hardware configuration, consider that the ESS *host adapters* are not an isolated factor when estimating the overall ESS performance. But instead, they must be considered together with other important factors like the I/O workload characteristics, the cache size, the ESS processors (Turbo or Standard), the disk drives capacity and speed, and the backend data layout and SSA loops.

In the following sections we present examples of ESS measurement results when using different ESS host adapter configurations. These examples and discussions will help you better understand what the performance implications of different possible ESS host adapters configurations are.

2.10.1 ESCON and FICON attachment

Figure 2-24 on page 44 illustrates examples of high levels of I/O throughput that are possible when using different numbers of 64-bit ESCON host adapters, and the workload is cache friendly—*cache hit reads*.

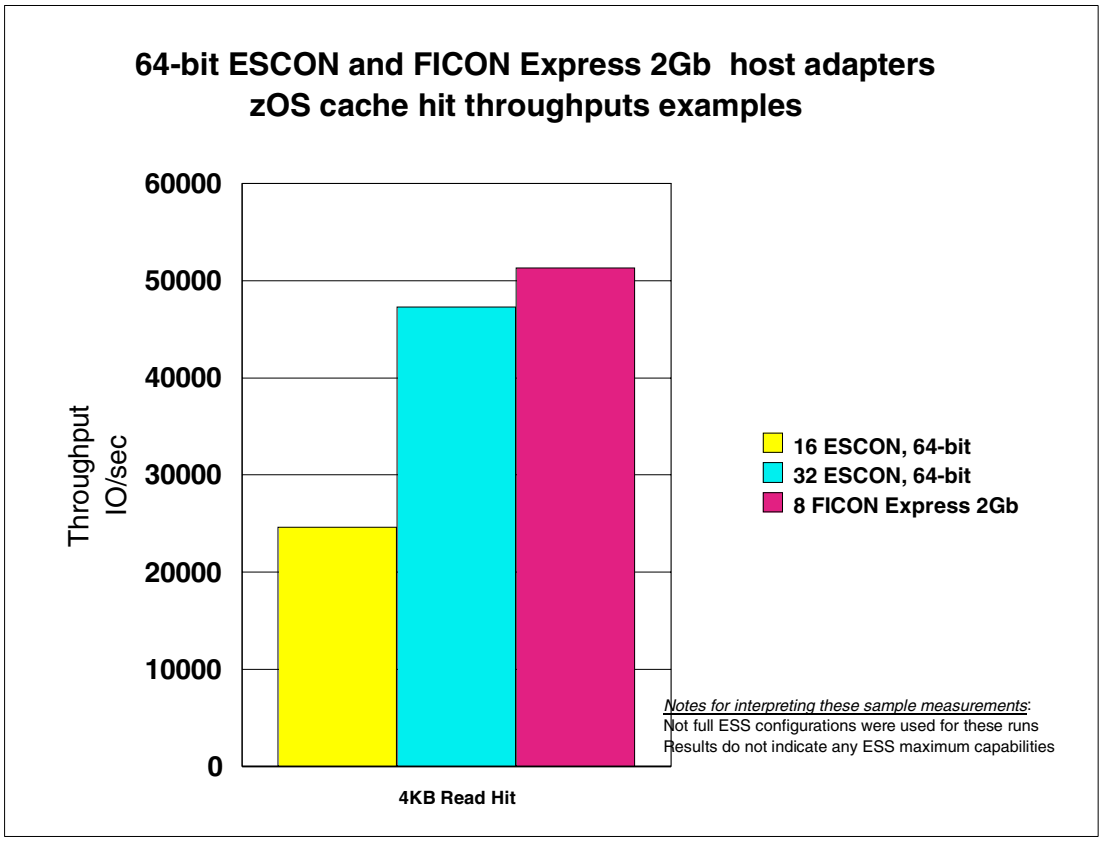


Figure 2-24 64-bit ESCON and FICON Express 2 Gb host adapters - 4 KB read hits

Figure 2-25 on page 45 illustrates examples of throughput results for a *sequential read* workload, when using different numbers of 64-bit ESCON host adapters.

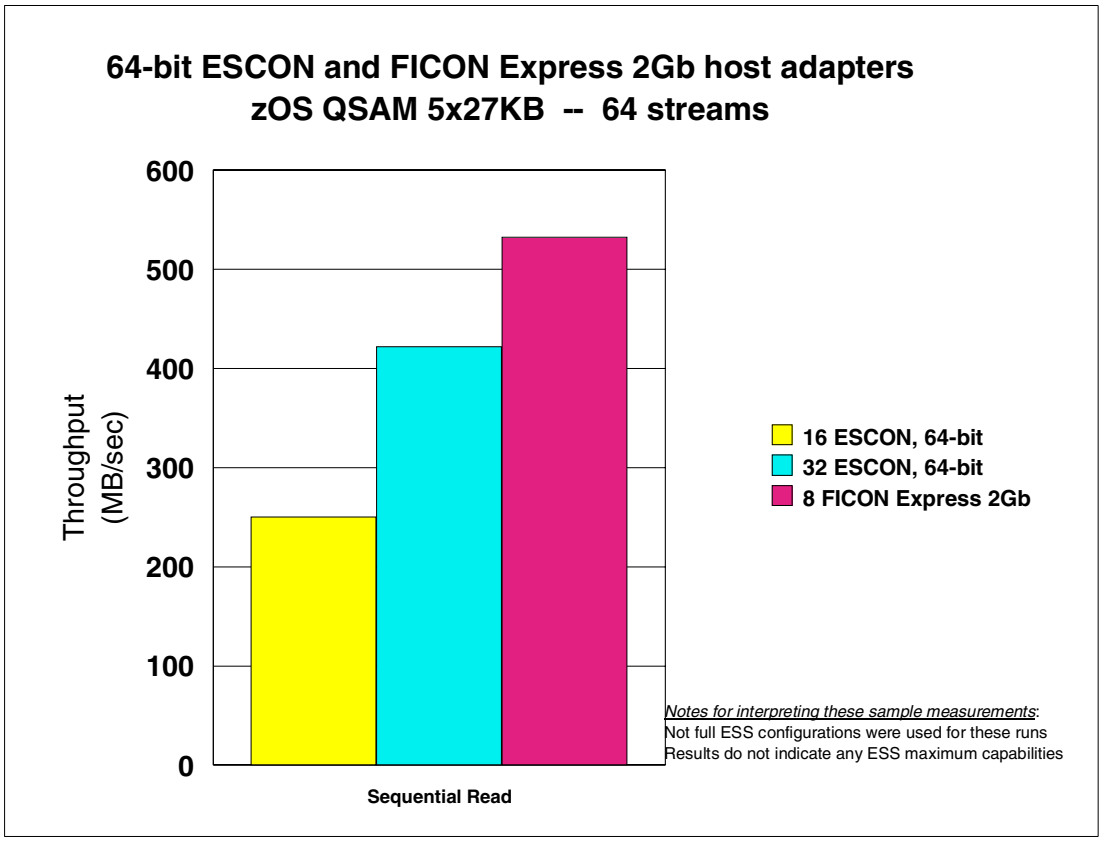


Figure 2-25 64-bit ESCON and FICON Express 2 Gb host adapters - Sequential reads

For the 64-bit ESCON host adapters you can expect a transfer rate of around 16 MB per second per ESCON port—for read workloads. If the ESCON links will be used for PPRC PPRC remote copy, you must estimate the number of the ESCON links required based on the number of writes.

2.10.2 SCSI attachment

When attaching SCSI servers, the following sustained data transfer rates can be used to determine the number of SCSI ports that will be required:

- 30 MB per second for Ultra SCSI ports
- 15 MB per second for SCSI Fast/Wide ports

We recommend configuring the maximum number of SCSI host adapters to achieve optimum performance and connectivity. At least one SCSI host adapter is required for each attaching server and two or more if Subsystem Device Driver (SDD) is used. When planning to use the multi-pathing facilities of SDD you should properly evaluate the optimal number of paths per vpath, as SDD operation has some associated overhead.

Daisy chaining is not beneficial from a performance perspective:

- Daisy-chained servers must arbitrate for control of the SCSI bus according to strict SCSI target hierarchy.
- One daisy chained server could monopolize the SCSI bus.

Figure 2-26 on page 46 gives an example of throughput results attainable on an Ultra SCSI port.

2.10.3 FCP attachment

When attaching FCP servers, the following sustained data transfer rates can be used to determine the number of FCP ports that will be required:

- 73 MB per second for 1 Gb Fibre Channel ports
- 145 MB per second for 2 Gb Fibre Channel ports

Figure 2-26 gives an example of throughput results attainable on a 1 Gb FCP port in an ESS model F20, and on a 2 Gb FCP port in an ESS Model 800.

To define the number of ESS host adapters you must know your servers peak workload and also understand if the peak hours are the same for all these servers. If you do not have a good understanding of the workload, be conservative and plan for the peak hours of all your servers together.

For SAN implementations, throughput among the components of the SNA network should be considered, ensuring that the fastest links are implemented among all the them. Also enough FCP ports from the ESS must be available to cope with the data transfer rate coming from all the interconnected servers that have LUNs defined on the ESS.

An example could be that you have to plan for 12 servers to connect to your ESS using two switches, and all the SNA components have ports of 2 Gbps Fibre Channel technology. For high availability all servers will have two paths to the ESS. Assuming that the aggregate peak I/O workload of the 12 open servers is 400 MB/second, you will need at least three Fibre Channel host adapters in the ESS Model 800. But the recommendation would be to configure four Fibre Channel host adapters. This way you will be evenly balancing the workload between the two switches, and you will also have a high-availability configuration.

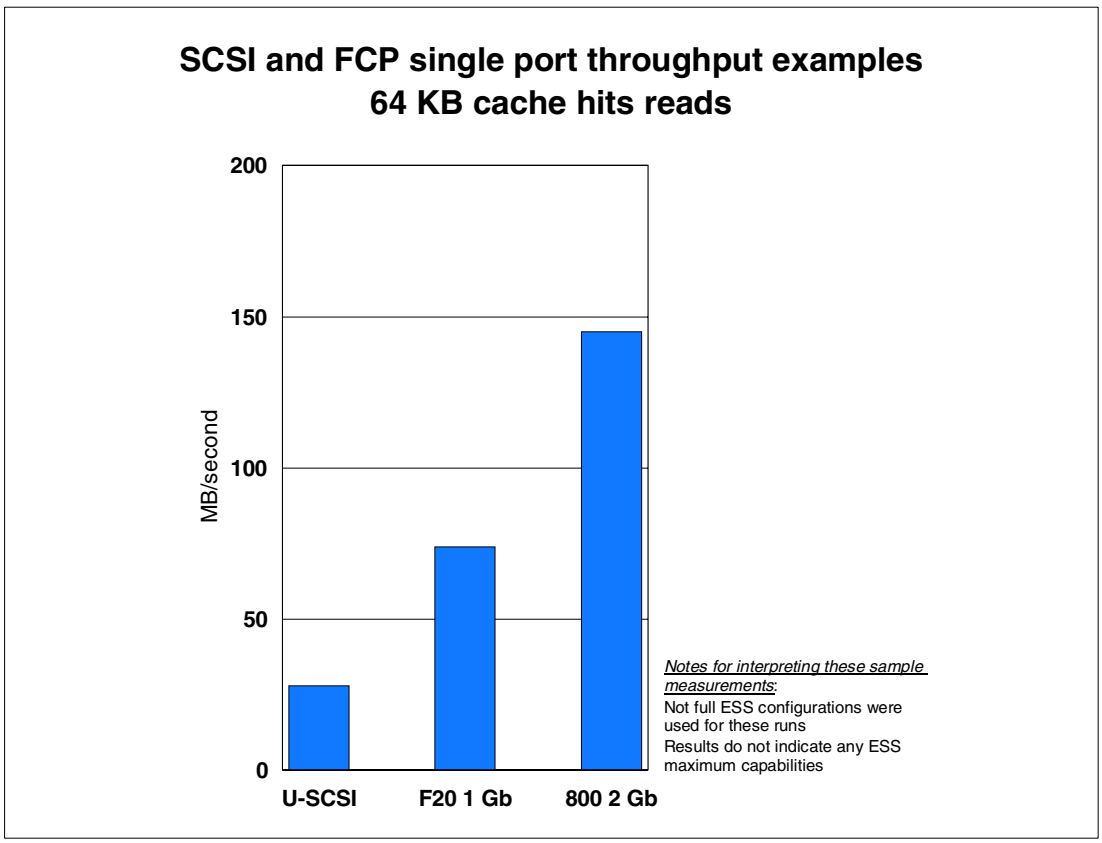


Figure 2-26 SCSI and FCP single port throughputs

2.10.4 FICON attachment

When attaching FICON servers, the following sustained data transfer rates can be used to determine the number of FICON ports that will be required:

- 75 MB per second for 1 Gb FICON ports
- 135 MB per second for 2 Gb FICON ports

Plan for peak hours, also plan for high availability and for keeping the aggregate transfer rate if a path is down.

If migrating to FICON, consider that depending on channel utilization 1 FICON port can consolidate a different number of ESCON ports:

- 6 ESCON for *low* utilization channels
- 4 ESCON for *average* utilization channels
- 2 ESCON for *high* utilization channels

When possible make FICON your choice. As mentioned before, FICON provides greater effective throughput compared to ESCON adapters and allows for simpler configurations.

Figure 2-24 on page 44 illustrates examples of the very high levels of I/O throughput that are possible when using different numbers of 2 Gb FICON host adapters, and the workload is cache friendly—*cache hit reads*. Figure 2-25 on page 45 illustrates examples of throughput results for a *sequential read* workload, when using different numbers of 2 Gb FICON host adapters.

3

Logical configuration planning

In this chapter we discuss considerations for an optimal *logical configuration* of the ESS, so that its performance and capacity characteristics can be efficiently exploited. Also in this chapter we review the ESS terminology that is necessary for describing the logical configuration.

This chapter discusses:

- The components and terminology involved in the logical configuration procedure
- Configuring the right number of spares
- Sizing and placing of logical disks
- Configuring RAID-5 and RAID-10 ranks
- Striping at the OS level

3.1 ESS logical configuration - Components and terminology

The building blocks of the ESS disk storage capacity are the *eight-packs*. Each eight-pack contains eight hard disk drives, also referred to as disk drive modules (DDMs), of the same capacity and speed. The ESS contains four Serial Storage Architecture (SSA) *device adapter* pairs (DA pairs), which total eight SSA *loops*. Each SSA loop can contain a maximum of six eight-packs. Figure 3-1 illustrates a full capacity ESS with the complete 48 eight-packs.

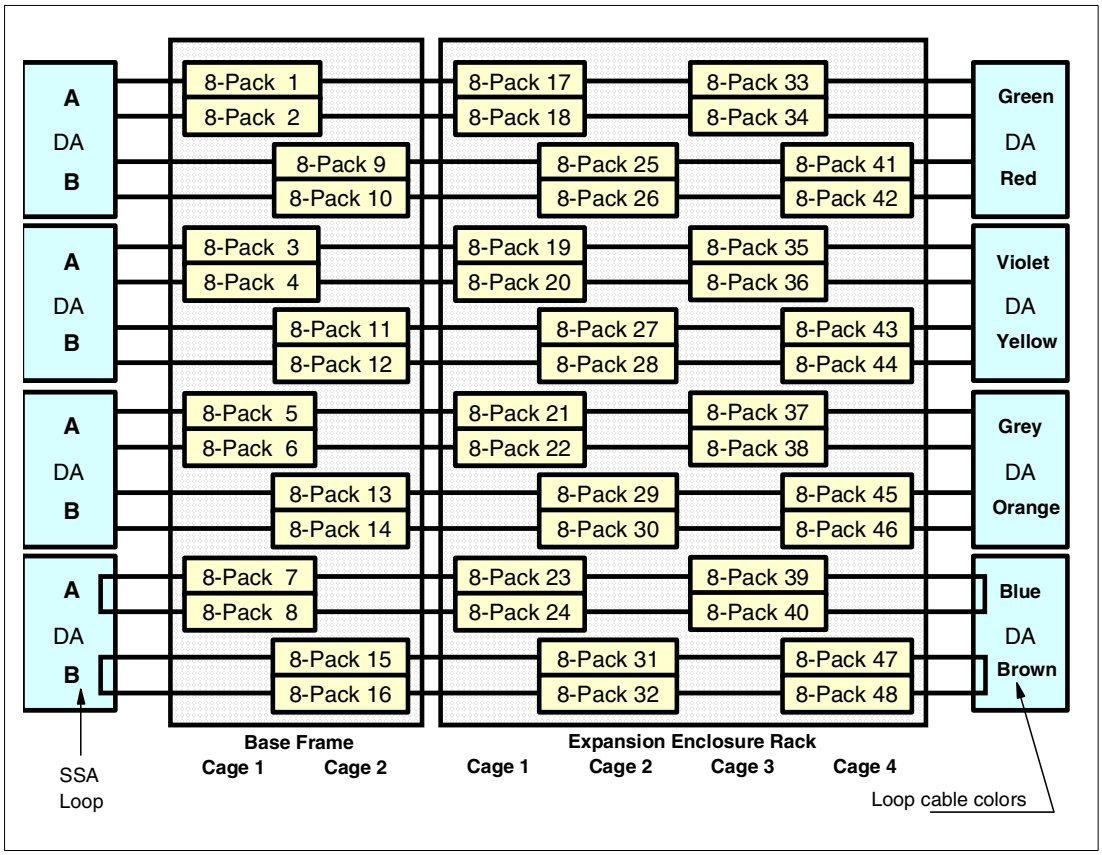


Figure 3-1 Fully configured ESS with 48 eight-packs and Expansion Enclosure

The full configuration illustrated in Figure 3-1 uses the ESS Expansion Frame to hold the eight-packs 17 to 48.

3.1.1 Eight-packs and disk drives

An *eight-pack* is a set of eight *disk drives*. All eight disk drives in an eight-pack are of the same capacity and speed. Disks are added to the ESS in pairs of eight-packs. Eight-packs are explained in detail in 2.6.2, “Disk eight-packs” on page 24.

The ESS Model 800 can contain up to 384 *disks drives* of different capacities. The ESS Model 800 supports 18.2 GB, 36.4 GB, 72.8 GB, and 145.6 GB capacity disk drives. The same disk technology and capacities are available for all ESS attachable servers—SCSI, FCP, ESCON, and FICON attachment. The ESS eight-packs that hold the disk drives are installed in the ESS *base frame* and—if needed—an *expansion rack* is used (as Figure 3-1 illustrates). The base frame of the ESS can hold up to 128 disk drives—in up to 16 eight-packs. The expansion frame can hold up to 256 disk drives—in up to 32 eight-packs.

3.1.2 SSA device adapters

The ESS Model 800 internally uses the Serial Storage Architecture (SSA) protocol for performing disk I/O operations and includes new, more powerful SSA *device adapters* (DAs, illustrated in Figure 3-1 on page 50) to further improve the backend disk performance. The SSA device adapters manage the SSA loops (A and B, as illustrated in Figure 3-1 on page 50) and perform all the RAID operations for the loops, including *parity generation* and *striping* for RAID-5, and *mirroring* and *striping* for RAID-10. This is done by the device adapters together with the data transfer reads and writes, as well as any disk sparing activity if needed.

Note: The SSA *device adapters* have on-board cache memory to hold data and effectively offload the RAID functions from the ESS clusters. No parity processing is done by the cluster processors or cache.

Whenever *sparing* is needed—the recovery of a failed disk drive onto one of the *spare* disk drives—it is also handled automatically by the SSA device adapter. The *sparing* process takes place in the background over a period of time, thus minimizing its impact on normal I/O operations. In fact, the sparing process dynamically adjusts its rate, slowing down when application I/O processing to the array increases and subsequently taking advantage when application I/O operations decrease. Then the failed disk drive can immediately be replaced, and this replacement disk drive automatically becomes the new spare—*floating spare*.

3.1.3 Arrays, ranks, and disk groups

The ESS Specialist presents disks from two physical eight-packs of the same loop as a logical *disk group*. A *rank* results from the ESS Specialist formatting of a disk group as either a RAID-5 or RAID-10 array. The terms rank and array are used interchangeably. These components and terms are illustrated in Figure 3-2 on page 52.

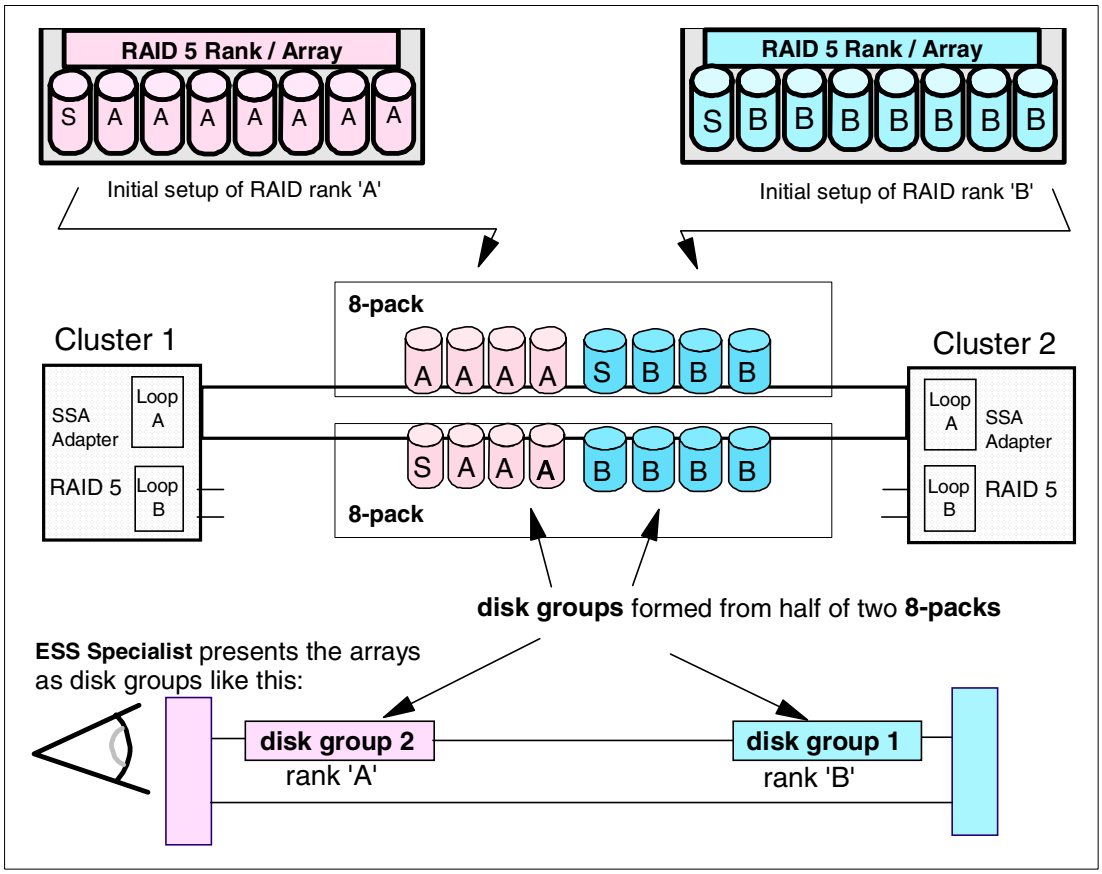


Figure 3-2 Eight-packs, ranks, arrays, and disk groups

Having performance in mind, RAID *ranks* are organized by grouping DDMs from two different eight-packs into a *disk group*, as Figure 3-2 illustrates. This allows the SSA adapters to achieve maximum throughput for each rank by having a path to each half of the rank down each leg of the loop. The ESS Model 800 arrays can be formatted as either RAID-5 or RAID-10 ranks, even on the same SSA loop.

Do not confuse eight-packs (the physical packaging) with arrays. Arrays typically span two eight-packs, but could span more than two eight-packs after a sparing operation.

3.1.4 ESS storage allocation - Logical disks

In the ESS, storage is allocated as *logical devices* (or *disks*) that appear as 3390 (or 3380) devices to the zSeries servers, and as *physical disks* to the open systems servers. The ESS Specialist refers to these logical disks as *logical volumes*. For the zSeries the terms *device*, *DASD*, and *volume* are used interchangeably. For the open systems the terms *logical disk*, *LUN*, *volume*, and *logical volume* are sometimes used interchangeably. This is a little bit confusing indeed.

For clarity, we use the term *logical device* (or *disk*) when generically referring to the unit of ESS-allocated storage as it is presented to the operating system server for its addressing. When speaking of open systems, we use the term *logical volume* when discussing the operating system commands used to map storage on top of the *logical disks* in order to create file systems. When speaking of zSeries servers, we use the terms *logical volume* and *logical device* interchangeably.

In the ESS, *logical disk* storage is allocated as a series of *hardware stripes* across the physical *disk drives* of a RAID *rank* (or *array*), and presented to the host operating system as a *physical device* —but in fact you can see that this is a *logical device*. An example of storage allocation in the ESS is illustrated in Figure 3-3.

Logical disks are always created by *striping* across an array. This means that a section of logical disk space, called a *strip*, is claimed first from one physical disk drive, then the next, then the next, in a round-robin fashion until a *stripe* is formed. For example, the ESS *strips* for FB devices (non S/390) are 32 kilobytes in size, so when data is read from or written to a logical disk, each 32 kilobytes transferred travels to or from a different physical disk drive. If the RAID array is RAID-5 (7+p), then each *stripe* will contain eight *strips* making 8*32 KB = 256 KB in size. A logical device is formed from multiple *stripes* across the disks drives that make the RAID array.

Logical disks are striped automatically by the ESS hardware, across the physical disk drives of an array. Logical disks *do not span* multiple ranks.

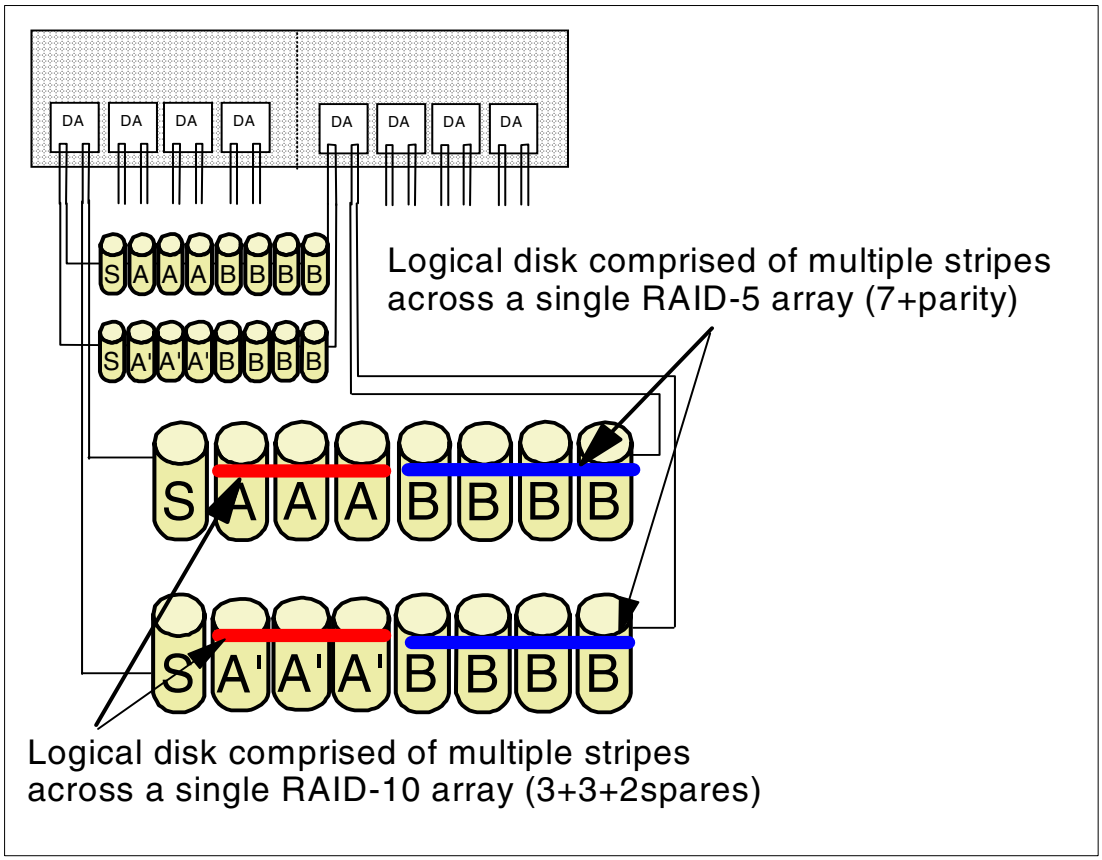


Figure 3-3 Logical disks within RAID-10 and RAID-5 arrays

3.1.5 Fixed Block and count-key data

The ESS arrays can be formatted into either CKD or FB format. We will see the characteristics of these two different kinds of formatting.

CKD

Count-key data (CKD) is the disk architecture used by zSeries (and S/390) servers. In this data organization, the *data* field stores the user data. Also, because the data records can be variable in length, they all have an associated *count* field that indicates the user data record

size. Then the *key* field is used to enable a hardware search based on a key. However, this is not generally used for most data anymore. ECKD™ is a more recent version of CKD that uses an enhanced S/390 channel command set.

Fixed Block architecture (FBA or FB)

The *Fixed Block architecture* is used for servers that attach to the ESS via SCSI or Fibre Channel (FCP). In the FB architecture, the data (hence the logical disks) is mapped over fixed-size blocks or sectors. The location of any block can be calculated to retrieve that block. The concept of tracks and cylinders also exists, because on a physical disk we have multiple blocks per track, and a cylinder is the group of tracks that exists under the disk heads at one point in time without doing a seek.

3.1.6 Logical subsystems (LSSs)

The *logical subsystem* (LSS) is a logical structure that is internal to the ESS. It is a logical construct that groups up to 256 logical disks of the same disk format (either CKD or FB) and is identified by the ESS with a unique ID. For the ESCON and FICON architectures the LSS relates directly to the *logical control unit* (LCU) concept of S/390.

The count-key data (CKD) *logical subsystems* are configured with the ESS Specialist at the time of configuring the *logical control units* (LCUs) at S/390 storage allocation time. The fixed block (FB) *logical subsystems* are configured by the ESS Specialist at the time of allocating the FB logical disks of the open systems. In PPRC environments, the *logical subsystems* are used for managing and establishing PPRC relationships. The relationship between the SSA *device adapters* (DAs) and the LSSs is predefined as shown in Figure 3-4.

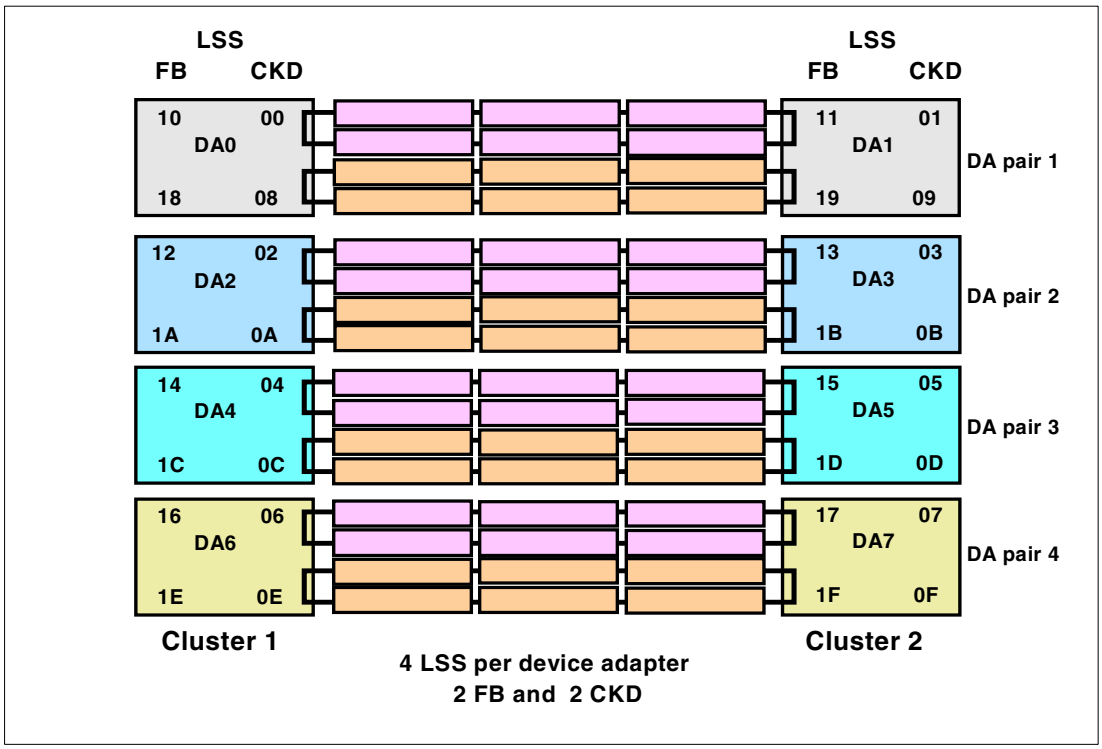


Figure 3-4 Mapping of LSSs to device adapters

Each SSA *device adapter* (DA) supports up to four LSSs—two CKD and two FB (refer to Figure 3-4). Also each DA connects two loops, each of which can be mapped by the four LSSs. You can chose whether to have eight or sixteen CKD LSSs, and also either eight or

sixteen FB LSSs. The LSS numbers are pre-defined (see Figure 3-4 on page 54). CKD subsystems are numbered from x'00' to x'0F', and FB logical subsystems from x'10' to x'1F'. If you chose only eight LSSs, only the first LSS number for that DA is generated. For example, DA0 will have only FB LSS 10.

During the initial installation of the ESS, the IBM Systems Support Representative will set the logical subsystem upper limit to 0, 8, or 16 for each type of LSS. If you do not plan to use more than eight LSSs, setting the upper limit to less than 16 will save storage in cache, up to 2 MB per LSS. However, be aware that adding more LSSs or types of LSSs later will require a IML of the ESS.

3.2 Optimizing storage allocation

Every SSA loop must have at least *two spare disks* for *each different drive capacity* installed in the loop. When ordering a new ESS Model 800 with mixed capacity drives, eight-packs will be installed in sequence from highest to lowest capacity.

When you are installing an ESS with heterogeneous capacity disks, or adding new disks of a different capacity—that is, when you are getting a *mixed capacity* configuration—the two (minimum) spare disks per capacity rule must me observed. But you will also want to format the disk groups so as to *minimize the number of spares*, as well as *balance the spares' distribution* across clusters, SSA loops, and LSSs. For a further discussion on sparing refer to the publication *IBM TotalStorage Enterprise Storage Server Model 800,* SG24-6424.

3.2.1 Minimizing number of spares

On a given loop there can be RAID-5 and RAID-10 ranks, as well as eight-packs that differ in their disk drive capacity and speed. When installing eight-packs of different capacity and speed certain rules apply, as explained in 2.6, "ESS disks" on page 23.

Each disk group is formatted independently within a loop to form either a RAID-5 or RAID-10 rank. As such, it is possible to define within an eight-pack *pair* both a RAID-5 rank and a RAID-10 rank. In this case, the sequence in which the two ranks are formatted will determine the number of spare DDMs allocated—assuming these are the first two eight-packs of a given capacity formatted within the loop.

For a given capacity, either the first two ranks that are configured (if RAID-5) will contain spares, or the very first rank that is configured (if RAID-10) will hold both spares. As Figure 3-5 on page 56 shows, formatting the RAID-10 array first will be done as 3+3+2S, which provides the required two spares for the loop. Hence the RAID-5 array is formatted as 7+P.

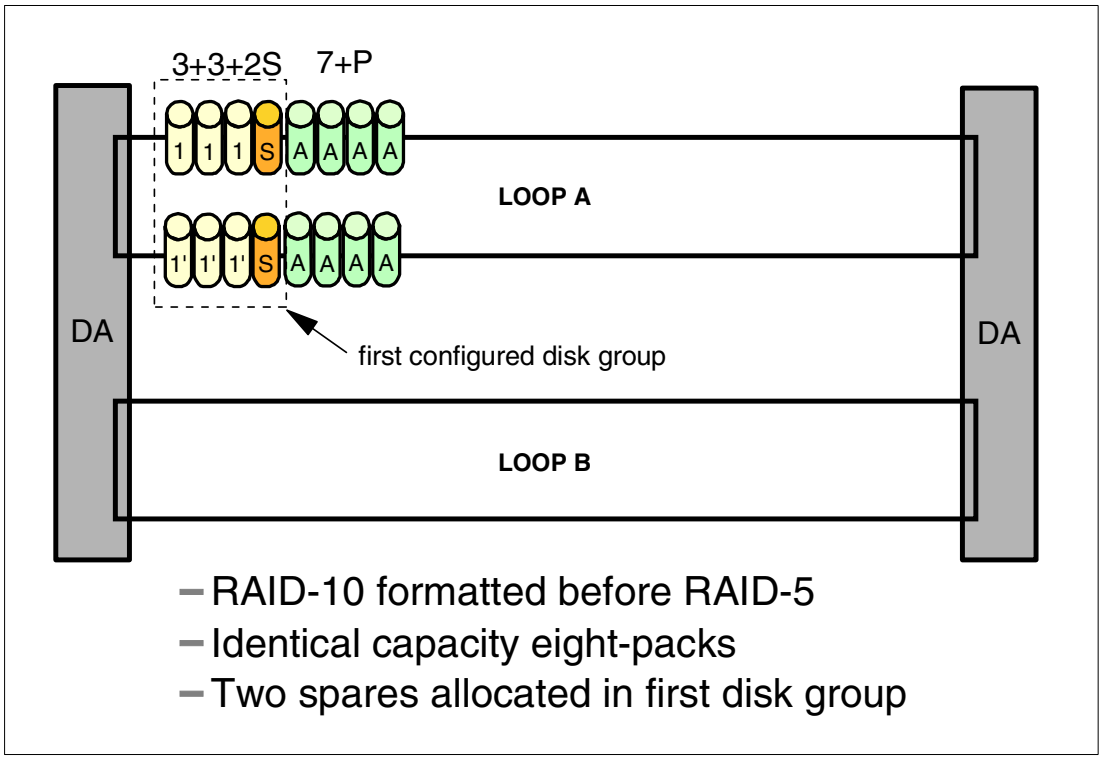


Figure 3-5 RAID-10 followed by RAID-5 formatting

Figure 3-6 shows that formatting the RAID-5 array first will be done as 6+P+S, which only provides one spare, and hence the RAID-10 array has to be formatted as 3+3+2S. This results in three spare DDMs in the loop.

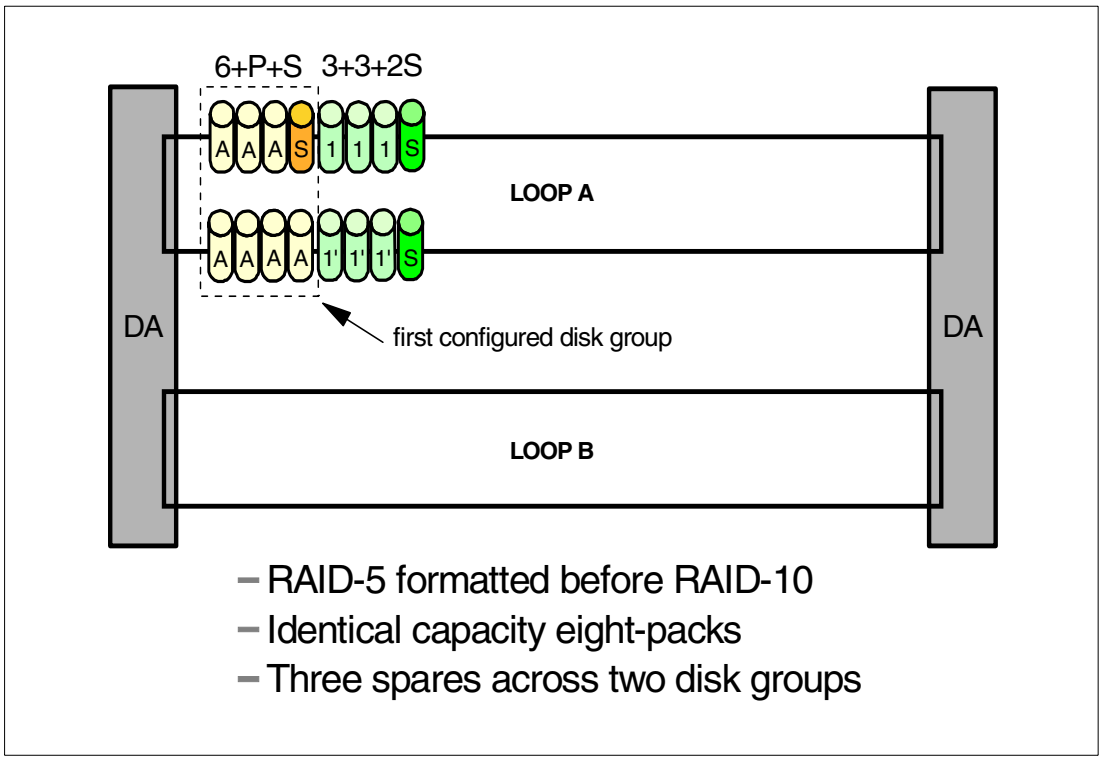


Figure 3-6 RAID-5 followed by RAID-10 formatting

Tip: To avoid *wasting* a DDM as an extra spare, consider carefully the sequence in which you format the arrays. Configure the RAID-10 array before the RAID-5 array whenever you are having a situation similar to this example.

An IBM Field Technical Sales Specialist or IBM Business Partner can use the IBM utility Capacity Magic to see the spares that result depending on the rank formatting sequence. Information on Capacity Magic is available in 4.3, “Capacity Magic” on page 98.

This discussion on the formatting sequence effect applies to each set of different eight-pack capacities installed in a loop.

3.2.2 Balancing logical subsystems

Normally, you will want to balance storage allocation within an ESS across LSSs—thus across the clusters. In this example we use RAID-10 ranks, but the same principle applies to RAID-5 arrays also.

Since the first RAID-10 rank in a loop will provide the required two spare DDMs (3+3+2S), subsequent RAID-10 arrays in the same loop will be 4+4. This means the first rank will have three DDMs for data and the others will have four DDMs for data. This can result in an *unbalance* of capacity allocated among LSSs for some loop configurations.

Unbalanced LSSs

Figure 3-7 on page 58 shows the potential scenario, where the disk groups are allocated in the numbered sequence. LSS 0 has ended up with two arrays of 3+3+2S, while LSS 1 has two arrays of 4+4, and hence has more usable capacity.

If all eight SSA loops in the ESS were configured this same way, Cluster 2 would be handling more capacity (16 times the DDM capacity) since all 16 spares (2 spares/loop * 8 loops) would be in arrays controlled by Cluster 1. For an uniform access density (I/O/sec/GB), this will result in more I/O activity rate on Cluster 1 as compared to Cluster 2.

Balancing I/O across clusters is important because cache and NVS are split between each cluster. By balancing I/O, you will benefit from the combined processing capabilities of both clusters.

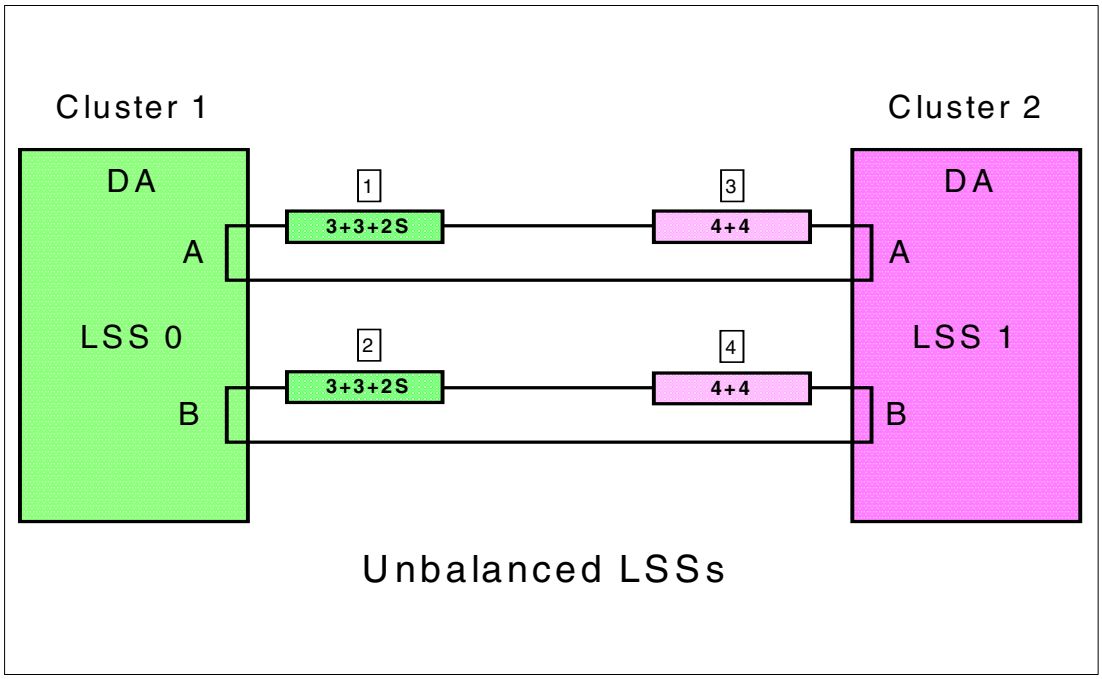


Figure 3-7 Unbalanced LSSs

Balanced LSSs

To balance the allocated capacity among the LSSs, allocate an initial RAID-10 array to each LSS within the DA pair from different loops before allocating subsequent arrays to an LSS, using the following process (refer to Figure 3-8 on page 59):

1. Add the first array for LSS 0 from loop A. The ESS creates the array as 3+3+2 (three pairs of mirrored drives and two spares).
2. Add the first array for LSS 1 from loop B. The ESS creates the array as 3+3+2.
3. Add the second array for LSS 0 from loop B. The ESS creates the array as 4+4 (four pairs of mirrored drives).
4. Add the second array for LSS 1 from loop A. The ESS creates the array as 4+4.

This assumes that you only have one LSS per cluster in the DA pair. If you have four LSSs per DA pair, then you should still ensure that capacity is spread evenly between clusters and loops, but it will not be possible to have the same capacity in all four LSSs since only two of the RAID-10 disk groups will be 3+3.

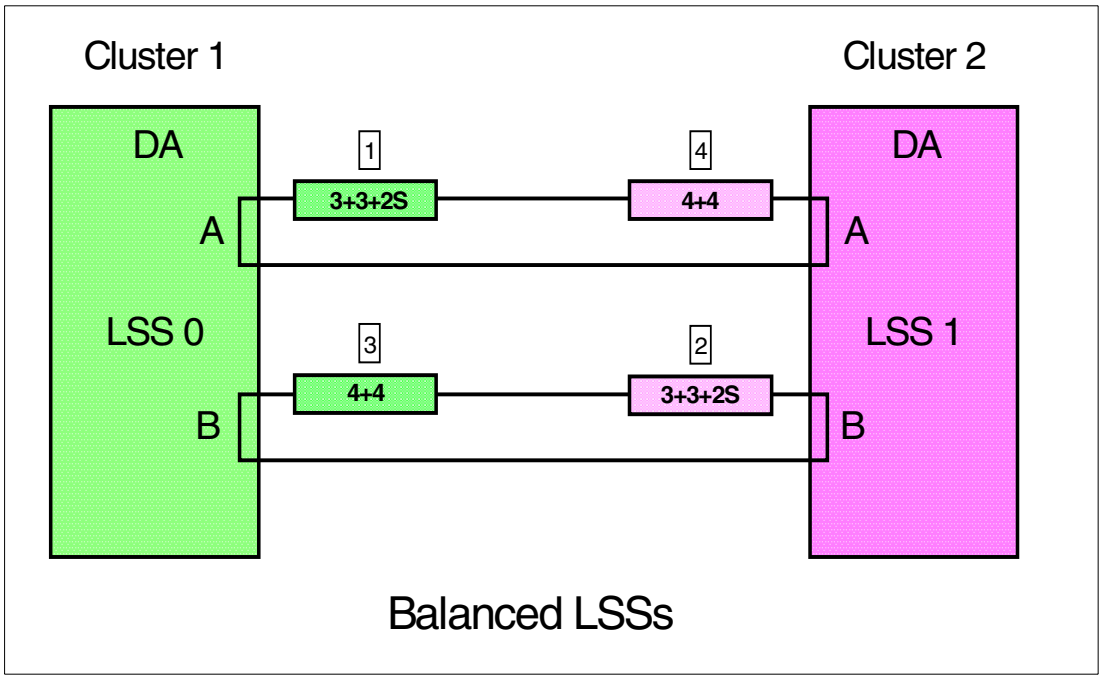


Figure 3-8 Balanced LSSs

3.3 Logical disks - Number and size

Generally, for easier administration and better *overall* performance, we recommend *spreading everything across everything*; meaning spread I/O across as many ranks, SSA loops and adapters, and clusters as possible. This will allow you benefits from the aggregated throughput provided by more SSA loops and arrays. It is also important to remember that each cluster has 1 GB of NVS and half of the cache. Since all writes must go through NVS and cache, you will want to balance I/O across both clusters.

The way to spread I/O is by assigning to servers, logical disks that have been created on several different ranks on different loops and clusters within the ESS.

Sometimes though, you may want to dedicate an array or all the arrays in a SSA loop for a given server or application. The overall I/O performance in that case may not be as great as spreading I/O across all ESS components, but will be predictable—especially for the application (or server) whose storage is isolated.

The ESS is very good at detecting I/O patterns. So if your environment does a lot of large sequential file copying from A to B, you might want to split A-reads and B-writes to different ranks. Let the reads come from logical disks on one group of ESS ranks, and the writes go to a separate set of ESS ranks.

The ESS is very good at detecting sequential I/O and adjusting I/O requirements accordingly; however, avoiding large reads and writes to the same ranks will improve performance.

It is a challenge to select logical disk sizes in a manner that:

- Allows you to spread I/O across multiple ESS ranks
- Does not proliferate the number of disk devices presented to a host

- Allows enough granularity for performance monitoring but not that much that analyzing the performance data gets too complex—for example, `iostats` becoming overwhelming.
- Allows for growth and re-assignment of logical disks from one host to another—when working with open systems
- Allows for growth and expansion of data sets (avoiding out-of-space cancellations)—when working with zSeries servers

Tip: Try to strike a reasonable balance between flexibility and manageability for your needs.

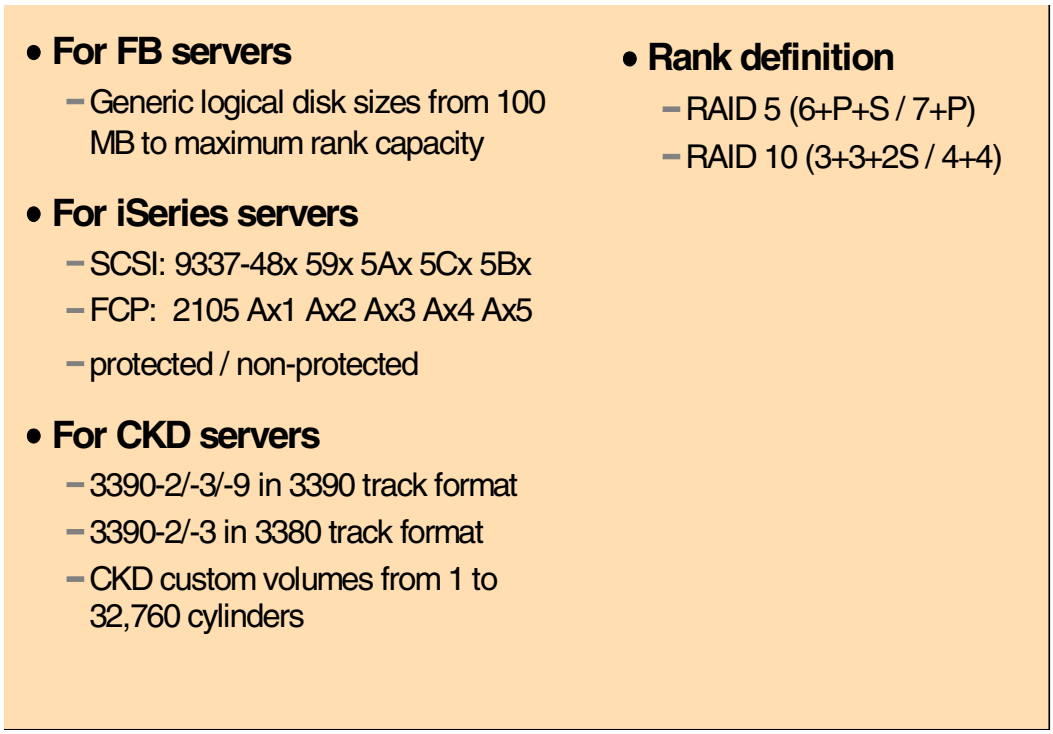


Figure 3-9 Logical disks configuration options

You can realize that the ESS gives you great flexibility when it comes to allocating disk space, as shown in Figure 3-9.

For FB ranks, logical disk sizes can vary from 100 MB to the full effective capacity of the rank—in increments of 100 MB. An ESS Model 800 rank of 145.6 GB disk drives has a full effective capacity of 982 GB—when configured as a 7+P RAID-5 rank.

For CKD servers, logical disk sizes can vary from a 1 cylinder 3390 device (that is, 849.9 KB), to a 32 K cylinder 3390 device (that is, 27.8 GB). For iSeries™ servers, there are five different logical disk sizes supported, which vary by attachment type.

SAN implementations

In a Storage Area Network (SAN) implementation, care must be take in planning the configuration to prevent the proliferation of disk devices presented to the attached hosts. In a SAN environment, each path to a logical disk on the ESS is presented to the host system as a physical device. The SAN zones will effect how many devices are presented to a server.

A detailed discussion of configuring logical disks in a SAN is discussed in 5.7.7, “Configuring logical disks in a SAN” on page 146.

3.4 Logical disk sizes - General considerations

The flexibility that you have with the ESS for choosing the logical disk size is particularly helpful when you need to satisfy dissimilar I/O processing requirements, but it can present a challenge as you plan for future needs. In this section we discuss the considerations that should be viewed when planning the size of you logical disks in the ESS.

3.4.1 Future requirements

The ESS supports a high degree of parallelism and concurrency on a single logical disk. Because the ESS does not serialize I/O on the basis of logical disks, the size of logical disks does not have an impact on performance internal to an ESS. Measurement results indicate that a single logical disk consuming an entire array can achieve the same performance as many smaller logical disks on that array.However, the size of logical disks can have an impact on *server* performance and also on the administration tasks.

The size of logical disks becomes very important when you want to re-assign ESS storage capacity. For example, in an open systems environment, if you have a 200 GB logical disk, and now you want to divide it into four 50 GB sections and assign to different hosts, you could be stuck. In order to change the size of the logical disk, the array that the logical disk resides on will have to be re-formatted.

In the zSeries environments you can go for the bigger volumes (3390-9 devices) without compromising server performance if you use Parallel Access Volumes (PAV). But as with open system, if for any reason you later prefer to use a different combination of capacity sizes within a specific rank, then the rank will have to be re-formatted.

Once created, logical disks cannot be re-sized or removed from an array without reformatting the entire array. So, it is important to plan for future requirements.

3.4.2 Maximum number of devices

There are maximum numbers of supported devices for both the ESS and also that result from the servers’ attachment type. These maximum numbers should be taken into consideration when planning the number and size of the devices that will be configured on the ESS.

The maximum number of logical disks an ESS can support is 4096. In the ESS, each LSS supports a maximum of 256 devices, so if you use all 16 LSSs, the maximum number of logical disks supported is 16 x 256 = 4096.

The ESS supports a maximum of 128 host login IDs per Fibre Channel/FICON host adapter port, and a maximum of 512 SCSI/FCP host login IDs or SCSI-3 Initiators per ESS.

These numbers are important when considering the implementation of ESS Copy Services, the maximum number of hosts to attach to a given ESS, and the number of logical disks to assign to each host.

When considering which logical disk size to use, it is also important to consider that the ESS attachment type a host uses will limit the number of logical disks that can be presented to the host. Typically, a SCSI attached host can support 8 or 32 logical disks. The limit for FCP attached hosts is typically 256 logical disks.

To check the attachment characteristics you should check both the IBM publication *IBM TotalStorage Enterprise Storage Server Host System Attachment Guide,* SC26-7446*,* and the vendor information for your host. This topic is also further discussed in Chapter 5, “Host attachment” on page 127.

3.4.3 System management

Consider how logical disk size affects systems management. Smaller logical disks allow for more granularity when managing storage, although it increases the number of logical disks seen by the operating system. Operating systems vary in the number of disk devices they can support. Select an ESS logical disk size that allows for granularity and growth without proliferating the number of logical disks. Specifically for open systems, as more logical disks are presented to a server, the longer device discovery and, therefore, system boots could take.

3.5 Logical disk sizes - LVM considerations

On open systems with a built-in Logical Volume Manager (LVM) (such as AIX, HP-UX, Windows NT, and Windows 2000), 8 GB or 16 GB logical disks will generally satisfy most requirements. The same is true for operating systems with a Logical Volume Manager add on, such as Veritas Volume Manager for Sun Solaris.

Eight or 16 GB logical disks usually work well with open systems because:

- The logical volume manager can group together logical disks from the ESS into a larger volume group.
- Typical volume group sizes for UNIX servers are on the order of 100–200 GB so you can group together 6–12 logical disks from the ESS, which is reasonable to manage.
- By using logical disks from different arrays, you will spread the I/O across ESS components. This is preferable to forming a volume group from one large logical disk on one array because:
  - The throughput of that volume group will be limited to the performance of a single array.
  - I/O transactions to that volume group will only utilize half of the available ESS cache and NVS since an array is managed by only one cluster at a time.
- 8 or 16 GB logical disks leading to the creation of multiple logical disks will help to implement OS level *striping* properly. OS level striping on the ESS should be done across logical disks from different arrays, which are preferably all the same size. See 3.10, “Open systems striping” on page 72.
- For striping at the OS level, eight and 16 GB are multiples of commonly used fine grain stripe sizes of 64 and 128 K. This helps prevents wasting disk space when striping.
- The size is large enough to prevent a huge of amount of devices from being presented to the operating system.
- The size is small enough to be eligible to re-assign to almost any other server in an enterprise environment (several different types of ESS-attached servers).
- If arrays contain several logical disks, it makes tuning hot spots (hot arrays) much easier. With the logical volume manager of AIX, for example, you can migrate data from one logical disk on a busy array to another logical disk on a less busy array—live with no downtime.

Keep in mind that each logical disk is presented to a server as a physical device, an **hdisk**, for example, to AIX. A large number of logical disks presented could negatively effect the time to boot because the scan for bootable disks will take longer, and high availability failover operations can take longer as each logical disk has to be unconfigured from one host and re-configured on another. See “SAN implementations” on page 60 for a discussion on how SAN configurations can cause the proliferation of disk devices to an operating system.

AIX, HP-UX, and Sun considerations

Currently, Subsystem Device Driver (SDD) for AIX, HP-UX, and Sun operating systems supports up to 600 **vpaths**, so these hosts should not have more than 600 logical disks assigned to them even if their attachment type supports such a large number.

Windows NT considerations

The attachment type of a Windows NT server will limit the number of disk devices and therefore effect the choice of logical device size. Windows NT has a limit of eight logical disks per SCSI target and a maximum of 255 logical disks per Fibre Channel port. A Windows NT server may also have a limit to the number of drive letters available for use.

Intel hosts running Linux

Be aware that the maximum number of devices that are supported on a Linux host without Linux extensions is 128. The standard Linux kernel uses a major and minor number address mechanism where a special device file represents each disk device. There is a maximum of 16 partitions per disk.

There are eight major numbers that are reserved for SCSI devices, numbered 8, 65, 66, 67, 68, 79, 70, and 71. In turn there are 256 minor numbers available for each of the eight major numbers.

The following formula provides the maximum number of devices for Linux host systems:

```
Number of devices = (number of major numbers) x (number of minor numbers) ÷ (number of
partitions) = 8 x 256 ÷ 16 = 128
```

There are several Linux extensions available to address this limitation. One approach is to use the major and minor number address spaces in different ways. You can use some of the minor number address spaces for the partitions for the major number address space to provide more devices with less partitions.

You can also use the device file systems **devfs** command. The **devfs** command uses a 32-bit device identifier, which enables the ESS to address many more devices. It shows only the devices that are available on the system, instead of listing device files for devices that are not attached to the system. The **devfs** command mounts over the /dev file and uses UNIX-like device identification.

Operating systems without an LVM

You should adjust the volume size to the given requirements on operating systems without an LVM (Sun Solaris and Windows NT in an MSCS environment). However, a bigger volume can be a better choice because most operating systems have the ability to divide a physical volume into smaller parts (partitions or slices).

3.6 Logical disk sizes - zSeries

Logical disks for zSeries servers are fomatted on CKD arrays. The CKD ranks can be configured either RAID-5 or RAID-10, and formatted into 3390 or 3380 track format. The ESS

supports 3390-3, 3390-9, or 3390 with 3380 emulation device types. All device types are implemented the same way on ESS and offer the same performance within the ESS.

When configuring your CKD volumes in the ESS, you also have the option to define *custom volumes*. You can define logical 3390 or 3380 type volumes that do not have the standard number of cylinders of a 3390 Model 3 or Model 9, for example, but instead have a flexible number of cylinders that you can choose (in fact, any number from 1 to 32,760 cylinders).

The choice of the device type and size depends on your needs, migration paths, and the application.

Parallel Access Volume (PAV)

The ESS also has a specific feature for z/OS (and OS/390) operating systems called Parallel Access Volume (PAV). PAV allows a zSeries server to run applications that can perform concurrent I/Os to the same logical disks. More information on using the PAV feature is available in 9.2, “Parallel Access Volumes” on page 308.

Large volume support (LVS)

For CKD devices there is also *large volume support* (LVS), which increases the size of the largest CKD logical disk supported from 10,017 to up to 32,760 cylinders (approximately 27.8 GB).

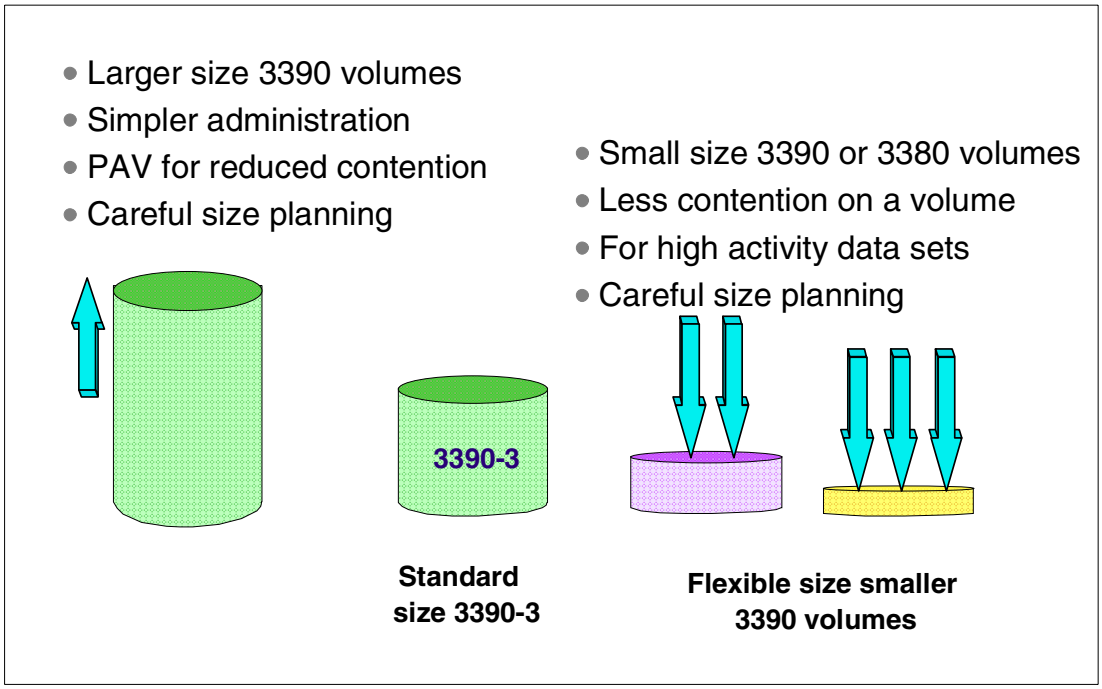


Figure 3-10 CKD custom volumes

Figure 3-10 shows the choices in creating CKD logical disks (volumes) for zSeries. You might want to consider larger capacity logical disks (for example, 3390-9) for one or more of the following reasons:

- Simpler storage administration with fewer total volumes.
- Fewer out-of-space conditions with larger volumes.
- Use of larger capacity device types, such as 3390-9, requires fewer device addresses to access your data.

- Large volumes used in combination with PAV will provide concurrent I/O to the volumes and reduce I/OS queue time.

An alternate approach is to spread high-activity data sets on separate smaller custom volumes, or even give each high-activity data set its own custom volume. For this situation remember that you cannot exceed the 256 device addressing limit per LSS.

You should carefully plan the size of the custom volumes and consider the potential growth of the data sets. You can adjust the size of each custom volume to the data set that you plan to put on this volume, but you might also come to the conclusion that you just need some standard small volume sizes of, perhaps, 50, 100, or 500 cylinders.

3.7 Logical disk sizes - iSeries

For the IBM @server iSeries, the choice in size of ESS logical disks will be determined by the operating system release level and attachment type (SCSI or FCP), as well as the array capacity.

3.7.1 SCSI attachment

When an iSeries host is SCSI attached, the ESS emulates a 9337 device.

The ESS defines protected 9337 Models xxC and unprotected 9337 Models xxA, as shown in Table 3-1. You can define a status of unprotected through the ESS Specialist when you assign the logical disks. The iSeries host supports only software mirroring on an unprotected 9337 model and prevents software mirroring on a protected 9337 model. The status for protected models is DPY (device parity).

Note: From an ESS physical configuration viewpoint, all IBM AS/400® volumes are RAID-5 or RAID-10 logical disks that are protected within the ESS. When you create the iSeries logical disks by using the ESS Specialist, you can create them as *logically* protected or unprotected.

Table 3-1 Capacity of logical disks for SCSI-attached iSeries

Size	Type	Protected	Unprotected
4.190 GB	9337	48C	48A
8.589 GB	9337	59C	59A
17.548 GB	9337	5AC	5AA
35.165 GB	9337	5CC	5CA
36.003 GB	9337	5BC	5BA

Note: You cannot specify a logical disk size of 70.564 GB for a SCSI-3 attachment.

You can specify 1–8 logical disks for each SCSI-3 attachment to a specific ESS host adapter port.

3.7.2 FCP attachment

When an iSeries host is FCP attached, the ESS presents logical disks to the host as device type 2105. A list of possible logical disk sizes is shown in Table 3-2.

Table 3-2 Capacity of logical disks for Fibre Channel attached iSeries

Size	Type	Protected model	Unprotected model	Release support
8.59 GB	2105	A01	A81	Version 5 Release 1 or later
17.548 GB	2105	A02	A82	Version 5 Release 1 or later
35.165 GB	2105	A05	A85	Version 5 Release 1 or later
36.003 GB	2105	A03	A83	Version 5 Release 1 or later
70.564 GB	2105	A04	A84	Version 5 Release 1 or later

Note: You cannot use SDD on the IBM iSeries host system.

You can specify 1–32 logical disks for each attachment to an iSeries Fibre Channel adapter.

You cannot specify a logical disk size of 4.190 GB for the FCP attachment.

For a list of AS/400 and iSeries models to which you can attach an ESS, see the following Web site:

http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105

3.8 Placement of logical disks

In this section we discuss the factors that should be considered when planning the distribution of the logical disks across the backend arrays.

3.8.1 RAID configuration

The ESS RAID-5 ranks can have two different arrangements: 6+P+S and 7+P, depending on if the array contains a spare or not. ESS RAID-10 arrays can also have two different configurations: 3+3+2S (3x2) or 4+4 (4x2). You might have realized that a 7+P RAID-5 array should perform better than a 6+P array, and a 4x4 RAID-10 should perform better than a 3x3. For random I/O specific benchmarking the 7+P and 4x4 ranks may show up to 15 percent greater throughput than their counterparts. For sequential applications, the differences are minimal. See Figure 3-13 on page 71 and Figure 3-14 on page 71 for an example of the performance differences between array configurations.

Keep in mind that some arrays in the ESS must have spares, so there is no way to avoid the 6+P+S and 3+3+2S arrays. Also consider that the larger arrays—the 7+P and the 4x4—will have more capacity, which typically means more I/O transactions for a given access density. So the gain in speed of the larger arrays is usually offset by an increase in demand of the array.

Note: Try to balance workload activity evenly across RAID arrays, regardless of the size. It is not worth the management effort to do otherwise.

3.8.2 Logical disk placement

With servers that have locally attached single SCSI disks, placement of data on different areas of the disks can be important. For example, hosts that have their operating system on internal SCSI drives may try to improve paging performance by placing swap space in the middle of drives.

Also, for some other storage array vendors, placement within disks is important. That is because parts of single SCSI disks are presented to hosts as logical disks. In that case, placement of data on the drives is important.

The logical disk placement considerations when using the older single non-raid disks were based in the following assumptions:

- Data placed on the outer edge can be transferred at a higher rate because the disk surface area covered by the read head in each revolution is greater. Every part of the disk surface is spinning at the same speed (revolutions per second), but the area covered by the read head per revolution is 2pi*R*W (where R is the radius from the center of the disk to the location of the read head and W is the width of the read head). More area is covered on the outer edge per revolution vs. inner areas of the drive where “R” is smaller.
- Seek times are minimized for data that resides in the middle of the referenced data on disk. (This avoids frequently seeking [moving the read/write head] from one extreme edge of the disk to the other.)

However, the logical disks presented by the ESS are not parts of a single disk, but sections of RAID-5 or RAID-10 arrays across multiple drives. It is natural to carry forward methods and assumptions concerning data placement that you might have used before for locally attached SCSI drives, but the ESS requires a new way of thinking about data placement.

Important: The logical disks that the ESS presents to hosts are really a series of hardware stripes across RAID-5 or RAID-10 arrays. Placement of data within an ESS array is *not* a performance issue.

3.8.3 Creating logical disks on different disk groups

While the location of the logical disks *within* an array is of no importance in the ESS, placing the data *across* the different arrays can deliver the performance benefit of their aggregate throughput. Spreading across arrays will allow both ESS clusters to fully contribute with the I/O processing activity.

When using the ESS Specialist to carve logical disks out of disk groups, you can spread I/O within the ESS by creating logical disks on disk groups on different clusters, SSA adapters, and loops, as the example in Figure 3-11 on page 68 illustrates.

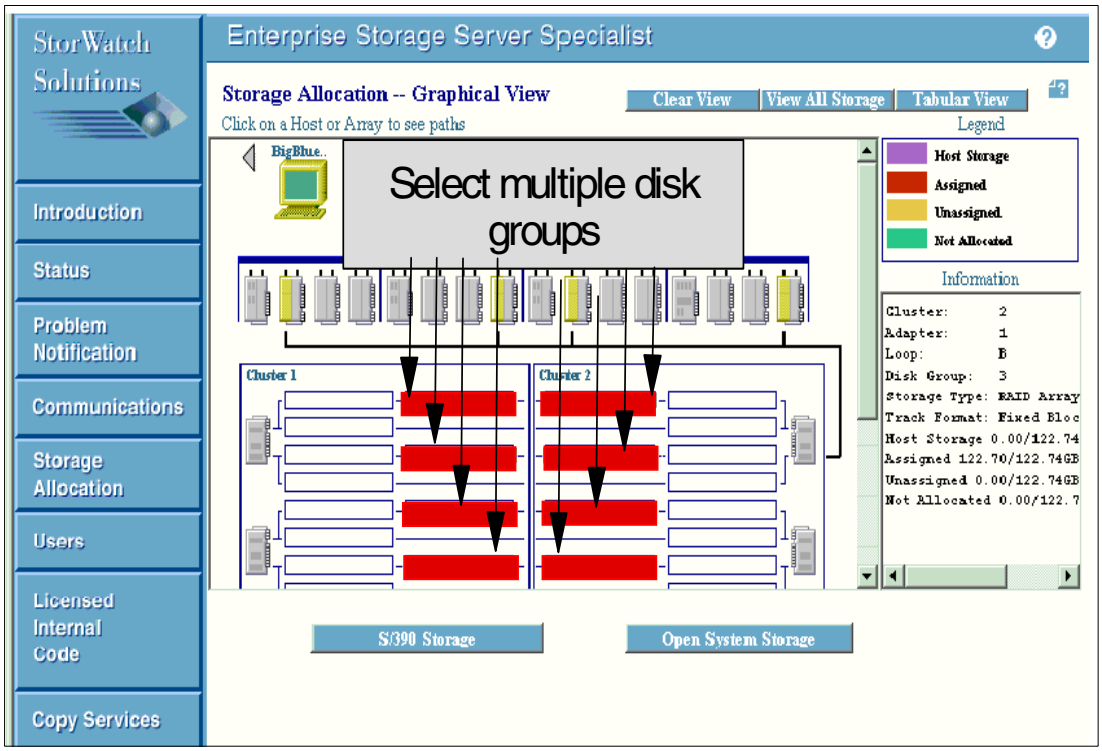


Figure 3-11 ESS Specialist - Selecting multiple disk groups

For the example in Figure 3-11 we arbitrarily chose eight-disk groups.

After selecting the disk groups, the next step is to carve logical disks. You can take advantage of the ESS Specialist option to *spread volumes across all selected storage areas*—as shown in Figure 3-12 on page 69 for our open systems storage allocation example.

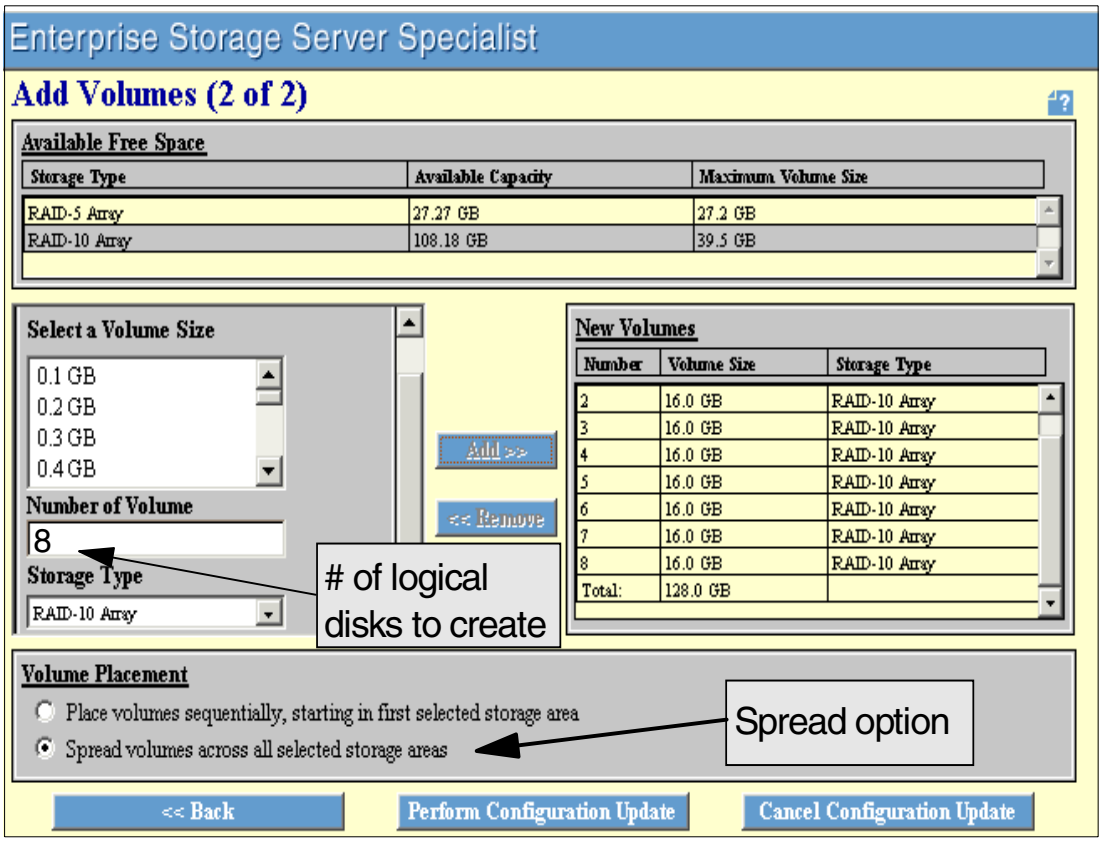


Figure 3-12 ESS Specialist spread logical disks option

Notice that the number of volumes (logical disks) to create is a multiple of the number of disk groups. In this case, eight logical disks will be created and spread—one logical disk at a time on each of the eight disk groups selected. If the *sequential* option had not been chosen, then all eight logical disks would have been created on the first disk group if space permitted before moving on to the next disk group.

Tip: The ESS Specialist will offer both options to either *Place volumes sequentially, starting in first selected storage area* or *Spread volumes across all selected storage areas* even when selecting a single disk group (storage area). There is no difference in the outcome for either option when using only one disk group. All logical disks (volumes) will be created on the disk group selected.

3.9 RAID-5 vs. RAID-10 considerations

RAID-5 optimizes storage far better than RAID-10 (less capacity overhead). See 2.6.3, “Disk eight-pack capacity” on page 24, for further discussion.

While RAID-5 remains a price/performance leader, offering excellent performance for most applications, RAID-10 can offer better performance for selected applications; in particular, in high random write content cache unfriendly applications, particularly in the open systems environment. This is so because of the I/O processing characteristics of the RAID-10 operations:

- Reads can be satisfied from either of the mirrored copies in a RAID-10 rank, so the I/O read request will be satisfied from the available copy.

- Writes do not need parity generation when done upon a RAID-10 rank.

The decision about whether to use RAID-5 or RAID-10 will depend principally upon your performance requirements and I/O workload characteristics, balanced against the extra cost of using RAID-10 over RAID-5. Although RAID-10 arrays will potentially deliver higher performance than RAID-5 arrays, it must be considered that the disk arrays are connected to the SSA loops and device adapters, which in turn are front-ended by a powerful pair of clusters with considerable amounts of cache memory and NV. For some applications, the majority of its reads and writes can be satisfied from cache and NVS, so these applications will see no discernible difference whether the ranks are RAID-5 or RAID-10.

Consider that the *array configuration*—whether RAID-5 or RAID-10—is not an isolated factor when estimating the overall ESS performance, but must be considered together with other important factors like the I/O workload characteristics, the disk drives capacity and speed, the ESS processors (Turbo or Standard), the cache size, the number and type of ESS host adapters, and the backend data layout and SSA loops.

Our recommendation is to use Disk Magic for properly determining how significant the difference between using RAID-5 or RAID-10 ranks on your configuration can be.

A more fair comparison - Same number of disk drives

If you only had one choice in disk drive capacity (say, for example, 36 GB disk drives) and if you were configuring the ESS solely based on required storage capacity, then a RAID-10 configuration would require roughly twice the number of disks as compared to RAID-5 just to meet the storage capacity that is needed. Because there would be more disks over which to distribute I/O activity, this would give the RAID-10 configuration an edge over the RAID-5 configuration, but also the cost would be substantially more.

This complicates matters a little more since two different variables are changing, *cost* and *performance*. The alternative and more simpler approach is to compare configurations of an equal number of disk drives, that sets aside the cost consideration. This simpler approach can be acceptable, and besides, the ESS offers different disk capacity choices, so you might consider for your comparison using larger capacity disks for RAID-10 than for RAID-5. In that case, the RAID-5 and RAID-10 configurations would have roughly equal capacity and cost.

RAID-5 vs. RAID-10 examples

Here we present some examples of ESS measurement results when using RAID-5 and RAID-10 arrays. The discussions on these examples will help you better understand the performance implications of the different rank configurations.

You can estimate that RAID-10 random and sequential *reads* will be very similar to RAID-5 reads. From the performance perspective, it is high levels of *random writes* that will show benefits for the RAID-10 operations, as shown in Figure 3-13 on page 71.

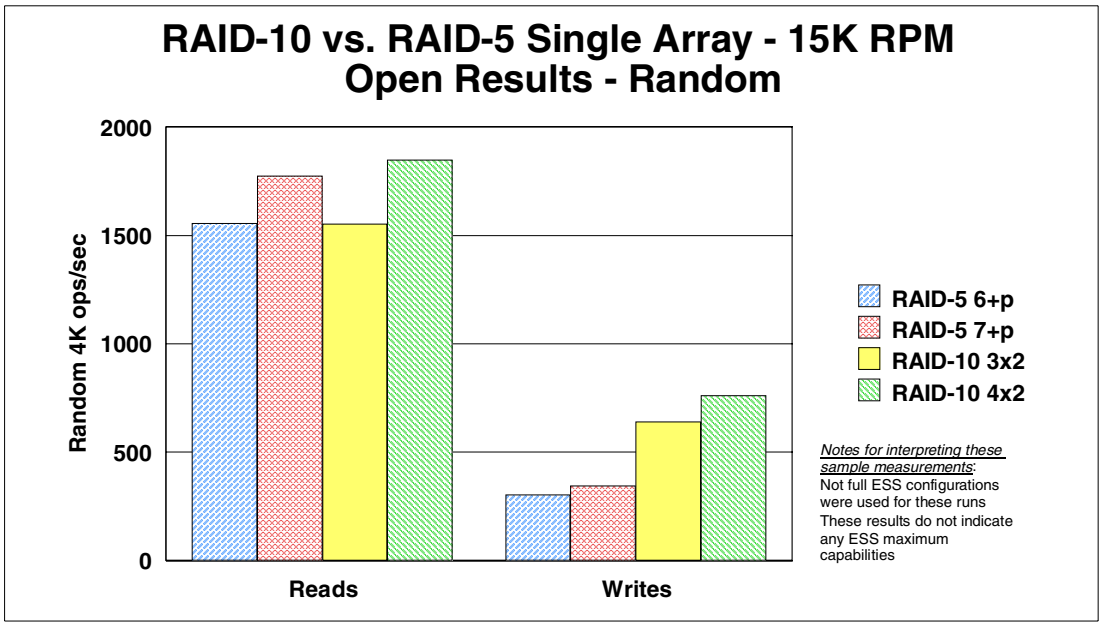


Figure 3-13 RAID-10 vs. RAID-5 - Random I/O workload

Sequential reads can be very similar between RAID-5 and RAID-10, as shown in Figure 3-14. However, you should consider that for *sequential writes*, RAID-5 can be faster because it stripes data across more disks. For example, writing data across a RAID-5 array (7+P) will stripe across eight disks, whereas a RAID-10 array (4x4) will write the same data twice-striped across two mirrors of four disks each.

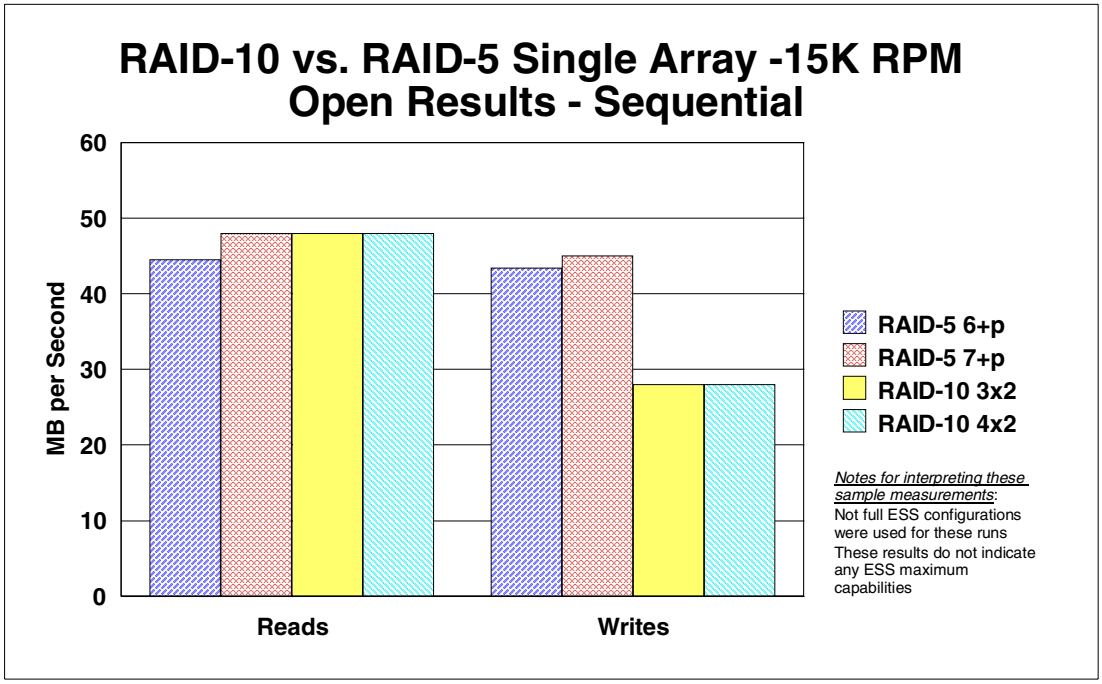


Figure 3-14 RAID-10 vs. RAID-5 - Sequential I/O workload

Since backend writes to the disk arrays on the ESS are asynchronous, whether it is RAID-10 or RAID-5 normally does not really affect the I/O response time seen by most applications.

However, RAID-10 can allow more random write operations (asynchronous destages) to the disks (since there is no parity calculation required), thus allowing higher throughput levels for random write-intensive workloads as compared to RAID-5 with the same number of disks. For a cache unfriendly environment with a large number of random writes, RAID-10 could offer a performance advantage.

The ESS has already proven that RAID-5 is suitable for the majority of the commercial applications and it would be wise to continue to evaluate specific application performance requirements and I/O characteristics to decide whether RAID-5 or RAID-10 is most appropriate. For instance, RAID-5 is very well suited for database logging since it is generally sequential in nature and enables the SSA device adapter to do full stripe writes.

Table 3-3 presents for the different types of I/O workloads, which is the RAID configuration that should be most beneficial. What this table is not showing is the *significance* of the gains, for which a Disk Magic should be run. Nor does the table reflect the greater number of disks (cost) that the RAID-10 configuration demands.

Table 3-3 I/O workload and RAID configuration - Performance expectations

Type of workload	Best RAID type
Random writes (very cache unfriendly)	RAID-10
Random reads (very cache unfriendly)	Slight edge to RAID-10
Sequential reads (large files)	Either RAID-5 or RAID-10
Sequential writes (large files)	RAID-5

The random workload must be sufficiently *intensive* and *random enough* so that I/O requests mostly will not be serviced from cache but a disk stage or destage must take place. If a large percentage of the I/O requests can be serviced from the ESS cache, then RAID-10 will not provide a significant improvement over RAID-5. Also, remember that the ESS tries to cache enough data so that it can destage RAID-5 data in full stripe writes.

Making the decision

Our recommendation is to run Disk Magic to more accurately estimate how significant, from a performance perspective, one RAID implementation is compared to the other one.

3.10 Open systems striping

The ESS Model 800 is capable of exceptional throughput for the different types of I/O. Understanding your I/O workload characteristics (see 2.2, “Understanding your workload characteristics” on page 12) will allow you to further maximize the performance gains you obtain from the ESS. It will be basic to know which of the following is more preponderant in each of your applications:

- Lots of small random I/O operations
- Large sequential I/O operations
- A combination of the above

With this knowledge, now we will see some implementations that will help you boost the performance of your applications.

3.10.1 Single rank file systems

After assigning *logical disks* (LUNs) to an open systems server, the next step is to create *logical volumes* and *file systems* (unless your application calls for just a *raw* logical volume). The easiest way to create a logical volume is to select one logical disk and use the *logical volume manager* (LVM) of the operating system to allocate space within a logical disk.

Figure 3-15 gives an example of creating a logical volume on a single logical disk (LUN). In this example we use vpath0, which is an 8 GB logical disk on the ESS. As we already know, the logical disk is *hardware striped* by the ESS in a RAID-5 or RAID-10 array. If the rank is RAID-5 (7+P), for example, then vpath0 will be striped across 8 DDMs.

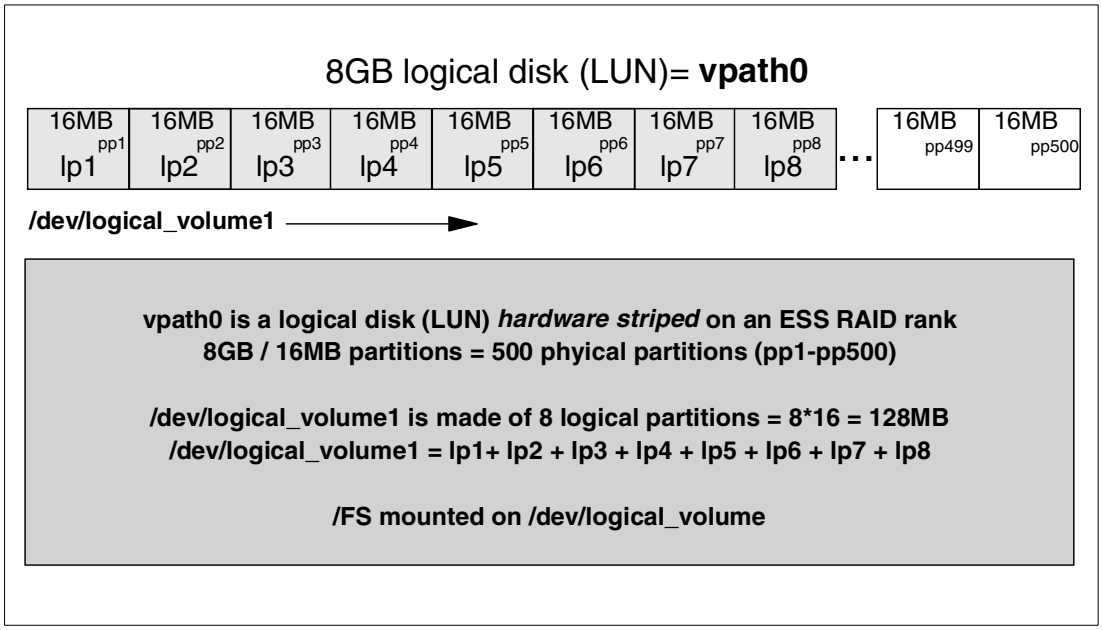


Figure 3-15 File system on a single logical disk

In this example the logical volume manager (LVM) of the host operating system has divided logical disk vpath0 into 16 MB physical partitions. Next, we grouped together eight of these partitions to create a 128 MB logical volume called /dev/logical_volume1.

We could then assign the raw logical volume /dev/logical_volume1 to a database application or create a file system on top. In this case, we created a file system called /FS mounted on top of /dev/logical_volume1.

I/O to file system /FS will be limited by the capacity of the array that vpath0 resides on. That is fine for cache friendly random I/O because most I/O operations can be completed in cache. When the ESS does need to read or write to disk for /FS, data is hardware striped across the RAID array disks vpath0 resides on. If /FS access is very cache unfriendly, then we prefer vpath0 is created on a RAID-10 array.

3.10.2 Striping for high sequential I/O

Now, if we wanted to read from /FS (in Figure 3-15) at sustained very high throughput rates, we could be in the situation of not being able to do so because the file system resides on a single logical disk and we could be limited by the rank capacity.

If you anticipate a need for high sequential read speeds, for example, environments that need to create offline backups at 1 TB/hour (1000 GB/hour or 277 MB/sec), then for optimum

performance you will prefer to *aggregate* the throughput capabilities of more than one ESS array. As powerful as the ESS is, for a given file system if it is only reading from a single array (eight or less disks) then you will not be taking full advantage of all the ESS capabilities.

In that case you will want to *software stripe* the file systems at the operating system level (fine grain stripe at 256 K or less) across logical disks on *different arrays* on *different loops*. By striping across multiple SSA loops, you can take advantage of the aggregate throughput of several arrays and SSA device adapters.

Tip: When *striping* at the operating system level, it is very important to stripe across logical disks from different arrays to avoid I/O serialization.

By using a fine grain stripe, you can achieve such high read/write speeds that the throughput for a given server will be limited by the number of host bus adapters it contains.

It is difficult to predict the overall throughput of a given array since the bandwidth of arrays and SSA loops will generally be shared among several applications and ESS-attached servers. For optimum performance, our recommendation is that when you are reading sequentially over 200 MB/sec, you should be striping file systems across 4–6 logical disks on different SSA device adapters and loops. To read over 400 MB/sec, your server will need two 2 Gb/sec Fibre Channel host bus adapters (or four 1 Gb/sec adapters; note, server I/O drawers and internal PCI buses can become a limit here—and striping across 8–10 ESS arrays.

To implement striping at the OS level, you will want to use logical disks that:

- Are all the same capacity
- Reside on different arrays on different SSA loops and different SSA adapters
- Are a multiple of the stripe size you plan to use (64 or 128 K, typically)

3.10.3 Spread vs. stripe

The stripe size chosen is very important when implementing striping, and must be carefully selected. If you choose a large stripe size on the order of MB, you will not see the performance increase like you will when using a *fine-grain* stripe of 256 K or less. For a large stripe size, like 4 MB, data is not really *striped*, but just *spread* from one ESS array to the next. In this case, sequential I/O will read/write 4 MB from one ESS rank before going to the next, which is too large of a size to make multiple arrays active at the same time. Even if your operating system offers a read-ahead function, most of the time you will still be accessing a single ESS array and for a brief moment two arrays.

For concurrent I/Os from multiple striped file systems, it is best to stripe each file system on a different set of ESS ranks or at least in a different order within the same set of ranks.

Figure 3-16 on page 75 shows an example of spreading a logical volume across different arrays. The logical volume is a group of 16 MB physical partitions from four different logical disks—vpath0, 1, 2, and 3. When writing to /SPREADFS, 16 MB will be written to lp1 on vpath0, then lp2 on vpath1, and so on. This logical layout does spread I/O across different ESS components but does not offer the aggregate throughput of multiple arrays for sequential I/O.

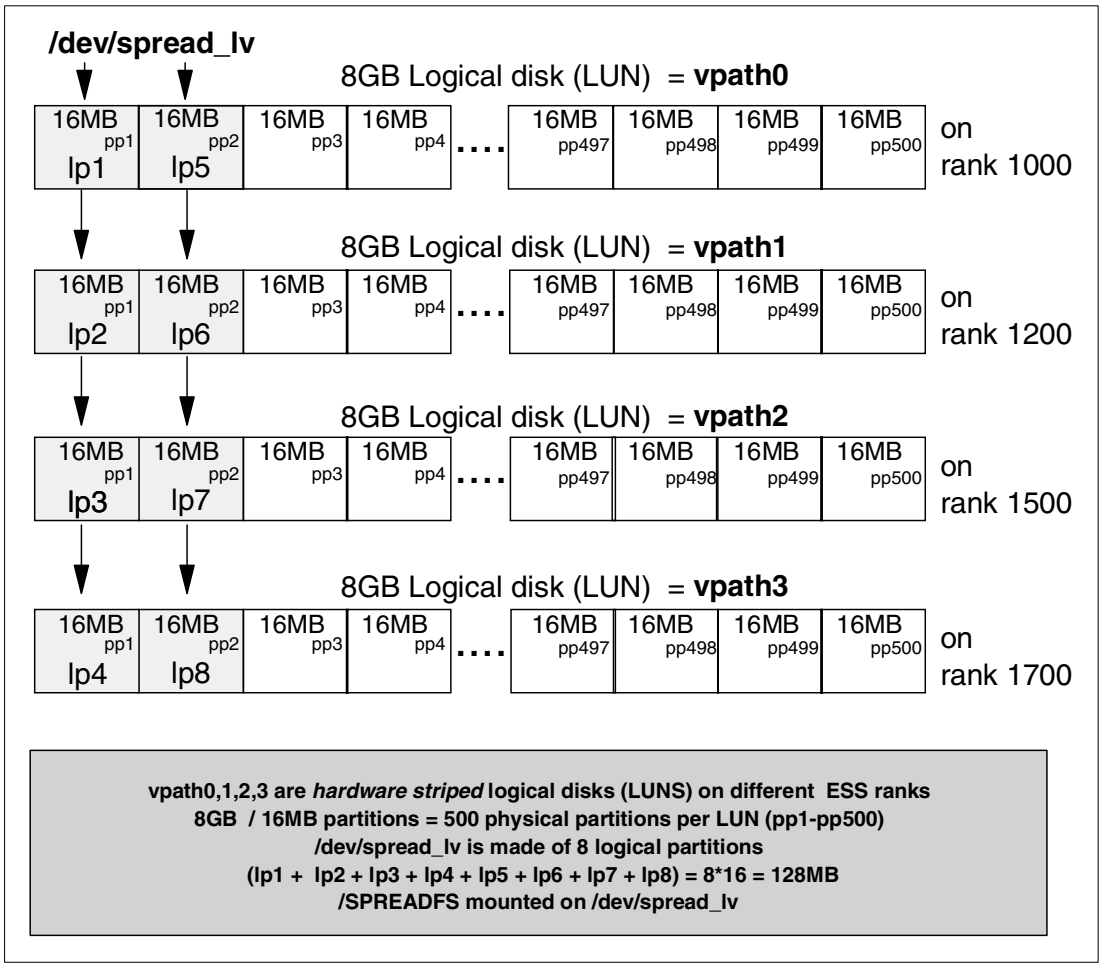


Figure 3-16 Spread file system

3.10.4 Striped file system

An example of a striped logical volume is shown in Figure 3-17 on page 76. The logical volume called /dev/striped_lv uses the same capacity as /dev/spread_lv (shown in Figure 3-16) but is created differently.

Notice that /dev/spread_lv is also made up of eight 16 MB physical partitions, but each partition is then subdivided into 64 chunks of 256 K—only 3 of the 256 K chunks are shown per logical partition for space reasons. Most operating systems include a -S flag or similar to create a striped logical volume.

On top of /dev/striped_lv we created and mounted a file system called /STRIPED_FS. Notice that the striping function takes place when the logical volume is created—not the file system. If you are going to implement striping, 128 K or 256 K stripe sizes work well. For RAID-5 arrays with 9.1 GB and 18.2 GB DDMs, the ESS stripes 32 KB across each DDM in an array (64 KB for 36.4 GB, 72.8 GB, and 145.6 GB disk drives).

Access to /STRIPED_FS will cause vpath0, 1, 2, and 3 to run in parallel. If each vpath resides on a RAID-5 rank with a 7+P configuration, then that is 8*4 = 32 disk drives running in parallel. You could expect to see four times the sequential throughput for /STRIPED_FS as compared to /FS or /SPREAD_FS in the previous examples.

Tip: If you need to read from one striped file system and write to another, then start the file systems on different logical disks. For example, /dev/striped_lv starts on vpath0 in our example in Figure 3-17. If you made a /dev/striped_lv_TWO, start that striped logical volume on vpath3.

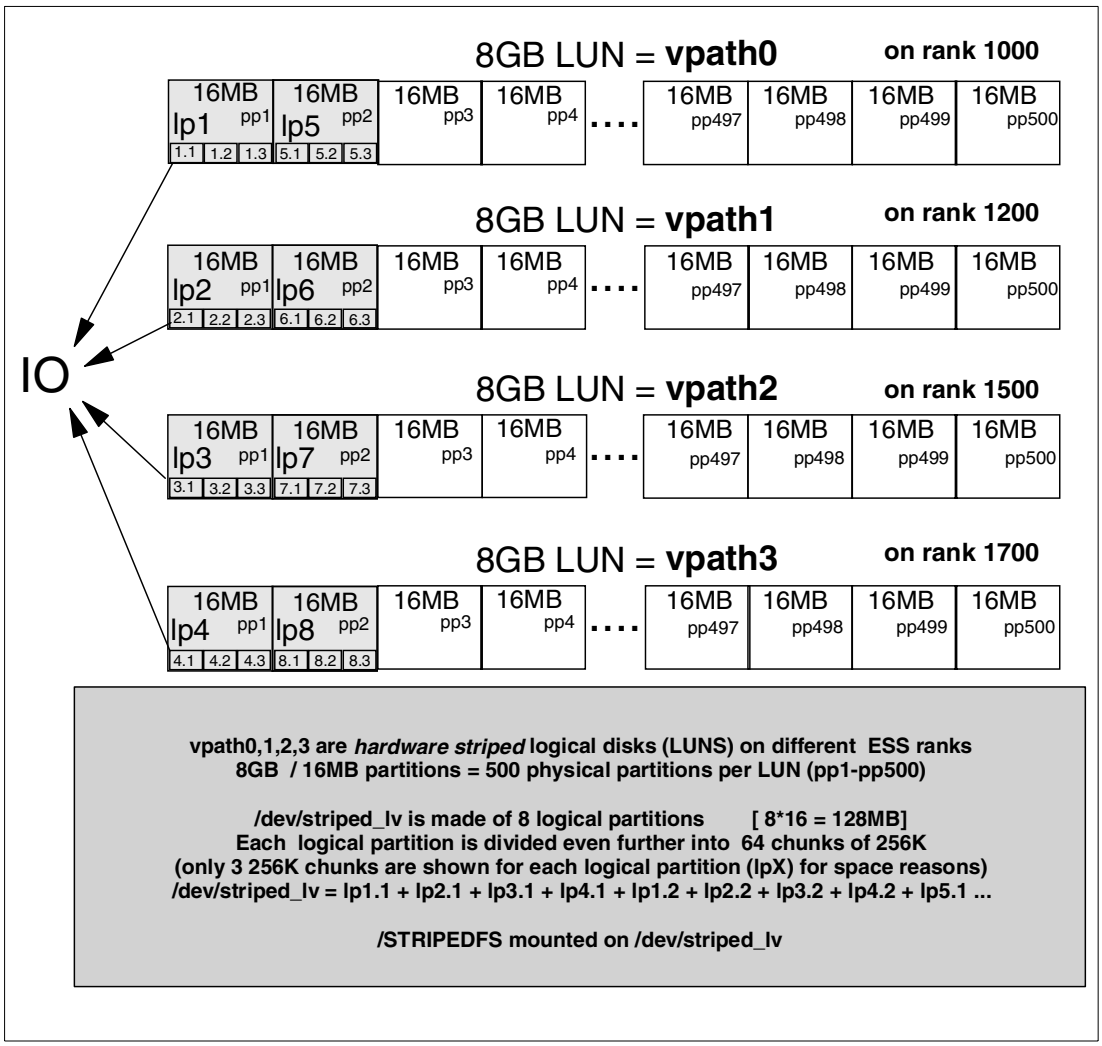


Figure 3-17 Striped file system

How striping affects random workload

Striping does not negatively affect random workload. Look at /STRIPED_FS in our example in Figure 3-17, which is striped across four RAID-5 (7+P) logical disks. That is 32 drives in the stripe set. When performing sequential reads or writes, all 32 drives are active. There will be some overhead to have all 32 drives set up for sequential I/O, but once the I/O starts, it will be about four times faster than using a single array.

For *random* I/O, all 32 drives will not be involved. Normally, random I/O for open systems is on the order of 32–64 KB for database applications. If your stripe size is 256 K, for example, random I/Os less than 256 K only go to a single logical disk, just as they would if you were not striping.

3.10.5 Striping logical volumes - Trade-offs

Keep in mind that *striping* file systems is not the answer to every I/O performance goal. Striping will definitely help sequential performance, but is not as easy to administer as other methods of creating file systems.

CPU load

When striping, the operating system has to perform more calculations to determine where to obtain data from in all those little chunks striped across multiple logical disks. You estimate an additional CPU load of about 1–2 percent when using several striped file systems.

Logical volume size increase

Increasing the size of a striped logical volume is possible if there is at least one free physical partition on each of the disks that make up the striped logical volume. For example, Figure 3-17 on page 76 shows /dev/striped_lv striped across four vpaths. As long as vpath0-3 each have one free physical partition, then the logical volume can be increased in size.

Note, however, it is not possible to increase the stripe *width* (that is, add another vpath to the logical volume /dev/striped_lv and stripe across 5 arrays instead of 4). Increasing the stripe width (adding another LUN) would require backing up the logical volume, re-creating it striped across 5 LUNs, and restoring data.

This is a trade-off when using striped logical volumes—you may not be able to increase the size very easily. If you do not need the gains in sequential I/O striping provided, then there is no need to go to the extra effort in planning and implementation.

3.10.6 Hardware and operating system considerations

There are some other factors to consider when implementing striping for high sequential I/O. Besides using ESS logical disks on different arrays and creating a striped logical volume, you typically need to tune the operating system to handle more I/O than the default settings allow.

I/O buffers

For an I/O operation, the operating system needs a memory buffer to store data read from disk. Depending on your operating system, you can increase the number or size of these buffers. For AIX, for example, there are pbufs that need to be increased. For more details, check the specific tuning sections in this publication for your operating system.

Read-ahead

A *read-ahead* option needs to be *turned on* by the operating system to detect sequential I/O patterns and start reading ahead, so data is already in memory when requested. Note, if your operating system allows it, you may want to have scripts that adjust the read-ahead option on the fly. For example, you may want read-ahead turned off during the day if a system is running random workloads, and then turn read-ahead back on at night during sequential offline backups.

Stripe size

Your operating system may present fine grain stripe sizes from 4 K to 256 K. The decision in stripe size should be made to match your applications—for example, if a database is issuing I/O in 64 K blocks, then use 64 K or 128 K stripe size. However, we do not recommend using a stripe size lower than 32 K.

Number and placement of adapter cards

For sequential I/O, you may find that the ESS can provide more I/O than your server can sustain. The number of I/O host bus adapter cards in your server and their placement becomes important. If your system has multiple PCI busses, for example, you may need to move adapter cards around to prevent a PCI bus from becoming a bottle-neck.

3.11 Logical configuration - Checklists and worksheets

In this section we include summary information and checklists that will help you when considering how to allocate storage in the ESS—for easier management and optimum performance. This is by no means a definitive list, but hopefully a good place to start.

Also included are two worksheets. One is for planning capacity requirements based on the server workload, and the other one is for planning which ESS arrays to use for each server.

Table 3-4 summarizes useful ESS information and facts, and lists recommendations for configuring logical components of the ESS.

Table 3-4 ESS logical configuration - Information and recommendations summary

The ESS contains 2 clusters; each cluster has 1 GB of NVS and half of the cache capacity.
All ESS writes go through NVS and cache. All reads go through cache.
The ESS contains eight SSA device adapters and loops. Each loop can contain up to 6 disk arrays.
An ESS can contain up to 48 eight-packs = 384 disk drives.
An ESS array is a collection of disks drives from two eight-packs.
ESS arrays are formatted in either FB or CKD format and either as RAID-5 or RAID-10.
ESS arrays are also referred to as ranks.
A logical disk is a series of hardware stripes across an ESS RAID array, and presented to a host operating system as a physical disk device.
For open systems, logical disks are also referred to as LUNS, volumes, and logical volumes. For zSeries, logical disks are referred to as 3390 devices (or 3380) or volumes.
Logical disks do not span multiple ESS RAID arrays. For open systems with a logical volume manager, logical disks can be grouped together and create file systems, so the data is spread or striped across multiple logical disks. For z/OS systems, data sets can span multiple 3390 volumes.
Each path from a host to ESS logical disk is represented as a physical device to the host.
The size of a logical disk cannot be changed without re-formatting the entire array it belongs to.
Open systems striping at the OS level: It is imperative to use logical disks from different ESS arrays. It is easier to stripe if all logical disks are the same size and a multiple of the intended stripe size. For example, 16 GB is an even multiple of 64 K or 128 K. For open systems sequential I/O reads over 200 MB/sec, striping at the OS level is recommended.
Attachment types differ in the number of devices that can be presented to a host.
The ESS supports up to 4096 logical disks.
The ESS can support a maximum 128 host login IDs per FCP port and a maximum of 512 SCSI-FCP attached hosts per ESS.
If the ESS needs to have a host adapter bay replaced, the entire bay must be quiesced.

The information in the following publications and URLs can be useful when planning the configuration and attachment of the ESS:
IBM TotalStorage Enterprise Storage Server Host System Attachment Guide, SC26-7446
IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide, GC26-7444
IBM TotalStorage Enterprise Storage Server User's Guide, SC26-7445
The IBM 2105 Technical Support pages under Documentation at:
http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105

Note: Some of the information in this table applies to either open systems or zSeries, but not both —although it is not specifically commented on every occurrence.

Table 3-5 is a checklist that summarizes the steps to take when planning the logical disk layout.

Table 3-5 ESS logical configuration - Checklist

Data distribution on the ESS ranks
Server and enterprise I/O requirements Workload types for each server. Random I/O requirements. Sequential I/O requirements. Maximum I/O performance required from the ESS. Plan for future growth in both ESS capacity and storage requirements for servers.
Did you plan with IBM and utilize Disk Magic and Capacity Magic?
How many arrays (FB and/or CKD) do you need for capacity and performance?
How many RAID-5 and/or RAID-10 arrays?
Logical disk planning: Size, number, placement, and paths from server to ESS
Size of logical disks Can you pick one size for the entire ESS or at least one size for each host system to use? Consider future requirements. Will you be able to re-assign the logical disks later? Will the size allow you to spread I/O across ESS components? Will you be striping at the OS level for sequential I/O? With which stripe size: 256 K? Is the logical disk size an even multiple of the stripe size? FlashCopy and PPRC considerations? For z/OS systems, for larger volume sizes, did you plan for PAV definitions?
Number of logical disks Host OS limitations. What is the limit on the number of disk devices supported? Attachment types: How many disk devices can each host support for the planned attachment type? How many logical disks will a host need for the selected logical disk size? How many times will each logical disk be presented to the host system? Will you be striping at the OS level for sequential I/O? If so, across how many different arrays?
Array configuration
Spares minimized? For a given capacity, either the first two ranks that are configured (if RAID-5) will contain spares, or the very first rank that is configured (if RAID-10) will hold both spares.
LSSs balanced?
Logical disks configuration
Take advantage of the *spread* option of the ESS Specialist to create logical disks on different disk groups.

ESS utilities installed for performance monitoring and tuning? (See 5.9, “ESSUTIL utility package” on page 155.)
SDD installed for SAN-attached hosts and verified working correctly? (See 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.)
Note: Some of the information in this table applies to either open systems or zSeries, but not both —although it is not specifically commented on every occurrence.

Figure 3-18 is a worksheet that can be used for each host system you anticipate attaching the ESS to. Use the worksheet to plan where logical disks for each server will reside so you can spread I/O across ESS components.

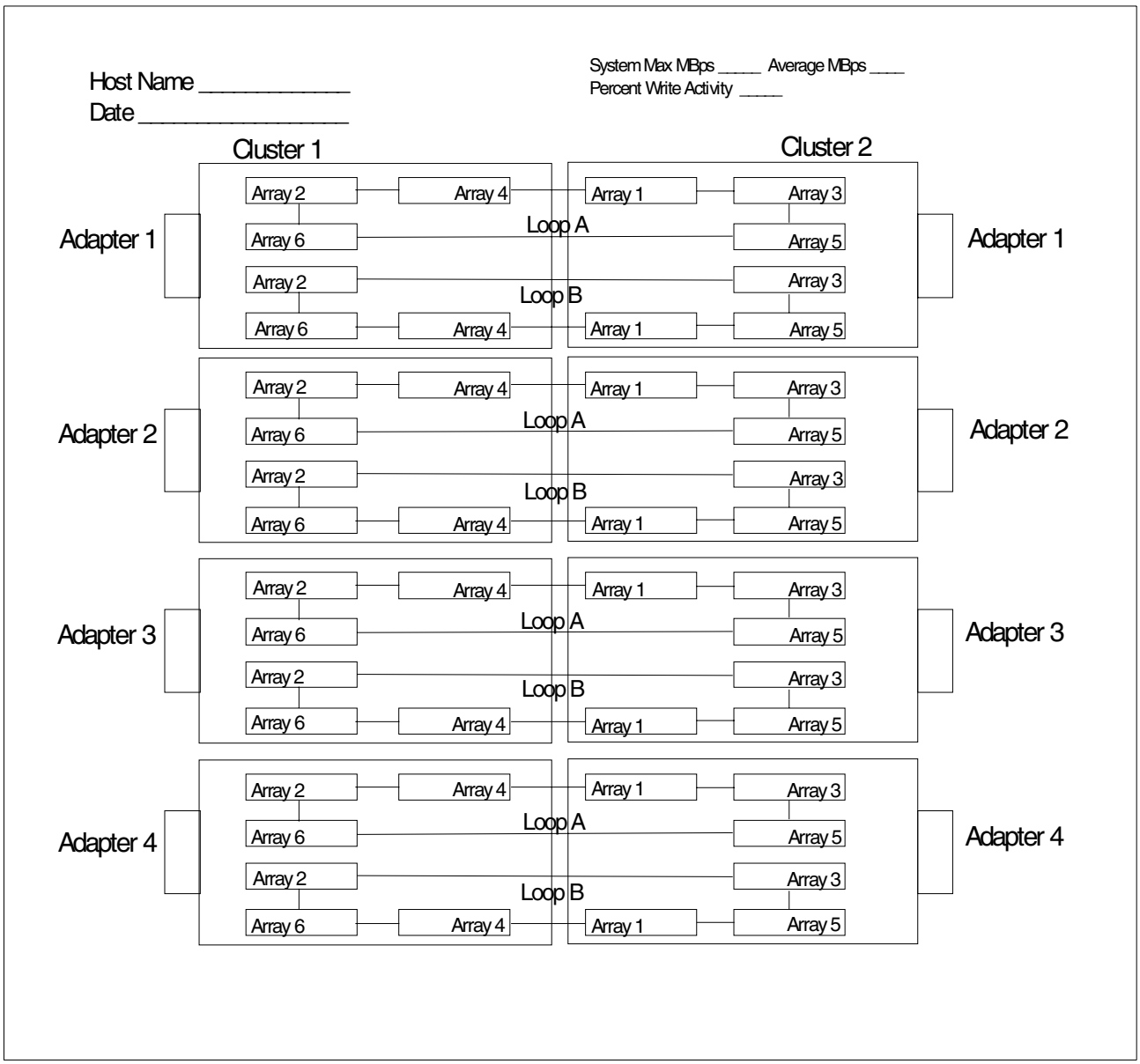


Figure 3-18 ESS logical planning worksheet

Figure 3-19 on page 81 shows a sample spread sheet you can model after, when planning for ESS capacity and performance requirements.

	ESS Storage Requirements							
Application	Hosts	Failover	Type of Workload	Capacity		RAID type	FlashCopy	Notes
PeopleSoft	people1	Clustered	Random / Sequential (backups)	450	GB	RAID-5	YES	limit disk
	people2	Clustered	Random / Sequential (backups)				YES	devices
SAP	sapprod1	Clustered	Random	970	GB	RAID-10	YES	presented
	sapprod2	Clustered	Random				YES	Special SDD
OLTP	oltp2003		Random	250	GB	RAID-10		
Data Warehouse	dwh1000		sequential	250	GB	RAID-5		
	dwh2000		sequential	250	GB	RAID-5		
	dwh3000		sequential	250	GB	RAID-5		
Data Miner	dmine1		Very sequential (READS)	1000	GB	RAID-5		
Digital Video Editor	vserv1		Very sequential	700	GB	RAID-5		
			Total Capacity	**4.12**	**TB**			

Figure 3-19 ESS capacity planning worksheet

The example shown in Figure 3-19 illustrates some important factors to consider when planning on migrating servers to an ESS. Remember, the type of workload is just as important to consider as the capacity requirements.

4

Planning and monitoring tools

This chapter describes the tools available for doing ESS capacity planning and performance monitoring. Some of these tools are for the IBM Business Partner or the IBM Field Technical Sales Specialist (FTSS) to use, but an interactive process like the ESS capacity estimation and sizing is much more efficient when both the customer and the IBM representative are familiar with the tool.

In this chapter we present the following:

- Disk Magic
- Sequential Sizer
- Capacity Magic
- IBM TotalStorage Expert, ESS feature
- IBM TotalStorage Expert use with other operating system specific tools

4.1 Disk Magic

In this section we describe Disk Magic and what Disk Magic is used for. We also include examples where we show the input data that is required, how it is fed into the tool, and also show the output reports and information that Disk Magic provides.

Note: Disk Magic is for the IBM representative to use. Nevertheless, the ESS capacity and sizing planning is done better when both the customer and the IBM representative are familiar with the tool. Customers should contact their IBM representative to do the Disk Magic runs, when planning for their ESS hardware configurations.

Disk Magic for Windows is a product of IntelliMagic, licensed exclusively to IBM and IBM Business Partners.

4.1.1 Overview and characteristics

Disk Magic is a tool that helps in planning the ESS hardware configuration. With Disk Magic you *model* the ESS performance behavior when doing changes in the ESS configuration and the I/O workload. Disk Magic is for use with both S/390 and open systems server workloads.

When doing the ESS modelling, you will be starting from either of these scenarios:

- An existing, non-ESS baseline model from which to migrate to an ESS. This could be an IBM product such as an old IBM 3990-6 or Ramac Virtual Array (RVA), or a non-IBM zSeries attachable disk. Because an ESS might have much greater storage and throughput capacity than other disk storage systems, with Disk Magic you can *merge* the workload from several existing disk storage subsystems into a single ESS.
- An existing ESS workload.

When modelling an open systems workload, you will always start by entering data into the Disk Magic dialogs. This should not be a problem since the amount of data entry is minimal. Performance information you need to gather in this case is block size, read/write ratio, read hit ratio, and I/O rate.

For zSeries workload modelling, Disk Magic can model performance at two levels: subsystem or device. Subsystem level performance modeling was designed to get realistic results quickly, with a minimal amount of data entry. Device level performance modeling requires more input data.

For device level modelling, data must supplied to Disk Magic by means of an automated input process. For this process a Disk Magic Control (DMC) file that is a Disk Magic automated input file can be used. This is a text file that is created with the CP2000 Data Extraction Program, CP2KEXTR, that is part of CP2000 application. A DMC file contains device level workload statistics or disk subsystem level also. At present, Disk Magic Control files can be created for zSeries workloads only.

Disk Magic contains advanced algorithms that can substitute data that normally would have to be entered manually, for both zSeries and open systems modelling. For instance, if cache statistics are not provided, then the Automatic Cache Modeling feature will generate realistic values based on other inputs provided.

Disk Magic is good for modeling *random* workloads. For *sequential* workloads Disk Magic tends to be too optimistic. Disk Magic is not recommended to model iSeries workloads.

Disk Magic is a product of IntelliMagic and is licensed to the IBM Storage Systems Division, to be used for marketing support purposes.

4.1.2 Output information

Disk Magic models the ESS performance, based on the I/O workload and the ESS hardware configuration. Thus it helps in the ESS capacity planning and sizing decisions. Major ESS components that can be modeled using Disk Magic are:

- SMP processor type: Standard or Turbo
- Cache size
- Number, capacity, and speed of DDMs
- Number of arrays and RAID type
- Type and number of ESS Host Adapters

When working with Disk Magic always make sure to feed accurate and representative workload information, because Disk Magic results depend on the input data provided. Also carefully estimate future demand growth, as this will be fed into Disk Magic for modelling projection on which the hardware configuration decisions will be made.

4.1.3 How Disk Magic works

Basically the process of modelling with Disk Magic consists in loading a *base* model configuration, for which you define the *hardware configuration* and then you enter the *workload information*. Once this information is entered, you create a *valid base model*—selecting the **Base** button at the bottom of the dialog panel. In this step, Disk Magic algorithms, validate the hardware and workload information you entered, and if everything is OK, then a valid base model is created. If not, Disk Magic will provide messages and warnings in its log.

Once the valid base model is created you proceed with your projections. Basically you will be changing hardware configuration options to the base model, to decide what is your best ESS configuration for a given workload. Or you can modify the workload values that you initially entered, so, for example, you can see what happens when your workload growth or its characteristics change.

4.1.4 Input dialogs

In this section we do an overview of the more relevant dialog panels that Disk Magic presents, and we discuss what information to complete in those panels.

Welcome to Disk Magic

In this example we are beginning a new Disk Magic project and entering the input data manually. We start with the Welcome to Disk Magic dialog (see Figure 4-1 on page 86), where we select **S/390 - Open Project**. Then we write the total number of S/390 and open systems that we are going to attach. In our example in Figure 4-1 on page 86 we select one S/390 and one open systems, but in fact you can select more.

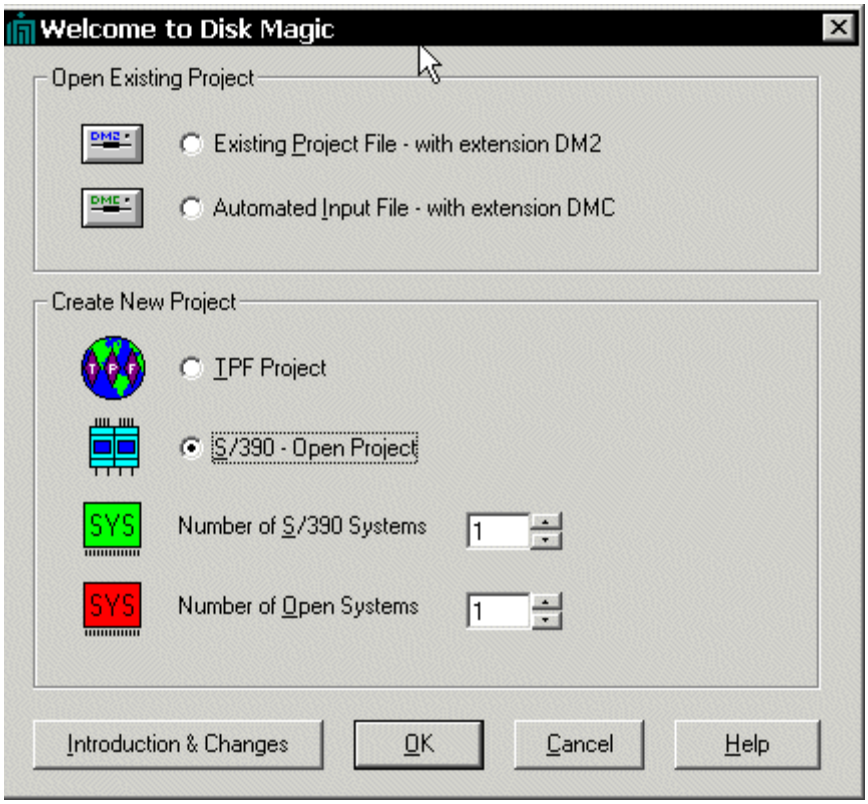


Figure 4-1 Welcome to Disk Magic

Hardware configuration

In the Disk Subsystem - CU1 dialog window, the General tab is used to enter hardware information like the Hardware Type (that basically identifies the *base model* machine), as well as the cache and NVS size information. Also in this dialog panel, if the disk subsystem has any S/390 storage, then the number of *logical control units* (LCUs) within the disk subsystem is entered; if this is an existing ESS, then this would be the number of CKD LSSs. Figure 4-2 on page 87 shows the Disk Subsystem - CU1 dialog panel, with the hardware type ESS 800 selected as an example.

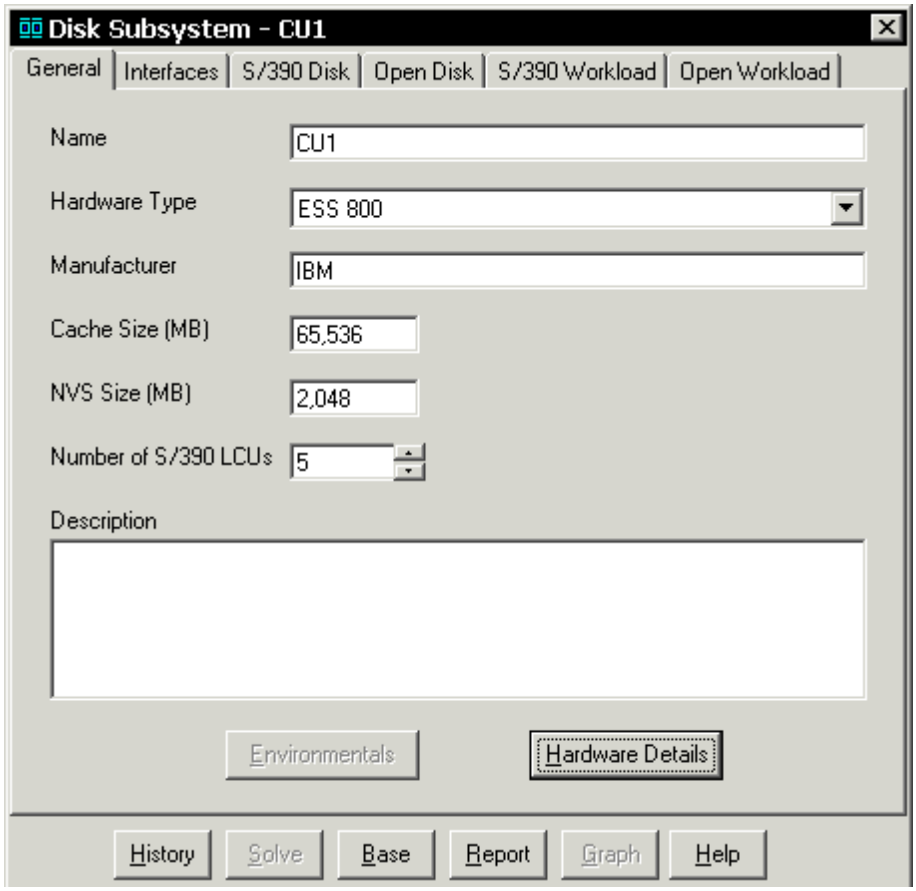


Figure 4-2 Disk Subsystems dialog - General page

The General tab in the Disk Subsystem - CU dialog window is used initially to identify the starting disk subsystem *hardware configuration*, to be used as the model *base* system. Later it can be used to make changes in the hardware configuration of the *base* model, to study the effect of those changes.

By selecting the **Hardware Details** button we open an ESS Configuration Details dialog window, shown in Figure 4-3 on page 88. This dialog window is used to provide further information about the disk subsystem hardware configuration. The fields displayed in this dialog window will depend on the hardware type (see Figure 4-2) that was initially selected. That is because in our example we initially selected ESS 800 as the hardware type, we are getting fields to complete like the Number of 8-Packs, the Parallel Access Volumes check box, as well as protected fields we do not need to complete, like the NVS size, which is fixed for the ESS.

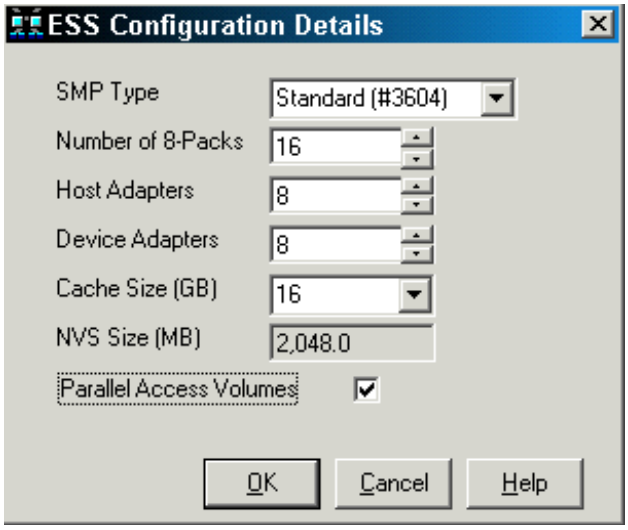


Figure 4-3 ESS Configuration Details

By selecting the **S/390 Disk** tab from the Disk Subsystem - CU1 dialog window, we move to the S/390 Disk page (see Figure 4-4), where we describe the configuration of the CKD arrays. We specify the type of DDMs (capacity and speed), the type of 3390 logical devices defined, and the RAID configuration of the array.

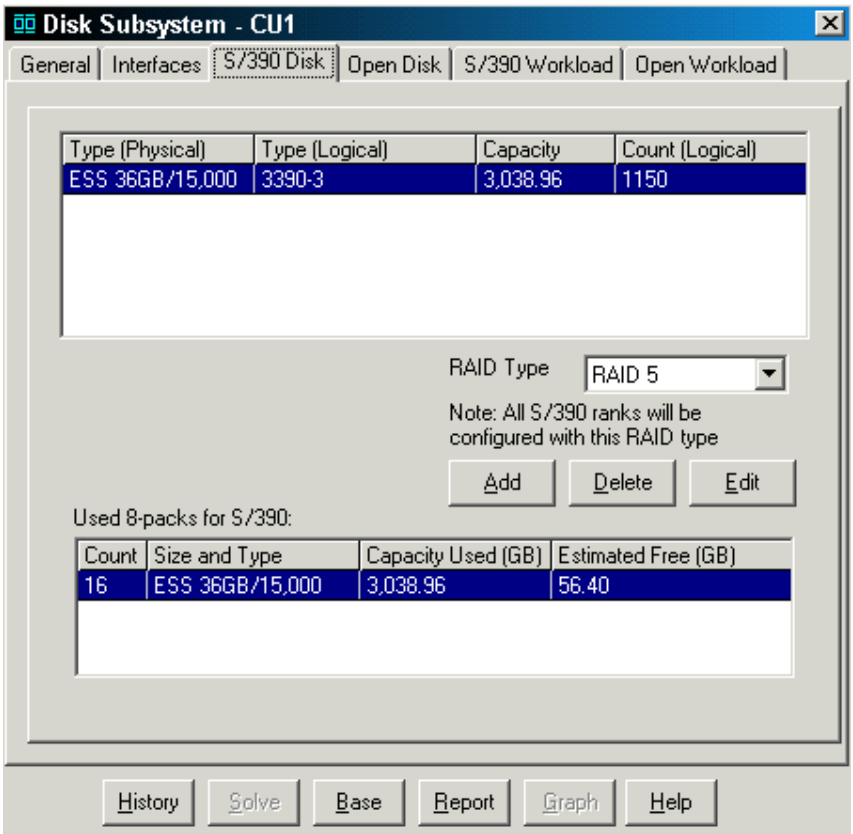


Figure 4-4 DIsk Subsystem - S/390 Disk page

By selecting the **Open Disk** page (see Figure 4-5 on page 89) we can specify the amount and characteristics of the storage capacity attached to a particular open systems server. We

specify the DDMs (speed and capacity), as well as the RAID configuration of the arrays assigned to that particular server. If in the Welcome to Disk Magic dialog panel (see Figure 4-1 on page 86) we would have indicated more than one open systems, then we would be having as many corresponding Open Disk tabs, so we could enter the information for each of those attached servers.

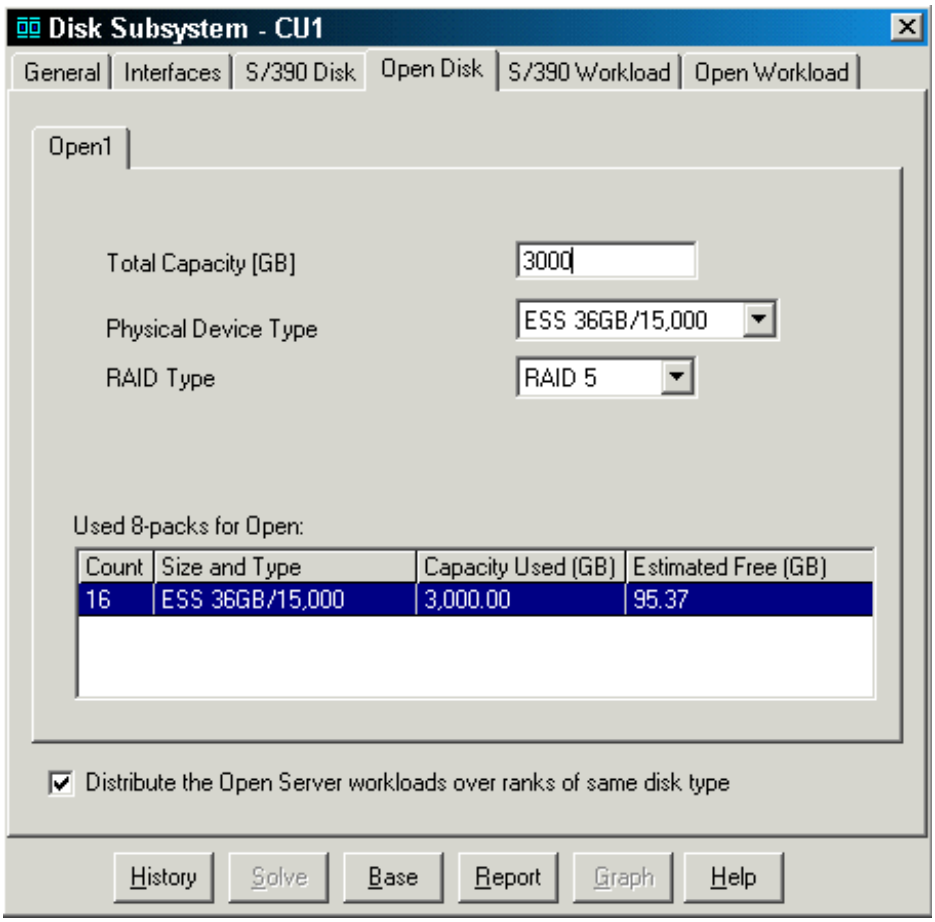


Figure 4-5 Disk Subsystem dialog - Open Disk tab

The type and number of interfaces that connect the ESS and the servers can be specified by selecting the **Interfaces** tab from the Disk Subsystem - CU1 dialog window, as shown in Figure 4-6 on page 90.

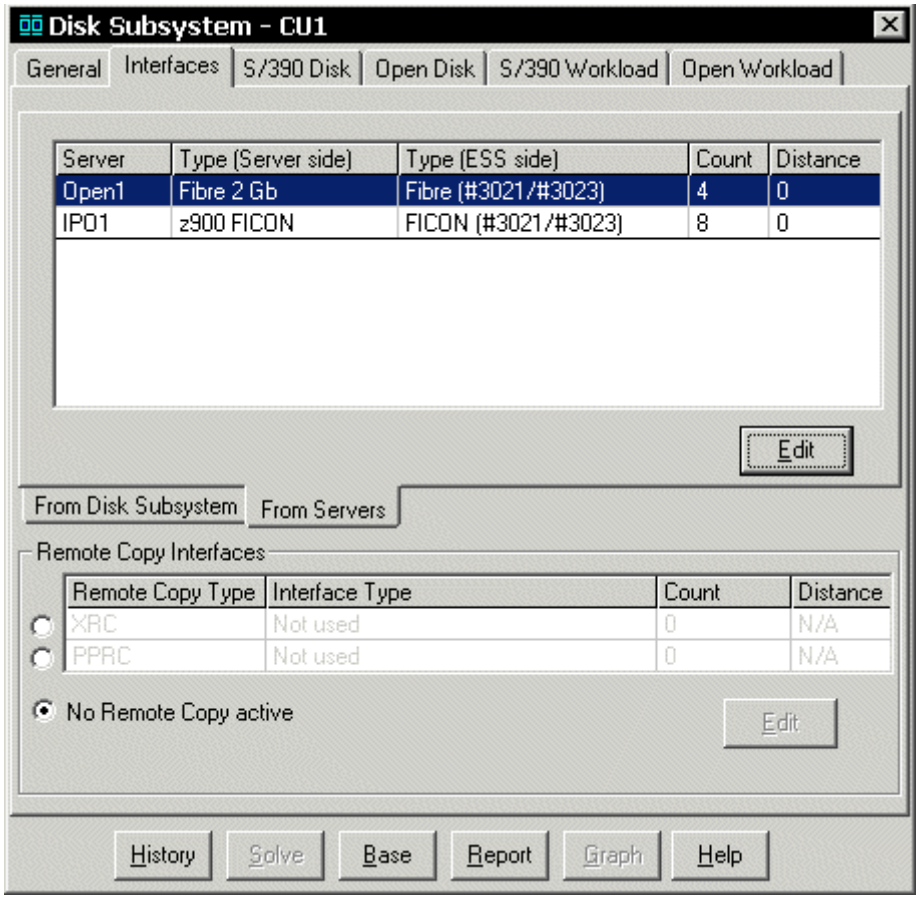


Figure 4-6 Disk Subsystem dialog - Interfaces tab

In the Interfaces tab of the dialog window you specify whether there is any storage being remote copied, PPRC or XRC, and in that case you will be able to specify the number of links and the distance between the primary and secondary ESSs. If you select remote copy for an ESS configuration, you will also enter the percentage of the storage capacity that is mirrored, using the S/390 Workload and/or Open Workload tabs.

S/390 workload

At the beginning of the session, the number of *logical control units* (LCUs) is entered in the General tab of the Disk Subsystem - CU1 dialog panel (see Figure 4-2 on page 87). The number entered here for a particular disk subsystem should correlate to the number of logical control units reported by RMF™ for that disk subsystem. For an ESS, this would be the number of CKD logical subsystems (LSSs) that match one-to-one to the LCUs reported by RMF device activity reports. Also there will be RMF data for each system image that is doing I/O, and this activity must be consolidated in the Disk Magic model.

The S/390 Workload tab of the Disk Subsystem - CU1 dialog panel (see Figure 4-7 on page 91) is used to enter the I/O workload factor values from each of the *system images* (IPOs) for each of the *logical control units* (LCUs). The IPO1 - IPO2 - IPO3 tabs in the example in Figure 4-7 on page 91 appear if you specify 3 in the Number of S/390 Systems field of the Welcome to Disk Magic dialog panel (shown in Figure 4-1 on page 86 with a value of 1). The LCU1, LCU2, LCU3, LCU4, and LCU5 tabs in the example in Figure 4-7 on page 91 appear if you select 5 in the Number of S/390 LCUs field of the General tab of the Disk Subsystem - CU1 dialog panel (shown in Figure 4-2 on page 87).

Using the S/390 Workload tab of the Disk Subsystem - CU1 dialog panel (see Figure 4-7), we use the tabs at the top of the panel to select the system image, and the tabs at the bottom to select the logical control units, for which RMF data will be entered. The values that are entered are I/O Rate, IOSQ Time, Pending Time, Disconnect Time, and Connect Time. This is done for all the system images (IPOs) presented, and their logical control units (LCUs).

From the Cache Statistics section of the panel, and according to the source for your cache statistics data, one button can be selected to feed the cache data into the *base* model: Cache RMF Reporter (CRR), Cache Analysis Aid (CAA), Report Management Facility (RMF), or CMF cache subsystem summary report.

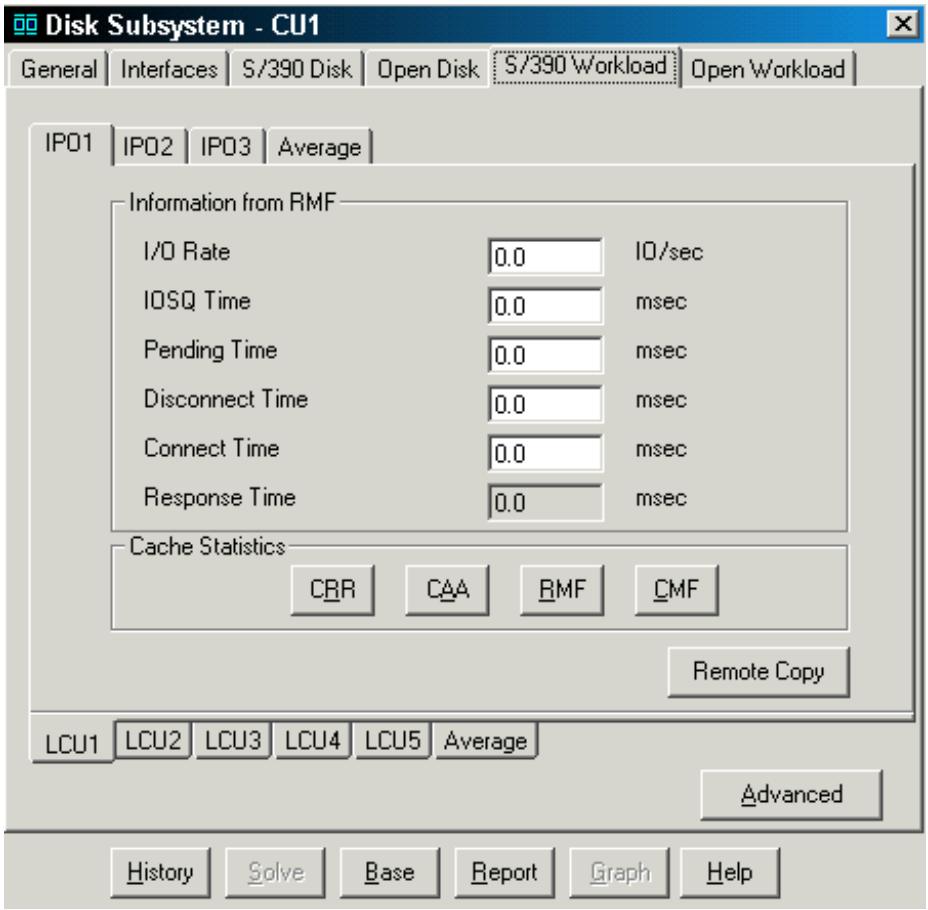


Figure 4-7 Disk Subsystem dialog - S/390 Workload page

The RMF reports that provide the data to complete the S/390 Workload tab fields are:

- Direct Access Device Activity Report
- Cache Subsystem Overview and Cache Subsystem Activity Report

Example 4-1 RMF control statements

```
//SYSIN DD *
DATE(mmddaaaa,mmddaaaa)
RTOD(hhmm,hhmm)
REPORTS(ALL,DEV(DASD))
DINTV(hhmm)
SUMMARY(TOT)
SYSOUT (X)
```

Example 4-1 on page 91 shows the control statements that can be used to get RMF reports with the required information.

Open workload

For open systems workloads, the I/O Rate and the Transfer Size (KB) can be entered in the Open Workload tab of the Disk Subsystem - CU1 dialog panel (see Figure 4-8). Alternatively, if the data you have is the throughput rate then you can enter the MB/sec. instead of the I/O rate information. In the example in Figure 4-8 you can see that the workload information must be completed for Open1 and Open2, that is the number of attached open servers, as per the specification that is initially done in the Number of Open Systems field of the Welcome to Disk Magic dialog panel.

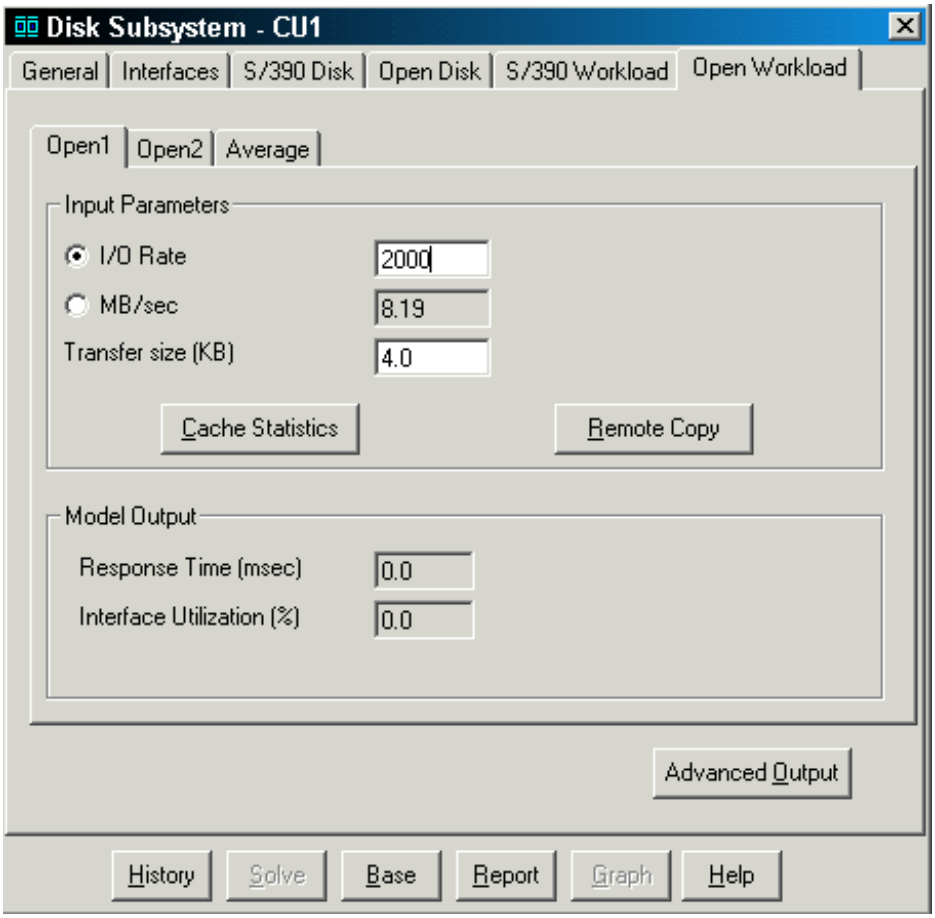


Figure 4-8 Disk Subsystem dialog panel - Open Workload page

Then if you have the necessary information, the Cache Statistics tab can be selected and the Read Write ratio and Cache Hit Ratio information can be entered (see Figure 4-9 on page 93).

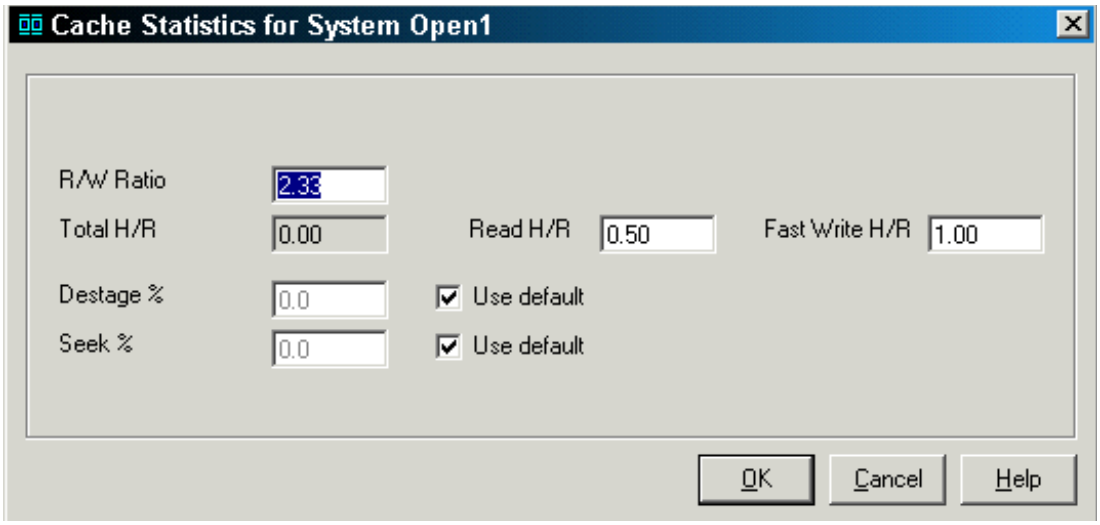


Figure 4-9 Cache Statistics for open workload

You can see (back in Figure 4-8 on page 92) that on the **Open Workload** tab, for each attached server, you can also select the **Remote Copy** button. This allows you to specify the percentage of the server's allocated storage that is using PPRC.

4.1.5 Output reports

You can ask Disk Magic to present the output information in several different formats. Chart data and formats can be selected from the Graph Options dialog panel, illustrated in Figure 4-10 on page 94.

When tailoring the output reports, you can ask for:

- Response time components
- Cache statistics
- Utilization number
- I/O rates
- Operational cost items

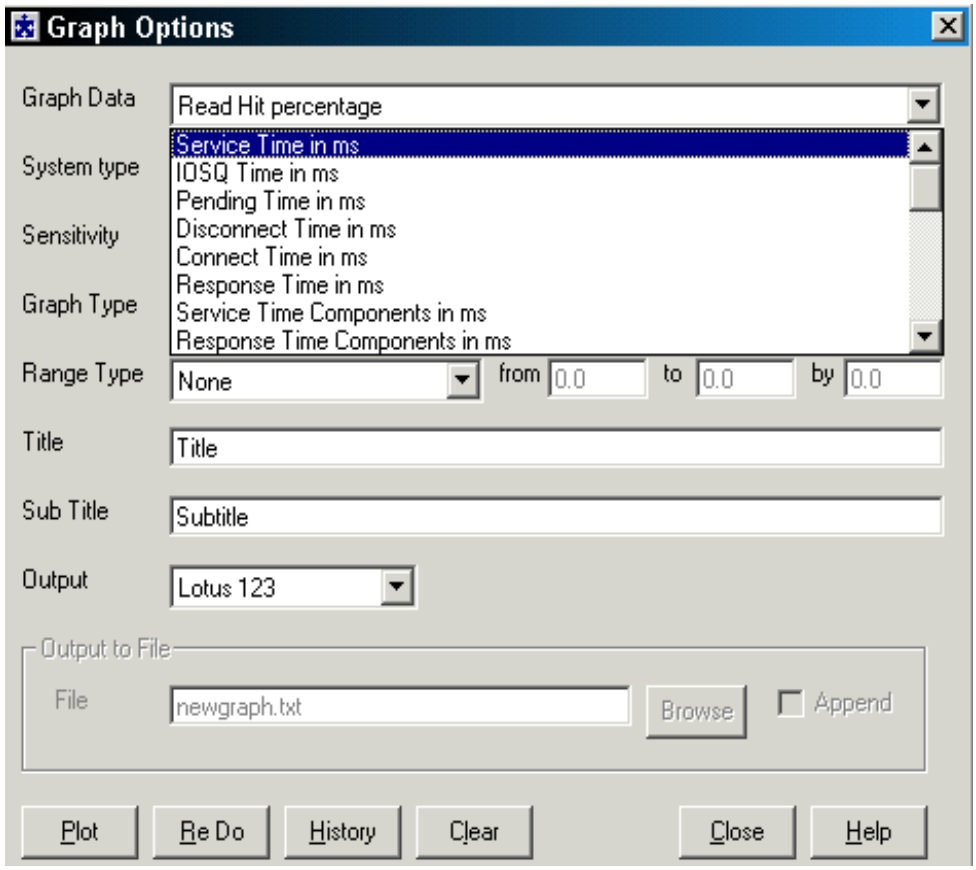


Figure 4-10 Graph Options

Three graph types that are supported:

- Line
- Stacked bar
- Pie chart

Figure 4-11 on page 95 illustrates an example of a Disk Magic stacked bar output report that shows—for a given I/O workload—what happens to the response time when the cache size changes. Notice that Disk Magic is able to present this information, detailing the amount each component adds to the total response time.

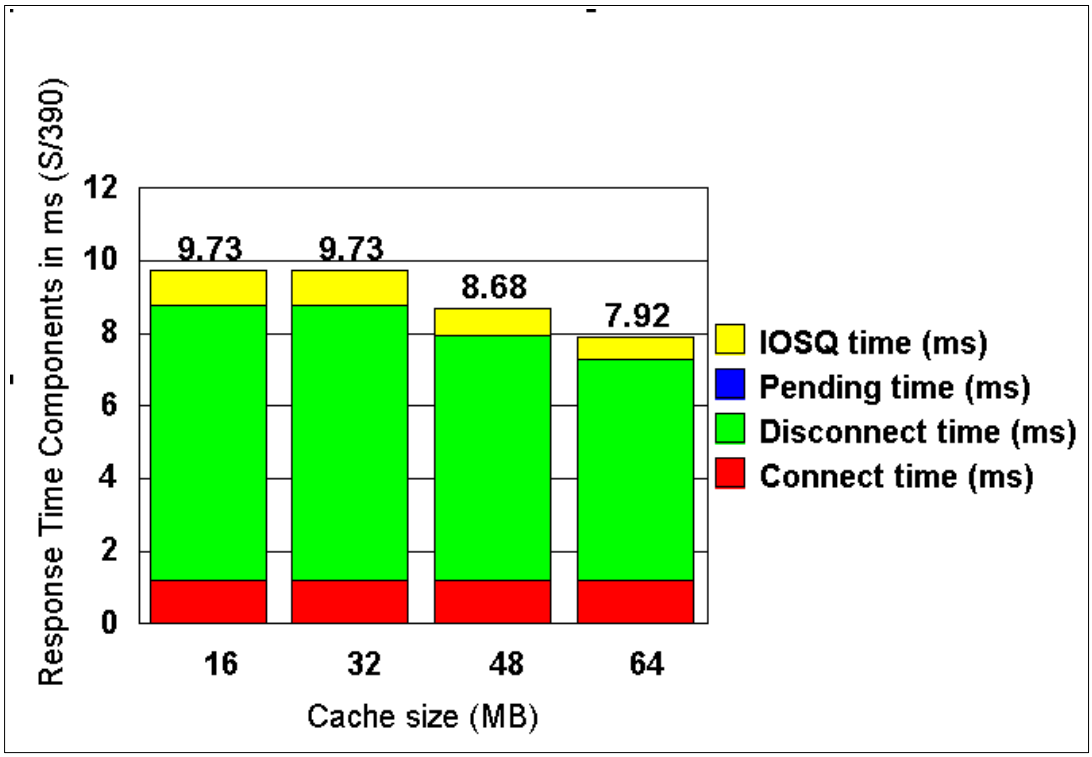


Figure 4-11 Disk Magic output report - Stacked bar example

Figure 4-12 illustrates a sample line graph output report from Disk Magic.

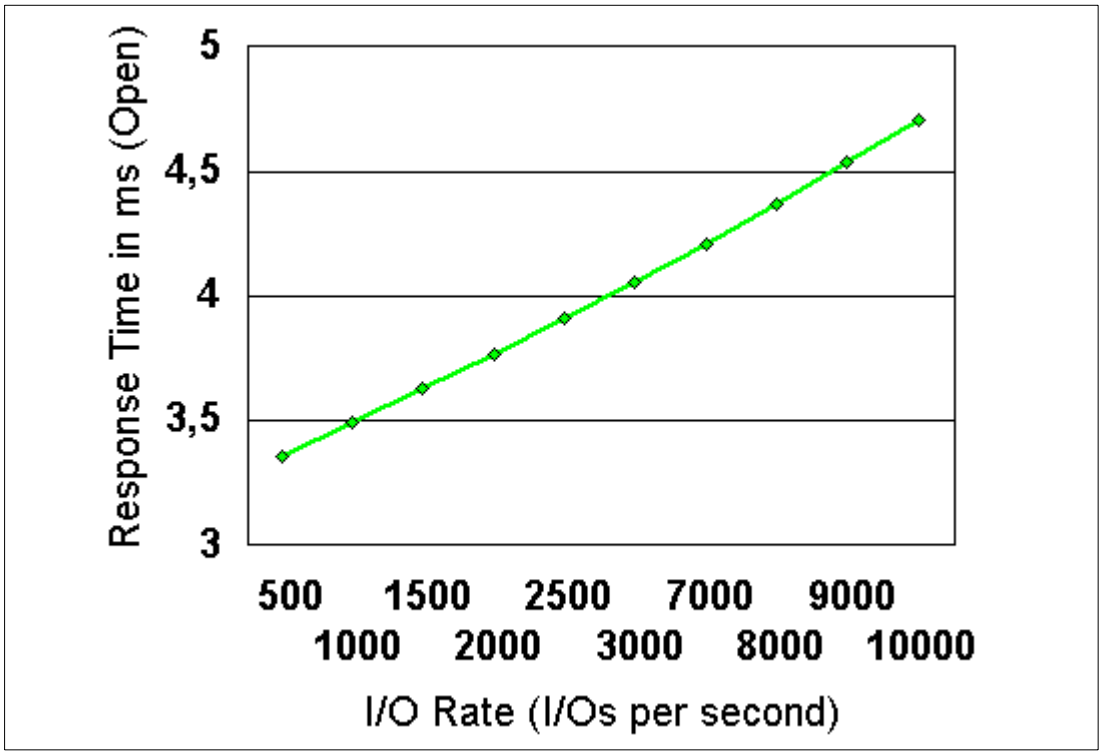


Figure 4-12 Disk Magic output report - Line graph example

4.2 Sequential Sizer

This section describes the Sequential Sizer tool. We also discuss when to use it, and what it will do for you. Finally we show an example spread sheet output, which is a piece of the tool.

Note: Sequential Sizer is for the IBM representative to use. Nevertheless, the ESS capacity and sizing planning is better done when both the customer and the IBM representative are familiar with the tool. Customers should contact their IBM representative to do the Sequential Sizer runs, when planning for their ESS hardware configurations.

4.2.1 Overview and characteristics

The sequential sizer (SeqSizer) is a collection of tools that help estimate the effects on a *sequential workload*, when it is migrated to other storage disk subsystems.This is a S/390 tool only.

Sequential Sizer also allows you to describe the data to be backed up, and estimate the time it would take with today's configuration, and the improvement as you move from your present storage disk subsystem to another.

Predicting the performance of batch (sequential) applications using traditional tools such as Disk Magic is not normally successful because those tools are not designed to handle sequential workloads. Sequential workload observations and results, and the tested approaches used in the past by IBM San Jose, Tucson, and field experts have been captured and integrated into Lotus® 1-2-3 spreadsheets. Using these spreadsheets that accept various levels of input, and a user's guide that provides step-by-step examples, IBM can credibly project *sequential batch* performance of a proposed IBM storage solution.

4.2.2 Spread sheets

The four major SeqSizer spreadsheets do the following tasks:

- Compare utilization of current and new storage controls based on user estimates of the workload.
- Compare utilization of current and new storage controls based on CRR or RMF cache data.
- Merge up to four storage controls to a new storage control based on CRR or RMF cache data.
- Estimate stand-alone volume backup times.

Figure 4-13 on page 97 shows an example where the user-provided input data has been entered into the lightly colored cell boxes. This example shows a migration from the current ESS model F, which has 16 ESCON host adapters, to a configuration with 4 FICON host adapters.

The Sequential Sizer					
					Analysis based on user estimates
	Customer:	Any Customer:	Analyst	R. Edison	
			Current date:	4-Feb-03	
Environment data:	Current CU:	ESS F**/16	GOOD	<--CU type OK?	CurrentLCU ID:
	New CU:	ESS F**/4F	GOOD		1000
Workload description:					
	56	PercentVSAM (4K records, 1 track/access)			Color code:
	44	PercentQSAM (27K records, 2.5 track/access)			User input field
	0	PercentDFDSS dump concurrent with other work			Optional inputs
	0	Percent Other work			Other results
					Other results
	30	CPU component of workload			Intermed. results
	70	IO component			
	108	I/O rate to the control unit			
	4.1	Read/Write ratio			

Figure 4-13 Sequential Sizer example spreadsheet with user input

In Figure 4-14 you can see the analysis results, with the values in the colored cell boxes. The current control unit limits show before the workload analysis of your current system. The new control unit limits show the potential performance increase, if you were to migrate to the new storage disk unit.

Analysis results:					
Current control unit limits:	VSAM workload (4K records, 1 tracks/access)				
	Read limits		Write limits		
Single stream performance	10.7	Megabytes/sec.	8	Megabytes/sec.	
Multi-stream performace	165	Megabytes/sec.	103.8	Megabytes/sec.	
	QSAM workload (27K records, 2.5 tracks/access)				
	Read limits		Write limits		
Single stream performance	13.9	Megabytes/sec.	12.2	Megabytes/sec.	
Multi-stream performace	204.3	Megabytes/sec.	150.4	Megabytes/sec.	
You are currently operating at	5.02%	Of your control unit sequential capability			
New control unit potential:	VSAM workload (4K records, 1 tracks/access)				
	Read limits		Write limits		
Single stream performance	17.7	Megabytes/sec.	10.78	Megabytes/sec.	
Multi-stream performace	169.41	Megabytes/sec.	126.79	Megabytes/sec.	
New control unit potential:	QSAM workload (27K records, 2.5 tracks/access)				
	Read limits		Write limits		
Single stream performance	31.83	Megabytes/sec.	20.71	Megabytes/sec.	
Multi-stream performace	236.88	Megabytes/sec.	152.1	Megabytes/sec.	
Based on the input you may see:	3.00%	reduction in signel job elapsed time if there are no other constrants			
New CU Utilization:	4.07%	At constant I/O rate			
	4.09%	With the increased I/O rate possible with better sequential performance			

Figure 4-14 Sequential Sizer example spreadsheet with analysis output

4.2.3 Input data

The Seqsizer tool automates procedures used by experts in sequential workload analysis. Input from the user is essential. Estimates provided by the user are needed to project the benefits of migrating from one storage control to another. You would use this spread sheet when you have no CRR or RMF cache statistics data available. To use this sheet, you need to know the following data:

- Percent of VSAM (4 K records, 1 track per access)

- Percent of QSAM (27 K records, 2.5 tracks per access)
- Percent of DFDSS concurrent dump work
- CPU content of the workload
- I/O rate to the storage control
- Read/write ratio

4.2.4 When to use Sequential Sizer

If you are considering migrating from one storage disk to another then you should contact IBM to perform an analysis of the batch I/O performance improvement on the new disk subsystem. For example, if you wanted to know if you would have increased performance on sequential workload by migrating from an ESS model F to an ESS Model 800, you could have IBM size the difference in performance with this tool.

This tool is for the IBM representative to use. Customers should contact their IBM representative to have an analysis done.

4.3 Capacity Magic

This section describes the Capacity Magic tool, and discusses when to use it, what and where do you get from it. We also show examples of the tool, and explain the intermix of RAID-5, RAID-10, and different drive capacities.

Note: Capacity Magic is for the IBM representative use. Nevertheless, the ESS capacity planning is better done when both the customer and the IBM representative are familiar with the tool. Customers should contact their IBM representative to do the Capacity Magic runs, when planning their ESS capacity.

Capacity Magic for Windows is a product of IntelliMagic, licensed exclusively to IBM and IBM Business Partners.

4.3.1 Overview and characteristics

IBM Capacity Magic for Windows is an easy-to-use tool that runs on a PC, requires a minimum of input, and calculates the *physical* and *effective* disk capacity for an IBM ESS 800.

Capacity Magic offers a graphical interface that allows IBM to enter the disk drive configuration of an ESS: The number of 8-packs, the type of disk drives (DDMs), and the RAID type. Using this information, Capacity Magic calculates the ESS *physical* and *effective* storage capacity (physical and effective capacity is discussed in 2.6.3, “Disk eight-pack capacity” on page 24).

The *effective* capacity represents the approximate portion of the ESS array that is usable for user data. Given a *physical* capacity, several factors determine what the resulting *effective* capacity will be. The combination and sequence in which the disk eight-packs are purchased and installed, and then logically configured, is one of such factors. Another factor is, for example, if the rank is RAID-5 configured, then the equivalent of one disk drive is used for the floating parity; and if it is a RAID-10 rank then three (or four) disk drives will be used for mirroring.

In summary, not all of the eight disk drives in the installed eight-packs of the ESS will be usable for user data. Table 4-1 on page 99 can be used to calculate the resulting *effective* capacities of the ESS disk eight-packs.

Table 4-1 Disk eight-pack - Physical and effective capacity chart

Disk size	Physical capacity	Raid 10 3X2 array (4)	Raid 10 4x2 array (5)	Raid 5 6+P array (6)	Raid 5 7+P array (7)
18.2 GB	145.6 GB	52.50 GB	70.00 GB	105.20 GB	122.74 GB
36.4 GB	291.2 GB	105.12 GB	140.16 GB	210.45 GB	245.53 GB
72.8 GB	582.4 GB	210.39 GB	280.52 GB	420.92 GB	491.08 GB
145.6 GB	1,164.8 GB	420.78 GB	561.04 GB	841.84 GB	982.16 GB

The following comments apply to Table 4-1:

- Array (4) consists of three data drives mirrored to three copy drives. The remaining drives are used as spares.
- Array (5) consists of four data drives mirrored to four copy drives.
- Array (6) consists of six data drives and one parity drive. The remaining drive is used as a spare.
- Array (7) consists of seven data drives and one parity drive.

The information in Table 4-1 can be used to calculate the effective capacities of an ESS. But, as you will see in the following examples, this can be done more efficiently with the Capacity Magic tool.

4.3.2 Input panels

In this section we will see the Capacity Magic panels and dialogs that are used to enter the configuration information.

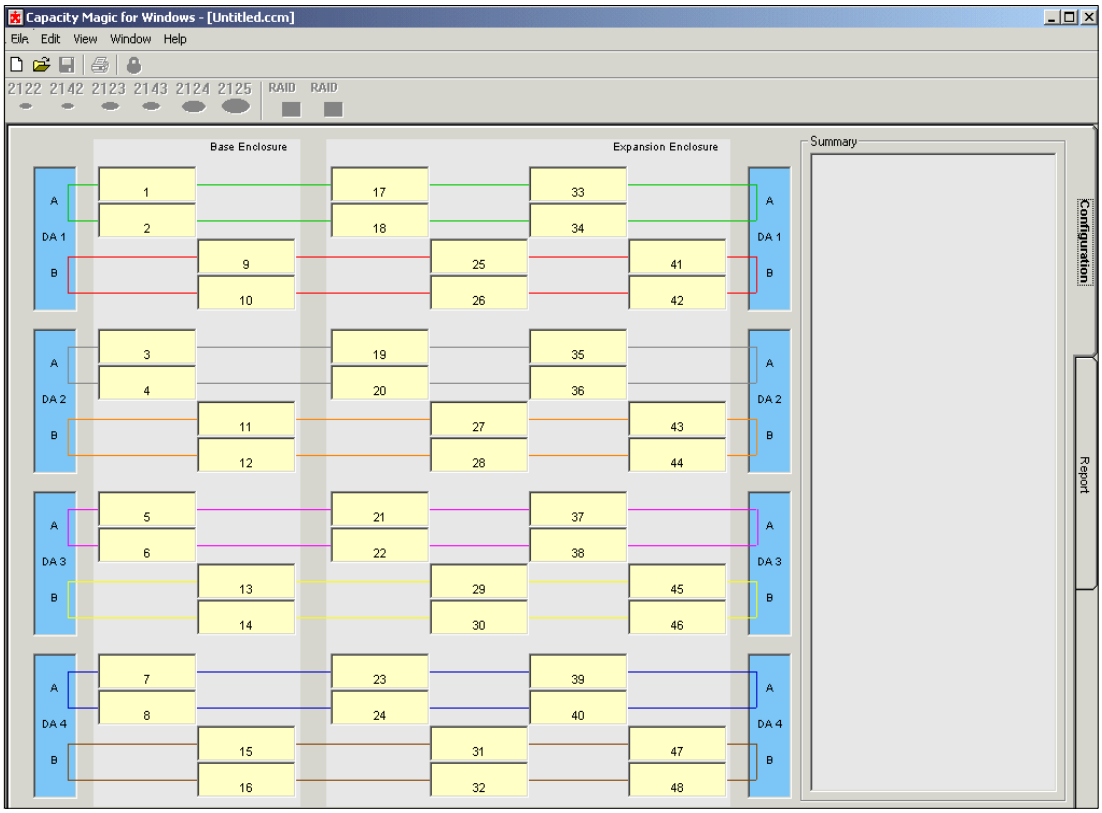


Figure 4-15 Capacity Magic, an example of the graphical tool

Figure 4-15 presents an example of a blank work page that shows the ESS device adapter pairs, the SSA loops, and 48 blank ESS disk groups. By clicking in the selected empty boxes, the disk groups can be populated with different combinations of disk capacity, speed, and RAID configuration options.

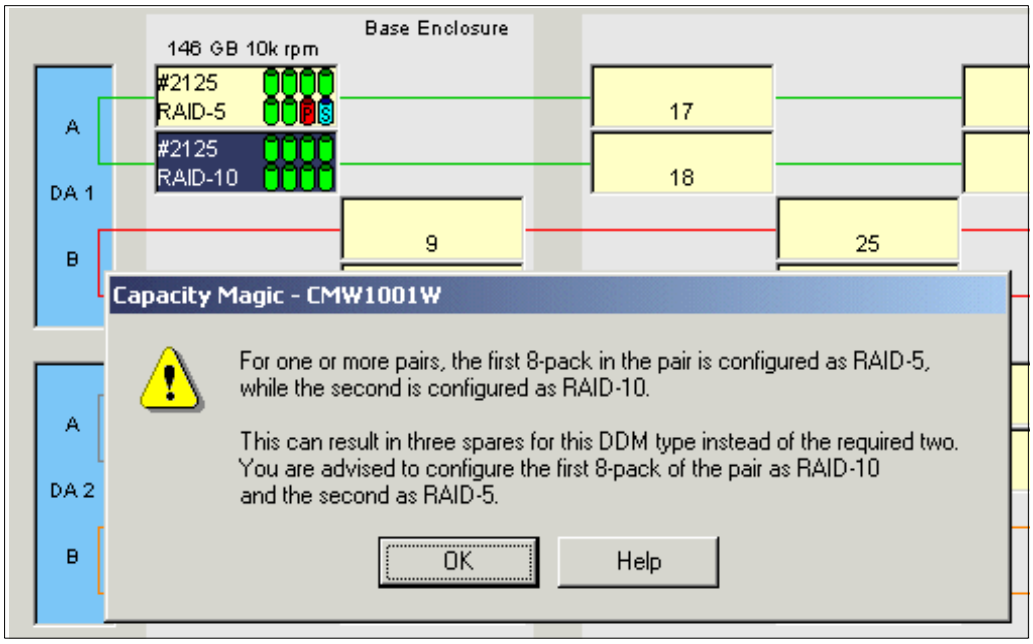


Figure 4-16 Capacity Magic example of RAID-5 and RAID-10 in the first two disk groups

In Figure 4-16 on page 100, the first disk group has been configured as RAID-5, and the second disk group is going to be configured as RAID-10. A screen has popped up to warn the user that this particular choice of configuration will result in extra spares being created. Then it gives a recommendation for optimizing the spare allocation.

Figure 4-17 illustrates the results of configuring RAID-5 and then RAID-10; we end up with three spares.

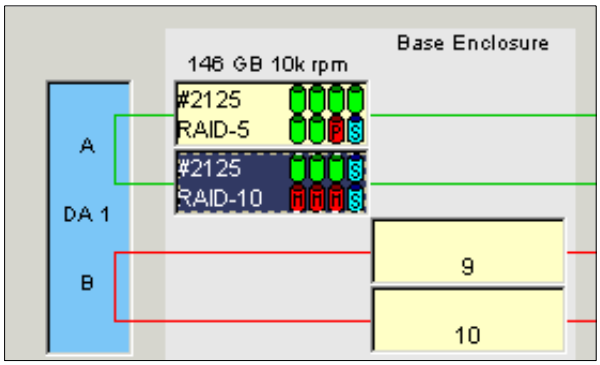


Figure 4-17 Capacity Magic - RAID-5 first and RAID-10 in the second disk group

In Figure 4-18 we see how using the tool allows us for an efficient planning. In this example we now configure first the RAID-10 rank and then the RAID-5 rank, as Capacity Magic advised us. This results in two spare drives, thus not wasting one disk.

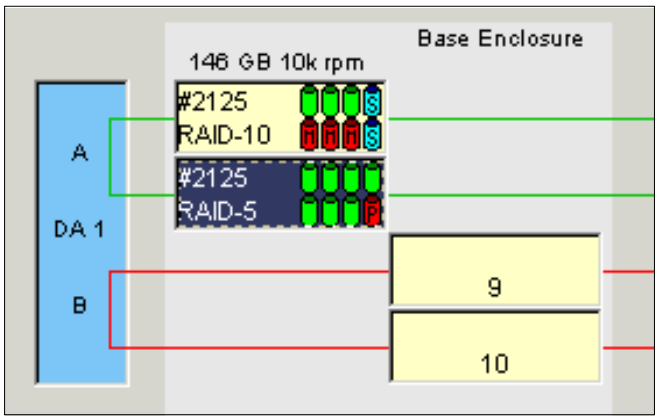


Figure 4-18 Capacity Magic RAID-10 configured before RAID-5

4.3.3 Examples

Just to illustrate the point we will show you an ineffectively planned example, as compared to an effectively planned example when using Capacity Magic.

The ESS hardware configuration we are going to work with consists of:

- 145.6 GB capacity disk drives in the first sixteen disk groups
- 72.8 GB capacity disk drives in the next sixteen disk groups
- 36.4 GB capacity disks drives in the next eight disk groups
- and 18.2 GB capacity disks drives in the last eight disk groups

In this example our goal is to show the difference that proper planning and layout can achieve in effective capacity. Now look at and compare the physical capacity and the effective capacity reports.

The configuration result illustrated in Figure 4-19 shows an ending *effective* capacity of 17,044.52 GB.

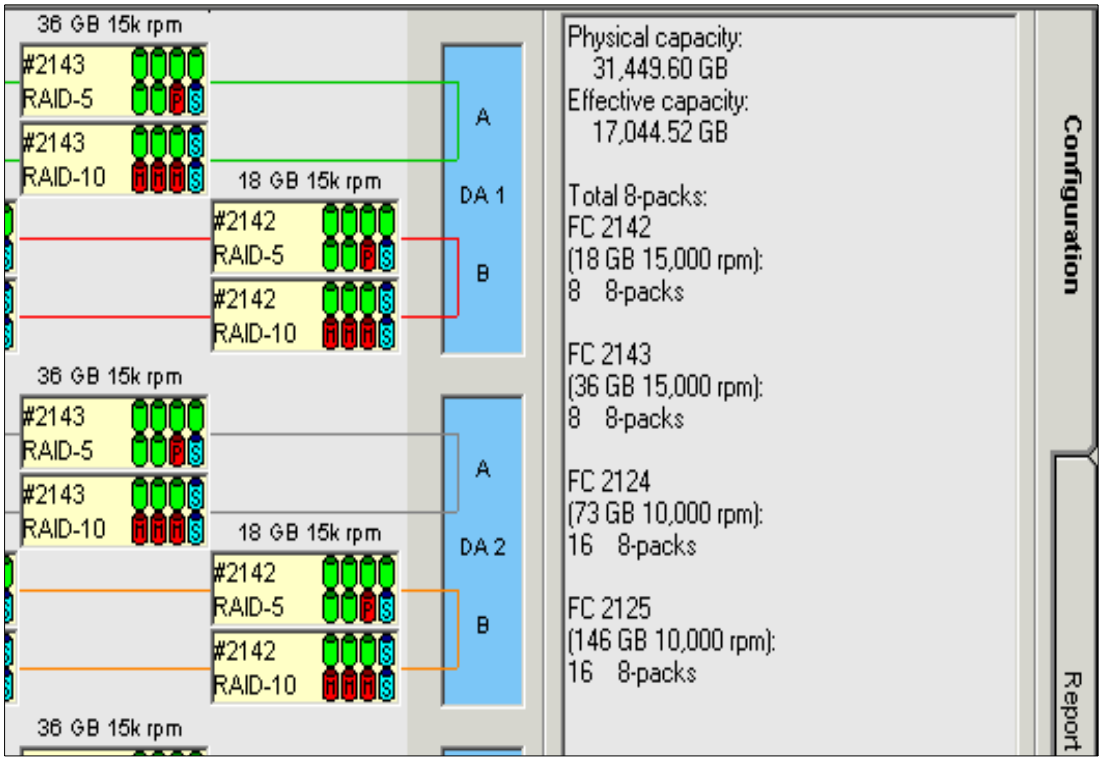


Figure 4-19 Configuration resulting in 17,044.52 GB of effective capacity

The configuration result illustrated in Figure 4-20 shows an ending effective capacity of 19,500.28 GB.

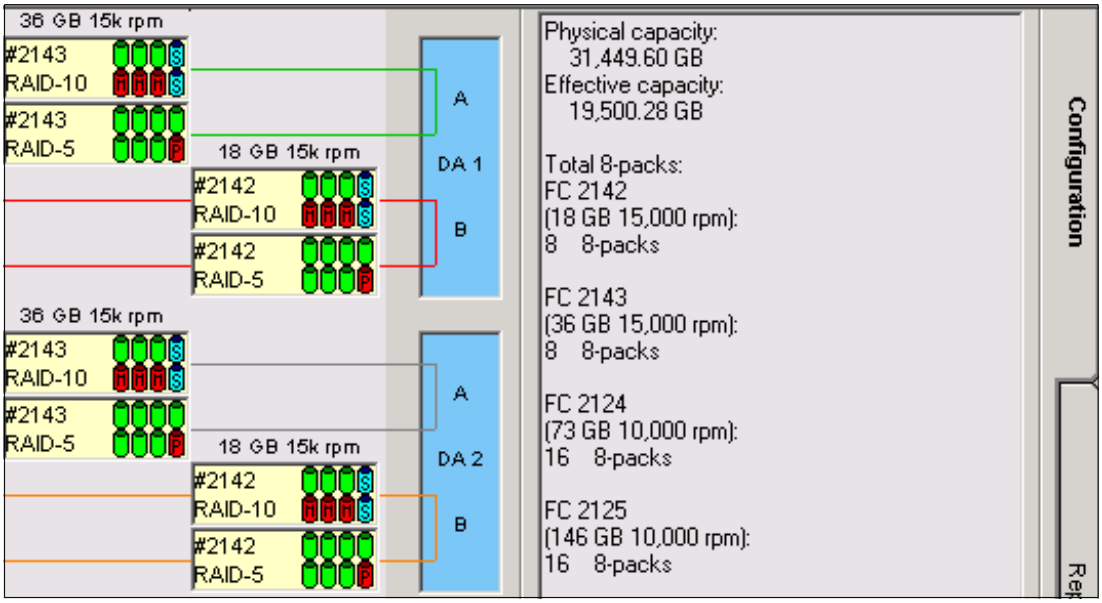


Figure 4-20 Configuration resulting in 19,500.52 GB of effective capacity

You will notice that the *physical* capacity is the same but the *effective* capacity is different between the examples. Proper planning and layout has allowed us to increase the effective capacity by 2455.76 GB.

Summary reports

The Capacity Magic summary reports help determine the efficiency of the layout of drive capacities and RAID setups. Figure 4-21 shows the left side of the summary report. You can see here the information about the number of disk drives in the configuration, by eight-pack characteristics (capacity and speed) and also by RAID configuration. For each, it also gives the physical capacity.

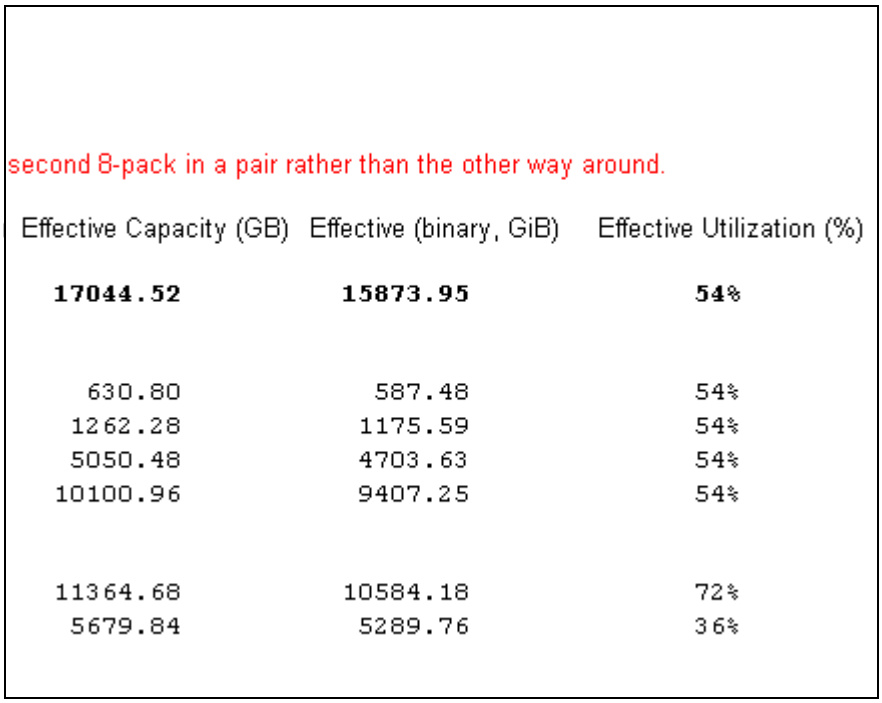
Consul/Capacity Magic v. 1.0.3 - Report for project Untitled
This report was generated on Friday, February 07, 2003 at 11:06:11 AM

Note that 8 loops have three spares for one DDM Type instead of two.
This can be avoided by specifying RAID-10 for the first and RAID-5 for the

Type	Count	Physical Capacity (GB)
All 8-packs	**48**	**31449.60**
By 8-pack type		
FC 2142 (18 GB 15,000 rpm)	8	1164.80
FC 2143 (36 GB 15,000 rpm)	8	2329.60
FC 2124 (73 GB 10,000 rpm)	16	9318.40
FC 2125 (146 GB 10,000 rpm)	16	18636.80
By RAID type		
RAID 5	24	15724.80
RAID 10	24	15724.80

Figure 4-21 Capacity Magic summary report - Left side of the page

On the right side of the report, shown in Figure 4-22 on page 104, you can see information about the effective utilization of the capacity. Thus, in one summary report you can easily realize how efficiently the storage capacity is being planned.



second 8-pack in a pair rather than the other way around.

Effective Capacity (GB)	Effective (binary, GiB)	Effective Utilization (%)
17044.52	**15873.95**	**54%**
630.80	587.48	54%
1262.28	1175.59	54%
5050.48	4703.63	54%
10100.96	9407.25	54%
11364.68	10584.18	72%
5679.84	5289.76	36%

Figure 4-22 Capacity Magic summary report - Right side of the page

4.3.4 Input data

In order to effectively use this tool, you need to provide the following data:

- Total physical capacity of the ESS
- The number of disk eight-packs, and their disks capacity and speed
- The desired RAID-5 and RAID-10 ranks distribution
- The disk capacity that will be using ESS Copy Services

4.3.5 When to use Capacity Magic

When you are planning to use an intermix of drive capacities and RAID array configurations (RAID-5 and RAID-10) in your ESS, you should contact IBM to use this tool. Configuring the IBM TotalStorage Enterprise Storage Server Model 800 becomes more complex due to the intermix of drive capacity sizes and RAID options.

Also remember that for intermixed configurations, in addition to the efficiency considerations, there are guidelines that must be followed (see 2.6, “ESS disks” on page 23). All these considerations and guidelines are automatically managed when using Capacity Magic, so Capacity Magic is especially recommended when combining configuration options.

This tool is for the IBM representative to use. Customers should contact their IBM representative to have an analysis done.

4.4 IBM TotalStorage Expert

The information and descriptions in this section are for a large part based on the information in the *IBM TotalStorage Expert Hands-on Usage Guide*, SG24-6102. We have summarized the information that is relevant for the ESS performance management discussion.

4.4.1 Overview and characteristics

IBM TotalStorage Expert is an application in the IBM TotalStorage software family that helps you manage your IBM TotalStorage Enterprise Storage Server (ESS) and IBM TotalStorage Enterprise Tape Library (ETL) using a Web browser user interface. IBM TotalStorage Expert has two features: The ESS feature, which supports the ESS; and the ETL feature, which supports the tape libraries. The two features are licensed separately.

The IBM TotalStorage Expert is designed to augment commonly used IBM performance tools such as the Resource Management Facility (RMF), DFSMS Optimizer, AIX Performance Toolkit, and similar host-based performance monitors. While these tools provide performance statistics from the *host* system's perspective, the IBM TotalStorage Expert provides statistics from the ESS and ETL system perspective. By complementing other performance tools, the IBM TotalStorage Expert provides a more comprehensive view of performance.

IBM TotalStorage Expert gathers and presents information that provides a complete management solution for storage monitoring and administration. IBM TotalStorage Expert helps storage administrators by increasing the productivity of storage resources.

Both tools can run on the same server, share a common database, efficiently monitor storage resources from any location within the enterprise, and provide a similar look and feel through a Web browser user interface. Together they provide a complete solution that helps optimize the potential of IBM disk and tape subsystems. In this chapter we will only discuss the ESS feature of the IBM TotalStorage Expert, referred shortly to as ESS Expert.

Figure 4-23 gives a graphical view of the IBM TotalStorage Expert product's operating environment.

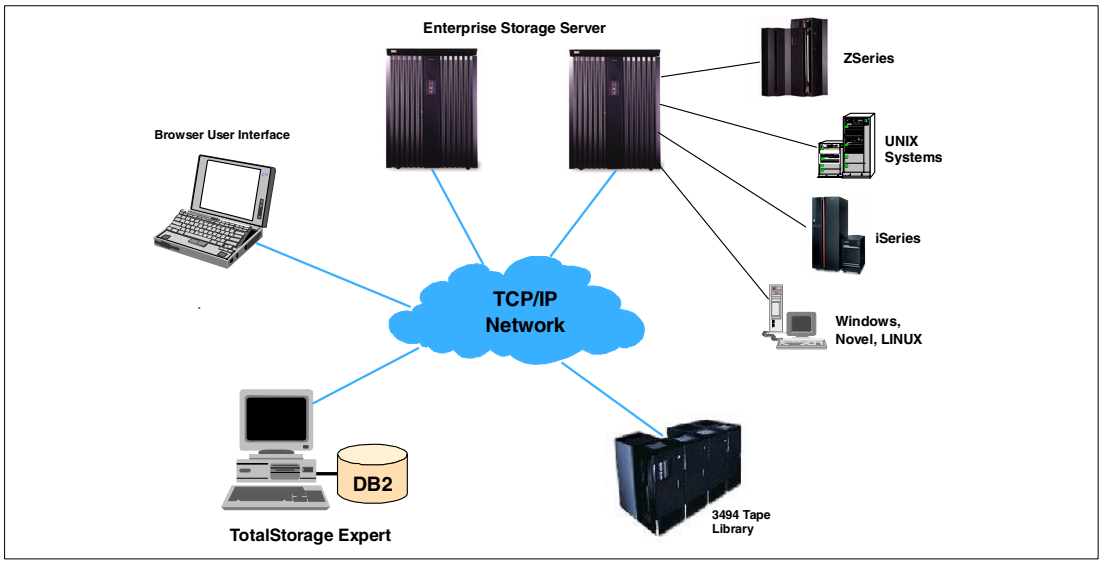


Figure 4-23 IBM TotalStorage Expert operating environment

ESS Expert provides capabilities for performance management, asset management, and capacity management.

- Performance management

  ESS Expert gathers performance information from the ESS and stores it in a relational database. You can generate and view reports of performance information to help you make informed decisions about volume placement and capacity planning, as well as

identify ways to improve ESS performance. Performance information is gathered over intervals of time. The information is summarized in reports that can include:

- Number of I/O requests for the entire storage server in total and separated among various physical disk groupings
- Read and write cache hit ratio
- Cache to/from disk operations (stage/destage)
- Disk read and write response time
- Disk utilization

You can use this information to determine ways to improve ESS performance, make decisions about where to allocate new space, and identify time periods of heavy usage.

- Asset management

  With the asset management capabilities of ESS Expert you can:

  - Discover (in the enterprise network) all of the ESS storage systems, and identify them by serial number, name, and model number.
  - Identify the number of clusters and expansion features on each.
  - Track the microcode level of each cluster.

  This information can save you time in keeping track of your storage hardware environment.

- Capacity management

  ESS Expert provides information that you can use in capacity management. Information includes:

  - Storage capacity, including total capacity, storage assigned to application servers, and storage that is free space.
  - Capacity assigned to each application server, capacity shared with other application servers, and the names and types of each application server that can access ESS.
  - A view of volumes-per-host, which lists the volumes accessible to a particular SCSI- or Fibre Channel-attached host, for each ESS.
  - A volume capability that provides information about a particular volume, and identifies all the application server hosts that can access it.
  - Trends and projections of total, assigned, and free space over time for each ESS.

  In the capacity management report, ESS Expert provides the view of capacity distribution and host connectivity. The logical volumes information, in the form of volume serial number, for each host attached to the ESS storage server is also provided. For IBM TotalStorage Expert V2.1, there is new correlation between the logical volume serial number and the disk name on the host operating system. This is provided with the support of ESS logical volume to host disk mapping.

In this chapter we will discuss only the ESS Expert Performance Management task. Before exploring the Performance Management task in detail, we will explain how the IBM TotalStorage Expert product can help you manage your ESS performance.

4.4.2 Performance management for your ESS

In your role as a system administrator, you may need to analyze performance information for the applications you maintain, or when an end user feels that something has gone wrong with an application's performance. In this case, you will need to determine which parts of your

installation may be degrading performance—for example, application programs, database management systems, host processors, or I/O subsystems.

ESS performance management

By using tools that are available on the host processors, you can investigate how many I/Os are issued, or how much data is being transferred during a certain period of time. However, suppose that you need to know how well your disk subsystems are working. In addition to hard disk drives, today's high-end disk subsystems have many complex features, such as large cache memory and multiple host bus adapter connectivity, so they can provide broad bandwidth to satisfy performance requirements. This multiplicity of features is especially true with the ESS.

Just looking at the physical specifications of the disk drive modules (DDMs) that are loaded on your ESS will not help you to analyze the ESS's performance—you may also need detailed performance data. One of the unique features of the ESS is its ability to maintain various performance statistics on a logical device basis. For example, it records the total number of I/O requests, as well as the number of I/O requests that completed within cache memory. This is referred to as *cache hit*.

IBM TotalStorage Expert provides you with the ability to customize and enable threshold events relating to ESS performance and utilization. In addition, default settings may be used which are derived from the IBM recommended thresholds. You can modify the default threshold settings to create user-defined general ESS thresholds (see customizing of ESS threshold alerts in the publication *IBM TotalStorage Expert Hands-on Usage Guide*, SG24-6102). The user-defined general ESS threshold settings are applied to all ESSs unless you set specific thresholds for an ESS (user-defined specific ESS). This feature will enable you to fine tune your ESS storage specifically to monitor application, cache, and disk performance from a host perspective.

The following parameters can be user-defined:

- Disk utilization
  - Customize this threshold to indicate the percent of utilization that indicates there is a bottleneck for a disk group.
  - Customize this threshold to prevent it from applying to disks with a particular percent of sequential I/O activity. This will prevent the application from identifying instances of high disk utilization where most of the I/Os are sequential as bottlenecks.
- Cache holding time

  Customize this threshold to monitor the time data remains in cache. Data should remain in cache for at least 30 seconds. If it does not, this indicates a possible bottleneck.
- NVS cache full

  Customize this threshold to indicate the percent of space temporarily unavailable in NVS memory.

Another IBM TotalStorage Expert performance monitoring feature is the Activate Simple Network Management Protocol (SNMP) Function panel to configure SNMP managers. This enables your ability to receive alerts and notify you of thresholds that have been exceeded and exception events that have occurred. You also have the option to view the alert logs and filter alert messages to meet your monitoring requirements. This will provide you with the administrative ability to react decisively and pro-actively to your enterprise performance demands.

4.4.3 Operation characteristics

Figure 4-23 on page 105 shows how the IBM TotalStorage Expert works with your ESS. The IBM TotalStorage Expert communicates with your ESS through the TCP/IP network. Therefore, you can gather information on any ESS around the world as long as you have a communication path between the IBM TotalStorage Expert and the ESS through your intranet or the World Wide Web (Internet).

You will need to provide a LAN link between the ESS and the IBM TotalStorage Expert so that the performance information, in particular, can be sent from the ESS to the IBM TotalStorage Expert. If you put the ESS on a private network (for example, one that has only the ESS and the ESSnet console), then you will need to provide a communication capability for the IBM TotalStorage Expert with that private network, or attach the private network to your existing intranet through a LAN router that does address translation.

The IBM TotalStorage Expert itself is installed in a Windows 2000 Server or AIX V4.3.3 or 5.1 operating system environment. However, you do not have to operate the ESS Expert right where it runs, since the user interface of the IBM TotalStorage Expert is the Web browser interface. In other words, you can operate the ESS Expert through Netscape Navigator or Microsoft Internet Explorer from any machine that has network access to the machine where the ESS Expert is installed.

The IBM TotalStorage Expert solicits your ESS to send information about its capacity or performance. When the IBM TotalStorage Expert receives this information, it is stored in tables within a DB2® database. Thus, you can prepare and produce customized reports, in addition to the built-in Expert reports, containing just the information you need using traditional DB2 commands.

4.4.4 Using the IBM TotalStorage Expert

To operate the IBM TotalStorage Expert, you define tasks on the IBM TotalStorage Expert server to perform the required data processing. They can be either one-time tasks that run only once, or tasks that you schedule to run repeatedly on predefined intervals. Use the following general process to begin using the IBM TotalStorage Expert in your environment.

1. Let the TotalStorage Expert know where your ESS resides.

   The first thing you need to do is to let the ESS Expert know about the existence of your ESS resources by defining and scheduling a Node Discovery task. In this task, you define the IP addresses of your ESS clusters. When the TotalStorage Expert runs the Node Discovery task, the task explores your network based on the task definition, and it registers the information in its database.

2. Have the IBM TotalStorage Expert gather information on your ESS.

   The next thing you need to do is to have the IBM TotalStorage Expert gather data from your ESS.

   In order to gather information from the ESS, you need to define and schedule Data Collection tasks. You define when and how long you want the IBM TotalStorage Expert to gather information, and which of your ESSs you want monitored.

   When the IBM TotalStorage Expert runs the Performance Data Collection tasks, it solicits the ESS to send performance information, and interacts with the software on the ESS to stop the data transmission when the task completes. When the IBM TotalStorage Expert receives information from the ESS, it is stored in a DB2 database in a raw format.

3. Let the IBM TotalStorage Expert pre-format the ESS performance data.

After the IBM TotalStorage Expert has collected data through data collection tasks, it needs to pre-format performance data through Data Preparation tasks. In order to view performance reports, you have to define and schedule the Data Preparation tasks.

4. Ask the IBM TotalStorage Expert to create reports to fit your needs.

   Now you are ready to review the ESS reports. Make sure that the Data Preparation tasks have been completed before you try to view ESS performance reports.

   The IBM TotalStorage Expert creates reports for viewing in the Web browser not only in tabular view, but also as graphical charts, such as bar graphs and pie charts.

4.4.5 Performance report types

ESS Expert collects performance data over intervals of time. The length of the data collection interval period is user definable, the minimum being five minutes. The interval sample data is then summarized to hourly and daily totals. You can produce performance reports from these totals. You can also create granular performance reports to view interval period data directly from performance sample data.

ESS Expert produces three main performance report types:

- Disk Utilization report

  This reports the Disk Drive Module (DDM) utilization report.

- Disk<>Cache Transfer report

  This shows the track movement per second from disk to cache, and also from cache to disk.

- Cache report

  The read-hit-ratio, write-hit-ratio, and total-hit-ratios are reported here, as well as cache holding time in seconds.

ESS Expert summarizes performance data to provide reports at different time levels:

- Summary reports show daily averages
- Detail reports show hourly data for the selected day
- Granular reports show interval period measurement data for the selected hour
- Ranked reports show top ten entries for each hour for the selected day

Expert produces detail reports by cluster, device adapter (DA), disk group (DG), and logical disk or volume (LV), but not by the overall ESS box combined. Actually, this is not a limitation of ESS Expert, since the two clusters in an ESS subsystem operate independently from each other. A logical disk (volume) can be accessed from one cluster only during normal operations, and the logical subsystem to which it belongs can only use one set of caches and NVS that belongs to one cluster, and cannot use the other set of caches on the other cluster. The only exception to this is when a failover occurs.

4.4.6 Viewing performance management reports

The information in performance reports can help you make informed decisions about volume placement, capacity, and isolate I/O constraints. In order to view performance management reports, you have to make sure that data preparation tasks have been run. Otherwise, the IBM TotalStorage Expert cannot produce any performance reports.

After you have scheduled data collection and data preparation, you can view detailed information about disk utilization and disk<>cache transfer rates. When you would like to see performance management reports, click **Manage ESS -> Manage Performance -> View**

Reports and the IBM TotalStorage Expert will prompt you to input how you want to create reports (see Figure 4-24).

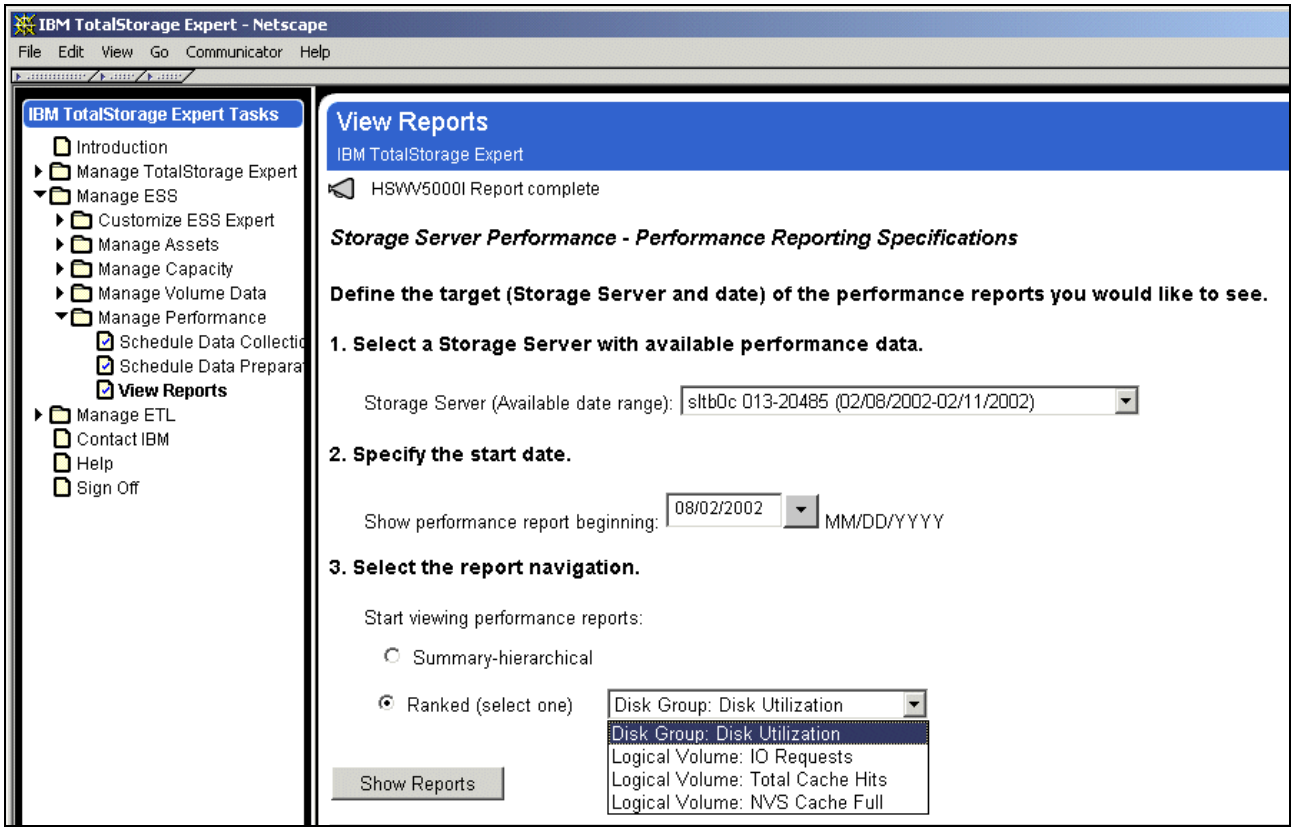


Figure 4-24 Specifying an ESS and a period - Summary Hierarchical or Ranked reports

After you have selected an ESS and reporting period and either the Summary or the Ranked report option, click the **Show Reports** button.

4.4.7 Interpreting the ESS performance reports

ESS performance analysis is an iterative process that requires you to do the following: Collect data over specified periods, analyze performance data, identify problems and conditions that seem out of the ordinary, and decide on a course of action. After making changes, begin collecting data again for the appropriate sample periods and verify and analyze the changes.

The IBM TotalStorage Expert helps you to see the overall performance of your ESSs. The ESS Expert supplies information at the ESS subsystem level; it does not directly connect the host view with the disk subsystem view. Using the IBM TotalStorage Expert in conjunction with host system monitors and available performance tools, you will receive the necessary picture of your ESSs' performance.

In general we recommend:

- Use the Cache Summary report and identify the busiest time interval of the day.
- Use the Ranked Disk Utilization report and identify the busiest adapters and disk group, and check the average response time for the disk lower interface.
- Use the Ranked by Total Cache Hits reports to identify logical disks (volumes) with low cache holding time.

- Use the Ranked by NVS Cache Full report to verify NVS cache full conditions on logical disk (volume) level
- Use the ESS Performance Statistics and Threshold Summary reports to identify the most recent threshold exceptions.

To understand how performance impacts occur, it is important to have a basic understanding of the logical characteristics of the ESS. The ESS Model 800 offers new capabilities to support the optimization of sequential performance and provide increased bandwidth.

Interpreting Disk Utilization reports

Disk utilization reports show you how busy the DDMs are on your ESSs. Figure 4-25 shows a Ranked Disk Utilization report on Disk Group level.

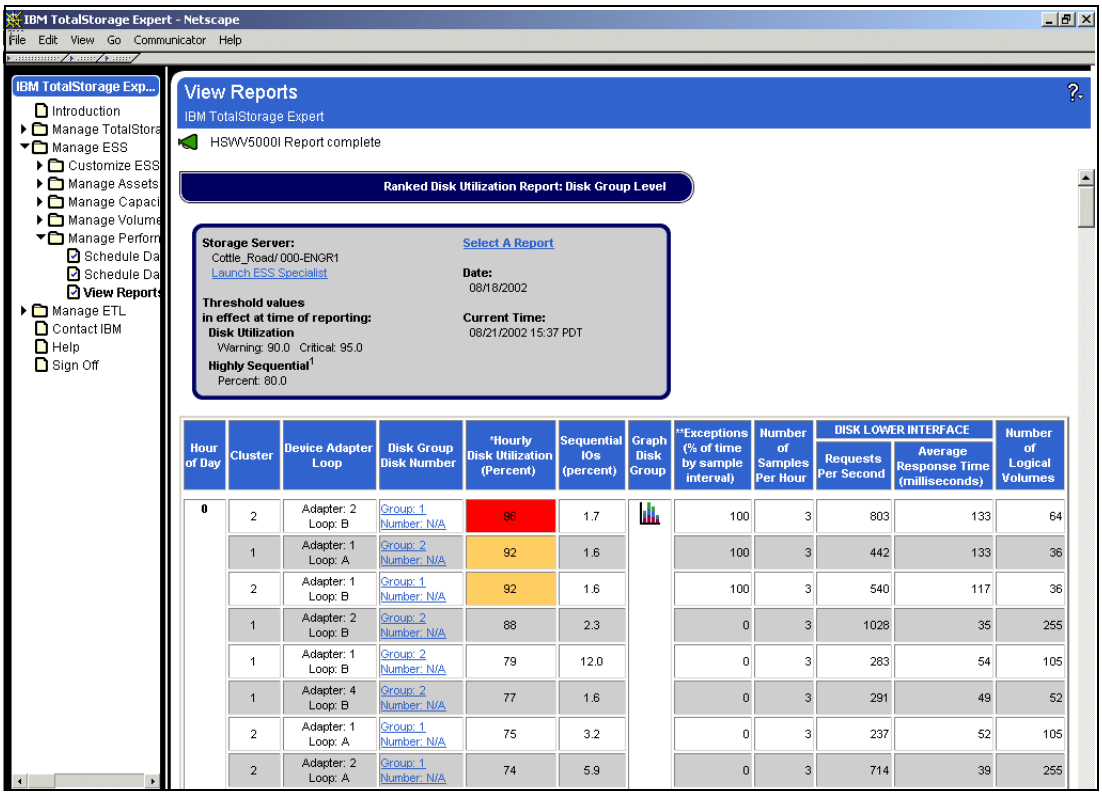


Figure 4-25 Ranked Disk Utilization report

Following is a discussion on several of the columns in this report:

- Disk utilization

  The disk utilization column in the reports indicates how busy the DDMs were during the measurement interval. The disk utilization takes account of all I/O activities against the DDMs, while Disk<>Cache reports do not show all types of destaging and staging operations.

  There is a relationship between disk utilization, cache hit ratio, cache holding time, and percent read requests. When cache hit ratio is low or cache holding time is short, this indicates that the ESS has frequent transfers from DDMs to cache and the disk utilization increases.

  When percent read requests is low, the ESS write activity to the DDMs can be high, and disk utilization may also go up. When cache, hit ratio is high, cache holding time is long,

and percent read request is also high, most of the I/O requests can be satisfied without accessing DDMs.

In general, the ESS should show good performance when disk group utilization is less than 80 percent. The IBM TotalStorage Expert recommends 50 percent as a warning threshold value. Physical I/Os against DDMs are background operations and are asynchronous to the host systems' I/O requests. If you have a volume and disk group utilization greater than 80 percent, this is an indication of high read and write activities caused by a high I/O workload.

Disk utilization greater than 50 percent does not necessarily mean your ESS's performance is poor. According to our IBM TotalStorage Expert analysis of the ESS Model 800, we still have good lower interface response times if disk group utilization is between 50 percent and 80 percent. A comparative analysis of disk utilization values across DAs and ranks within the ESS provides more value in terms of workload balance.

- Lower interface requests per second

  This shows the number of I/O requests made by DAs, including both read and write requests. The number is an indication of internal ESS operations and does not represent host I/O requests per second to the ESS. I/O requests from host systems are based on the characteristics of logical volumes. On the other hand, I/O requests to DAs are based on characteristics of DDMs. When you see a higher disk utilization, this number should also reflect a higher value.

- Lower interface average response time

  This shows an average response time to complete each lower interface request. The number is not the average response time of a DDM. Since a cluster makes various kinds of I/O requests, some of the requests access all DDMs in a rank, while others may involve just one DDM. Taking this into consideration, you cannot use this number to measure performance without knowing how each cluster made I/O requests to the DAs.

Interpreting Disk<>Cache Transfer reports

Disk<>Cache reports show you how much data is transferred between cluster cache and ranks. Figure 4-26 on page 113 shows a Disk<>Cache Transfer Summary report.

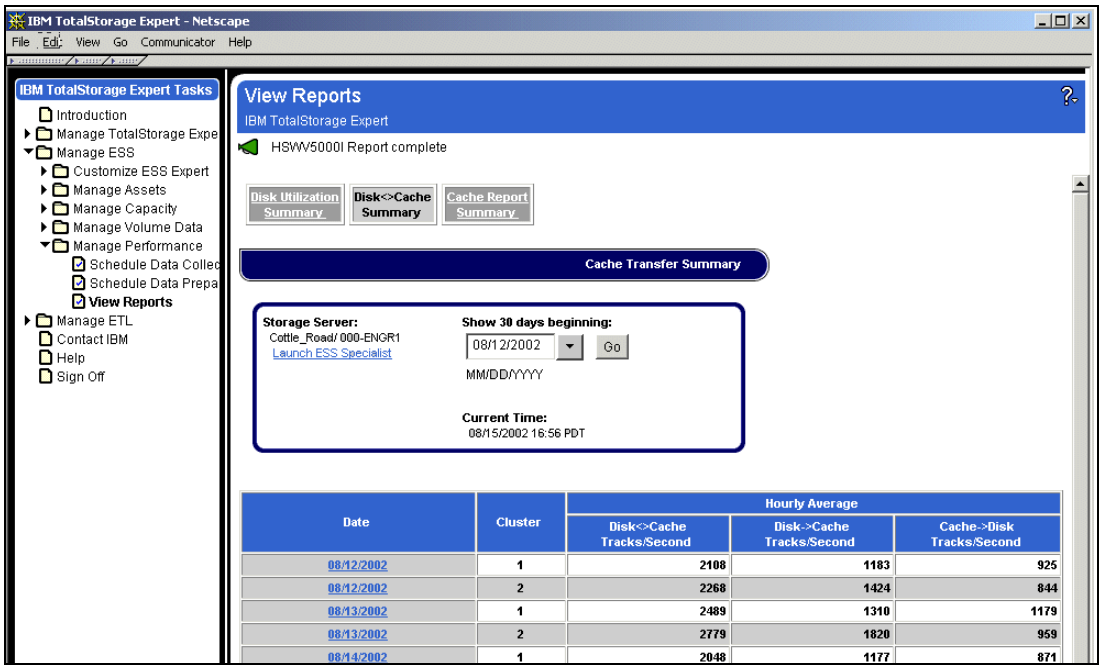


Figure 4-26 Disk<>Cache Transfer Summary report

Now we discuss two of the columns in the report:

- Disk -> Cache

  This column shows the number of data transfer operations from a logical volume—logical disk—in a rank to cache, referred to as *staging*. A cache-miss condition causes a staging operation. The unit of the staging operation is a logical track. A full logical track can contain 56 KB of data when a rank is in 3390 format, and 32 KB when in fixed block format.

  The ESS does not always perform full track staging. Depending on the position where the cache-miss I/O request is issued, the ESS performs *partial track* staging, that is, from a record to the end of track, or from the beginning of a track to right before the record. The ESS counts both full track staging and partial track staging for this number.

  The DA level of this report accumulates numbers for all logical volumes (disks) that are accessed through a DA. To find the distribution, you can use the logical volume level of this report.

- Cache -> Disk

  This column shows the number of data transfer operations from cache to a logical disk (volume) in a rank—*destaging*. The value is the number of logical tracks transferred per second. The ESS destages modified data only.

  There are several conditions that cause the ESS to destage modified data. Typically it is caused by the NVS threshold conditions. If you see a high value for Cache-> Disk while the Percent Read Requests in the corresponding Cache Report is low, you could get the NVS full threshold condition. Like the Disk -> Cache column, the DA level of this report accumulates numbers for all of logical volumes that are accessed through a DA. To find the distribution, you can use the logical volume level of this report.

Interpreting Cache reports

The Cache reports tell you how your ESS has processed I/O requests in terms of cache effectiveness. Read hits occur when all of the data requested for a data access is located in

cache. The ESS improves the performance of read caching by using algorithms to store in cache tracks that have the greatest probability of being accessed by a read operation. Figure 4-27 shows a Cache Detail report.

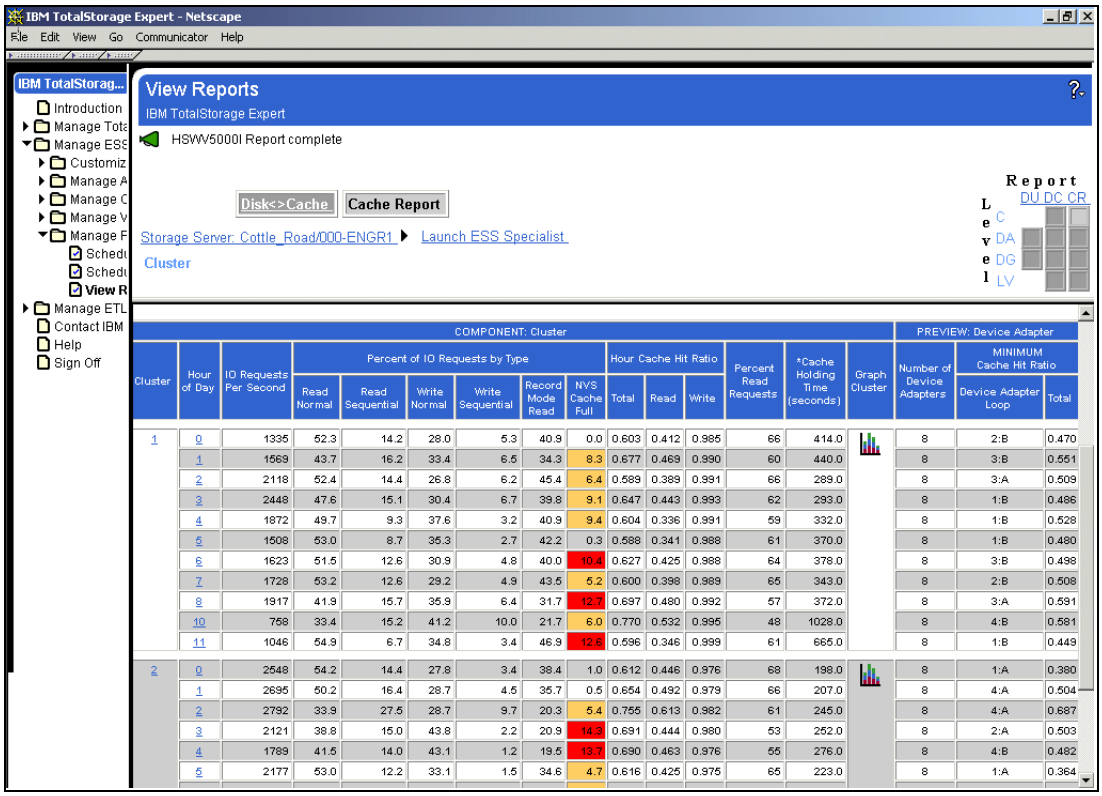


Figure 4-27 Cache Detail report Cluster level

The ESS caching benefits write performance since almost all writes are at cache speeds. Write performance is enhanced by striping. The ESS automatically stripes logical volumes across all the drives in the RAID array. This provides automatic load balancing across the disk in the array, and an elimination of hot spots. The following are important fields in the cache reports.

- I/O requests per second

  This number tells you how many I/O requests are processed. By comparing I/O rates at the device adapter, disk group, and logical disk levels, it is possible to identify where the greatest demand is occurring for storage system resources. To assess whether the I/O rate of a given adapter, disk group, or logical volume (disk) is excessive, see the corresponding component disk utilization report. A small or high I/O rate depends on the I/O workload type you have. However, consider that the ESS Model 800 Turbo can process more than 20,000 open random reads (4 KB I/Os) per second.

  When applications do not perform heavy I/Os to the subsystem, the number of I/O requests per second are small and the disk group utilization is low. There may be large data transfers per I/O, for example, an application transfers 10 MB of data at a time through one 40 MB/s SCSI port. In this case, the ESS can process only four I/O requests per second, per port. Theoretically, 128 I/O requests per second are the maximum value in a configuration with 16 SCSI host attachments and 32 SCSI ports. In general, reading or writing sequential data, or a deferred write from database management systems, transfers a lot of data at a time.

There are several factors that can affect the I/O requests per second:

– I/O contention on the host systems.

  Parallel Access Volume (PAV) for z/OS sharply reduces volume contention (IOSQ) within a system. Multiple reads are executed in parallel and multiple writes are executed in parallel and serialized by extent specification in the Define Extent command. You need to perform additional subsystem tuning actions if your operating system does not allow parallel reads.

  You can check the performance measurement tool available at the host system to determine if there is a performance bottleneck on the host system side.

– Host attachment considerations.

  If you have only one SCSI host attachment, and your ESS can accept only two I/O requests, each is transferring 10 MB data transfer at a time, so you must have a sufficient number of host attachments. Using a performance measurement tool at the host system, for example, iostat for open systems and RMF for S/390, will help you to understand how busy the installed host attachments are.

– The IBM Subsystem Device Driver may not be installed.

  If you have configured your ESS to have multiple paths to an open host, you have to make sure that you have installed the IBM Subsystem Device Driver (SDD), which comes with your ESS. SDD can balance workload among paths, and pump more I/O requests to the ESS; refer to the publication *IBM TotalStorage Subsystem Device Driver User's Guide,* SC26-7478.

To get useful reference data we recommend that you monitor your subsystem during regular work days and peak workload activities when there are no reported user issues or performance constraints. If subsystem I/O performance degrades, user complaints and response times increase, you can compare the actual and historical figures and take appropriate actions, for example, spread workload across clusters, across DAs, and across logical disks where possible. To determine how well the I/O workload is distributed across the ESS resources, you can compare these numbers, for example, within the Cluster and the Disk Utilization reports.

- Read hit ratio

  Read hit ratio shows how efficiently your cache works on the ESS. For example, the value of 1.00 indicates that all read requests are satisfied within the cache. If the ESS cannot complete I/O requests within the cache, it asks for a lower interface to transfer data from the DDMs. The ESS suspends the I/O request until it has read the data. This situation is called *cache-miss*. If an I/O request is cache-miss, the response time will include not only the data transfer time between host and cache, but also the overhead.

  The read hit ratio depends on the characteristics of data on your ESS and applications that use the data. If you have a database and it has the locality of reference, it will show a high cache hit ratio, as most of the data referenced could remain in the cache. If your database does not have the locality of reference, but it has the appropriate sets of indexes, it will also show a high cache hit ratio, as the entire index could remain in the cache.

  A database could be *cache-unfriendly* to applications by nature. An example would be if a large amount of sequential data is written to a highly fragmented file system in an open system environment. If an application reads this file, the cache hit ratio will be very low, because the application never reads the same data, due to the nature of sequential access. In this case, de-fragmentation of the file system would improve the performance.

  You cannot determine if increasing the size of cache improves the I/O performance, without knowing the characteristics of data on your ESS and without using a capacity planning tool like Disk Magic.

We recommend that you monitor the read hit ratio over an extended period of time:

- If the cache hit ratio has been low historically, it is most likely due to the nature of your data, and you do not have much control over this. You can first try to perform de-fragmentation on a file system, making indexes if none exist, rather than considering increasing the cache size.
- If you have a high cache hit ratio initially, and it is decreasing as you load more data with the same characteristics, then adding cache or moving some data to another cluster that uses the other cluster's cache could improve the situation.

- Percent read requests (R/W ratio)

  This number shows the read-to-write ratio. This number depends on how the application programs issue I/O requests. In general, the overall average read-to-write ratio that you can find is in the range of 3 to 5 (75 percent to 80 percent reads). This is why you would see in most cases a larger value (greater than 1 ratio or greater than 50 percent reads) for the read-to-write ratio. A read-to-write ratio of about 1 indicates already heavy write activity and you should probably distribute heavy write activity data to different volumes or disk groups.

  If applications use a database, this is different. A database management system has its own caching mechanism using the host processor's memory (also called the database buffer pool). A database management system can defer the write operation until the modified data occupies a certain amount of this buffer pool. The duration between read and the corresponding write burst to the ESS subsystem can be long. You can see first a high percentage of read requests, then less than 50 percent because of the write requests later on.

  For a logical volume that has sequential files, you need to understand what kind of applications access those sequential files. Normally, these are used for either read-only or write-only at the time of use. The ESS monitors the channel program patterns to determine if the data access pattern is sequential or not. If the access is sequential, then contiguous data is *pre-fetched* into cache in anticipation of the next read request.

  The ESS has the 100 percent write hit function; therefore, all I/O requests against logical disks are completed without accessing DDMs. The most important policy that we have to consider on exploiting write cache function is to protect data integrity. For this reason, the ESS maintains a *two secured copies* policy. This ensures that modified data is stored in two different places in the ESS and a single component failure does not cause the loss of data.

  When the ESS accepts a write request, it will process it without writing to the DDMs physically. The data is written into both cache and NVS. Later, the ESS asynchronously destages the modified data out to the DDMs.

  The ESS's lower interfaces use SSA loops, which provide a high data transfer bandwidth. In addition, the destage operation is designed to avoid the write penalty of RAID-5, if possible. For example, there is no write penalty when modified data to be destaged is contiguous enough to fill the unit of RAID-5 stripe. However, when all of the write operations are completely random across a RAID-5 rank, and the ESS cannot avoid the write penalty, you could get some DDM level of I/O contention.

  The ESS Model 800 provides RAID-10 and the RAID-5 capability. The RAID-10 implementation can provide up to 75 percent greater throughput but less capacity for selected database workloads compared to equal configured RAID-5 configurations.

- Cache holding time

  This number shows how long your data was in cache. Like the read hit ratio, this indicates how efficiently your cache is working. The IBM TotalStorage Expert uses 30 seconds as a

default threshold value. This time is sufficient to ensure that at least cache-friendly data will be able to make effective use of the cache.

After you have reviewed the cluster level report, you can go to the lower level detailed reports to see how these values are distributed to device adapters and logical volumes.

4.4.8 Performance summary reports - Considerations

All average numbers appearing in the summary reports are daily averaged values, so we recommend that you *drill down to detail reports*, even though the summary reports appear to be adequate. For example, let us say you take performance sample data every 5 minutes and monitor for 24 hours (therefore, taking 288 samples a day), and 12 contiguous samples show the disk utilization as 90 percent for an hour, while the rest show zero (0) percent. In this case, the Disk Utilization report shows a 3.75 percent daily average with 90 percent as an hourly maximum. So we recommend checking both of these values.

Be careful when a *configuration change* has occurred. Either the Disk Utilization Summary report or Cache Summary report shows you whether a configuration change on the ESS has occurred during the monitor interval. If this happens, you need to reschedule Data Collection tasks. For example, if you have added a logical volume and begun using it while a Data Collection task is running, performance sample data from the task will not contain any data regarding the logical disk. As a nature of the ESS architecture, access to the volume also affects the ESS resources, such as DDMs and cluster cache. Therefore, performance sample data for the a new logical disk should be taken into account to see how your ESS works accurately.

4.4.9 Viewing performance detail reports

As you go through the performance summary reports, when you click date column, the IBM TotalStorage Expert gives you the respective *detail* reports for that day. For example, if you click a date field on the Cache Summary report, you will see the Cache Detail report. Therefore, we have three types of detail reports: Disk Utilization, Disk<>Cache, and Cache Detail.

Four levels of detail reports per type

The IBM TotalStorage Expert provides four levels of reports under each type. These levels are Cluster, Device Adapter, Disk Group, and Logical Volume.

Figure 4-28 on page 118 shows the relationship between those report levels and the ESS components, as well as its location information appearing in the IBM TotalStorage Expert reports.

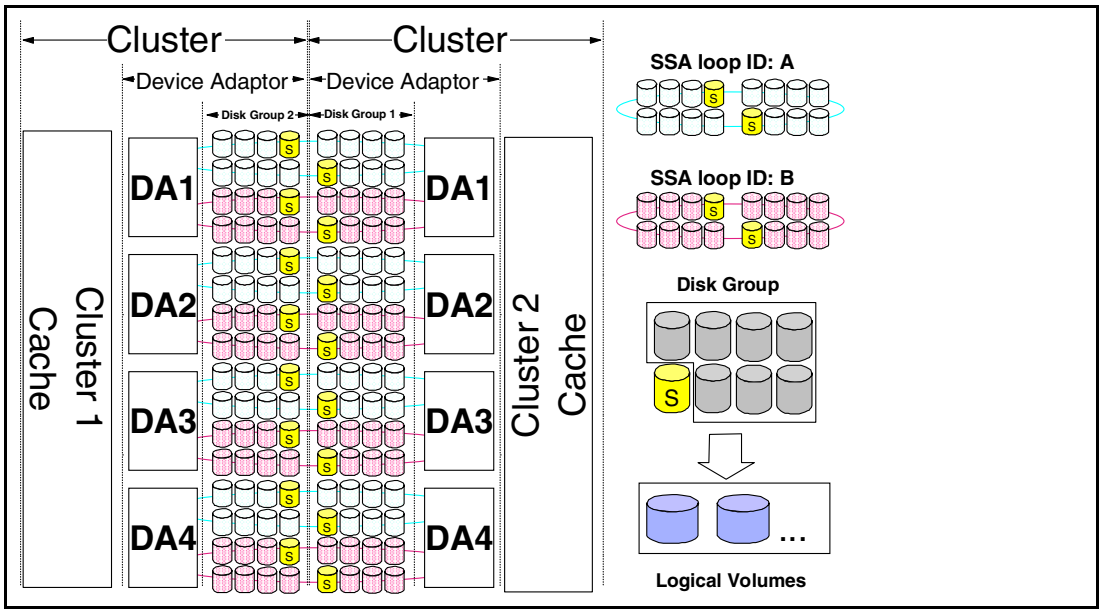


Figure 4-28 Scope of each level of report, and ESS component location information

Cluster reports show entire subsystem performance

An ESS has two clusters, and each cluster independently provides major functions for the disk subsystem. Some examples include directing host adapters for data transferring to and from host processors, managing cache resources, and directing lower device interfaces for data transferring to and from physical disks. Thus, this level of report gives you a performance overview from the viewpoint of a subsystem.

Note that Disk Utilization reports do not have cluster level reports, as Disk Utilization reports deal with physical device performance.

Device Adapter reports show I/O distribution across cluster

Each cluster on an ESS has four device adapters. As you can see in Figure 4-28, a device adapter in a cluster is paired with the other cluster's device adapter, and each device adapter pair has up to two SSA disk loops. Though they make a pair, they usually work independently, and each manages separate groups of DDMs under normal operation. Device adapter level reports help you understand the I/O workload distribution among device adapters and loops on either cluster on an ESS.

Disk Group reports show I/O distribution in a device adapter

A *disk group* is a collection of seven or eight DDMs that are behind the same SSA loop, as represented by the ESS Specialist. It is also a unit of logical disk emulation control, also known as an *array* or ESS *rank*. A rank can be either RAID-5 format or RAID-10. Please refer to 3.1.3, "Arrays, ranks, and disk groups" on page 51, for further discussion. The rank is associated to a device adapter, and the device adapter controls it under normal operation. So this level of reports helps you understand I/O workload distribution among disk groups in a loop in a certain device adapter.

Logical Volume reports show I/O distribution in a disk group

This level of reports helps you understand I/O workload distribution among logical volumes in a disk group. A logical volume (logical disk) belongs to a disk group. Host systems consider a logical disk as if it were a physical hard disk unit. A disk group has multiple logical volumes, depending on your definition. Please refer to 3.1.4, "ESS storage allocation - Logical disks" on page 52, for further discussion.

Note that Disk Utilization reports do not have this level of report, as Disk Utilization reports deal with physical device performance.

Navigation among performance detail reports

There is a collection of small boxes at the top right corner of each detail report, as in Figure 4-27 on page 114. We refer them to as the Performance Navigator Matrix. Figure 4-29 shows an expanded image of the Performance Navigator Matrix.

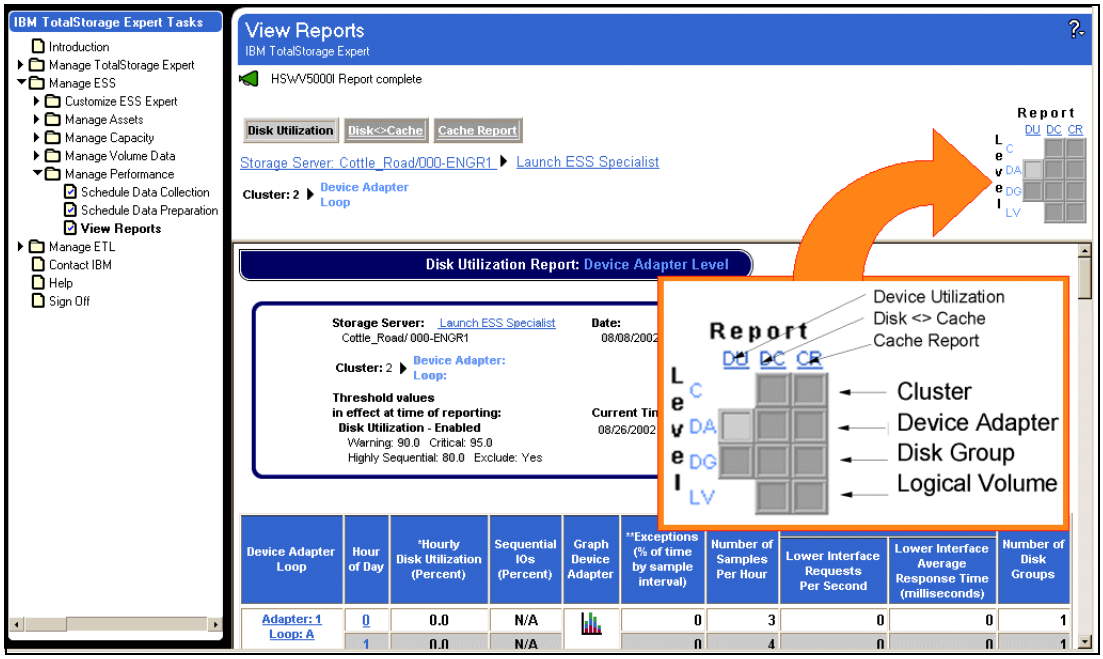


Figure 4-29 Performance Navigator Matrix

Every performance detail report has a Performance Navigator Matrix at the top right corner of the report, and you can simply click whichever one of the boxes you would like to see.

The columns DU, DC, and CR stand for disk utilization, Disk to/from Cache, and Cache report respectively. So these refer to the type of report. The rows C, DA, DG, and LV stand for cluster, device adaptor, disk group, and logical volume, respectively.

The box in a light gray color indicates the report that you are currently looking at. For example, in Figure 4-29 it shows the Device Adapter level of the Disk Utilization detail report. You can click on other boxes in a dark gray color to see the other detail reports.

Creating a graph from a detail report

Every performance detail report shows hourly average statistics and creates graphical reports for you. In order to view graphical reports, just click the graph icon appearing in the detail report of your choice.

Figure 4-30 on page 120 shows an example of graphical reports. We created this graph by clicking the graph icon that appears in Figure 4-27 on page 114.

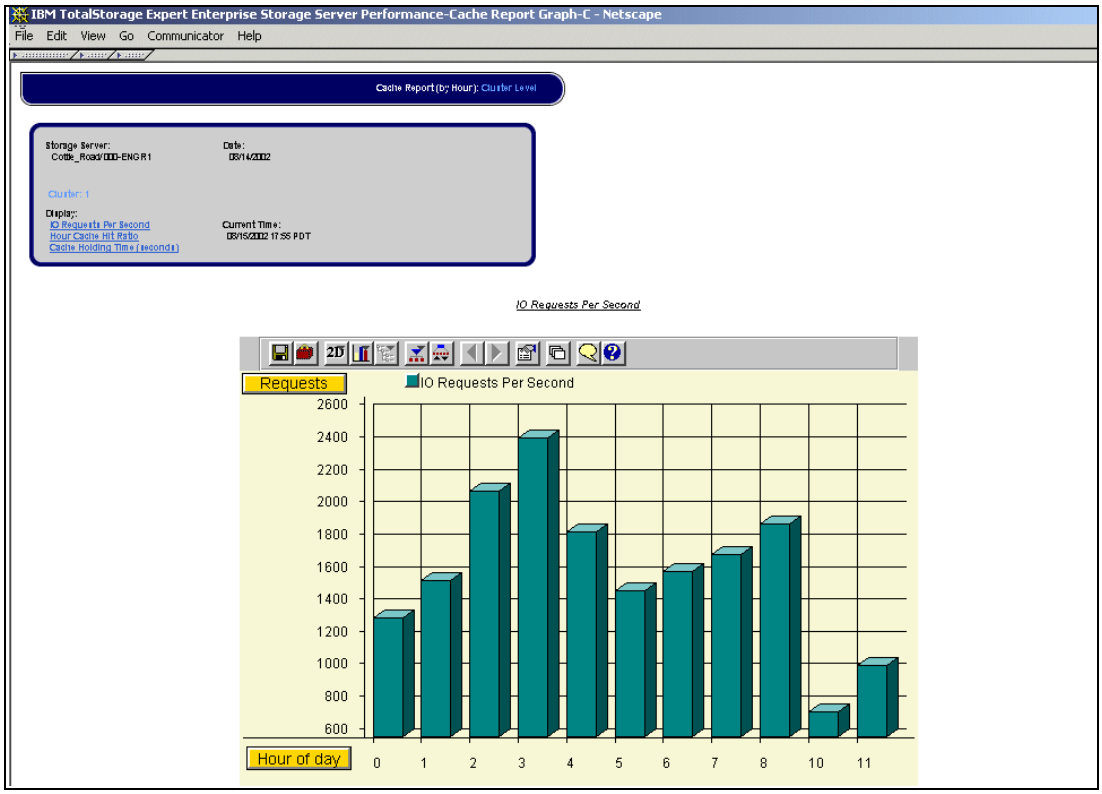


Figure 4-30 An example of a performance graph chart

Viewing a granular report

The IBM TotalStorage Expert has the capability to produce granular performance reports that are created directly from performance sample data. For example, if you collect performance data every 15 minutes, you can view performance reports showing 15-minute intervals.

When you click the Hour of Day column on a detail report, you can see the exact performance sample data taken for the respective hour. Figure 4-31 on page 121 is an example of the granular report.

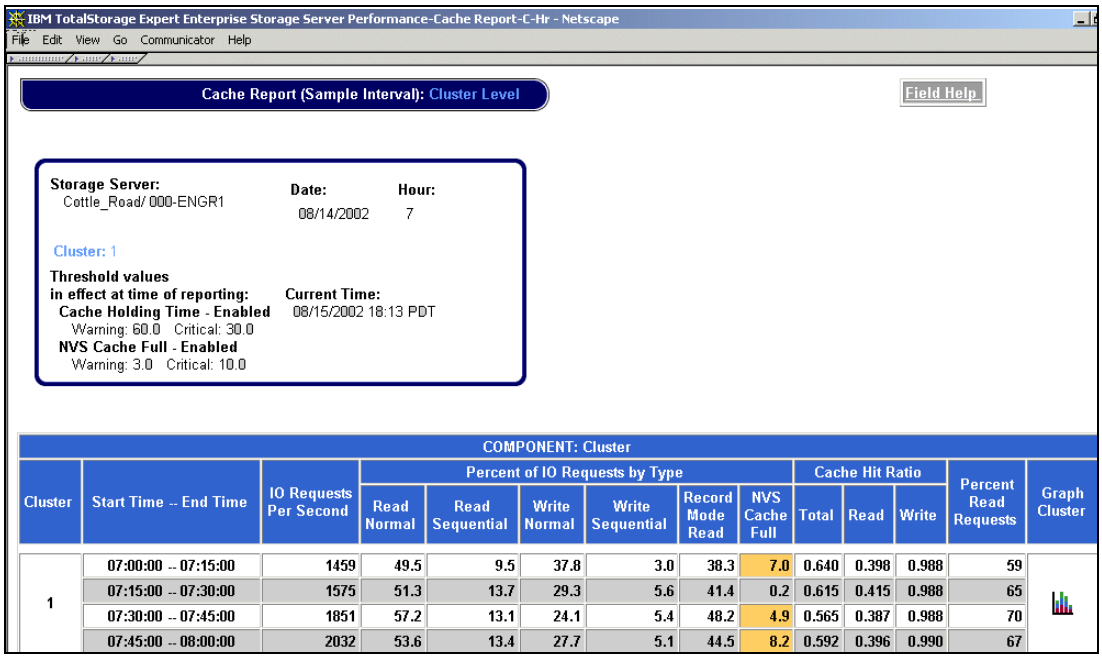


Figure 4-31 An example of a granular report

When you click the graph icon appearing in the figure, the IBM TotalStorage Expert shows you graph charts from the report.

4.4.10 Performance detail reports - Considerations

In this section we discuss some considerations that you should be aware of when viewing performance detail reports.

Go up to the cluster level, then drill down to see other cluster reports

Device Adapter or lower level reports have an affinity with either Cluster 1 or Cluster 2 of the ESS. Therefore, you cannot drill down to see the device adapter, disk group, or logical volume report for Cluster 2 while you are looking at one of these level reports for Cluster 1. If you would like to view the other side of a cluster's data, you have to refer back to the cluster level of reports, then drill down.

Note that the IBM TotalStorage Expert will not break this affinity even if performance data is captured after fail over. For example, if Cluster 1 fails and failover has completed, Cluster 2 will control all of the disk groups (ranks). However, views from the IBM TotalStorage Expert will not be changed.

Number of samples may vary

Assume you have scheduled a data collection task, which asks the ESS Specialist to collect performance sample data every 5 minutes. Some detail reports show you the actual number of samples the IBM TotalStorage Expert has received, and you might expect that the IBM TotalStorage Expert would show that it got exactly 12 samples per hour. But the actual number varies, for the following reasons:

- The ESS Specialist sends performance data on a *best effort* basis.

  The ESS Specialist runs under the ESS, so it cannot sacrifice its functions as an I/O subsystem. If the ESS has higher priority tasks than sending performance data to the IBM TotalStorage Expert, it cannot send the data.

- Network traffic problems may impact the sending of data.

  If very heavy network traffic or some network problem prevents a set of performance data from reaching the IBM TotalStorage Expert in the correct order, the IBM TotalStorage Expert will discard data that is out of sequence. For example, if the IBM TotalStorage Expert gets data for 11:05, 11:15, and 11:10, the ESS will ignore the data for 11:10. However, notice that the ESS keeps updating performance statistics. Therefore, the activity between 11:05 to 11:10 will be accumulated into the data for 11:15. This means only a level of granularity will be lost.

Daily reports show you an hourly average

Regardless of the number of performance data samples available per hour, the IBM TotalStorage Expert shows you average data on an hourly basis. We recommend that you collect as much performance sample data as possible and view detail reports along with granular reports so that you can get a more accurate measurement of the performance of your ESS.

4.5 TotalStorage Expert performance reports and other tools

The IBM TotalStorage Expert provides storage subsystem centric performance data. We receive, for example, the number of I/O requests the ESS has processed, the cache holding time, and the NVS use conditions. We cannot get the host system view from the ESS Expert reports, like I/O activity rate, I/O response time, or data transfer rate. If there is a performance problem with your applications, you could see a delay of batch jobs and slower response times during online transaction processing.

To determine if the I/O behavior is the reason for the problem, you need to gather the information about I/O profiles on the host systems. For example, it is possible that one application cannot get I/O services, while another application dominates I/O services. The I/O response time and its breakdown for each of the logical volumes helps you to isolate the source of the performance problem.

The following sections describe how to use host-based performance measurements and reporting tools, in conjunction with the ESS Expert under UNIX, Windows 2000, and z/OS environments.

4.5.1 Using the ESS reports under UNIX systems

Most application I/O requests against disk subsystems are through either database management systems or file systems. It can be difficult to associate the application or operating system I/O performance with that of the I/O subsystems directly. Why? Because they have their own internal caching mechanisms. An I/O request from an application does not always go directly to the I/O subsystems. You may see an I/O subsystem experiencing poor performance while applications are not affected.

To get host information about I/O subsystems, CPU activities, virtual memory, and physical memory use, you can use the following commands:

- **`iostat`**
- **`vmstat`**
- **`sar`**

These three commands are standard tools available with most UNIX systems. We recommend using **`iostat`** for the data you will need to evaluate your UNIX host I/O levels.

Use data transfer rates from iostat

With enough **iostat** data, you can see trends in the normal workload and then establish a baseline of host-side activity. Focus your attention on the data transfer rate figures.

Correlate with the ESS reports

When user complaints about host system response times increase, and when you see a downward trend on the data transfer rate of a logical volume on a host, you can check Disk Utilization reports. If Disk Utilization reports continue to show low values for the rank, and you still suspect I/O performance after you have investigated the applications, you may need more host adapters on your ESS, or you should implement the IBM Subsystem Device Driver if it has not already been installed. The bottleneck is likely the paths between the host and the ESS. High Disk Utilization on a particular rank does not mean you have a performance issue, unless the activity can be correlated with some host-side performance problem that is identified through proactive monitoring or by a user complaint.

When you see a downward trend on effective data transfer rate for a logical volume on a host and disk utilization is going up, you need to perform further analysis even after you have concluded that your ESS is not performing well. If you have other host systems, you also need to check them, as the source of poor performance could be at one of the other hosts. As we described in “Interpreting Disk Utilization reports” on page 111, cache-unfriendly applications or host systems could be a reason for high disk utilization. For this reason, we suggest that you check Cache reports for all logical disks that are behind the rank showing high disk utilization.

If the logical disk you are concerned about is in the rank, and other logical disks in the same rank show poor cache statistics (such as low read hit ratio, short cache holding time, or low percent read requests), moving the volumes to another rank would be worth considering. This may relieve the performance degradation condition, as the rank or below level of I/O delay probably caused the situation.

4.5.2 Using the ESS reports under Windows 2000 systems

Basically, almost all of the discussion in 4.5.1, “Using the ESS reports under UNIX systems” on page 122, is also applicable for Windows 2000 systems. You can substitute the word `file systems` for `NTFS`. The standard performance measurement tool available for Windows 2000 systems with Service Pack 2 or later is Performance Monitor.

Performance Monitor gives you the flexibility to customize the monitoring to capture various categories of Windows 2000 system resources, including CPU and memory. You can also monitor disk I/O through Performance Monitor.

Using the Performance Monitor

To initiate Performance Monitor, click **Start -> Administrative Tools -> Performance Monitor**. You can add a physical disk or logical disk object to a chart, log, or report, as well as other objects to be monitored. Figure 4-32 on page 124 shows an example of adding a physical disk object to a chart. Remember that the term *physical disk* is from a Windows 2000 perspective. When you define a logical disk in an ESS and assign it to a Windows 2000 system, the ESS logical disk is a physical disk for the host, and you may create one or more logical disk objects on it.

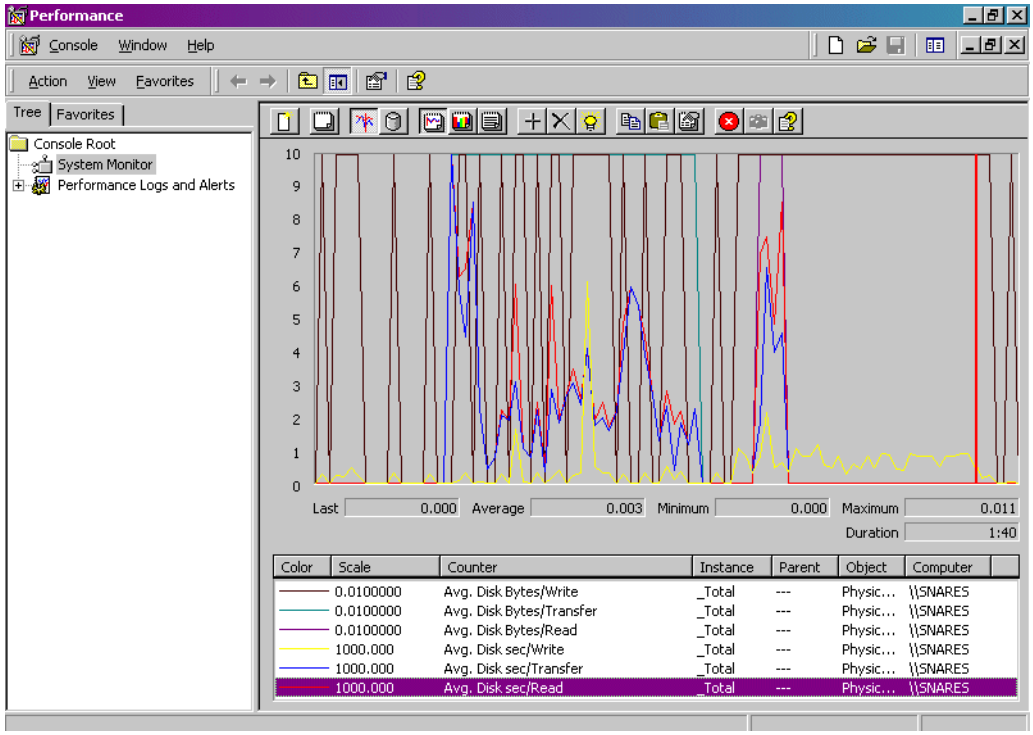


Figure 4-32 Windows 2000 Performance Monitor

The Performance Monitor shows the response time (Avg. Disk sec/I/O). The minimum monitor interval is one second when you log performance data. A one-second response time may not be a valid reflection of your system's performance. When you use the Performance Monitor in real time, you can set the monitor interval in the increment of one millisecond. If you set the monitor interval to one millisecond, the value will be closer to the actual response time.

Increasing the sample count will impact your system's performance and it will also affect the accuracy of these performance counters. This is most likely not acceptable for your production applications. In addition, it is not as convenient for historical analysis, since the real time monitor provides just one screen of data and it wraps around. Although you can log the performance data, the data is saved at a minimum of one second intervals, so the values may not be as accurate.

We suggest that you use the same approach as for a UNIX system, that is, to monitor data transfer rate trends over an extended period of time.

4.5.3 Using the ESS reports in an S/390 environment

The z/OS systems have proven performance monitoring and management tools available to use for performance analysis. RMF, a z/OS performance tool, collects performance data and reports it for the desired interval. It also provides cache reports. The cache reports are similar to Cache reports and Disk<>Cache reports available in the ESS Expert, except that RMF's cache reports are provided in text format. The RMF provides the DA level statistics as SMF records. These SMF records are raw data that you can run your own post processor against to generate reports. RMF is discussed in detail in Chapter 9, "zSeries servers" on page 307.

4.5.4 IBM TotalStorage Expert and mixed operating systems

A benefit of the IBM TotalStorage Expert is that you can analyze both open systems fixed block (FB) and S/390 CKD workloads. When the ESS subsystems are attached to multiple hosts running on different platforms, open systems hosts may affect your S/390 workload and vice versa. If this is the case, taking a look at RMF reports will not be sufficient. You need also the information about the open systems LSSs. The IBM TotalStorage Expert informs you about the cache and lower interface level of I/O activity.

Before beginning the diagnostic process, you must understand your workload and your physical configuration. You need to know how your system resources are allocated, as well as understand your path and channel configuration—for all attached servers.

Let us assume that you have an environment with an ESS attached to a z/OS host, an AIX pSeries™ host, and several Windows 2000 hosts. You have noticed that your z/OS online users experience a performance degradation between 7:30 a.m. and 8:00 a.m. each morning.

You may notice that there are 3390 volumes indicating high *disconnect* times, or high *device busy delay* time for several volumes in the RMF device activity reports. Unlike UNIX or Windows 2000, you may notice *response time* and its breakdown to *connect*, *disconnect*, *pending*, and IOS *queueing*.

Disconnect time is an indication of cache miss activity or destage wait (due to NVS high utilization) for logical disks behind the ESSs.

Device busy delay is an indication that another system locks up a volume, and an *extent conflict* occurs among S/390 hosts or applications in the same host when using Parallel Access Volumes. The ESS's multiple allegiance or Parallel Access Volume capability allows it to process multiple I/Os against the same volume at the same time. However, if a read or write request against an extent is pending while another I/O is writing to the extent, or if a write request against an extent is pending while another I/O is reading or writing data from the extent, the ESS will delay the I/O by queuing. This condition is referred as *extent conflict*. Queuing time due to extent conflict is accumulated to *device busy* (DB) *delay* time. An extent is a sphere of access; the unit of increment is track; usually I/O drivers or system routines decide and declare the sphere.

To determine the possible cause of high *disconnect* times, you should check the read cache hit ratios, read-to-write ratios, and bypass I/Os for those volumes. If you see the cache hit ratio is lower than usual while you have not added other workload on your S/390 environment, I/Os against open systems fixed block LSSs might be a cause of the problem. Possibly fixed block (FB) LSSs on the same side of the cluster had a cache-unfriendly workload, thus impacting your S/390 volumes hit ratio.

In order to get more information on cache usage, you can check the Cache reports in the IBM TotalStorage Expert. If you check the cache statistics of the logical volumes, which belong to the same side of clusters but behind FB LSSs, on the same side of the cluster's cache statistics, you may be able to point out the logical volumes that have a low *read hit ratio* and short *cache holding time*. If you can move the workload of these open systems logical disks, or the S/390 CKD volumes you are concerned about to the other side of the cluster, so that you can concentrate cache-friendly I/O workload to either cluster, this would improve the situation. If you cannot do this, or if the condition has not improved after moving, then balancing I/O distribution among SSA device adapters (DA) is worth considering. This will optimize the staging and destaging operation. Disk Utilization reports in the IBM TotalStorage Expert will help you with this type of analysis.

The approaches for using other tools' data in conjunction with the IBM TotalStorage Expert, as described in this chapter, do not cover all the possible situations you will encounter. But if you basically understand how to interpret the ESS performance reports, and you also have a good understanding of how the ESS works, then you will be able to develop your own ideas on how to correlate the ESS performance reports with other performance measurement tools when approaching specific situations in your production environment.

5

Host attachment

This chapter discusses the attachment considerations between host systems and the ESS for availability and performance. Topics include:

- ESS attachment types
- SAN fabrics: Zoning and cabling
- Configuring logical disks in a SAN
- SDD for multi-path failover and load balancing
- ESS utilities

5.1 Attachment architectures

The IBM TotalStorage Enterprise Storage Server provides heterogeneous host attachments, allowing you to consolidate storage capacity and workloads for the different types of servers. The ESS supports a maximum of 16 host adapters and can be configured for any intermix of the following system adapter types and protocols:

- Small computer system interface (SCSI)
- Fibre Channel protocol (FCP)
- Fibre connection (FICON)
- Enterprise Systems Connection® Architecture (ESCON)

An example of the different host attachment types is shown in Figure 5-1.

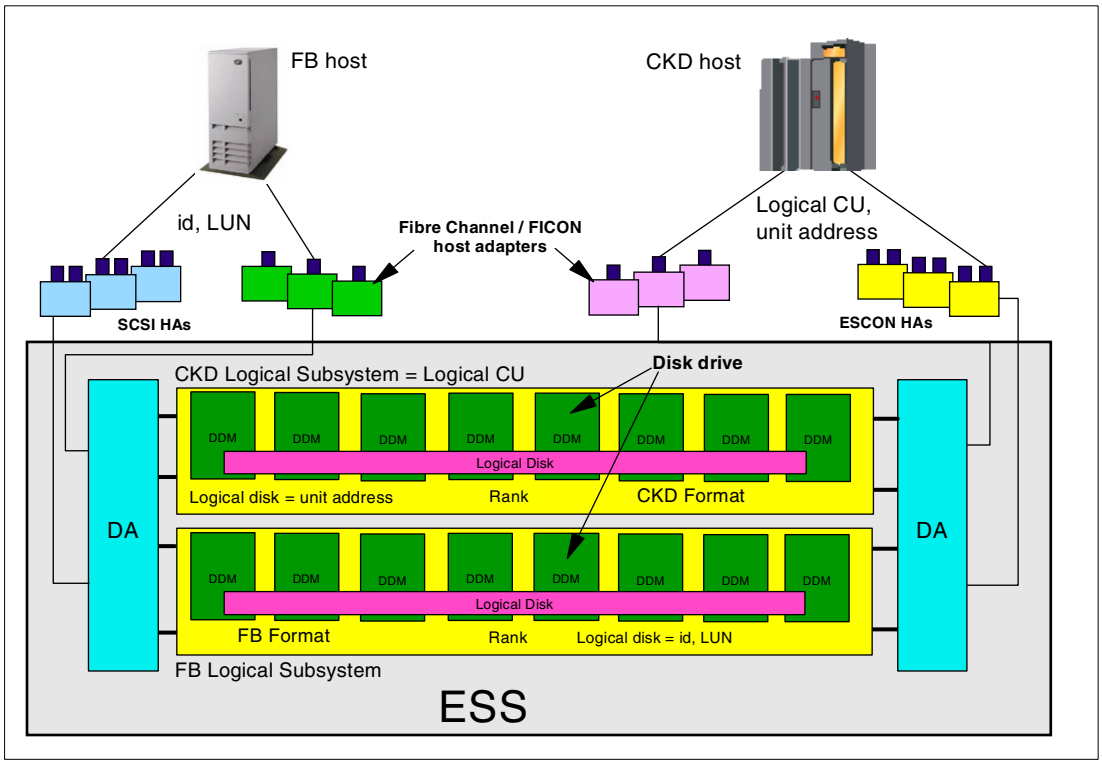


Figure 5-1 ESS attachment types: SCSI, FCP, ESCON, FICON

This chapter provides some tips on host attachments for availability and performance. For a detailed guide on attaching hosts to the ESS please refer to the following publications: *IBM TotalStorage Enterprise Storage Server Host System Attachment Guide,* SC26-7446, and *IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide,* GC26-7444.

The publications, as well as the most current data on ESS support, can be found at:

`http://www.storage.ibm.com/disk/ess/supserver.htm`

5.2 Multipathing

For whichever host attachment method you use, we recommend that whenever possible, you use two or more paths from each SCSI, FCP, ESCON, or FICON host to the ESS, and balance the host connections across ESS adapter bays.

By attaching a host with dual paths to the ESS, you can increase *availability* by avoiding single points of failure. Besides using two or more cables from host to ESS, each connection should be made to different host adapter bays in the ESS. This provides redundancy in the event of a cable failure or an adapter failure in either the host or the ESS.

For example, if a host adapter card in the ESS needs to be replaced, the entire host bay must be quiesced; if your hosts are cabled to two different host adapter bays and running failover software (SDD or native to the OS), the ESS card can be replaced without any disruption to hosts.

Note: If the ESS is not set for 'locally administered' WWPN then the ESS adapters cannot be replaced concurrently. Locally administered WWPNs start with '5000'. The ESS Model 800s are set to local from the plant.

Additionally, I/O *performance* can be improved by configuring multiple physical paths to groups of heavily used logical disks. For open systems, we recommend the use of SDD (subsystem driver) for load balancing and failover. For the zSeries servers the multi-pathing facility is already part of the I/O subsystem—I/O operations can be started on one channel and reconnect on a different one; so the recommendation is to lay out alternate paths to the devices.

Balancing connections across ESS bays

When attaching hosts to the ESS, it is important to use the following recommendations:

- Always spread the host connections across all the ESS host adapter bays if possible.
- If, for example, there are only 2 host bus adapters (HBA) in each of yours open systems servers, then distribute the connections to the host adapter bays in the following sequence: ESS Bay1, ESS Bay4, ESS Bay2, ESS Bay3.

  An example is shown in Figure 5-2 on page 130.

There are three reasons for this recommendation:

1. The 2 bays in each cluster are connected to different PCI busses. The host adapters in bays 1 and 3 share the same internal bus and the host adapters in bays 2 and 4 share a different internal bus. By spreading the adapter connections to the host adapters across the bays, you spread the load and improve overall performance and throughput.
2. If you need to replace or upgrade a host adapter in a bay, then you have to quiesce all the adapters in that bay.
3. ESS host bays 1 and 2 share some power distribution components. Host bays 3 and 4 also share some power distribution components. No matter that you rarely have to power off a host bay to do service actions, it is better to plan for this.

Figure 5-2 on page 130 has an example of two servers, host A and host B, and an ESS with four Fibre Channel/FICON host adapters (one in each bay). In this example of open systems that use FCP point-to-point connections, you would want to establish two paths from server A to the ESS through host bays 1 and 4 and two paths from server B to the ESS using adapters in host bays 2 and 3.

With this configuration, you do not risk losing the paths to one of those servers in the rare event that you must quiesce a host bay. Also, I/O from each host uses both internal busses in the ESS.

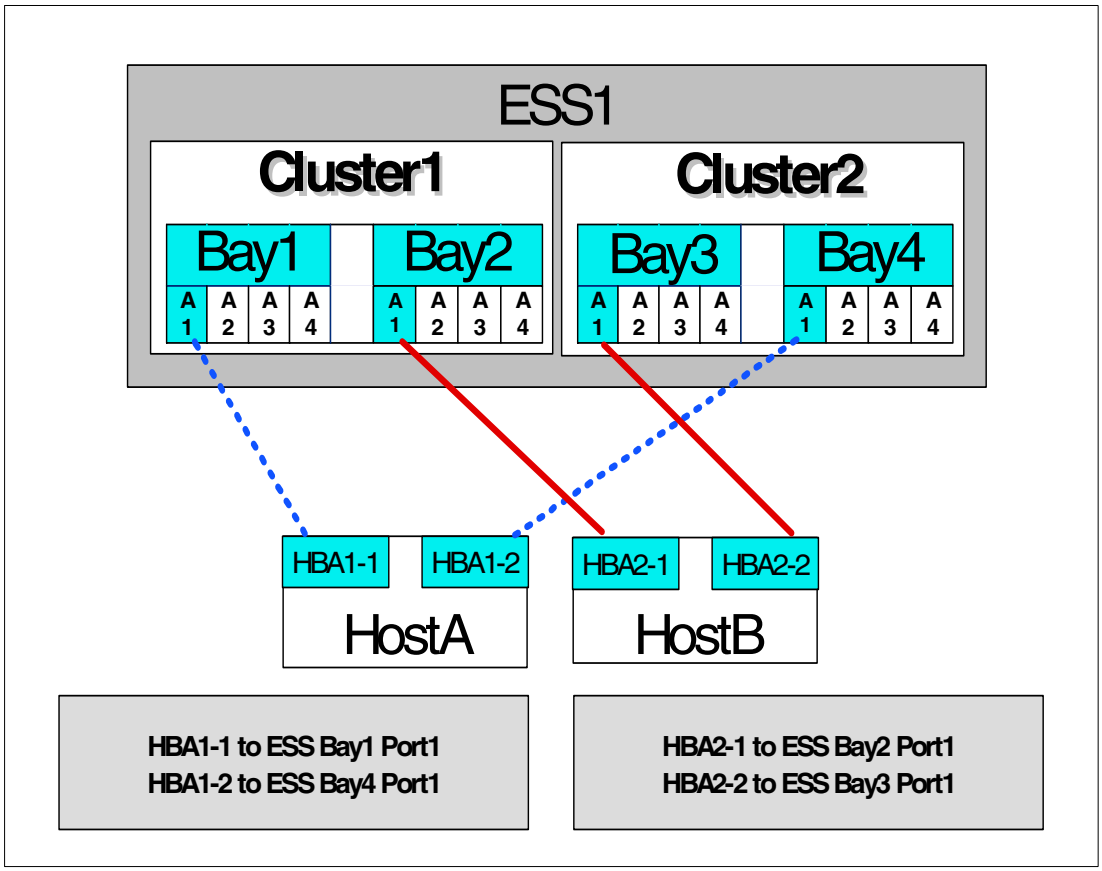


Figure 5-2 Balancing fiber connections across ESS bays in different clusters

Throughout this chapter, the examples of ESS attachment types will balance across ESS adapter bays.

5.3 ESCON

zSeries hosts can attach to the ESS using Enterprise Systems Connection Architecture (ESCON) channels (see 2.9.1, “ESCON attachment” on page 37). With ESCON adapters, the ESS supports:

- A maximum of 64 logical paths per port
- A maximum of 2048 logical paths across all ESCON ports
- A maximum of 256 logical paths per control unit image (or LSS)
- Access to all 16 control unit images (4096 CKD devices) over a single ESCON port
- Up to 32 ESCON links; two per ESCON host adapter
- 17 MB/sec data rate

For the zSeries environments, there are certain specific recommendations in order to fully take advantage of the ESS performance capacity. When configuring for ESCON, consider these general recommendations (refer to Figure 5-3 on page 131):

- Eight to 16 ESCON channels is recommended, to have a good I/O bandwidth.
- Use 4- or 8-path groups (preferably 8) between each zSeries host and LSS.
- Plug channels for an 8-path group into four host adapters (that is, use one HA per bay).
- Each 8-path group should access its logical control unit (or LSS) on one cluster.

- If possible, spread the ESCON connections equally across the four ESS adapter bays. That way, you can connect them to different internal busses. As already discussed, when you spread the adapters across the four bays, you can optimize performance and availability.
- If you need to replace or upgrade a host adapter in a bay, then you have to quiesce all the adapters in that bay. If you spread ESCON connections evenly, then you will only have to quiesce a quarter of your adapters. For example, for an ESCON configuration with 8 ESCON links spread across the 4 ESS bays, then the loss of 2 ESCON links out of 8 may have only a small impact, compared with all 8 if they were all installed in one bay.
- One 8-path group is better than two 4-path groups. This way, both the channels connected to the same host adapter will serve only even or odd LCUs, which is the best, and access will be distributed over the four HA bays.

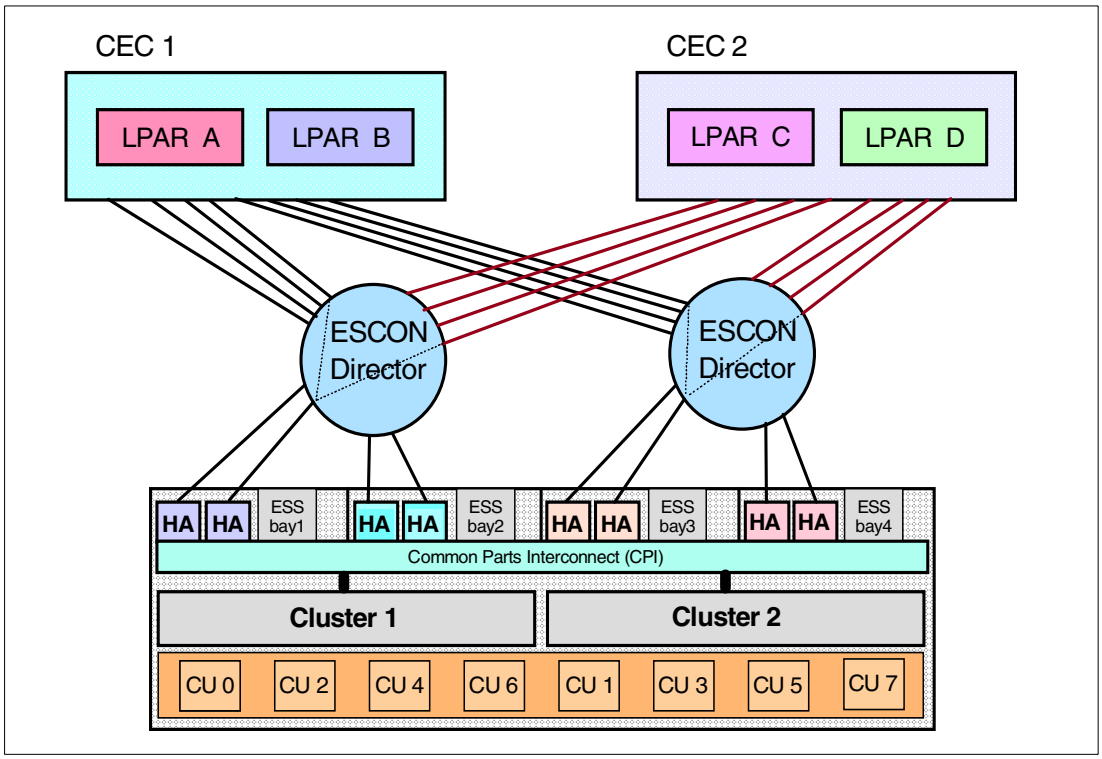


Figure 5-3 ESS ESCON attachment

ESCON cables may be used to attach the ESS directly to an S/390 host, or to an ESCON director, channel extender, or a dense wave division multiplexer (DWDM). They may also be used to connect to another ESS, either directly or via ESCON director or DWDM, for Peer-to-Peer Remote Copy (PPRC).

ESCON cables come in a standard length of 31 meters, two for each ESCON adapter. Cables can also be ordered from IBM Global Services (IGS) in various lengths. The maximum length of an ESCON link from the ESS to the host channel port, ESCON switch, or extender is 3 km. This is using 62.5 micron fiber, or 2 km using 50 micron fiber.

5.4 FICON

FICON is a Fibre Connection used with zSeries servers (see 2.9.4, “FICON attachment” on page 40). The connection speeds are 100–200 MB/sec similar to Fibre Channel for open systems.

You can attach the FICON channels directly to an ESS or you can attach the FICON channels to a Fibre Channel switch. When you attach the FICON channels directly to an ESS, the maximum number of FICON attachments is 16 since that is the maximum number of Fibre Channel/FICON host adapters you can have in an ESS.

When you use an ESS Fibre Channel/FICON host adapter to attach to FICON channels, either directly or through a switch, the port is dedicated to FICON attachment and may not be simultaneously attached to FCP hosts. When you attach an ESS to FICON channels through one or more switches, the maximum number of FICON *logical paths* is 256 per ESS port. The directors provide very high availability with redundant components and no single points of failure.

Figure 5-4 shows an example of FICON attachment to connect a zSeries server through FICON switches, using eight FICON channel paths to eight LSSs—this is half of the possible CU images on an ESS.

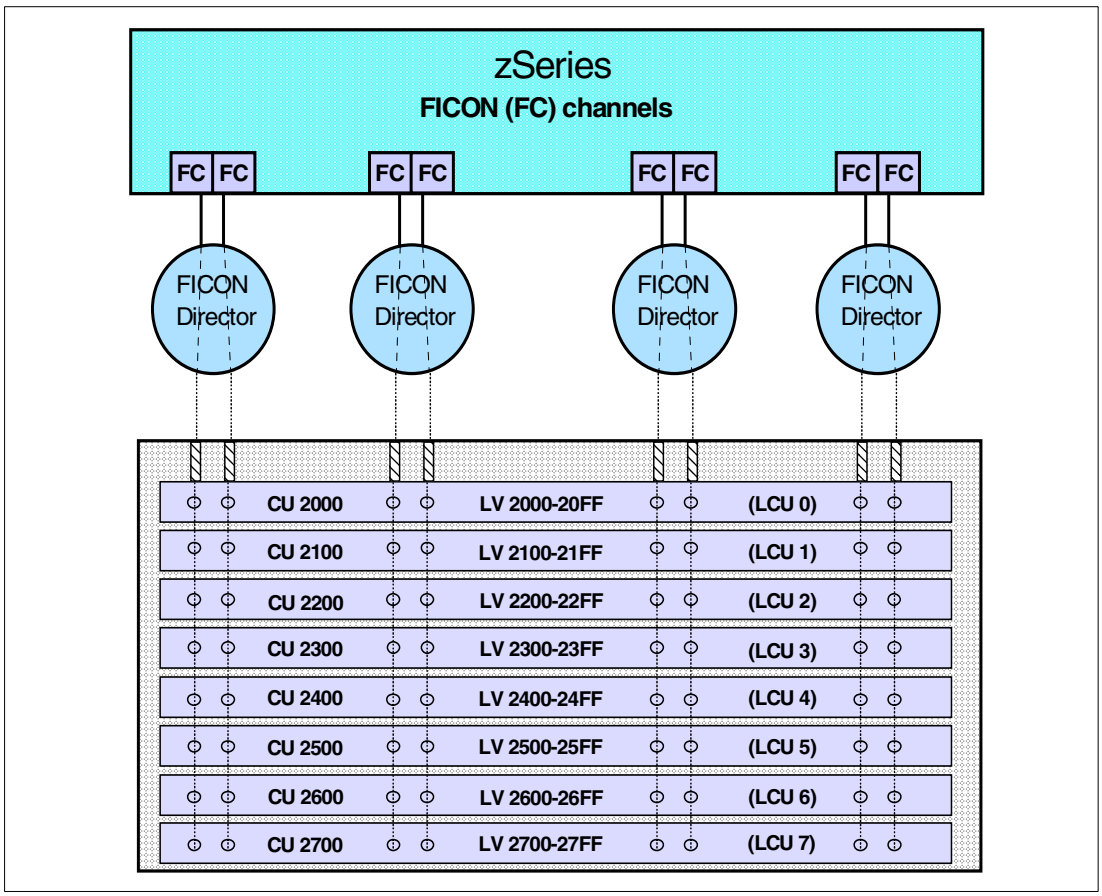


Figure 5-4 ESS FICON attachment

5.4.1 FICON benefits

FICON attachment for zSeries servers provides many improvements over ESCON. Some of the benefits you should consider are:

- Increased number of concurrent I/O connections (operations) over the link. FICON provides channel-to-ESS multiple concurrent I/O connections. ESCON supports only one I/O connection at any one time (these are logical connections, not links).
- Increased distance. With FICON, the distance from the channel to the ESS, or channel to switch, or switch to ESS link is increased. The distance for ESCON of 3 km is increased to up to 10 km (20 km with RPQ) for FICON channels using long wavelength lasers with no repeaters.
- Increased link bandwidth. FICON has up to 10 times the link bandwidth of ESCON (2 Gbps full duplex, compared to 200 MBps half duplex). FICON has up to more than four times the effective channel bandwidth (135 MB/sec compared to 17 MB/sec for ESCON). The FICON adapter on the ESS will also transmit/receive at 2 Gb if the switch/director supports 2 Gb links.
- No data rate droop effect. For ESCON channels, the droop effect started at 9 km. For FICON, there is no droop effect even at a distance of 100 km.
- Increased channel device-address support, from 1,024 devices for an ESCON channel to up 16,384 for a FICON channel.
- Greater exploitation of Parallel Access Volume (PAV). FICON allows for greater exploitation of PAV in that more I/O operations can be started for a group of channel paths.
- Greater exploitation of I/O Priority Queuing. FICON channels use frame and Information Unit (IU) multiplexing control to provide greater exploitation of the I/O Priority Queuing mechanisms within the FICON-capable ESS.
- Better utilization of the links. Frame multiplexing support on FICON channels, switches, and FICON control units such as the ESS provides better utilization of the links.

These improvements can be realized in more powerful and simpler configurations with increased throughput. The z/OS user will notice improvements over ESCON channels, with reduced bottlenecks from the I/O path, allowing the maximum control unit I/O concurrency exploitation:

- IOSQ time (UCB busy) can be reduced by configuring more alias device addresses using Parallel Access Volumes (PAVs). This is possible because FICON channels can address up to 16,384 devices (ESCON channels address up to 1,024; the ESS has a maximum of 4096 devices).
- Pending time can also be reduced:
  - Channel busy conditions are reduced by FICON channel's multiple starts.
  - Port busy conditions are eliminated by FICON switches' frame multiplexing.
  - Control unit busy conditions are reduced by FICON adapters' multiple starts.
  - Device-busy conditions are also reduced, because FICON's multiple concurrent I/O operations capability can improve the Multiple Allegiance (MA) function exploitation.

All these factors allow you to lay out more simple redundant configurations using FICON and access more data with better performance than what is possible with ESCON.

FICON vs. ESCON

Table 5-1 on page 134 summarizes the maximum number of ports, devices, and logical paths supported by the FICON and ESCON attachments.

Table 5-1 FICON and ESCON comparison

	FICON	ESCON
Host adapter ports per ESS	16	32
Devices per channel	16384	1024
Logical paths per host adapter port	256	64
Logical paths per LCU	256	256

The ESS Model 800 supports 256 *logical paths* per LCU; so a fully configured ESS with 16 FICON ports (256 logical paths per port) can *logically* daisy-chain the maximum 4096 devices in an ESS. On the other side, a fully configured ESS with 32 ESCON ports (64 logical paths per port) can only logically daisy-chain 2048 devices. This is additional configuration flexibility that FICON provides.

FICON and ESCON intermixing

The intermixing of ESCON and FICON channels to the same LCU from the same operating system image is only supported for migration purposes. It is not a recommended configuration for the production environment. The coexistence is useful during the transition period from ESCON to FICON channels. The mixture allows you to dynamically add FICON native channel paths to a control unit while keeping its devices operational. A second dynamic I/O configuration change can then remove the ESCON channels while keeping devices operational.

5.4.2 FICON recommendations

When configuring for FICON, consider the following recommendations:

- Define a minimum of four channel paths per control unit. Fewer channel paths will not allow exploitation of full ESS bandwidth. A more typical configuration would have eight FICON channels.
- Spread FICON host adapters across all adapter bays. This should result in minimally one host adapter per bay, or in a typically configured ESS, two host adapters per bay.
- Define a minimum of four FICON channels per path group.

For more information on ESS FICON support, see the documents *IBM TotalStorage Enterprise Storage Server Model 800,* SG24-6424, and *FICON Native Implementation and Reference Guide,* SG24-6266.

5.5 SCSI

An ESS can attach to open systems hosts with its two-port SCSI host adapters (see 2.9.2, “SCSI attachment” on page 38). SCSI ports are 2-byte wide, differential, fast-20. With SCSI adapters the ESS supports:

- A maximum of 15 targets per SCSI adapter (depending on the host type)
- A maximum of 64 logical units per target, depending on host type

The SCSI adapter always provides termination and therefore the ESS needs to be at the end of the SCSI bus.

5.5.1 Supported SCSI attached hosts

Currently the ESS supports the following host systems for SCSI attachment:

- Data General with the DG/UX operating system
- Hewlett-Packard with the HP-UX operating system
- Hewlett-Packard AlphaServer with the OpenVMS operating system
- Hewlett-Packard AlphaServer with the Tru64 UNIX operating system
- IBM iSeries and AS/400 with the OS/400® operating system
- IBM pSeries, RS/6000, and RS/6000 SP™ with the AIX operating system
- Intel-based servers with the Novell Netware operating system
- Intel-based servers with Microsoft Windows NT operating system
- Intel-based servers with Microsoft Windows 2000 operating system
- Sun servers with the Solaris operating system

For specific considerations that apply to each server platform, as well as for the most current information on supported servers—the list is updated periodically—please check:

`http://www.storage.ibm.com/disk/ess/supserver.htm`

5.5.2 SCSI attachment recommendations

There are four different types of possible SCSI attachments (refer to Figure 5-5):

- Direct single SCSI connection
- Daisy-chaining
- Multi SCSI connection with no redundancy
- Multi SCSI connection with redundancy via the IBM Subsystem Device Driver (SDD)

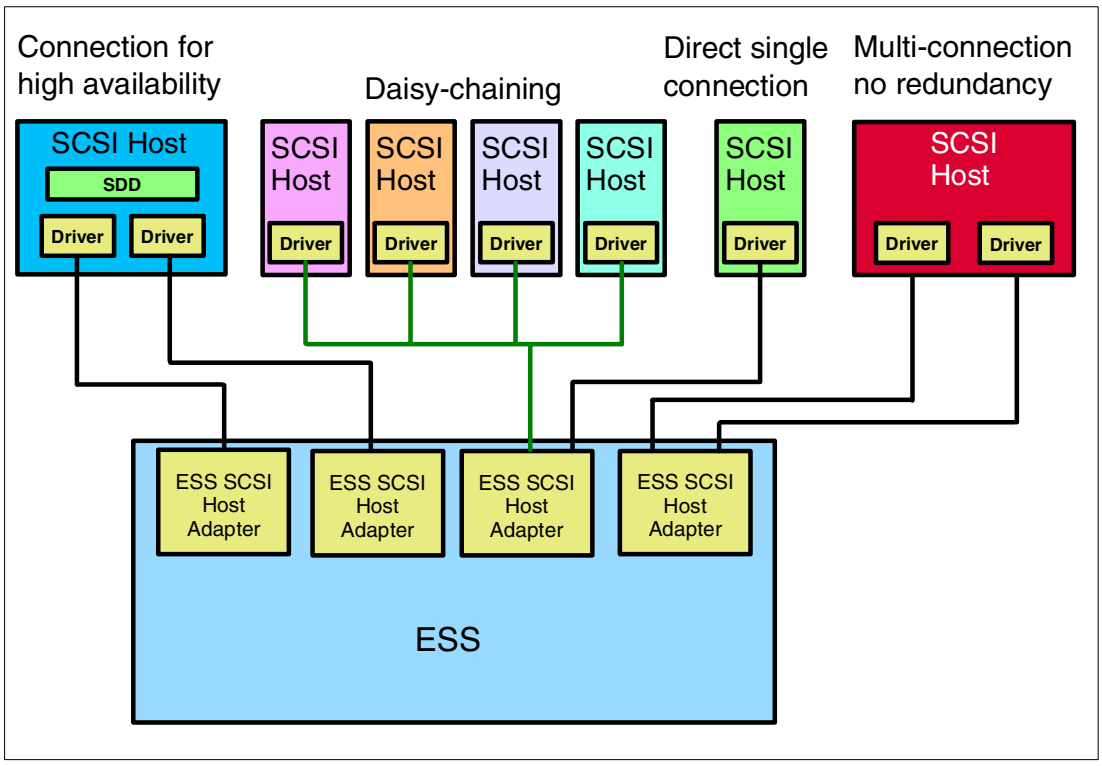


Figure 5-5 Types of ESS SCSI connections

When implementing SCSI attachments to the ESS, consider the following recommendations.

Daisy chaining SCSI adapters

We do not recommend daisy chaining SCSI host adapters together. Although it is technically possible, SCSI bus arbitration considerations make it problematic.

Multi-SCSI connections

For hosts with multi-connections, we recommend the use of the IBM Subsystem Device Driver (SDD), and connections to adapters in different host bays in the ESS. SDD provides load balancing and failover protection in the case of a SCSI adapter failure (refer to Figure 5-5 on page 135). Section 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149, further discusses SDD characteristics.

Unique target IDs

Care must be taken to ensure that the target ID of each adapter on the bus is unique. A single ESS SCSI port may contain up to 15 target IDs. You need to ensure that each target ID assigned to logical devices attached to that port is unique on that SCSI bus. Use the ESS Specialist, Open Systems Storage panel -> Configure Host Adapter Ports panel, to display internally assigned target IDs for a SCSI adapter port in the ESS.

It is preferable to define first the hosts that will access the ESS on the SCSI bus, then the IDs of all other initiators and non-ESS devices on the bus. On the ESS Specialist, Configure Host Adapter Ports panel, define your host IDs, then enter the other IDs as unrelated hosts or devices, before adding logical volumes to the port. This will ensure that the ESS assigns non-conflicting IDs for the logical volumes you add.

5.6 Fibre Channel

Fibre Channel is a 100 MBps or 200 MBps, full-duplex, serial communications technology to interconnect I/O devices and host systems that are separated by tens of kilometers. The ESS Model 800 supports 100 MBps and 200 MBps connections; negotiating automatically to determine whether it is best to run at 100 MBps (1 Gbps link) or 200-MBps (2 Gbps link). This means that you can connect the ESS Model 800 to a 1 Gbps link or a 2 Gbps link and the ESS will detect and operate at the higher link speed (see 2.9.3, “FCP attachment” on page 39).

Fibre Channel adapters that are configured for SCSI-FCP (Fibre Channel protocol) provide:

- A maximum of 128 host logins per Fibre Channel port
- A maximum of 512 SCSI-FCP host logins or SCSI-3 initiators per ESS
- A maximum of 4096 LUNs per target (one target per host adapter), depending on host type
- Either arbitrated loop, switched fabric, or point-to-point topologies

5.6.1 Supported Fibre Channel attached hosts

Currently the ESS supports the following host systems for FCP attachment:

- Hewlett-Packard with the HP-UX operating system
- Hewlett-Packard AlphaServer with the OpenVMS operating system
- Hewlett-Packard AlphaServer with the Tru64 UNIX operating system
- IBM iSeries and AS/400 with the OS/400 operating system
- IBM iSeries and AS/400 with the Linux operating system
- IBM pSeries, RS/6000, and RS/6000 SP with the AIX operating system
- IBM NUMA-Q with the DYNIX®/ptx operating system

- Intel-based servers with the Linux operating system
- Intel-based servers with the Novell Netware operating system
- Intel-based servers with Microsoft Windows NT operating system
- Intel-based servers with Microsoft Windows 2000 operating system
- SGI Origin servers with the IRIX operating system
- Sun servers with the Solaris operating system

For specific considerations that apply to each server platform, as well as for the most current information on supported servers—the list is updated periodically—please check:

http://www.storage.ibm.com/disk/ess/supserver.htm

5.6.2 Fibre Channel topologies

The ESS architecture supports all three Fibre Channel interconnection topologies:

- Direct connect
- Arbitrated loop
- Switched fabric

The three Fibre Channel topologies are discussed briefly below.

Direct connect

This is the simplest of all the Fibre Channel topologies. By using just a fiber or copper cable, two Fibre Channel adapters (one host and one ESS) are connected. The Fibre Channel host adapter card C in Figure 5-6 is an example of a direct connect connection.

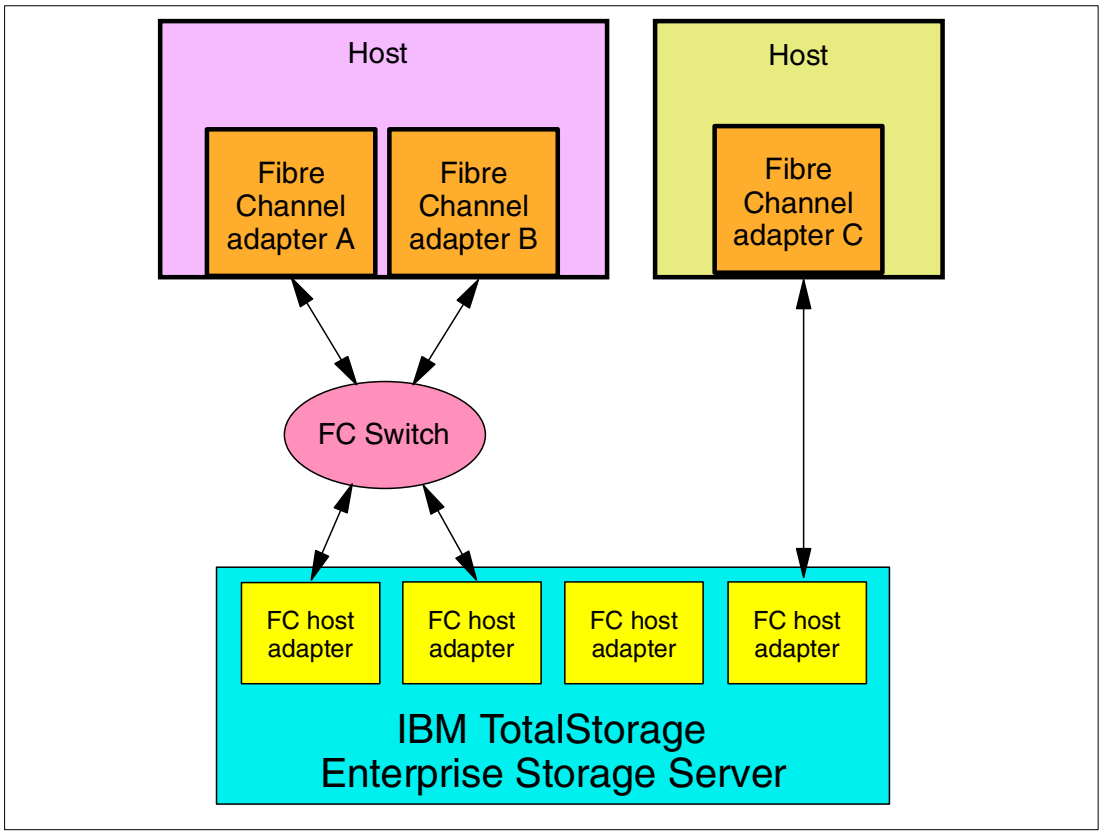


Figure 5-6 Fibre Channel connections with an ESS

This topology supports the maximum bandwidth of Fibre Channel, but does not exploit any of the benefits that come with SAN implementations.

Tip: When using the ESS Specialist to connect directly to a host HBA, set the ESS Fibre Channel port attribute to match the requirements of the host HBA configuration.

The ESS supports direct connect at a maximum distance of 500 m (1500 ft) at 1 Gb and 300 m (984 ft) at 2 Gb with the shortwave adapter. The ESS supports direct connect at a maximum distance of 10 km (6.2 mi) with the longwave adapter.

Arbitrated Loop

Fibre Channel Arbitrated Loop (FC-AL) is a uni-directional ring topology very much like token ring. Information is routed around the loop and repeated by intermediate ports until it arrives at its destination. If using this topology, all other Fibre Channel ports in the loop must be able to perform these routing and repeating functions in addition to all the functions required by the point-to-point ports.

Up to a maximum of 127 ports can be interconnected via a looped interface. All ports share the FC-AL interface and therefore also share the bandwidth of the interface. Only one connection may be active at a time, and the loop must be a private loop. An example of Fibre Channel arbitrated loop topology is shown in Figure 5-7. Note how the three servers with host adapters X, Y, and Z share a single port to the ESS.

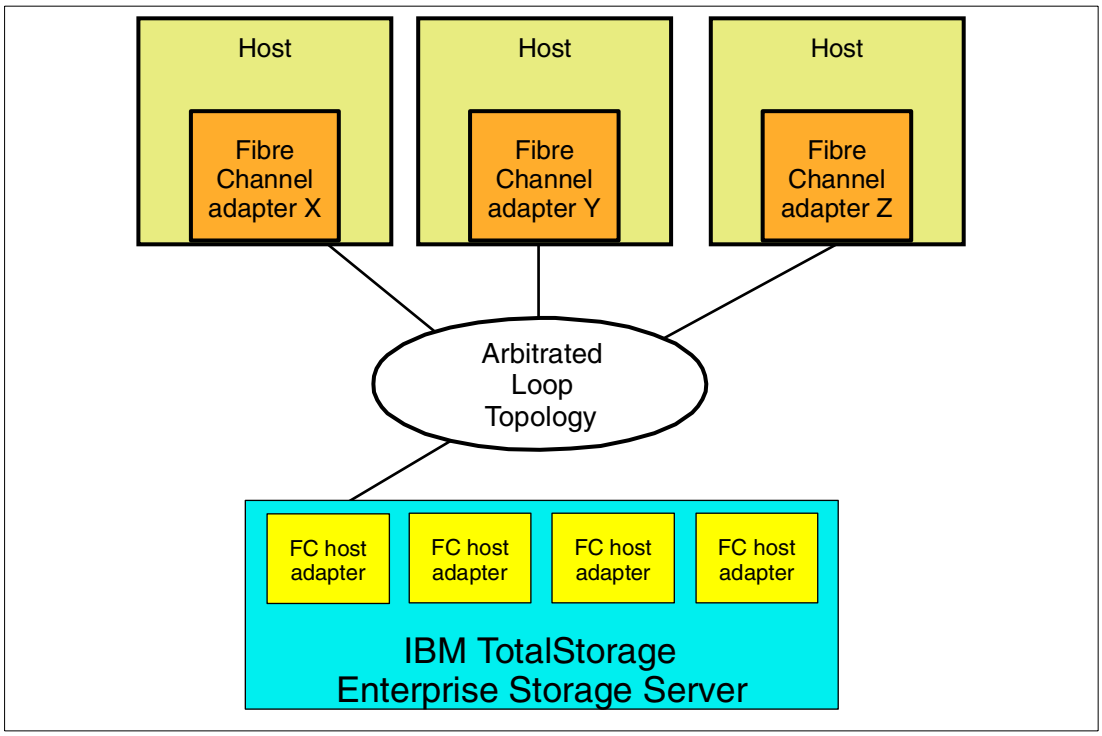


Figure 5-7 Fibre Channel arbitrated loop topology

The ESS does not support FC-AL topology on adapters that are configured for FICON protocol.

When using the ESS Specialist to configure a loop, always use Arbitrated Loop as the Fibre Channel port attribute. The ESS supports up to 127 hosts or devices on a loop. However, the loop goes through a loop initialization process (LIP) whenever you add or remove a host or

device from the loop. LIP disrupts any I/O operations currently in progress. For this reason, we recommend that you only have a single host and a single ESS on any loop.

iSeries arbitrated loop connection

Specifically for iSeries, remember that it has to connect via FC-AL and that the ESS Fibre Channel port cannot be shared with any other platform type. Additionally:

- The iSeries is tolerant of occasional LIP disruption and can recover I/O operations affected by such disruptions. More than one iSeries host might be placed within a loop with the ESS if your configuration meets the following requirements:
  - Hosts are rarely rebooted.
  - Only iSeries hosts are in the loop.
  - Few configuration changes occur.
- The iSeries supports multiple hosts or ESSs on a loop. For Version 5 Release 1, the iSeries supports only one ESS on a loop.

Refer to the *IBM TotalStorage Enterprise Storage Server Host System Attachment Guide,* SC26-7446, for more details on iSeries attachment.

Switched fabric

A switched fabric is an intelligent switching infrastructure that delivers data from any source to any destination. Figure 5-6 on page 137, with Fibre Channel adapters A and B, shows an example of a switched fabric. A switched fabric is the basis for a Storage Area Network (SAN), as shown in Figure 5-8 on page 141.

Tip: When using the ESS Specialist to configure a switched fabric, always use Point-to-Point as the Fibre Channel port attribute, as this is the protocol used in switched fabrics.

The supported distance between the host and the ESS depends on the speed of the host adapters (currently 2 Gb or 1 Gb), and whether short-wave or long-wave host adapters are being used. See Table 5-2 for a list of supported distances between hosts and the ESS Model 800. Long-wave adapters support greater distances than short-wave, while higher link speeds reduce the maximum distance.

Table 5-2 Distances supported by Fibre Channel cables for the ESS Model 800

Fibre Channel host adapter	Transfer rate	Cable type	Distance
FC 3025 (shortwave)	1 Gbps	62.5 µm, multimode	300 m (984 ft)
	2 Gbps	62.5 µm, multimode	150 m (492 ft)
	1 Gbps	50 µm, multimode	500 m (1640 ft)
	2 Gbps	50 µm, multimode	300 m (984 ft)
FC 3024 (longwave)	1 Gbps	9-µm fibre cable (single mode)	10 km (6.2 mi)
	2 Gbps	9-µm fibre cable (single mode)	10 km (6.2 mi)

Notes® for reading the information in Table 5-2:

- These adapters have auto-sensing that automatically negotiates with the attached unit to determine whether the link operates at 1 or 2 Gps.

- When attaching new equipment that is capable of operating at 2 Gbps to old equipment that only operates at 1 Gbps, the connection length might be a problem. The length might be adequate for 1 Gbps, but inadequate for 2 Gbps. You are responsible for determining the supported distance for the cable that you use.

Recommendations for implementing a switched fabric are covered in more detail in the following section.

5.7 SAN implementations

In this section we describe a basic SAN network and how to implement it for maximum performance and availability. We show some examples of a properly connected SAN network to maximize the throughput of disk I/O.

5.7.1 Description and characteristics of a SAN

SAN stands for Storage Area Network. A SAN allows you to connect heterogeneous open system servers to a high speed (1 Gb or 2 Gb/sec now; 10 GB/sec in future) network, sharing storage devices such as a disk storage and tape libraries. Instead of each server having its own locally attached storage and tape drives, a SAN allows you to share centralized storage components and easily allocate storage to hosts.

Figure 5-8 on page 141 shows an example of a SAN *switched fabric*. It is called a *switched fabric* because the SAN switches allow any Fibre Channel port to connect to any other Fibre Channel port. All the Fibre Channel adapters in the servers and storage in this example are running in switched fabric mode. There are four main components of the SAN:

- The servers.
- The storage subsystems—in this case an ESS Model 800 and a tape library.
- The SAN fabric switches in-between. The switches are the heart of the SAN network.
- The connection medium, which consists of cables, and the host adapter cards on each end (Qlogic, Emulex, or JNI), that reside in the server and the ESS.

In this example there are AIX, HP-UX, Sun, and Linux servers directly attached to the SAN via Fibre Channel adapters. There are also some PCs that do not have Fibre Channel adapters, but can access disk storage on the ESS via a NAS-SAN gateway. All servers also attach to an Ethernet network.

Notice each server is dual attached to the SAN for availability and load balancing. The storage devices—the ESS Model 800 and the tape library—have multiple SAN connections for availability and performance.

This is just one example of a SAN, and there are a myriad other ways to create a SAN with different types and numbers of servers, storage devices, and switches.

5.7.2 Benefits of a SAN

One of the biggest benefits of a SAN is centralized storage. Before SANs were available, it was common for servers in a data center to each have their own dedicated storage. Backups were difficult because the servers had to use their own tape drives, or backup over an Ethernet network. Sharing a tape drive could be possible, but usually meant moving the tape drive from one server to another, and reboots were required to attach the tape drive. Re-assigning disk storage meant making cable changes and possibly moving storage or servers.

Some of the benefits a SAN offers are:

- The ability to assign and un-assign a large (TB) amount of storage to a host with just the click of a mouse. No cabling changes required.
- Hosts can share storage for failover environments, or actually work on the same data using software that provides locking protection.
- Availability: Multiple connections between servers and storage.
- Performance: Each Fibre Channel adapter runs at 100 MB/sec or 200 MB/sec.
- LAN congestion is reduced by moving backups off the LAN and onto the SAN.
- Faster backups: It is even possible to implement *server free* backups, which copy data directly (at the block level) from the ESS to tape with no load on servers.
- Increased distance (10 km +) between servers and storage for convenience and security.

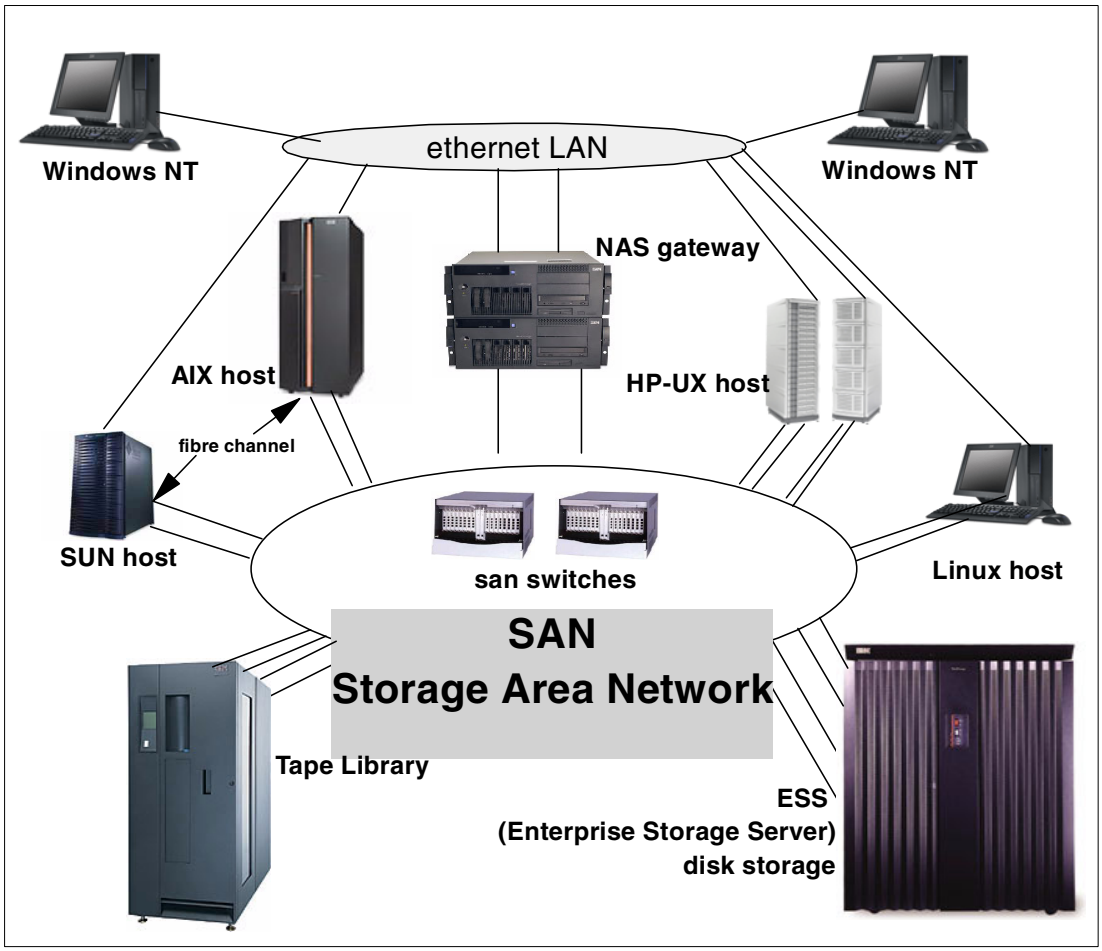


Figure 5-8 Example of a Storage Area Network

Notice in Figure 5-8 how many different types of servers there are sharing the same ESS storage and tape library.

5.7.3 SAN cabling for availability and performance

Figure 5-9 on page 143 is an example of four AIX hosts cabled to two SAN switches for high availability and performance.

At the top of the diagram you see the ESS with four host adapter bays. For simplicity in our example, each host adapter bay has only two Fibre Channel/FICON host adapter cards configured for FCP attachment. The other adapter slots could have more Fibre Channel/FICON, ESCON, or SCSI host adapters.

Next, you see the SAN switches. In this case there are two switches without ISLs (Inter-switch links), which keeps the switches logically separated. With the two switches separated, they each have their own zoning information as if they were two separate SANs. If the Zoning information on one switch became corrupt to due human error or a HW failure, the other switch would not be effected. Notice each switch has four connections to the ESS.

Next we have the cables from the hosts. Each host is dual attached with a fiber cable to each SAN switch.

Always spread the host connections across all the host adapter bays if possible. In a SAN fabric, hosts are not directly connected to the ESS with point-to-point connection, so the SAN switch *zones* determine the paths hosts use.

If you only have 2 HBAs in each host with point-to-point connections, then distribute the connections to the host adapters across the bays in the following sequence: Bay 1, Bay 4, Bay 2, Bay 3.

More information on the reasons behind using all four ESS host bays is in “Balancing connections across ESS bays” on page 129.

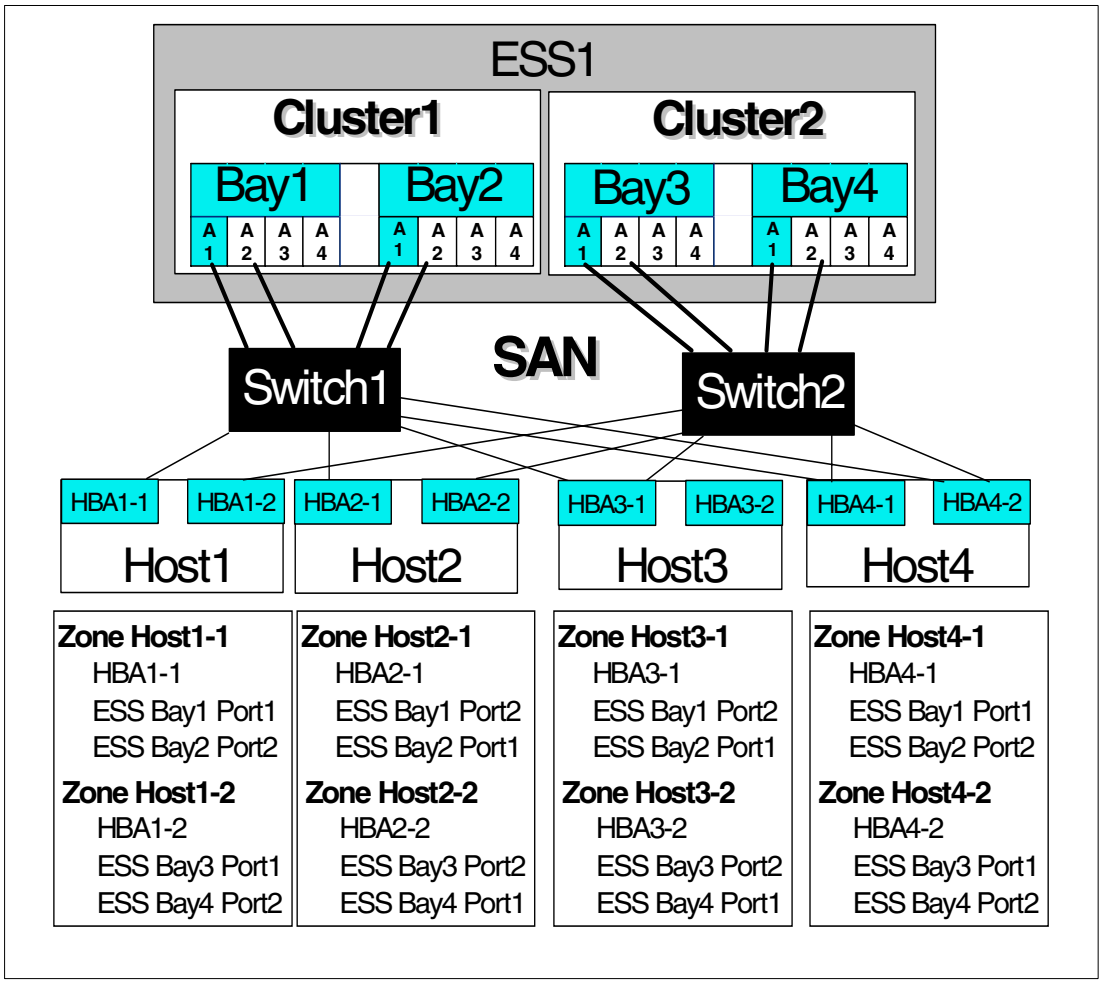


Figure 5-9 SAN cabling and zoning with four paths per host

Figure 5-9 shows a SAN fabric in which *zoning* is used to make sure each host has four connections to the ESS using all four ESS adapter bays. In this configuration, each host will see each logical disk (LUN) assigned to it from the ESS four times.

If you wanted hosts to only see each logical disk twice, then you would need to change the *zoning* information, as shown in Figure 5-10 on page 144. Notice that each host is zoned to use two paths to the ESS. Also, a given host uses ESS bays 1 and 4, or bays 2 and 3, for performance and availability reasons.

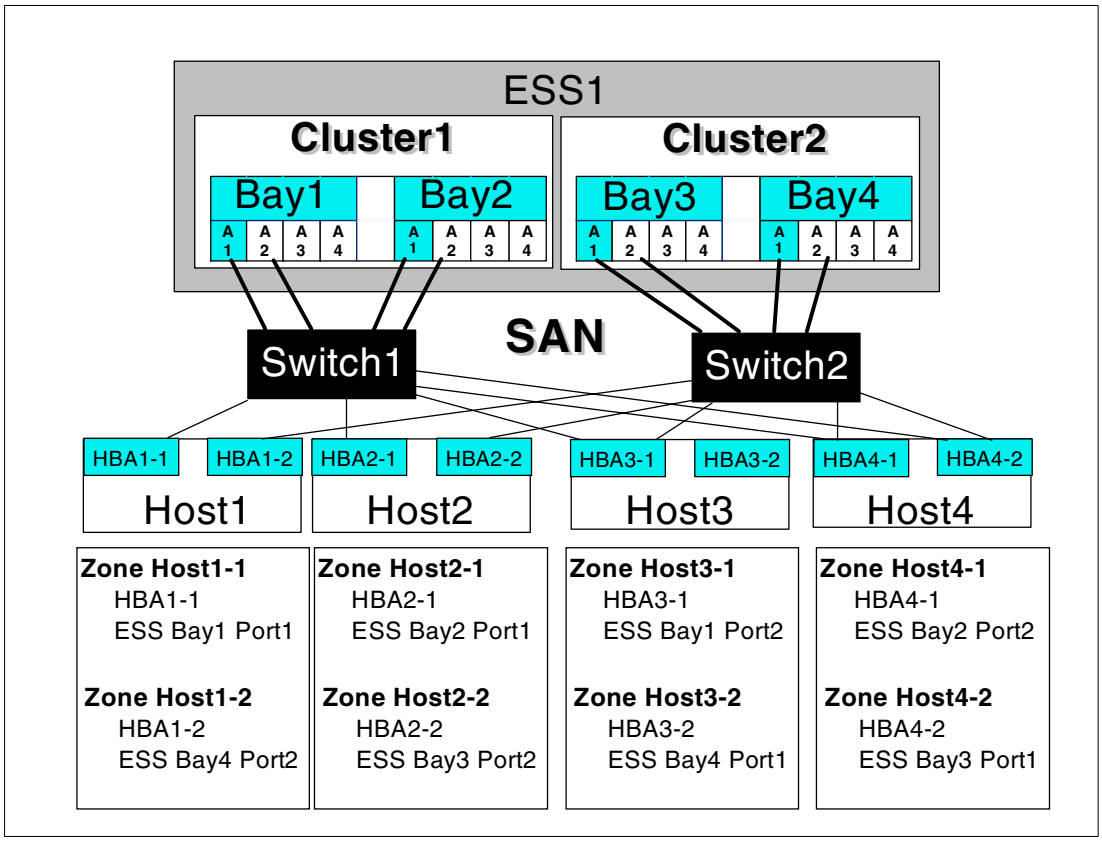


Figure 5-10 SAN cabling and zoning with two paths per host

Either configuration is valid and prevents a single point of failure. However, we generally recommend using all four host bays in the SAN zones, and let SDD balance across all four ESS host bays.

5.7.4 Tasks for a SAN implementation

When a SAN is going into an environment, you should make sure all the team players are in place and assign the areas of responsibility. In this section we review the major tasks that you will be involved in when implementing a SAN.

Plan in advance for the cabling layout scheme and provide it to the IBM System Support Representative (SSR). The IBM SSR will physically install the host adapter cards, cable connections, and switches. A System Administrator will typically install the drivers and file sets on the hosts to support the adapter cards and connections to the SAN.

The networking team will assign IP addresses and will *zone* the switches for *logical* connectivity and disk isolation of the LUNs to the hosts in the switches. Provide the world wide port numbers (WWPNs) to the networking team to do the zoning. You will then configure the RAID sets, carve the LUNs, and make the LUN assignments to the various hosts on the SAN. This includes LUN *masking* and isolation of the disks to the hosts on the ESS. Next you will bring in the LUNs to the hosts and create the file systems.

The summary in Table 5-3 on page 145 may help you to visualize our discussion. You may have only one or all of the responsibilities shown in the table, or several process owners may have to work together to implement and support the SAN implementation.

Table 5-3 SAN implementation matrix

Component	Tasks to implement and support	Responsible owner
Adapters that reside in the host server	The adapters are installed and drivers are loaded on the servers to support the adapter and connect to the switches. The drivers are upgraded as needed.	Server services/system administration
SAN switches	IP address and zoning must be loaded and configured into the switches, to define the SAN attached hosts and paths. The configuration is maintained and changed as required.	Networking services
ESS	The disk Raid sets are configured and LUNs are carved and LUN masked for presentation to the hosts. Upgrades to microcode are scheduled and coordinated with the IBM CE. LUN assignments are made and changed as needed.	Disk administration
Cables and hardware	Installed and maintained.	Hardware CE
Host configuration of the disks	The disks are brought into the host OS. File systems are configured and changed as needed.	Storage services/system administration/disk administration

It takes the effort and coordination of all the responsible owners to implement and support the ongoing integrity and performance of the SAN.

5.7.5 Importance of establishing zones

For Fibre Channel attachments in a SAN, it is important to establish *zones* to prevent interaction from host adapters. Every time a host adapter joins the fabric, it issues a Registered State Change Notification (RSCN), which does not cross zone boundaries, but will effect every device or host adapter in the same zone.

If a host adapter should go bad and start logging in and out of the switched fabric, or a server must be rebooted several times, you do not want it to disturb I/O to other hosts. Figure 5-9 on page 143 shows zones that only include a single host adapter and multiple ESS ports. This is the recommended way to create zones to prevent interaction between server host adapters.

Tip: Each zone should contain a single host system adapter with the desired number of ports attached to the ESS.

By establishing zones, you reduce the possibility of interactions between system adapters in switched configurations. You can establish the zones by using either of two zoning methods:

- Port number
- Worldwide port name (WWPN)

You can configure switch ports that are attached to the ESS in more than one zone. This enables multiple system adapters to share access to the ESS Fibre Channel ports. Shared access to an ESS Fibre Channel port might be from host platforms that support a combination of bus adapter types and the operating systems.

5.7.6 LUN masking

Assigning ESS LUNs (logical disks, volumes) in Fibre Channel attachments is a lot different than in SCSI attachments. In SCSI, LUNs are assigned based on SCSI ports, independent of which hosts may be attached to those ports. So if you have multiple hosts attached to a single SCSI port (ESS supports up to four hosts per port), all of them will have access to the same LUNs available on that port.

In Fibre Channel attachment, LUN affinity is based on the *world-wide port name* (WWPN) of the adapter on the host, independent of to which ESS Fibre Channel port the host is attached. In a switched fabric with multiple connections to the ESS, this concept of LUN affinity enables the host to see the same LUNs on different paths.

If the host is not capable of recognizing that the set of LUNs seen via each path is the same, this may present data integrity problems when the LUNs are used by the operating system. To get around this problem, you can do *switch zoning*, or you can install the IBM Subsystem Device Driver (SDD), which is preferred. Aside from preventing the above problem, SDD also provides multipathing and load balancing, which improves performance and path availability. SDD is covered in 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.

When configuring the ESS, the LUN masking function is done by the ESS when you assign logical disks (volumes) to servers using the ESS Specialist.

To complement the information discussed in this section, please refer to the IBM publication *Implementing Fibre Channel Attachment on the ESS*, SG24-6113.

5.7.7 Configuring logical disks in a SAN

In a SAN, care must be taken in planning the configuration to prevent the proliferation of disk devices presented to the attached hosts. A large number of disk devices presented to a host can cause longer failover times in cluster environments. Also boot times could take longer because the device discovery steps will take more time.

The number of times an ESS logical disk is presented as a disk device to an open host depends on the number of paths from each host adapter to the ESS. The number of paths from an open server to the ESS is determined by the following:

- The number of host adapter cards installed in the server
- The number of connections between the SAN switches and the ESS
- The zone definitions created by the SAN switch software

Note: Each physical path to a logical disk on the ESS is presented to the host operating system as a disk device.

Consider a SAN configuration as shown in Figure 5-11 on page 147:

- The host has two connections to SAN switches and each SAN switch in turn has four connections to the ESS.
- Zone A includes one Fibre Channel card (FC0) and two paths from SAN switch A to the ESS.
- Zone B includes one Fibre Channel card (FC1) and two paths from SAN switch B to the ESS.
- This host is only using four of the eight possible paths to the ESS in this zoning configuration.

By cabling the SAN components and creating zones as shown in Figure 5-11, each logical disk on the ESS will be presented to the host server four times since there are four unique physical paths from host to ESS. If zone A and zone B were modified to include four paths each to the ESS, then the host would have a total of eight paths to the ESS. In that case, each logical disk assigned to the host would be presented as eight physical disks to the host operating system.

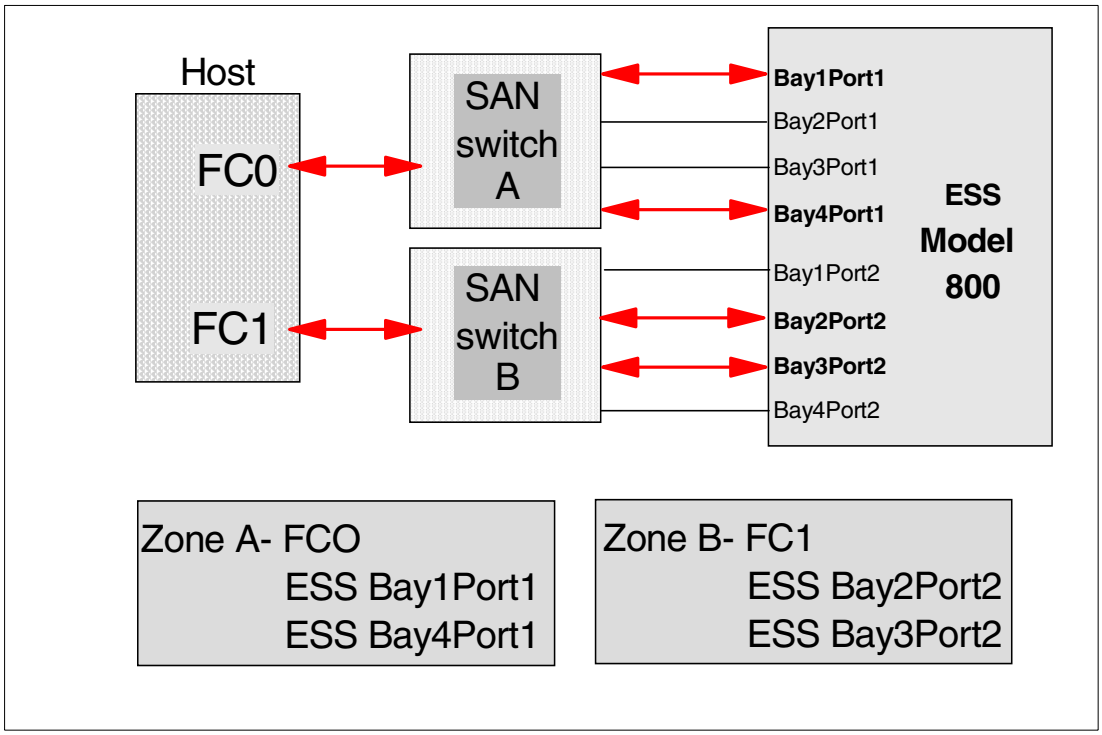


Figure 5-11 Zoning in a SAN environment

In a SAN environment, Subsystem Device Driver (SDD) is used to provide load balancing and failover. SDD also adds another device to the host operating system for each logical disk presented from the ESS. Figure 5-12 on page 148 shows how SDD adds a pseudo device called a vpath (virtual path) on top of the disk devices. The host operating system issues I/O calls to vpath0 in the example, and SDD in turn picks the best physical path (disk0, disk1, disk2, or disk3) to use at a given time.

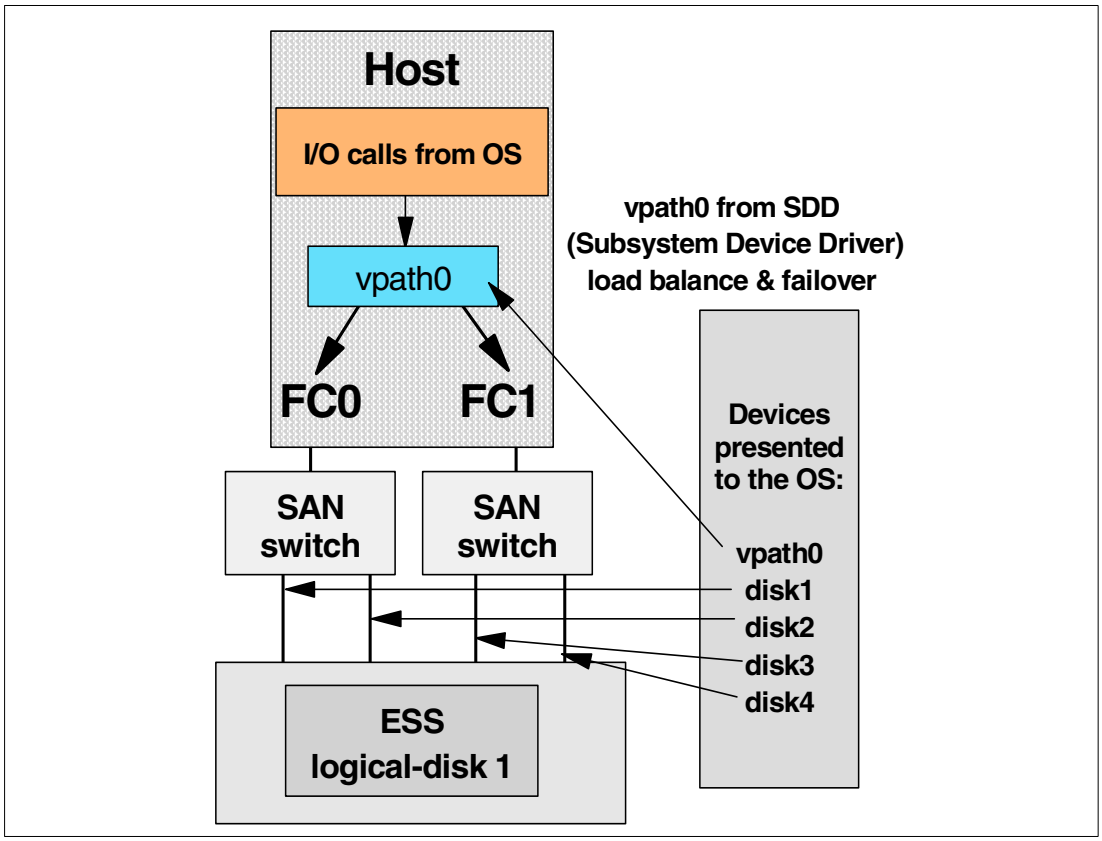


Figure 5-12 SDD with multiple paths to an ESS logical disk

In the example in Figure 5-12, the number of devices presented to the host operating system for each ESS logical disk is limited to five (four disk devices + 1 vpath).

The installation and use of SDD is covered in detail in 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.

Figure 5-13 on page 149 shows an ESS Specialist perspective on how zoning effects the number of paths from the host to each ESS logical disk. When selecting the host BigBlue, notice how the ESS Specialist displays connections from four host adapters to the disk group containing logical disk 1. The four connections are a result of the zones defined in Figure 5-11 on page 147. The four paths from BigBlue to the same logical disk are presented as disk devices (disk1, disk2, disk3, disk4) to the operating system.

You can see how the number of devices presented to a host could proliferate rapidly in a SAN environment if care is not taken in selecting the size of logical disks and the number of paths from host to ESS.

Typically, we recommend for dual attached hosts, cable the switches and create zones in the SAN switch software so that each host adapter has two paths from the switch to the ESS. Figure 5-11 on page 147 shows an example of hosts using four paths to the ESS; one path to each host bay. With hosts configured this way, you can let SDD balance the load across all four host adapter bays in the ESS.

As mentioned previously, for a host bay adapter card to be replaced in the ESS, the entire bay must be quiesced. By using all four ESS host bays (two for each zone) an entire host bay on the ESS could be down, but your host would still have three paths to the ESS.

Maintenance could be done on an ESS bay and your host could still benefit from SDD load balancing to the three remaining bays.

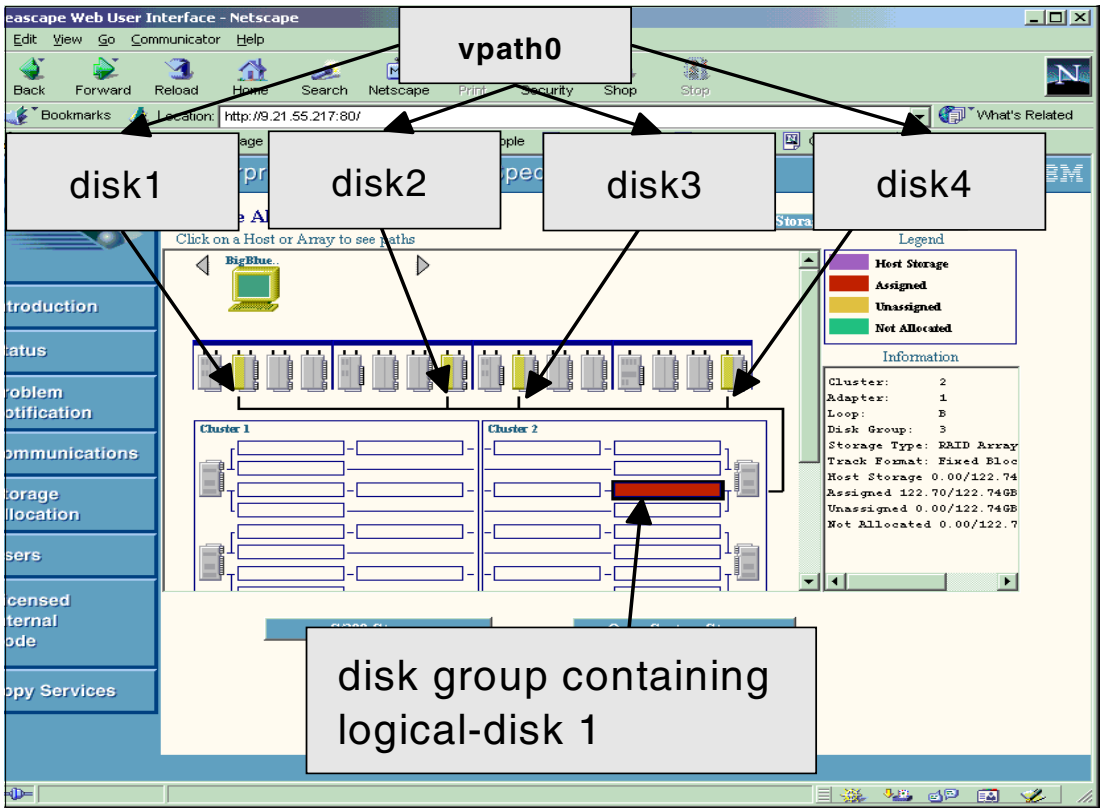


Figure 5-13 Logical disk-to-host device mapping

Zoning more paths, such as eight connections from the host to ESS, will generally not improve SAN performance and will only cause twice as many devices to be presented to the operating system.

A discussion on limiting the UNIX server to four to eight paths is included in 6.2, “Planning and preparing UNIX servers for performance” on page 164.

5.8 Subsystem Device Drivers (SDD) - Multipathing

The IBM Subsystem Device Driver (SDD) software is a host-resident pseudo device driver designed to support the multipath configuration environments in the ESS. SDD resides in the host system with the native disk device driver and manages redundant connections between the host server and the ESS, providing enhanced performance and data availability.

Some operating systems and file systems natively provide similar benefits provided by SDD, for example, z/OS, OS/400, NUMA-Q Dynix, and HP/UX.

SDD provides ESS-attached hosts running Windows NT and 2000, AIX, HP/UX, Sun Solaris, or Linux with:

- Dynamic load balancing between multiple paths when there is more than one path from a host server to the ESS. This may eliminate I/O bottlenecks that occur when many I/O operations are directed to common devices via the same I/O path, thus improving the I/O performance.

- Automatic path failover protection and enhanced data availability for users that have more than one path from a host server to the ESS. It eliminates a potential single point of failure by automatically rerouting I/O operations to remaining active paths from a failed data path.

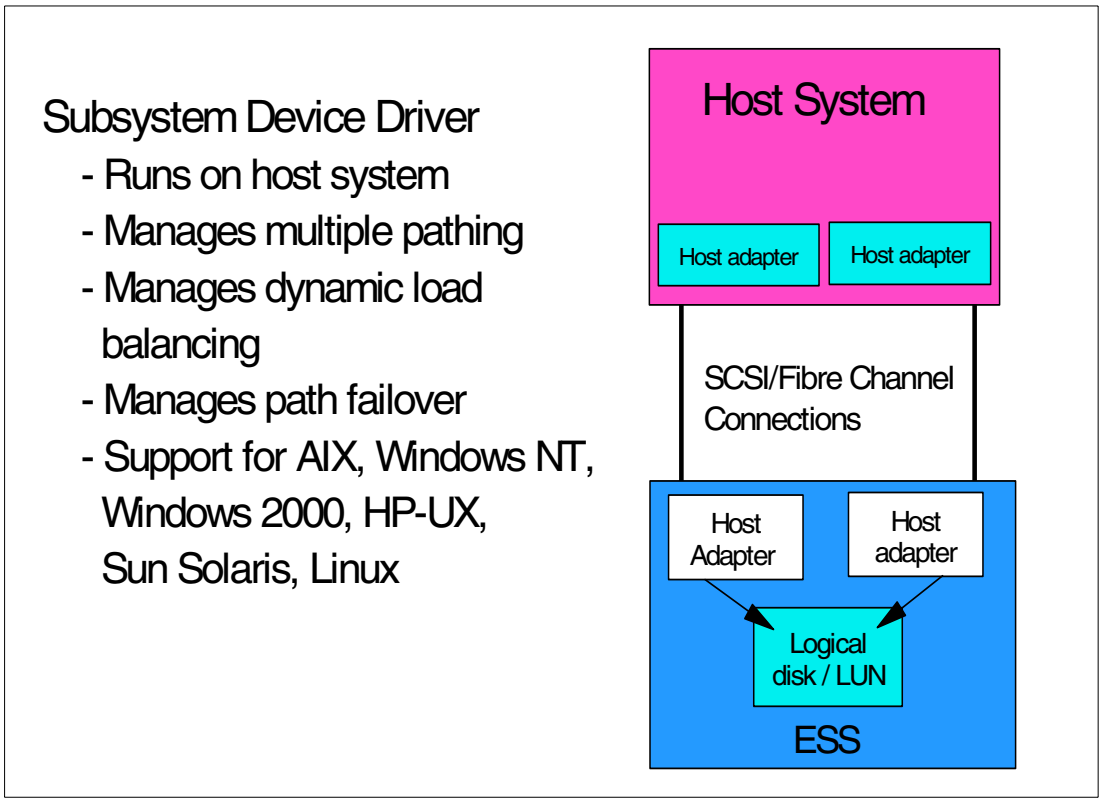


Figure 5-14 Subsystem Device Driver configuration

An example of a dual attached host that can benefit from SDD is shown in Figure 5-14.

SDD supports either SCSI or FCP connections from host to ESS. However, SDD does not support a host system with both a SCSI and Fibre Channel connection to a shared ESS LUN.

Note: You should not be sharing LUNs between multiple hosts without the protection of Persistent Reserve (PR). If you are sharing LUNs between hosts without HACMP, you are exposed to data corruption situations. PR should also be used when using FlashCopy.

The Subsystem Device Driver may operate under different modes/configurations:

- Concurrent data access mode: A system configuration where simultaneous access to data on common LUNs by more than one host is controlled by system application software such as Oracle Parallel Server, or file access software that has the ability to deal with address conflicts. The LUN is not involved in access resolution.
- Non-concurrent data access mode: A system configuration where there is no inherent system software control of simultaneous access to the data on a common LUN by more than one host. Therefore, access conflicts must be controlled at the LUN level by a hardware-locking facility such as SCSI Reserve/Release.

It is important to note that the IBM Subsystem Device Driver does not support booting from or placing a system primary paging device on an SDD pseudo device.

For some servers, like selected pSeries and RS/6000 models running AIX, booting off the ESS is supported. In that case LUNs used for booting are manually excluded from the SDD configuration by using the **querysn** command to create an exclude file. More information can be found in “QUERYSN for multi-booting AIX off the ESS” on page 180.

For more information on installing and using SDD, refer to *IBM TotalStorage Subsystem Device Driver User's Guide,* SC26-7478. This publication and other information are available at:

http://www.ibm.com/storage/support/techsup/swtechsup.nsf/support/sddupdates

5.8.1 SDD load balancing

SDD automatically adjusts data routing for optimum performance. Multipath load balancing of data flow prevents a single path from becoming overloaded, causing input/output congestion that occurs when many I/O operations are directed to common devices along the same input/output path.

The path selected to use for an I/O operation is determined by the policy specified for the device. The policies available are:

- Load balancing (default). The path to use for an I/O operation is chosen by estimating the load on the adapter to which each path is attached. The load is a function of the number of I/O operations currently in process. If multiple paths have the same load, a path is chosen at random from those paths.
- Round robin. The path to use for each I/O operation is chosen at random from those paths not used for the last I/O operation. If a device has only two paths, SDD alternates between the two.
- Failover only. All I/O operations for the device are sent to the same (preferred) path until the path fails because of I/O errors. Then an alternate path is chosen for subsequent I/O operations.

Normally, path selection is performed on a global rotating basis; however, the same path is used when two sequential write operations are detected.

5.8.2 Concurrent LIC load

With SDD you can concurrently install and activate the ESS Licensed Internal Code while applications continue running if multiple paths from the server have been configured. During the activation process, the host adapters inside the ESS might not respond to host I/O requests for up to 30 seconds. SDD makes this process transparent to the host system through its path-selection and retry algorithms.

5.8.3 Single path mode

SDD does not support concurrent download and installation of the Licensed Internal Code (LIC) to the ESS if hosts are using a single-path mode. An example of a host using a single path is shown in Figure 5-15 on page 152.

However, SDD does support single-path SCSI or Fibre Channel connection from your host system to an ESS. It is possible to create a volume group or a **vpath** device with only a single path.

Note: With a single-path connection, SDD cannot provide failure protection and load balancing and this is not recommended.

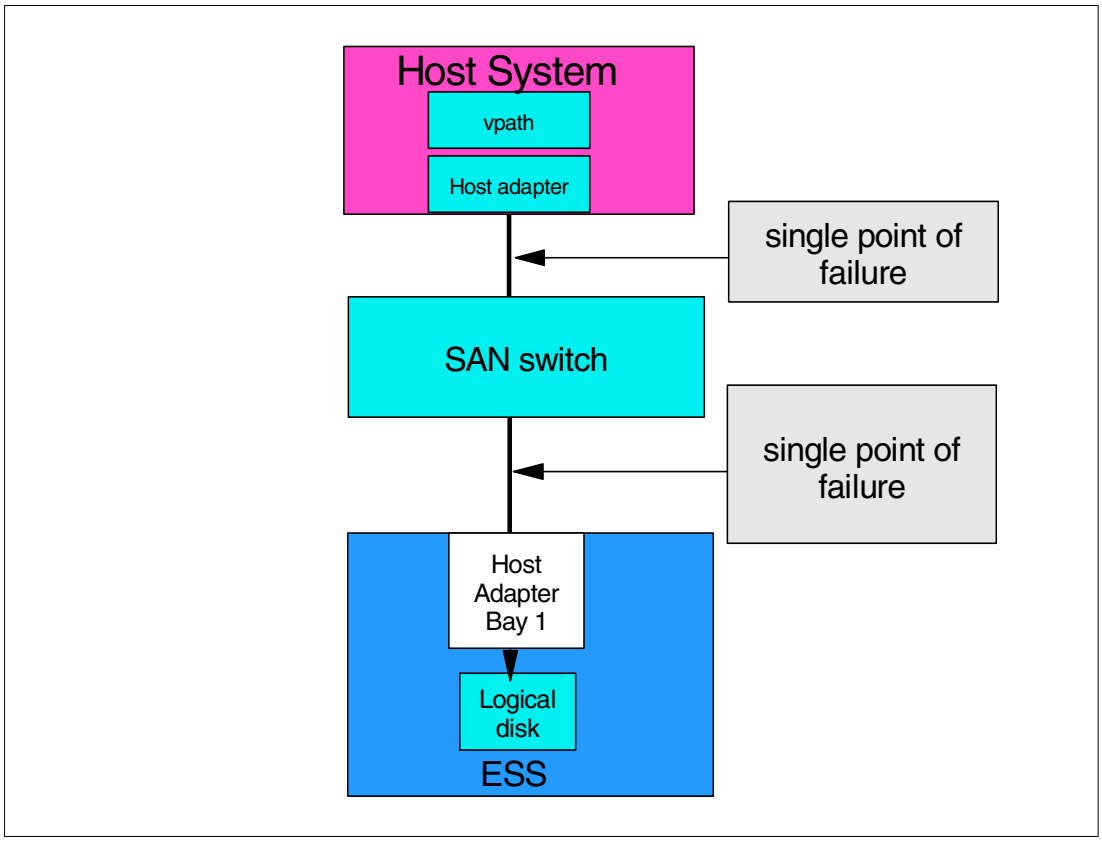


Figure 5-15 SAN single-path connection

5.8.4 Single FC adapter with multiple paths

A host system with a single Fibre Channel adapter that connects through a switch to multiple ESS ports is considered to have multiple Fibre Channel paths. Figure 5-16 on page 153 illustrates an example.

From an availability point of view, the configuration is not good because of the single fiber cable from the host to the SAN switch. However, this configuration is better than a single path from host to ESS and can be useful for preparing for maintenance on the ESS. This configuration prevents a loss of connection to the ESS if a host bay is down for maintenance.

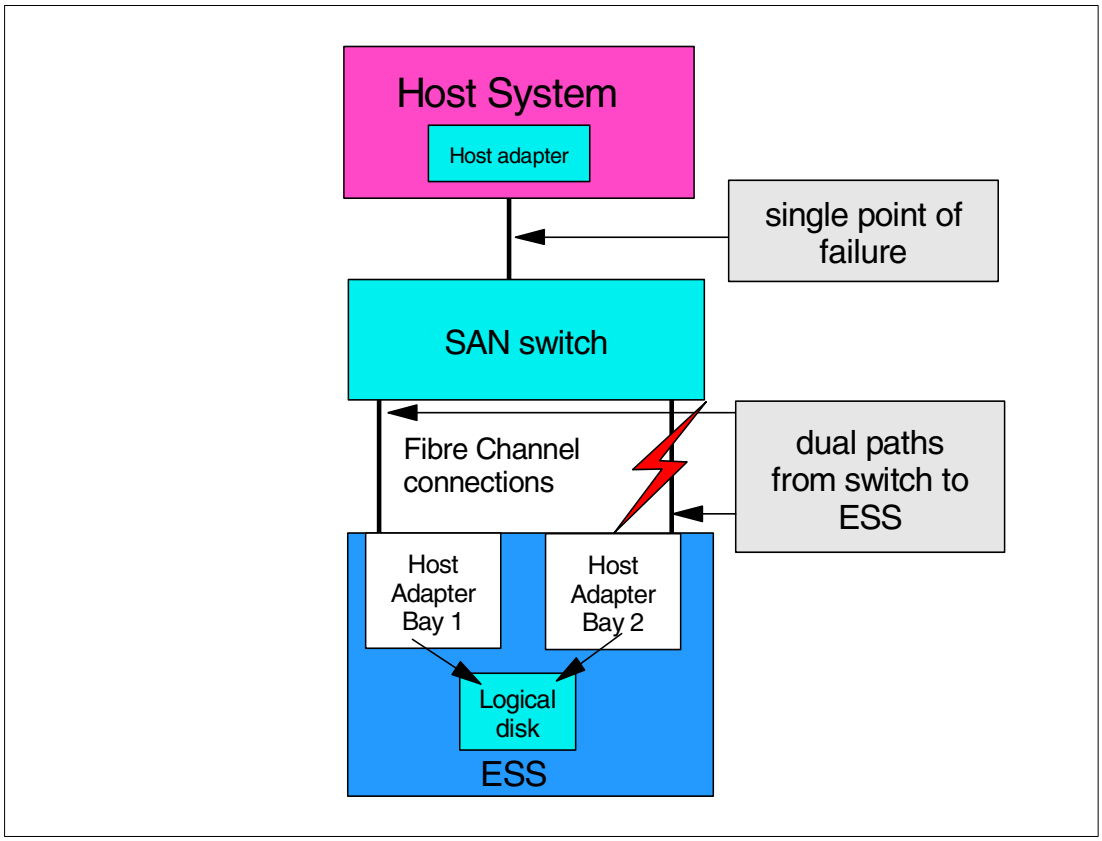


Figure 5-16 SAN multi-path connection with single fiber

5.8.5 Path failover and online recovery

SDD automatically and non-disruptively can redirect data to an alternate data path. In most cases, host servers are configured with multiple host adapters with either SCSI or Fibre Channel connection to an ESS that, in turn, would provide internal component redundancy. With dual clusters and multiple host adapters, the ESS provides more flexibility in the number of input/output paths that are available.

When a path failure occurs, the IBM SDD automatically reroutes the I/O operations from the failed path to the other remaining paths. This eliminates the possibility of a data path being a single point of failure.

5.8.6 SDD datapath command

SDD provides commands that you can use to display the status of adapters that are used to manage disk devices, or to display the status of the disk devices themselves. You can also set individual paths online or offline and also set all paths connected to an adapter online or offline at once.

A summary of the **`datapath`** commands is listed in Table 5-4.

Table 5-4 datapath command options

Command	Description
datapath query adapter	Displays information about adapters.

Command	Description
datapath query adapstats	Displays performance information for all SCSI and FCP adapters that are attached to SDD devices.
datapath query device	Displays information about devices.
datapath query devstats	Displays performance information for a single SDD device or all SDD devices.
datapath set adapter	Sets all device paths that are attached to an adapter to online or offline.
datapath set path	Sets the path of a device to online or offline.
datapath open path	Dynamically opens a path that is in an invalid state.
datapath set device policy	Dynamically changes the path-selection policy of the SDD devices. Choices are round-robin, load balance, default, failover.

Example 5-1 illustrates the command **datapath query adapter**. Notice that this host has four adapters, all functioning normally. There are two SCSI adapters (scsi2 and scsi3) and two Fibre Channel adapters (fcssi0 and fscsi2).

Example 5-1 DATAPATH QUERY ADAPTER command example

```
$ datapath query adapter
Active Adapters :4
Adpt#       Adapter Name     State        Mode      Select    Errors    Paths    Active
    0        scsi3             NORMAL    ACTIVE   129062051    0       64          0
    1        scsi2             NORMAL    ACTIVE   88765386    303      64          0
    2        fscsi2            NORMAL    ACTIVE   407075697   5427     1024        0
    3        fscsi0            NORMAL    ACTIVE   341204788   63835    256         0
```

The terms used in the output of **datapath query adapter** are defined as follows:

Adpt# The number of the adapter.

Adapter Name The name of the adapter.

State The condition of the named adapter. It can be either:

-Normal, adapter is in use.

-Degraded, one or more paths are not functioning.

-Failed, the adapter is no longer being used by SDD.

Mode The mode of the named adapter, which is either Active or Offline.

Select The number of times this adapter was selected for input or output.

Errors The number of errors on all paths that are attached to this adapter.

Paths The number of paths that are attached to this adapter. In the Windows NT host system, this is the number of physical and logical devices that are attached to this adapter.

Active The number of functional paths that are attached to this adapter. The number of functional paths is equal to the number of paths attached to this adapter minus any that are identified as failed or offline.

An example of the **datapath query 0** command is shown in Example 5-2 on page 155. The output shows the status of paths for vpath0. Notice that vpath0 has four paths that are functioning normally. In this case we have an AIX system that sees four hdisks: hdisk1 to

hdisk4. There are two different Fibre Channel adapters in the host: fscsi0 and fscsi1. The switch zones are configured to give each Fibre Channel adapter two paths to the ESS.

Example 5-2 datapath query device output

```
$ datapath query 0
datapath query device 0
DEV#:   0    DEVICE NAME: vpath0   TYPE: 2105F20 SERIAL: 5049900
POLICY: Optimized
============================================================
Path# Adapter/HardDisk State Mode Select Errors
    0 fscsi0/hdisk1    OPEN NORMAL 0
    1 fscsi0/hdisk2    OPEN NORMAL 0
    2 fscsi1/hdisk3    OPEN NORMAL 0
    3 fscsi1/hdisk4    OPEN NORMAL 0
```

Operating system specific SDD commands

There are other operating system specific commands that SDD adds for AIX, HP/UX, Solaris, Linux, Windows NT, and Windows 2000. Some of the commands for UNIX systems are covered in 6.4, “SDD commands for AIX, HP-UX, and Sun Solaris” on page 176. For more information please refer to *IBM TotalStorage Subsystem Device Driver User’s Guide,* SC26-7478.

5.9 ESSUTIL utility package

In this section we discuss the tools included with the ESSUTIL package (ESS utilities) and how to implement them to verify SAN connections for better performance. We will show some example outputs on some UNIX servers.

The ESSUTIL package is a set of utilities written by Ian Mac Quarrie in the IBM Product Engineering group, to retrieve and use hdisk/vpath/ESS LUN association information. The utilities display information about OS devices (hdisks/vpaths) and their connectivity to the ESS. This includes information such as:

- ESS LUN information (size, LSS, volume number, LUN serial number, etc.)
- Connectivity information (host adapter location, ESS port connection)
- Host associations (VG/vpath/hdisk associations)
- Ability to automate the offlining/onlining of SDD paths

5.9.1 Using ESSUTIL for performance enhancement

The commands included in the ESSUTIL package do not monitor performance like **iostat** does or the ESS Expert does. They do however, enable you to verify connections between hosts and the ESS. Once the SAN connections and zoning are complete, the ESS utilities are used to verify proper connection and zoning of the hosts to the ESS Model 800 through the SAN switches.

A performance tool called **ess_iostat**, presented later in 6.8, “Viewing iostats based on ranks - ess_iostat script” on page 196, depends on the ESS utilities in order to present I/O stats based on ranks instead of individual disk devices like **iostat** does.

5.9.2 ESS utilities supported servers

Presently the ESSUTIL package is available for AIX, Solaris, and HP/UX. Check for current supported operating systems and download the utility from the Web site:

http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105

5.9.3 Implementing and using the ESS utilities

When you download the ESSUTIL package, check the associated README file for the platform you intend to install it on. The installation is simple and follows the normal install procedures for each operating system:

- **smit** for AIX
- **sam** for HP/UX
- **pkgadd** for Solaris

Note: For AIX and HP/UX, you should also install an associated Host Attachment script for use with the ESSUTIL package.

The Host Attachment script adds 2105 device information, allowing your HP/UX or AIX host to properly:

- Identify logical disks presented from the ESS as 2105 disks.
- Set the default disk attributes such as queue_depth and timeout values. (The queue_depth determines how many I/Os can be issued to a single disk device at a time and is important for ESS performance).

For AIX, the host attachment script is packaged in the ibm2105.rte file set.

The ESSUTIL package includes the following utilities, which can be run from the command line:

lsess	This program displays data collected from the essmap program. The essmap program issues SCSI commands directly to the disks (inquery, read capacity, and log sense) to collect the information displayed.
ls2105 and **lssdd**	For AIX these programs display data collected by AIX and stored in the ODM when the disks were configured to the host. The last four columns contain data collected from lsess (because it is not available from AIX). For Solaris, these program format data collected by Solaris. For HP-UX, this program formats data collected by HP-UX and essmap.
lsvp	This program (list virtual path) shows vpath status from an ESS physical location code perspective. Included is the capability to set SDD paths offline using an ESS location code.

Installing ESSUTIL for AIX

When installing the ESSUTIL package for AIX you can install support to query both SCSI and Fibre Channel attached disks. If all drives are attached via SCSI or Fibre Channel, then you only need to select one of the following file sets:

- ibmpfe.essutil.scsi.data for SCSI attachment, adds the scsimap program to cfgmgr
- ibmpfe.essutil.fibre.data for FCP attachment, adds the fcmap program to cfgmgr

Choosing a file set for an attachment type you are not using will not cause serious problems, but every time you run **cfgmgr** you will receive an error similar to the following:

```
Method error (/usr/lib/methods/scsimap >> /var/adm/essmap.out):
        0514-023 The specified device does not exist in the
                 customized device configuration database.
```

LSSDD output example

An example of **lssdd** output for all three supported platforms is shown in Example 5-3:

Example 5-3 LSSDD output on AIX, Sun, and HP-UX platforms

```
AIX
Hostname   VG         vpath    hdisk    Location       LUN SN   S  Connection  Size  LSS  Vol  Rank
--------   --         -----    -----    --------       ------   -  ----------  ----  ---  ---  ----
aix_shark sharkvg    vpath0   hdisk1   10-68-01       0O8FC106 Y  R1-B1-H2-ZA 16.0   10   8    1001
aix_shark sharkvg    vpath0   hdisk9   17-08-01       0O8FC106 Y  R1-B4-H2-ZA 16.0   10   8    1001


SUN
Hostname   vpath    hdisk    Location                    LUN SN   S  Connection  Size  LSS  Vol  Rank
--------   -----    -----    --------                    ------   -  ----------  ----  ---  ---  ----
sun_host_1 vpath3   c1t1d0   /sbus@a,0/fcaw@2,0/sd@1,0   30412342 N  R1-B1-H4-ZA 12.0  19   004  1301
sun_host_1 vpath3   c1t0d0   /sbus@a,0/fcaw@2,0/sd@0,0   30412342 N  R1-B2-H4-ZA 12.0  19   004  1301
sun_host_1 vpath3   c1t15d0  /sbus@a,0/fcaw@2,0/sd@f,0   30412342 N  R1-B3-H4-ZA 12.0  19   004  1301
sun_host_1 vpath3   c1t6d0   /sbus@a,0/fcaw@2,0/sd@6,0   30412342 N  R1-B4-H4-ZA 12.0  19   004  1301


HP
Hostname   VG       vpath    hdisk    Location             LUN SN   S  Connection  Size  LSS  Vol  Rank
--------   --       -----    -----    --------             ------   -  ----------  ----  ---  ---  ---
hp_ess    sddvg    vpath0   c31t2d1  0/7/0/0.2.19.0.32.2.1 011FC106 Y R1-B3-H4-ZA  008.0 16  017  1001
hp_ess    sddvg    vpath0   c30t2d1  0/7/0/0.2.18.0.32.2.1 011FC106 Y R1-B4-H4-ZA  008.0 16  017  1001
```

The following is an explanation of the output header definitions in Example 5-3.

Hostname	The host name of the SAN connected server
VG	Volume group the LUN resides in
vpath	The virtual path number that the LUN is associated with
hdisk	hdisk name assigned by the host for each ESS logical disk
Location	The physical location code of the host adapter the LUN is accessed through
S	Shared by two or more ESS ports, yes or no
Connection	Physical location code of ESS adapter LUN is accessed through
LUN SN	Unique serial number for each LUN within the ESS
Size	Configured capacity of the LUN in gigabytes
LSS	Logical subsystem where the LUN resides
Vol	Volume number within the LSS
Rank	Unique identifier for each raid-array within the ESS

In Example 5-3, notice that you can check the zoning and connection of vpath0 for AIX. Notice the location includes two different Fibre Channel adapters (10-68-01 and 17-08-01 are the slot numbers for two different Fibre Channel cards). Also notice how path hdisk1 is connected to the ESS host adapter card in Bay1-Port2 (B1-H2). The other path, hdisk9, is

connected to ESS host adapter card in Bay4-Port2 (B4-H2). This shows that the host *aix_shark* has two paths to the ESS, and each path is from a different Fibre Channel adapter to two different bays on the ESS. There is no single point of failure in this configuration (Fibre Channel adapter, fiber cable, or ESS host bay). Host aix_shark can take advantage of SDD and load balance over both paths.

Notice how the server sun_host_1 in the example is zoned so that it uses all four host adapter bays in the ESS.

LSESS and LS2105 output example

An example of the **lsess** output is shown in Example 5-4. Notice the output is very similar to **lssdd**, but volume group and vpath information is not listed. **ls2105** output is very similar to **lsess**. Before running **lssdd** for the first time, run **lsess**, which will build some essmap files.

Example 5-4 LSESS output on AIX, SUN, and HP-UX platforms

```
AIX
Disk        Location       LUN SN       Type       Size    LSS    Vol    Rank   C/A   S    Connection
------      ------------   --------  ------------   ----    ---    ---   ----   ---   -   -----------
hdisk1      10-68-01[FC]   008FC106  IBM 2105-E20 16.0      10      8    1001    01   Y   R1-B2-H4-ZA

SUN
 Disk        Location                 LUN SN   Type   Size  LSS Vol Rank C/A S Connection
------     ------------              -------- ------- ----- --- --- ---- --- - -----------
c1t0d0   /sbus@a,0/fcaw@2,0/sd@0,0   30412342 2105F20  12.0  19 004 1301 2/2 N R1-B2-H4-ZA

HP/UX
Disk        Location                LUN SN   Type   Size  LSS Vol Rank C/A S Connection
------     ------------             -------- ------- ----- --- --- ---- --- - -----------
c12t2d1  0/4/0/0.2.16.0.32.2.1 [FC]  011FC106 2105E20 008.0  16 017 1001 1/1 Y R1-B2-H4-ZA
c14t2d1  0/4/0/0.2.17.0.32.2.1 [FC]  011FC106 2105E20 008.0  16 017 1001 1/1 Y R1-B1-H4-ZA
```

LSVP output example

The **lsvp** tool is useful to verify all paths to vpaths are working. One function of **lsvp** is similar to the **datapath** command that SDD provides, but **lsvp** shows/alters vpath status from an ESS physical location code perspective. The **datapath** command shows paths from a host system adapter perspective.

To prepare for SAN or ESS maintenance, **lsvp** makes it easier to set all paths using a particular ESS bay offline. The **lsvp** command can also test the physical access to each LUN down each physical path and display a report.

The command usage is:

```
Usage: lsvp [-adt] [-o] [-l location_code]
```

Where:

-a	Display SDD logical path to ESS physical location code
-d	Display path status for each vpath device
-t	Test physical access to hdisks, can only be used with -d option
-o	Attempts to set all SDD paths ONLINE
-l	Set SDD paths OFFLINE for specified ESS location code location_code must be in one of the following formats: **single port:** R1-Bx-Hy-Zz

all ports on card: R1-Bx-Hy

all ports on bay: R1-Bx

An example of verifying the SAN paths for a host is shown in Example 5-5. The output pertains to the server sun-host-1 from Example 5-3 on page 157. Notice the vpath numbers are listed on the left hand side (0 - 11 and the word vpath is not listed). You can see that the vpaths 0–7 are functioning fine and include four paths to the ESS. The host is using ESS Bay1,2,3,4-Port4 for SAN connections. Vpaths 8–11 have been closed using **`datpath set path offline`**.

Example 5-5 LSVP output to verify SAN paths

```
lsvp -d
            BAY-1(B1)          BAY-2(B2)          BAY-3(B3)          BAY-4(B4)
          H1  H2  H3  H4     H1  H2  H3  H4     H1  H2  H3  H4     H1  H2  H3  H4
         A B A B A B A B    A B A B A B A B    A B A B A B A B    A B A B A B A B

       0 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       1 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       2 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       3 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       4 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       5 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       6 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       7 - - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - Y -
       8 - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -
       9 - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -
      10 - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -
      11 - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -    - - - - - - 0 -

Y  =  online/open
O  =  online/closed
N  =  offline
X  =  not-available
-  =  path not configured
?  =  path information not available
PD =  path down
```

The **`lsvp`** command can be very useful when performing maintenance on the ESS or SAN. You can bring a path down, move fiber cables around, or perform maintenance on an ESS bay, and then bring paths back online.

The **`addpaths`** command from SDD for AIX adds paths dynamically and can be verified using **`lsvp`** as well.

5.9.4 Mapping ranks to ESS Specialist disk groups

The ESS utilities **`lsess`** and **`lssdd`** will help you visualize where host storage resides in the ESS. Figure 5-17 on page 160 shows a map that helps understand how the *rank IDs* presented in the output of **`lsess`** or **`lssdd`** map to *disk groups* presented by the ESS Specialist. Notice that a rank ID is unique for each array and based on the two-digit LSS number plus a two-digit position within the LSS.

Note: The mapping shown in Figure 5-17 describes the layout when the ESS is configured in the usual way. In the unlikely circumstance that you choose not to alternate between loops when defining the ranks within the LSS, then the mapping would be different.

The ESS in this example contains array sizes based on 36 GB disks with all arrays formatted RAID-5. Also, for space considerations, we included 32 arrays instead of a full ESS of 48 arrays. We recommend making a similar picture of your environment when using the ESS utilities. This will help you visualize how data resides on the ESS from different servers, and see more clearly how the I/Os from different servers could impact each other.

If you plan on implementing striping at the OS level, this type of figure will also be very helpful so you can make sure you are using logical disks from different arrays on different SSA loops.

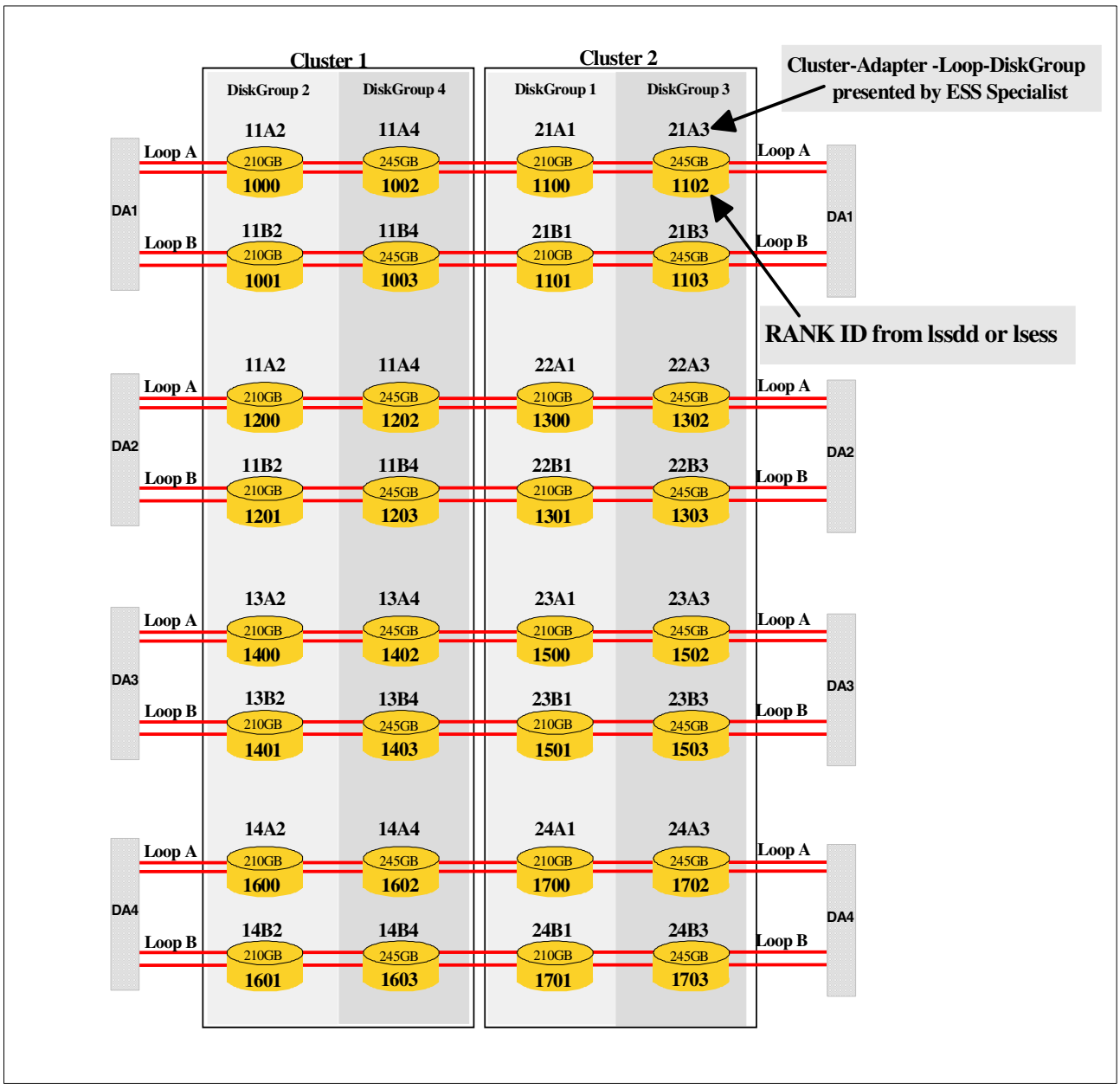


Figure 5-17 Mapping rank IDs to disk groups

You can gather the *rank IDs* based on Cluster-Adapter-Loop-DiskGroup from the ESS Specialist, as shown in Figure 5-18 on page 161.

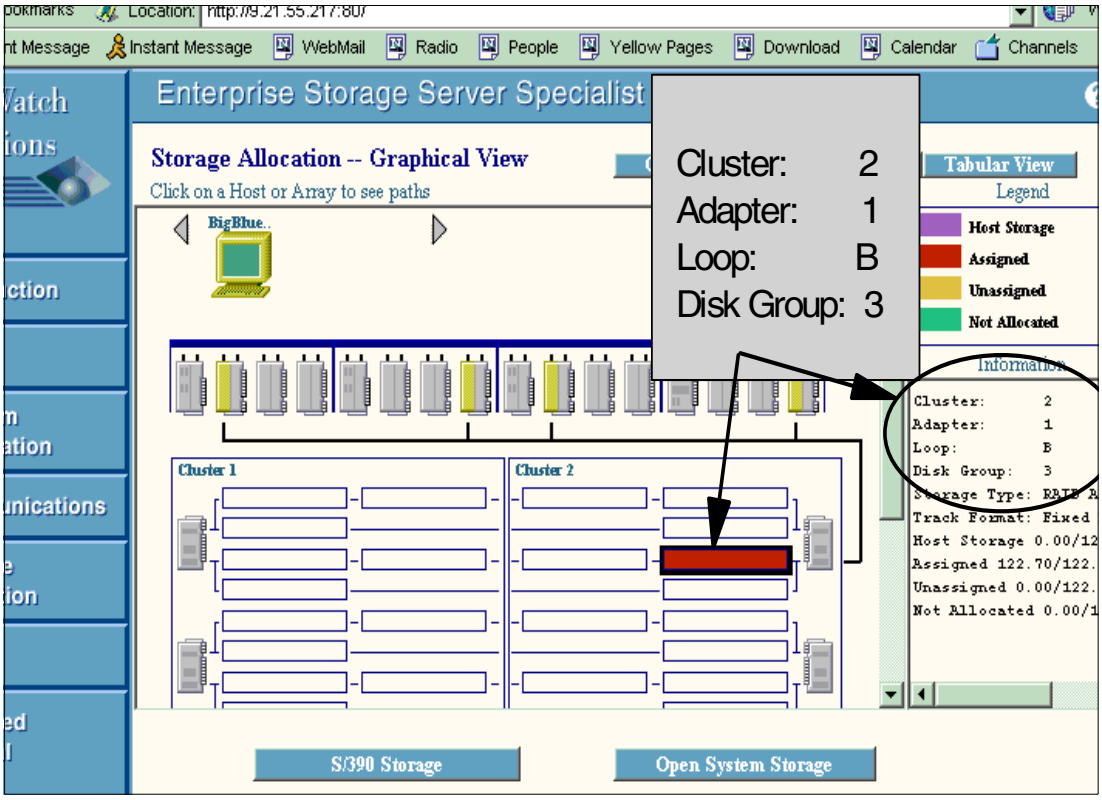


Figure 5-18 ESS Specialist array IDs - Cluster-Adapter-Loop-DiskGroup

Simply click a *disk group* and the Information panel on the right side will update the Cluster-Adapter-Loop-Disk_Group information.

6

Open systems servers - UNIX

This chapter contains information about monitoring and tuning the ESS performance when attaching AIX, HP-UX, and Sun Solaris servers. In this chapter we discuss:

- Preparing the ESS, the UNIX servers, and the SAN components
- Common UNIX tools to monitor I/O performance
- Specific AIX, HP-UX, Sun Solaris tools, and tuning options
- How to view host iostats by vpaths or ranks in real time
- Implementing OS level striping for faster sequential I/O
- How to test sequential read/write speeds
- How to gather iostats from ESS-attached hosts to create an enterprise view, detailing which servers are driving the I/O and how they effect each other

6.1 UNIX performance monitoring and tuning

In the following sections we discuss configuring UNIX servers to improve ESS I/O performance. We also present methods to monitor and tune disk I/O from AIX, HP-UX, and Sun servers attached to an ESS.

The tips and tools presented in this chapter will allow you to:

- Collect host I/O stats for:
  - Individual disk devices (paths to ESS LUNs)
  - Vpaths
  - Ranks
- Develop an iostat report for all ranks in the ESS (enterprise iostats) from a host perspective.
- Create baseline measurements of performance.
- Test and improve sequential I/O.

Performance tuning is an iterative process. It is very important to have a baseline measurement so you can:

- Verify if changes actually improved performance or not.
- Know if that DBA complaining of an I/O problem at 5 p.m. Friday is correct.
- Ask for a X percent raise after you have made the entire enterprise X percent faster!

Remember, when making tuning changes:

- Make sure you have a baseline I/O performance measurement under the *same* workload to compare to after making tuning changes.
- Plan changes to be made and include a back out plan.
- Make changes one at a time.
- Document changes.

Sometimes you will want to view performance of a specific host, and other times you will want to view performance stats of ESS components. Remember that multiple hosts can be using *logical disks* (LUNs) from the ESS that reside on the same array. You will want to correlate iostats gathered from servers with the ESS Expert stats. The ESS Expert monitoring tool is covered in 4.4, “IBM TotalStorage Expert” on page 104.

Keep in mind that the most important I/O measurements to gather from a server’s disk subsystem are:

- Number of I/O transactions per second
- Total MB/sec transferred
- KB/sec read
- KB/sec written
- KB/transaction = [(KBread/sec + KBwritten/sec) / (transactions/sec)]

6.2 Planning and preparing UNIX servers for performance

Before delving into iostat numbers and performance tools, it is important to consider some configuration factors that effect I/O performance on the ESS.

6.2.1 I/O balanced across ESS components

When configuring UNIX servers attached to an ESS for performance, it is important to remember that several servers are now sharing common disk storage. If one server is just “sitting” on an ESS array (that is, consuming the majority of the I/O bandwidth), it could negatively affect the performance of other hosts sharing the array.

It is important to spread I/O from ESS-attached servers across ESS components as discussed in 5.7.7, “Configuring logical disks in a SAN” on page 146.

6.2.2 Number of paths from host to ESS

Especially important in a SAN environment is to limit the number of disk devices presented to a host. In a SAN, every extra path from host to ESS will cause another disk device to be presented to the host OS for every ESS LUN assigned to it.

The more devices that are presented to a host, the longer boot times, LVM commands, and failovers will take. Also, with a huge number of disk devices (individual paths to the same ESS LUN), it makes gathering meaningful iostats more difficult.

We generally recommend that a host that is dual attached to SAN switches, be zoned to use four paths to the ESS. Also, each of the four paths for the host from SAN switch to ESS should use a different ESS host adapter bay.

In the SAN, try to refrain from using Internet switch links (ISLs) unless you have to. If you need ISLs so hosts can see certain devices such as SAN-attached tape drives, create separate zones so that I/O from host to ESS does not use a path across an ISL.

An example of SAN that has been configured for availability and load balancing, but limits the number of paths per host to four, is shown in Figure 6-1. Notice that the ESS has 12 paths: Six to each SAN switch.

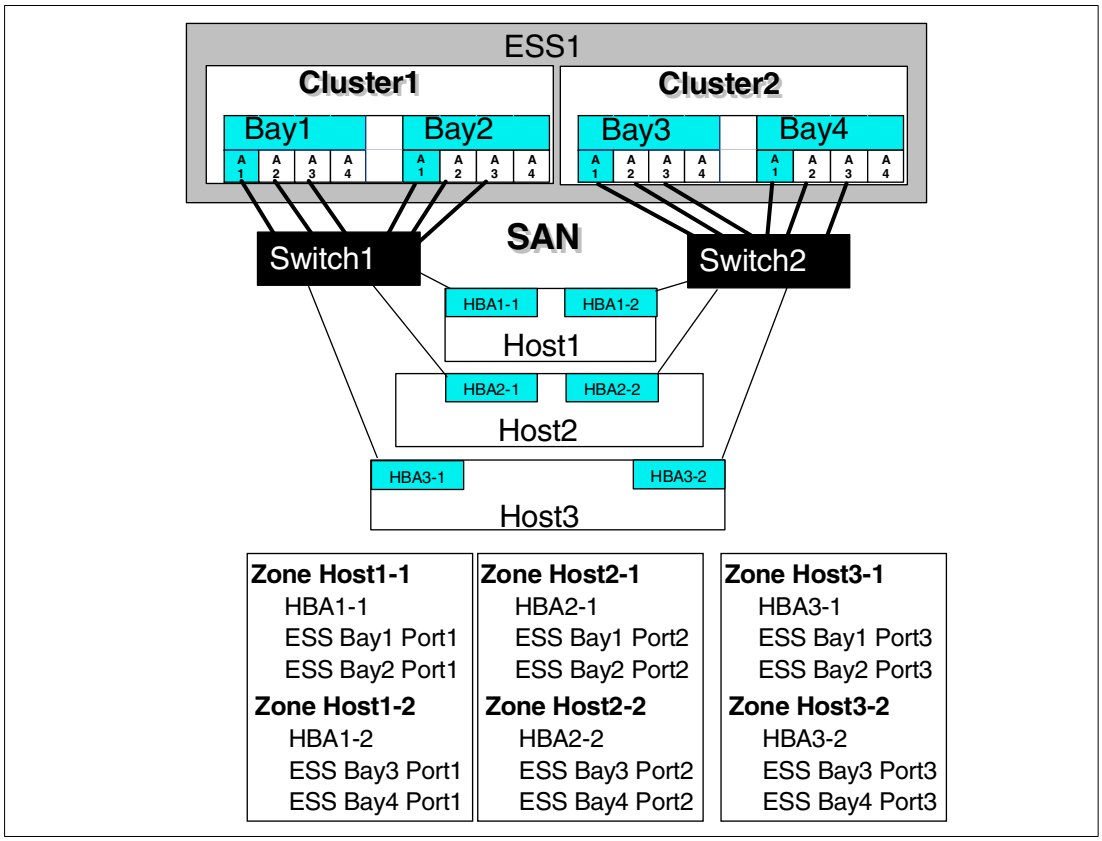


Figure 6-1 SAN paths - Four is enough

Notice that all three hosts in the example in Figure 6-1 use four paths to the ESS, but each host has a unique set of four paths or *path set*. You could simply create *zones* so that each host has access to all 12 paths and just let SDD balance over all the paths. However, that means that each logical disk assigned to a host will be presented 12 times to the host OS. While the SDD software is capable of load balancing over lots of paths, it is optimized to work with 4–8 paths.

We recommend manually balancing paths by assigning a four paths to each host and rotating the *path sets* around. If there was a host4 in the example of Figure 6-1, it could use the same paths as host1, or a a new combination of four paths out of the 12. Rotate the path sets around among servers and remember to make sure each host has a path to each of the four ESS host adapter bays.

With 1 Gbit Fibre Channel, four paths from ESS to SAN switches provides four different 100 MB/sec paths from the ESS to choose from. Unless you are trying to reach over 135 MB/sec of sequential read/write speeds with more Fibre Channel adapters, you do not need more paths. If you must have more paths, we do not recommend more than eight from host to ESS.

For more information on SAN zoning for performance and availability, refer to 5.7.7, “Configuring logical disks in a SAN” on page 146.

6.2.3 ESS LUN size

When implementing an ESS, the question of which logical disk (LUN) size to use for best performance often comes up. Internal to the ESS, there is no performance difference for

logical disk (LUN) sizes. For HP-UX, AIX, and SUN Solaris we believe 8 GB or 16 GB LUN sizes will satisfy most requirements.

For a more thorough discussion on selecting LUN sizes, refer to section 3.3, “Logical disks - Number and size” on page 59.

6.2.4 System and adapter code level

Before trying to tune disk I/O by moving data around or making kernel changes to the UNIX OS, it is important to make sure all components (servers, ESS, SAN switches) are prepared and have the latest firmware/microcode.

In a SAN environment, the microcode levels on the ESS, on the SCSI and Fibre Channel adapters on the servers, and SAN switches code, all effect each other.

Before implementing the ESS, be sure to verify/update:

- System and adapter microcode on the host servers
- Device drivers on the servers
- SAN switch software
- ESS Licensed Internal Code (LIC) level: Verify level with IBM support representative

You can find information on microcode levels for RS/6000 and pSeries servers and adapters at:

`http://techsupport.services.ibm.com/server/mdownload`

For HP-UX servers, download device drivers from:

`http://www.hp.com/country/us/eng/support.html`

For Sun Solaris servers, download device drivers from:

`http://wwws.sun.com/software/download/`

For the ESS, be sure to check for general updates, and the latest levels of the SDD and ESS utilities (ESSUTIL) at:

`http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105`

6.2.5 Subsystem Device Driver (SDD)

It is important to use SDD, and not only for load balancing and failover, but for the tools it provides such as **`datapath query`**.

Note: There are different versions of SDD for concurrent and noncurrent HACMP. Be sure to install the correct version for your environment.

We will cover some specific SDD commands for AIX, HP-UX, and Sun Solaris in 6.4, “SDD commands for AIX, HP-UX, and Sun Solaris” on page 176. For more details on SDD see 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.

6.2.6 ESSUTIL package

The ESSUTIL package is very important to verify and modify ESS connections. The tools included in ESSUTIL do not directly report ESS performance, but can be used to create performance monitoring scripts that report iostats based on vpaths or ranks.

Information on downloading and installing the ESSUTIL package is available in 5.9, “ESSUTIL utility package” on page 155.

For AIX and HP-UX servers, remember to install the ibm2105 attachment script along with the ESSUTIL package.

6.3 Common UNIX performance monitoring tools

Some common UNIX commands to monitor system performance are:

- **iostat**
- **sar** (System Activity Report)
- **vmstat** (Virtual Memory Statistics)

Keep in mind that when these tools were created, UNIX servers had their own locally attached storage and did not use disk devices presented from centralized disk storage servers like the ESS, which is full of RAID arrays.

We would not call these tools legacy yet, but some of their features do not work well with storage from RAID arrays. When looking at the output from these commands keep in mind that the numbers presented are not for a single disk anymore, but for a logical disk (LUN) on an ESS array (RAID-5 or RAID-10 rank).

These tools are worth discussing because they are almost always available and system administrators are accustomed to using them. You may have to administer a server, and these are the only tools you have available to use. These tools offer a quick way to tell if a system is I/O bound.

6.3.1 IOSTAT

The base tool for evaluating I/O performance of disk devices for UNIX operating systems is **iostat**. Although available on most UNIX platforms, **iostat** varies in its implementation from system to system.

The **iostat** command is useful to determine whether a physical volume is becoming a performance bottleneck. The **iostat** command is a fast way to get a first impression of whether the system has an I/O-bound performance problem or not. The tool reports I/O statistics for TTY devices, disks, and CD-ROMs. It is used for monitoring system I/O device utilization by observing the time physical disks are active in relation to their average transfer rates.

Tip: I/O activity monitors, such as **iostat**, have no way of knowing whether the disk they are seeing is a single physical disk or a logical disk striped upon multiple physical disks in a RAID array. Therefore, some performance figures reported for a device, for example, % busy, could appear high.

It would not be unusual to see a device reported by iostat as 90 percent to 100 percent busy because a ESS volume that is spread across an array of multiple disks can sustain a much higher I/O rate than for a single physical disk. Having a device 100 percent busy would generally be a problem for a single device but probably not for a RAID-5 device.

Tip: When using **iostat** on a server that is running SDD with multiple attachments to the ESS, each disk device is really just a single path to the same logical disk (LUN) on the ESS. To understand how busy a logical disk is, you need to sum up **iostats** for each “disk device” making up a vpath.

Figure 6-2 shows an example of how multiple paths to the ESS affect information presented by **iostat**. In the example, a server has two Fibre Channel adapters and is zoned so that it uses four paths to the ESS.

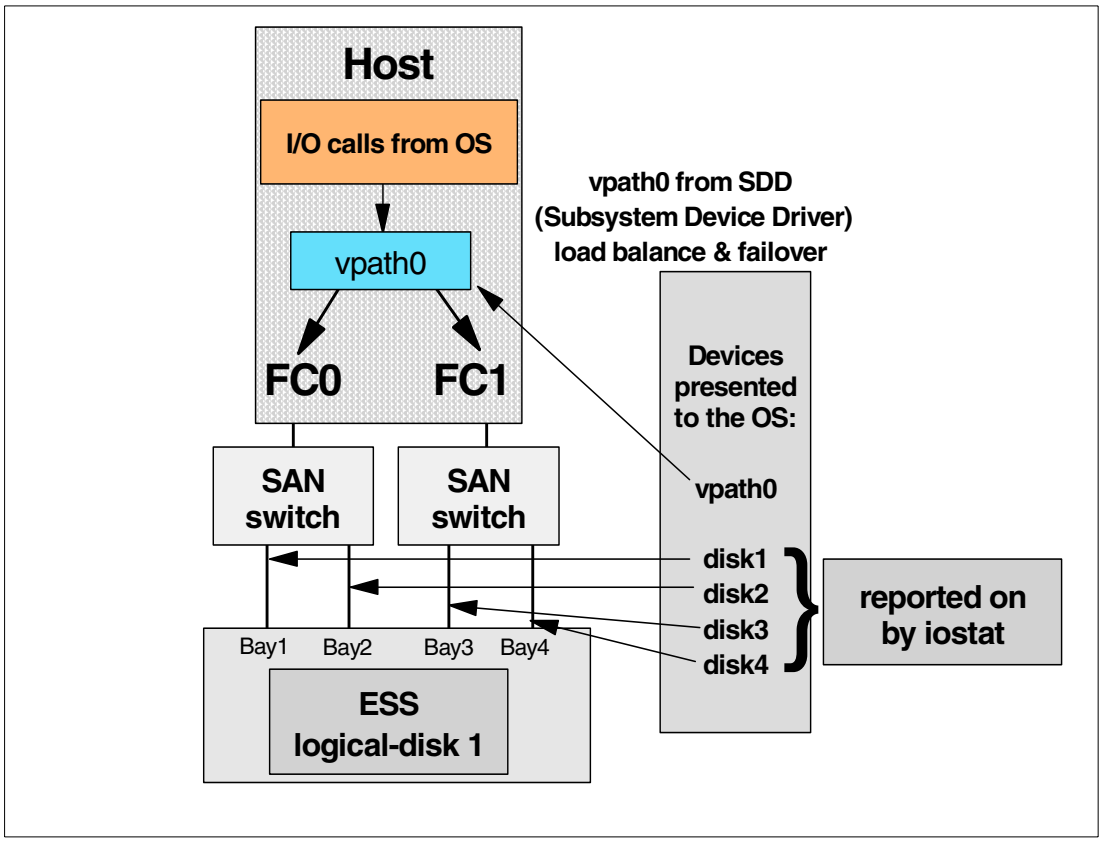


Figure 6-2 Devices presented to iostat

In order to determine the I/O stats for vpath0 for the example given in Figure 6-2, you would need to add up the iostats for disk1–4. One way to find out which disk devices make a vpath is to use the **lssdd** command included in the ESSUTIL package or the **datapath query <device>** command included with SDD.

Example 6-1 shows the output of **datapath query 0**, which lists the “disks” (paths to vpath0). In this example, the logical disk on the ESS has LUN serial number 5049900. The disk devices presented to the operating system are hdisk1, hdisk2, hdisk3, and hdisk4, so we could add up the iostats for hdisk1–4 to see how busy vpath0 is.

We could also use the ESS Expert and check the performance information for volume: 5049900. One drawback to ESS Expert, however, in this scenario is that the minimum time interval setting between reporting stats is 5 minutes.

Example 6-1 Datapath query device

```
$ datapath query 0
datapath query device 0
```

```
DEV#:   0    DEVICE NAME: vpath0   TYPE: 2105F20 SERIAL: 5049900
POLICY: Optimized
============================================================
Path# Adapter/HardDisk State Mode Select Errors
    0 fscsi0/hdisk1    OPEN NORMAL 0
    1 fscsi0/hdisk2    OPEN NORMAL 0
    2 fscsi1/hdisk3    OPEN NORMAL 0
    3 fscsi1/hdisk4    OPEN NORMAL 0
```

For a system with a large number of disk devices presented from the ESS, **iostat** can lose its effectiveness. You may want to try running **iostat** and then sort the output by **%busy**.

If your AIX system is in a SAN environment, you may have so many hdisks that **iostat** presents too much information. We recommend using **nmon**, which can report iostats based on vpaths or ranks, as discussed in 6.5, “AIX-specific I/O monitoring commands” on page 184.

Another idea, which we discuss in 6.7, “Viewing iostats based on vpaths - vpath_iostat script” on page 195, and 6.8, “Viewing iostats based on ranks - ess_iostat script” on page 196, is to feed the **iostat** data to a script that will use ESS utilities like **lsess** to present the iostat information based on vpaths or ranks.

The tables that follow show sample **iostat** reports from IBM AIX, Sun Solaris, and HP-UX systems.

IOSTAT report from AIX

Example 6-2 shows AIX **iostat** output.

Example 6-2 AIX iostat output

```
#iostat
Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk0           0.2       0.9       0.1    42253    459672
hdisk1           2.4       2.9       1.3    81008   1610432
hdisk2           0.2       0.9       0.1    52353    469672
hdisk3           2.8       2.9       1.3    81008   1710332
hdisk4           1.2       0.8       0.1    62253    359672

( **Notice the first stanza of the disk report output is history information **)
```

Together with the KB read and written, the output reports the following:

- %tm_act column indicates the percentage of the measured interval time that the device was busy.
- tps column shows the transactions per second over the interval period for the device. (Please note that a I/O transaction is a variable length of work assigned to a device.) This field may also appear higher than would normally be acceptable for a single physical disk device.

IOSTAT report from Sun Solaris

Figure 6-3 shows an **iostat** report from Sun Solaris. Here you can see an example of a device that appears to be very busy, device sd1.

The r/s column shows 124.3 reads per second; the %b column shows 90 percent busy for the device;but the svc_t column shows a service time of 15.7 ms, quite reasonable for 124 I/Os per second.

The calculations for service time that **iostat** presents are based on a single physical volume, and, as previously mentioned, the physical volume that the ESS presents to the host is in reality comprised of multiple physical volumes.

With RAID disks, the %b figure can be misleading and should not be relied on. To figure out how busy the individual disks are in RAID-array in the ESS, we would need to add up all the iostats for LUNs on that array and divide by the number of disks in the array. The ESS Expert can provide statistics based on RAID arrays and individual logical disks.

Example 6-3 SUN Solaris iostat output

```
#iostat -x
                                 extended disk statistics
disk      r/s   w/s   Kr/s    Kw/s wait actv   svc_t  %w  %b
fd0       0.0   0.0    0.0     0.0  0.0  0.0     0.0   0   0
sd1      124.3  14.5  3390.9   399.7  0.0  2.0   15.7   0  90
sd2       0.7   0.4   13.9     4.0  0.0  0.0     7.8   0   1
sd3       0.4   0.5    2.5     3.8  0.0  0.1     8.1   0   1
sd6       0.0   0.0    0.0     0.0  0.0  0.0     5.8   0   0
sd8       0.3   0.2    9.4     9.6  0.0  0.0     8.6   0   1
sd9       0.7   1.3   12.4    21.3  0.0  0.0     5.2   0   3
The fields have the following meanings:
          disk    name of the disk
          r/s     reads per second
          w/s     writes per second
          Kr/s    kilobytes read per second
          Kw/s    kilobytes written per second
          wait    average number of transactions waiting for ser-
                  vice (queue length)
          actv    average number of transactions  actively  being
                  serviced  (removed  from  the queue but not yet
                  completed)
          svc_t    average service time
          %w      percent of time there are transactions  waiting
                  for service (queue non-empty)
          %b      percent of time the disk is busy  (transactions
                  in progress)
```

Notice that for Sun Solaris, **iostat** uses disk aliases like sdX for disk devices like cxtydz.

Depending on which version of Sun Solaris you are running, you may be able to use an -n flag for **iostat** to list devices in the cXtYdZ format. Figure 6-4 shows the output of **iostat -n** for a Sun Solaris server.

Example 6-4 SUN Solaris iostat output using cxtydz devices

```
example% iostat -xnp
extended device statistics
device    r/s    w/s   kr/s   kw/s wait actv %w  %b
c0t0d0    0.4    0.3   10.4    7.9  0.0  0.0 0   1
c0t0d0s0  0.3    0.3    9.0    7.3  0.0  0.0 0   1
c0t0d0s1  0.0    0.0    0.1    0.5  0.0  0.0 0   0
```

There are also scripts available from Sun or in Sun Solaris user groups to map from sdX aliases to cxtydz devices. Search on the internet for sd_to_cxtydz.sh.

IOSTAT report from HP-UX

Example 6-5 shows an example of an **iostat** report from HP-UX. This is a fairly simple format with three columns of statistics: bps indicates the kilobytes transferred per second; sps indicates the seeks per second; and msps indicates milliseconds per average seek. The first two numbers, bps and sps, are subject to the effects of the ESS RAID architecture.

The man page for the **iostat** command on HP-UX states that the msps field is set to 1.0. With the advent of new disk technologies, such as data striping, where a single data transfer is spread across several disks, the number of milliseconds per average seek becomes impossible to compute accurately. At best it is only an approximation, varying greatly, based on several dynamic system conditions. For this reason, and to maintain backward compatibility, the milliseconds per average seek (msps) field is set to the value 1.0.

Example 6-5 HP-UX iostat output

```
# iostat 1
device    bps   sps        msps
c4t6d0    127     28.5     1.0
c3t6d0    118     24.5     1.0
c6t9d5    10252   44.4     1.0
c5t10d4   135      6.0     1.0
c30t4d1   148      8.0     1.0
c22t6d0   138      8.0     1.0
c31t10d2  138      8.0     1.0
c26t1d6   138      8.0     1.0

Column Heading     Interpretation
   device          Device name
   bps             Kilobytes transferred per second
   sps             Number of seeks per second
   msps            Milliseconds per average seek
```

For HP-UX, you may prefer to use **vmstat -d** to view disk stats, or use both **vmstat** and **iostat**. Details on the HP-UX **vmstat** output are shown in 6.3.3, “VMSTAT” on page 175.

IOSTAT summary

In a SAN environment with the ESS presenting several disk devices to a host, **iostat** output is not as easy to evaluate as when using individual SCSI disks. You will probably want to use another tool that presents iostats based on vpaths or ESS ranks.

With an ESS, also remember that typically the majority of random writes are happening at cache speeds. Data is written to the ESS and stored in cache to be destaged to disks later. For example, you can run a command in one window to copy a large file between file systems on ESS disks. Then in another window, watch **iostat** output. You will see that the write comes back as *complete* before the disk activity has stopped; this is due to the ESS reporting to the host system, that the write is complete as soon as all data was written to ESS cache. **iostat** will show disk activity still taking place as data is destaged from cache to disk.

Taken alone, there is no unacceptable value for any of the above **iostat** fields because statistics are too closely related to application characteristics and system configuration. Therefore, when evaluating data look for patterns and relationships. The most common relationship is between *disk utilization* and *data transfer rate*.

To draw any valid conclusions from **iostat** data, you have to understand the application’s disk data access patterns such as sequential, random, or combination, and the type of physical disk drives and adapters on the system.

For example, if an application reads/writes sequentially, you should expect a high disk transfer rate when you have a high disk busy rate. **Kb_read** and **Kb_wrtn** can confirm an understanding of an application's read/write behavior. However, they provide no information on the data access patterns.

Generally you do not need to be concerned about a high disk busy rate as long as the disk transfer rate is also high. However, if you get a high disk busy rate and a low disk transfer rate, you may have a fragmented logical volume, file system, or individual file that is causing the bottleneck.

Discussions of disk, logical volume, and file system performance sometimes lead to the conclusion that the more drives you have on your system, the better the disk I/O performance. This is not always true since there is a limit to the amount of data that can be handled by the adapter performing the I/O.

6.3.2 SAR

System Activity Report (SAR) provides a quick way to tell if a system is I/O bound. SAR has numerous options, providing paging, TTY, CPU busy, and many other statistics.

There are three possible modes in which to use the **sar** command:

- Real-time sampling and display
- System activity accounting via cron
- Display previously captured data

We will discuss these three modes of using the **sar** command.

Real-time sampling and display

One way you can run **sar** is by specifying a sampling interval and the number of times you want it to run. To collect and display system statistic reports immediately, run `# sar -u 2 5`.

An example of **sar** output is shown in Example 6-6.

Example 6-6 SAR sample output

```
# sar -u 2 5
AIX aixtest 3 4 001750154C00 2/5/03

17:58:15    %usr    %sys         %wio    %idle
17:58:17      43       9            1       46
17:58:19      35      17            3       45
17:58:21      36      22           20       23
17:58:23      21      17            0       63
17:58:25      85      12            3        0

Average       44      15            5       35
```

Not all **sar** options are the same for AIX, HP-UX, and Sun Solaris, but the **sar -u** output is the same. The output in the example shows CPU formation every 2 seconds, 5 times.

To check if a system is I/O bound, the important column to look at is %wio. The %wio includes time spent waiting on I/O from *all* drives, including internal and ESS logical disks. Generally, if a server has over 40 percent waiting on I/O, it is I/O bound. *You need to understand your workload though to make a judgement*. If you are running a video file server, then serving I/O may be the only activity the machine does and you would expect high %wio values.

Also remember that a system with busy CPUs can mask I/O wait. The definition of %wio is: Idle with some process waiting for I/O (only block I/O, raw I/O, or VM pageins/swapins indicated). If the system is CPU busy and also is waiting on I/O, the system accounting will increment the CPU busy but not the %wio column.

The other column headings mean (refer to Example 6-6):

- %usr time system spent executing application code
- %sys time system spent executing operating system calls
- %idle time the system was idle with no outstanding I/O requests

System activity accounting via cron

sar is an un-intrusive program because it just extracts data from information collected by the system. You do need to configure a system to collect data however, and the frequency of the data collection could effect performance and the size of data files collected.

To configure a system to collect data for **sar**, you can run the **sadc** command or the modified **sa1** and **sa2** commands. Here is more information on the **sa** commands and how to configure **sar** data collection:

- The **sa1** and **sa2** commands are shell procedure variants of the **sadc** command.
- The **sa1** command collects and stores binary data in the /var/adm/sa/sadd file, where dd is the day of the month.
- The **sa2** command is designed to be run automatically by the **cron** command and run concurrently with the **sa1** command. The **sa2** command will generate a daily report called /var/adm/sa/sardd. It will also remove a report more than one week old.
- /var/adm/sa/sadd contains the daily data file, dd represents the day of the month. /var/adm/sa/sardd contains the daily report file, dd represents the day of the month. Note the r in /var/adm/sa/sardd for **sa2** output.)

To configure a system to collect data, edit the root crontab file. For our example, if we just want to run **sa1** every 15 minutes every day, and the **sa2** program to generate ASCII versions of the data just before midnight, we will change the cron schedule to look like the following:

```
0,15,30,45 * * * 0-6 /usr/lib/sa/sa1
55 23 * * 0-6 /usr/lib/sa/sa2 -A
```

Display previously captured data

After the **sa1** and **sa2** commands are configured in cron and data collection starts, you will see binary report files in /var/adm/sa/sadd, where dd represents the day of the month.

You can view performance information files from these files with:

```
sar -f /var/adm/sa/sadd where dd is the day you are interested in.
```

You can also focus on a certain time period, say 8 a.m. to 5:15 p.m. with:

```
sar -s 8:00 -e 17:15 -f  /var/adm/sa/sadd
```

Remember, **sa2** will remove the data collection files over a week old as scheduled in cron.

You can save **sar** info to view later with the commands:

```
sar -A -o data.file interval count > /dev/null & [SAR data saved to data.file]
sar -f data.file [Read SAR info back from saved file:]
```

All data is captured in binary form and saved to a file (data.file). The data can then be selectively displayed with the **sar** command using the -f option.

SAR summary

SAR helps to tell quickly if a system is I/O bound. Remember though, that a busy system can mask I/O issues since io_wait counters are not increased if the CPUs are busy. Compare **sar -d** to **iostat** on your system and check the man pages for the different options to use. You may prefer the **sar -d** output to **iostat**.

SAR can help to save a history of I/O performance so you have a baseline measurement for each host. You can then verify if tuning changes make a difference or not. You may want, for example, to collect **sar** data for a week and create reports: 8am-5pm Mon-Friday if that is prime time for random I/O; 6 p.m.–6 a.m. Sat–Sun if those are batch/backup windows.

6.3.3 VMSTAT

The **vmstat** utility is a useful tool for taking a quick snapshot or overview of the systems performance. It is easy to see what is happening to the CPU, paging, swapping, interrupts, I/O wait, and much more. There are several reports that **vmstat** can provide. These reports vary slightly between the different versions of UNIX. Some of the I/O-related system information can be gathered by entering the following options:

vmstat	To display a summary of the statistics since boot
vmstat 2 5	To display five summaries at 2-second intervals
vmstat scdisk13 scdisk14	To display a summary of the statistics since boot including statistics for logical disks scdisk13 and scdisk14

Tip: **vmstat** presents an average-since-boot on the first line. When running **vmstat** over an interval, just disregard the first line of the **vmstat** output.

An example of **vmstat** output for Sun Solaris is shown in Example 6-7.

Example 6-7 SUN Solaris vmstat output

```
# vmstat 5
 procs     memory            page            disk          faults       cpu
 r b w   swap  free    re  mf pi po fr de sr s0 s1 s2 --  in   sy   cs  us sy id
 0 0 0  53218 49592   0   1  5  0  0  0  0  0  1  0  0  116  106   30  1  1  99
 0 0 0 285144 57916   0   1  0  0  0  0  0  2  0  0  0  110    5   19  0  1  99
 0 0 0 285144 57916   0   0  0  0  0  0  0  0  0  0  0  114    8   19  0  0 100
 0 0 0 285144 57916   0   0  0  0  0  0  0  0  0  0  0  103    9   20  0  0 100
```

HP-UX has similar **vmstat** output as shown in Example 6-8. Notice that with the -d flag, you can see transfer stats for disks.

Example 6-8 HP-UX vmstat output

```
vmstat -d

procs           memory                  page  faults                          cpu
r   b   w     avm    free   re   at    pi   po    fr   de    sr  in      sy    cs  us sy id
0   0   0    9158     721    0    0     0    0     0    0     0  101     18   7    0  0  100

      Disk Transfers
        device    xfer/sec
        c0t6d0         0
```

AIX **vmstat** output is shown in Example 6-9.

Example 6-9 AIX vmstat output

```
kthr     memory             page              faults           cpu
----- ----------- ------------------------ ------------      -----------
 r  b   avm   fre     re  pi  po  fr   sr  cy  in   sy     cs  us sy id wa
 0  0 87360 3161051   0   0   0  58  105   0  132   369    169 12  4 75  9
 0  4 85150 3163595   0   0   0   0    0   0  2476  8534   207  1  1 98  0
 1  4 85628 3162817   0   0   0   0    0   0  2510  17963  619  3  3 94  0
 0  4 85002 3163764   0   0   0   0    0   0  2417  13762  90   0  2 98  0
 0  4 85002 3163764   0   0   0   0    0   0  2412   439   39   0  0 99  0
```

The **vmstat** output for HP-UX, AIX, and Sun Solaris are all similar. Some important fields are:

r - runque Shows the number of tasks waiting for CPU resources.

b- blocked Indicates processes are waiting on a resource, usually I/O related.

pi- page in Page-ins from paging space indicate a shortage of free memory and swapping is occurring. Swapping activity can incur I/O costs.

us- user CPU Shows the amount of CPU used by user application code.

sy Shows the percent of CPU being used to service the operating system.

id - idle The percent of CPU that is idle.

wa-wait The percent of time the CPUs are idle, waiting on I/O to complete (AIX only).

vmstat reports are vital in determining what is happening to the system on a real-time basis. Some of the things to look out for in an I/O bound system are:

- High I/O wait percent (AIX includes this information in **vmstat** output in the wa column), which indicates that the majority of the CPU cycles are being wasted waiting for I/O operations to complete.
- High number of blocked processes. This normally indicates that there are a lot of processes waiting on a single resource; usually it is I/O related.
- High paging space paging rate, which indicates an overload on the system memory.
- High number of page faults, which could mean that the system is not making efficient use of memory for caching files.

The **vmstat** command is only the first step to look for performance problems. It gives an indication of where the performance problem could be located. With this in mind, choose a resource-specific command and take a deeper look into the system behavior.

6.4 SDD commands for AIX, HP-UX, and Sun Solaris

For availability and performance, we recommend dual attaching the host to the ESS or SAN fabric, and using SDD. A description of SDD and the common commands it provides is covered in 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.

There are some commands SDD provides that are specific for each platform, and we will cover some of the AIX, HP-UX, and Sun Solaris SDD commands here. All three platforms have the SDD command **datapath** available for use, which is discussed in 5.8.6, “SDD datapath command” on page 153.

6.4.1 AIX SDD commands

There are some particular SDD commands for AIX, which you will want to use for:

- Verifying that a host is using vpath devices properly for redundancy and load-balancing
- Configuring volume groups using vpaths instead of hdisks
- Adding paths to vpaths dynamically

The AIX SDD commands are listed in Table 6-1.

Table 6-1 AIX SDD commands

Command	Description
addpaths	Dynamically adds paths to SDD devices while they are in the Available state
lsvpcfg	Queries the SDD configuration state
dpovgfix	Fixes a SDD volume group that has mixed vpath and hdisk physical volumes
hd2vp	The SDD script that converts an ESS hdisk device volume group to a Subsystem Device Driver vpath device volume group
vp2hd	The SDD script that converts a SDD vpath device volume group to an ESS hdisk device volume group
querysn	The SDD driver tool to query unique serial numbers of ESS devices
lquerypr	The SDD driver persistent reserve command tool
mkvg4vp	Creates a SDD volume group
extendvg4vp	Extends SDD devices to a SDD volume group
savevg4vp	Backs up all files belonging to a specified volume group with SDD devices
restvg4vp	Restores all files belonging to a specified volume group with SDD devices

ADDPATHS

In a SAN environment, where servers are attached to SAN switches, the paths from server to ESS are controlled by *zones* created with the SAN switch software. You may want to add a new path and remove another for planned maintenance on the ESS or for proper load balancing. You can take advantage of the **addpaths** command to make the changes live.

For example, **lsvp** shows that the server in Example 6-10 has four connections to the ESS, but the zones have been configured so that it only uses three ESS host bays and has two paths to ESS Bay-1.

Example 6-10 AIX server with unbalanced connections to ESS host bays

```
lsvp -d
          BAY-1(B1)          BAY-2(B2)          BAY-3(B3)          BAY-4(B4)
        H1  H2  H3  H4     H1  H2  H3  H4     H1  H2  H3  H4     H1  H2  H3  H4
       A B A B A B A B    A B A B A B A B    A B A B A B A B    A B A B A B A B

     0 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     1 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     2 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     3 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     4 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     5 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     6 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
     7 Y - - - - - Y -    - - - - - - Y -    - - - - - - Y -    - - - - - - - -
```

```
Y  =  online/open
```

To add a new path to ESS Bay4-Port4 and remove the path to Bay1-Port1 in this case, you could:

1. Add a fiber cable between the SAN switch and ESS Bay4-Port4 if necessary.
2. Re-zone the server using SAN software to add a path from server to ESS Bay4Port4.
3. Run the **addpaths** command on the server.
4. Verify that the new path is added with **lsvp**.
5. Bring down the path to ESS Bay1-Port1 by using “lsvp -l R1-B1-H1”.
6. Re-zone the server so that the path to Bay1-Port1 is removed.

After the changes, **lsvp** output will look similar to what is shown in Example 6-11.

Example 6-11 AIX server with balanced paths to the ESS

```
lsvp -d
            BAY-1(B1)           BAY-2(B2)           BAY-3(B3)           BAY-4(B4)
          H1  H2  H3  H4      H1  H2  H3  H4      H1  H2  H3  H4      H1  H2  H3  H4
         A B A B A B A B     A B A B A B A B     A B A B A B A B     A B A B A B A B

       0  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       1  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       2  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       3  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       4  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       5  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       6  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -
       7  N - - - - - Y -     - - - - - - Y -     - - - - - - Y -     - - - - - - Y -

Y  =  online/open
N  =  offline
```

After a reboot of the host system, the offline status for the path to ESS Host Bay1-Port1 in the **lsvp** output will disappear.

LSVPCFG

To display which ESS vpath devices are available to provide fail over protection, run the **lsvpcfg** command. You will see output similar to that shown in Example 6-12.

Example 6-12 LSVPCFG for AIX output

```
# lsvpcfg
vpath0 (Avail pv vpathvg)018FA067=hdisk1 (Avail )
vpath1 (Avail )019FA067=hdisk2 (Avail )
vpath2 (Avail )01AFA067=hdisk3 (Avail )
vpath3 (Avail )01BFA067=hdisk4 (Avail )hdisk27(Avail )
vpath4 (Avail )01CFA067=hdisk5 (Avail )hdisk28 (Avail )
vpath5 (Avail )01DFA067=hdisk6 (Avail )hdisk29 (Avail )
vpath6 (Avail )01EFA067=hdisk7(Avail )hdisk30 (Avail )
vpath7(Avail )01FFA067=hdisk8 (Avail )hdisk31 (Avail )
vpath8 (Avail )020FA067=hdisk9 (Avail )hdisk32 (Avail )
vpath9 (Avail pv vpathvg)02BFA067=hdisk20 (Avail )hdisk44 (Avail )
vpath10 (Avail pv vpathvg)02CFA067=hdisk21 (Avail )hdisk45 (Avail )
vpath11 (Avail pv vpathvg)02DFA067=hdisk22 (Avail )hdisk46 (Avail )
vpath12 (Avail pv vpathvg)02EFA067=hdisk23 (Avail )hdisk47(Avail )
vpath13 (Avail pv vpathvg)02FFA067=hdisk24 (Avail )hdisk48 (Avail
```

Notice in the example that **vpath0**, **vpath1**, and **vpath2** all have a single path (hdisk device) and, therefore, will not provide fail over protection because there is no alternate path to the ESS LUN. The other SDD vpath devices have two paths and, therefore, can provide fail over protection and load balancing.

DPOVGFIX

It is possible for certain commands such as **chdev** on an hdisk device to cause a pvid (physical volume ID) to move back to an hdisk (single path to ESS LUN) instead of remaining on the vpath device. For example, look at the output shown in Example 6-13. The **lsvpcfg** command shows that hdisk46 is part of the volume group **vpathvg** and has a pvid assigned.

The command **lsvg -p vpathvg** lists the physical volumes making up the volume group **vpathvg**. Notice that hdisk46 is listed among the other vpath devices. This is not correct for fail over and load balancing because access to the ESS logical disk with serial number 02DFA067 is using a single path hdisk46 instead of vpath11. The system is operating in a mixed-mode with vpath pseudo devices and partially uses hdisk devices.

Example 6-13 AIX loss of a device path

```
#lsvpcfg
vpath11 (Avail pv vpathvg)02DFA067=hdisk22 (Avail )hdisk46 (Avail pv vpathvg)
vpath12 (Avail pv vpathvg)02EFA067=hdisk23 (Avail )hdisk47(Avail )
vpath13 (Avail pv vpathvg)02FFA067=hdisk24 (Avail )hdisk48 (Avail )

lsvg -p vpathvg
vpathvg:
PV_NAME PV STATE TOTAL PPs FREE PPs FREE DISTRIBUTION
vpath10 active 29 4 00..00..00..00..04
hdisk46 active 29 4 00..00..00..00..04 !!! MIXED MODE- HDISKs and VPATHS !!!
vpath12 active 29 4 00..00..00..00..04
vpath13 active 29 28 06..05..05..06..06
```

To fix this problem, run the command **dpovgfix volume_group_name**. Then re-run the **lsvpcfg** or **lsvg** command to verify.

Note: In order for the dpovgfix shell script to be executed, all mounted file systems of this volume group have to be unmounted. After successful completion of the dpovgfix shell script, mount the file systems again.

HD2VP and VP2HD

SDD provides two conversion scripts to move volume group devices to vpaths or hdisks:

- The hd2vp script converts a volume group from ESS hdisks into SDD vpaths. The syntax for the hd2vp script is as follows: hd2vp vgname.
- The vp2hd script converts a volume group from SDD vpaths into ESS hdisks. Use the vp2hd program when you want to configure your applications back to original ESS hdisks, or when you want to remove the SDD from your host system. The syntax for the vp2hd script is: vp2hd vgname.

These two conversion programs require that a volume group contain either all original ESS hdisks or all SDD vpaths. The program fails if a volume group contains both kinds of device special files (mixed volume group). You may need to use **dpovgfix** first to fix a volume group to contain all of one kind of device or another.

QUERYSN for multi-booting AIX off the ESS

With the maintenance level of the AIX 4.3.3 and AIX 5.1 operating systems, AIX supports Fibre Channel boot capability for selected pSeries and RS/6000 systems. This allows you to select ESS Fibre Channel devices as the boot device. However, a multi-pathing boot device is not supported. If you plan to select a device as a boot device, you should not configure that ESS device with multi-path configuration.

The SDD driver will automatically exclude any ESS devices from SDD configuration if these ESS boot devices are the physical volumes of an active rootvg.

Tip: If you require dual or multiple boot capabilities on a server, and multiple operating systems are installed on multiple ESS boot devices, you should use the **querysn** command to manually exclude all ESS boot devices that belong to multiple non-active rootvg volume groups on the server.

SDD V1.3.3.3 allows you to manually exclude ESS devices from the SDD configuration. The **querysn** command reads the unique serial number of an ESS device (hdisk) and saves the serial number in an exclude file, /etc/vpexclude.

During the SDD configuration, SDD configure methods read all the serial numbers in this exclude file and exclude these ESS devices from the SDD configuration.

The exclude file, /etc/vpexclude, holds the serial numbers of all inactive ESS devices (hdisks) in the system. If an exclude file exists, the **querysn** command will add the excluded serial number to that file. If no exclude file exists, the **querysn** command will create one. There is no user interface to this file.

Tip: You should not use the **querysn** command on the same logical device multiple times. Using the **querysn** command on the same logical device multiple times results in duplicate entries in the /etc/vpexclude file, and the system administrator will have to administer the file and its content.

The syntax for **querysn** is:**querysn <-d> -l device-name**.

Managing secondary system paging space for AIX

For better performance, you may want to place a secondary paging space on the ESS. SDD 1.3.2.6 (or later) supports secondary system paging on a multi-path Fibre Channel vpath device from an AIX 4.3.3 or AIX 5.1.0 host system to an ESS.

SDD supports secondary paging on ESS. The benefits are multi-pathing to your paging spaces. All the same commands for hdisk-based volume groups apply to using vpath-based volume groups for paging spaces.

Important: IBM does not recommend moving the primary paging space out of rootvg. Doing so may mean that no paging space is available during the system startup. Do not redefine your primary paging space using vpath devices.

The following steps provide information about managing secondary system paging space.

Listing paging spaces

You can list paging spaces by typing the following:

```
lsps -a
```

Adding a paging space

You can add a paging space by typing the following:

```
mkps -a -n -s NN vg
```

The **mkps** command recognizes the following options and arguments:

-a	The -a option makes the new paging space available at all system restarts.
-n	The -n option activates the new paging space immediately.
–sNN	The -s argument accepts the number of logical partitions (NN) to allocate to the new paging space.
vg	The volume group name in which a paging logical volume is to be created.

Removing a paging space

You can remove a specified secondary paging space that is not active. For example, to remove paging space PS01, type:

```
rmps PS01
```

LQUERYPR

The **lquerypr** command implements certain SCSI-3 persistent reservation commands on a device. The device can be either hdisk or SDD vpath devices. This command supports persistent reserve service actions or read reservation key, release persistent reservation, preempt-abort persistent reservation, and clear persistent reservation.

The syntax and options are:

```
lquerypr [[-p]|[-c]|[-r]][-v][-V][-h/dev/PVname]
```

Flags:

–p	If the persistent reservation key on the device is different from the current host reservation key, it preempts the persistent reservation key on the device.
–c	If there is a persistent reservation key on the device, it removes any persistent reservation and clears all reservation key registration on the device.
–r	Removes the persistent reservation key on the device made by this host.
–v	Displays the persistent reservation key if it exists on the device.
–V	Verbose mode. Prints detailed message.

To query the persistent reservation on a device, type:

```
lquerypr -h /dev/vpath30
```

This command queries the persistent reservation on the device without displaying. If there is a persistent reserve on a disk, it returns 0 if the device is reserved by the current host. It returns 1 if the device is reserved by another host. You must be extra careful when using the **lquerypr** command.

MKVG4VP, EXTENDVG4VP, SAVEVG4VP, and RESTVG4VP

mkvg4vp, **extendvg4vp**, **savevg4vp**, and **restvg4vp** have the same functionality as their counterpart commands without the 4vp extension; use the 4vp versions when operating on

vpath devices. These commands will maintain pvids on vpaths and keep SDD working properly.

It is a good idea to check periodically to make sure none of the volume groups are using hdisks instead of vpaths. You can verify the path status several ways. Some commands are:

- **lspv** (look for hdisk with Volume Group names listed)
- **lssdd**
- **lsvpcfg**
- **lsvg -p <vgname>**

Remember to change any scripts you may have that call savevg or restvg and change the calls to savevg4vp and restvg4vp.

6.4.2 HP-UX SDD commands

SDD for HP-UX adds the specific commands shown in Table 6-2.

Table 6-2 HP-UX SDD commands

Command	Description
showvpath	Lists the configuration mapping between SDD devices and underlying disks
hd2vp	Converts a volume group from ESS hdisks into SDD vpaths
vp2hd	Converts a volume group from SDD vpaths into ESS hdisks

SHOWVPATH

The **showvpath** command for HP-UX is similar to the **lsvpcfg** command for AIX. Use **showvpath** to verify that an HP-UX vpath is using multiple paths to the ESS. An example of the output from **showvpath** is displayed in Example 6-14.

Example 6-14 SHOWPATH command for HP-UX

```
#/opt/IBMdpo/bin/showvpath
vpath0:
   /dev/dsk/c3t4d0
   /dev/dsk/c3t6d0
vpath1:
   /dev/dsk/c4t6d0
```

Notice that vpath0 in the example has two paths to the ESS. vpath1, however, has a single point of failure since it is only using a single path.

Tip: You can use the output from **showvpath** to modify **iostat** or **sar** information to report stats based on vpaths instead of hdisks. Gather iostats to a file, and then replace the disk names with the corresponding vpaths.

HD2VP and VP2HD

The **hd2vp** and **vp2hd** commands work the same for HP-UX as they do for AIX. Use **hd2vp** to convert volume groups to use vpaths and take advantage of the fail over and load balancing

features of SDD. When removing SDD, you can move the volume group devices back to disk devices using the **vp2hd** command.

6.4.3 Sun Solaris SDD commands

On Sun Solaris, SDD resides above the Sun SCSI disk driver (sd) in the protocol stack. For more information on how SDD works, refer to 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149. SDD is supported on Solaris Version 2.6, Solaris 7, and Solaris 8.

Some specific commands SDD provides to Sun Solaris are listed below as well as the steps to update SDD after making ESS logical disk configuration changes for a Sun server.

CFGVPATH

The **cfgvpath** command configures vpath devices using the following process:

- Scan the host system to find all ESS devices (LUNs) that are accessible by the Sun host.
- Determine which Shark devices (LUNs) are the same devices that are accessible through different paths.
- Create configuration file /etc/vpath.cfg to save the information about ESS devices.
- With the -c option: **cfgvpath** exits without initializing the SDD driver. The SDD driver will be initialized after reboot. This option is used to reconfigure SDD after a hardware reconfiguration.
- Without the -c option: **cfgvpath** initializes the SDD device driver vpathdd with the information stored in /etc/vpath.cfg and creates pseudo-vpath devices /devices/pseudo/vpathdd*.

Note: **cfgvpath** without the -c option should not be used after hardware reconfiguration since the SDD driver is already initialized with previous configuration information. A reboot is required to properly initialize the SDD driver with the new hardware configuration information.

VPATHMKDEV

The **vpathmkdev** command creates files vpathMsN in the /dev/dsk/ and /dev/rdsk/ directories by creating links to the pseudo-vpath devices /devices/pseudo/vpathdd*, which are created by the SDD driver.

Files vpathMsN in the /dev/dsk/ and /dev/rdsk/ directories provide block and character access to an application the same way as the cxtydzsn devices created by the system. The **vpathmkdev** command is executed automatically during SDD package installation and should be executed manually to update files vpathMsN after hardware reconfiguration.

SHOWVPATH

The **showpath** command lists all SDD devices and their underlying disks. An example of the **showpath** command is displayed in Example 6-15.

Example 6-15 SUN Solaris showpath command output

```
# showpath
vpath0c
c1t8d0s2 /devices/pci@1f,0/pci@1/scsi@2/sd@1,0:c,raw
c2t8d0s2 /devices/pci@1f,0/pci@1/scsi@2,1/sd@1,0:c,raw
```

Tip: Note that you can use the output from **showvpath** to modify **iostat** or **sar** information to report stats based on vpaths instead of hdisks. Gather iostats to a file, and then replace the disk device names with the corresponding vpaths.

Changing an SDD hardware configuration in Sun Solaris

When adding or removing multi-port SCSI devices from a Sun Solaris system, you must reconfigure SDD to recognize the new devices. Perform the following steps to reconfigure SDD:

1. Shut down the system. Type `shutdown -i0 -g0 -y` and press Enter.
2. Perform a configuration restart. From the OK prompt, type `boot -r` and press Enter.

   This uses the current SDD entries during restart, not the new entries. The restart forces the new disks to be recognized.
3. Run the SDD configuration utility to make the changes to the directory /opt/IBMdpo/bin. Type `cfgvpath -c` and press Enter.
4. Shut down the system. Type `shutdown -i6 -g0 -y` and press Enter.
5. After the restart, change to the /opt/IBMdpo/bin directory by typing `cd /opt/IBMdpo/bin`.
6. Type `devfsadm` and press Enter to reconfigure all the drives.
7. Type `vpathmkdev` and press Enter to create all the vpath devices.

For specific information on SDD commands, check the latest *IBM TotalStorage Subsystem Device Driver User's Guide,* SC26-7478.

6.5 AIX-specific I/O monitoring commands

In this section we discuss some tools unique to AIX for monitoring system performance. The commands featured are:

- **topas**
- **nmon**
- **filemon**
- **lvmstat**

The **topas** and **nmon** commands are very thorough, providing an overall view of system performance including such perf stats as cpu busy, memory usage, disk I/O, adapter I/O, top processes, and paging activity. The **filemon** and **lvmstat** tools look at I/O performance in more detail and can be used to see for which applications and file systems a host spends the most time handling I/O for.

The **nmon** command is especially good for monitoring ESS activity because it can report:

- iostats based on either:
  - hdisks
  - vpaths
  - ranks (currently limited to 32 ranks at a time)
- Adapter stats including SCSI and Fibre Channel adapters

6.5.1 TOPAS

The interactive AIX tool, **topas**, is convenient if you want to get a quick overall view of the system's current activity. A fast snapshot of memory usage or user activity can be a helpful

starting point for further investigation. However, **topas** is of very limited use as a diagnostic tool when you are dealing with a large number of logical disks on an ESS since it reports I/O on hdisks. Example 6-16 contains a sample **topas** screen shot.

For monitoring ESS I/O on AIX hosts, we recommend the use of another tool called **nmon**, which is discussed in the next section.

Example 6-16 TOPAS output

```
Topas Monitor for host:    bigbluebox           EVENTS/QUEUES    FILE/TTY
Thu May 31 10:44:28 2001   Interval:  2         Cswitch     528  Readch    28532
                                                Syscall    1269  Writech     140
Kernel    0.1   |                             | Reads       104  Rawin        0
User      0.8   |                             | Writes        3  Ttyout       94
Wait     40.7   |###########                  | Forks         2  Igets         0
Idle     58.2   |###############              | Execs         2  Namei        42
                                                Runqueue    0.0  Dirblk        0
Interf   KBPS   I-Pack  O-Pack   KB-In  KB-Out  Waitqueue  12.0
en2       0.6     2.0     2.5     0.2     0.4
lo0       0.0     0.0     0.0     0.0     0.0  PAGING           MEMORY
                                                Faults      321  Real,MB   49151
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  Steals        0  % Comp     18.2
hdisk1    0.0      0.0     0.0     0.0     0.0  PgspIn        0  % Noncomp  11.9
hdisk2    0.0      0.0     0.0     0.0     0.0  PgspOut       0  % Client    0.8
hdisk4    0.0      0.0     0.0     0.0     0.0  PageIn        0
hdisk0    0.0      0.0     0.0     0.0     0.0  PageOut       0  PAGING SPACE
hdisk3    0.0      0.0     0.0     0.0     0.0  Sios          0  Size,MB   33824
                                                                 % Used      0.5
topas    (52976) 19.0% PgSp: 1.8mb perfpol2                      % Free     99.4
dtgreet  (8552)   1.0% PgSp: 1.3mb root
db2sysc  (46044)  0.5% PgSp: 0.4mb reg64
db2sysc  (37534)  0.5% PgSp: 0.4mb reg64          Press "h" for help screen.
init     (1)      0.0% PgSp: 0.6mb root           Press "q" to quit program.
```

6.5.2 NMON

The new **nmon** tool for AIX servers is available for free on the internet at:

http://www-1.ibm.com/servers/esdd/articles/analyze_aix/index.html

Note: The **nmon** tool is not formally supported. No warranty is given or implied, and you cannot obtain help or maintenance from IBM.

The **nmon** tool currently comes in two versions for running on different versions of AIX:

- **nmon** for use with 32-bit kernels (AIX 4.3 and AIX 5.1 on 32- or 64-bit hardware)
- **nmon64** for use with the 64-bit kernel (AIX 5.1)

The **nmon** tool is very similar to **monitor** or **topas**, which you may have used before to monitor AIX interactively, but it offers many more features that are useful for monitoring the ESS and SAN performance.

Unlike **topas**, **nmon** can also record data that can be used to establish a baseline of performance for comparison later. Recorded data can be saved in spreadsheet format for easy graphing also.

The **nmon** tool is an excellent way to show a lot of system monitoring information, of your choice, on one screen. When used interactively, **nmon** shows stats on an ASCII display and updates every few seconds (2 seconds default) unless you select a different refresh rate. To

run the tool, just enter `nmon` and press a key corresponding to each option you want to see performance for, for example, A for Adapter performance.

The different options you can select when running **nmon** version 8 are shown in Example 6-17.

Example 6-17 NMON version 8: Options for interactive mode

```
nmon v8a   Hostname=garmo-aix Refresh=2.0secs  22:07.4
HELP key  --- statistics which toggle on/off ---
     h = This help information
     r = RS6000/pSeries CPU/cache/AIX/kernel/hostname details + LPAR
     t = Top Process Stats 1=basic  2=CPU-Use 3=CPU 4=Size 5=Disk-I/O
         u = shows command arguments (hit u again to refresh)
     c = CPU by processor             l = longer term CPU averages
     m = Memory and Paging stats      k = Kernel Internal stats
     n = Network stats                j = JFS Usage Stats
     d = Disk I/O Graphs D=Stats      o = Disks %Busy Map
     a = Adapter I/O Stats            e = ESS vpath Stats
     A = Async I/O Servers           g = Disk Groups (see cmdline -g)
     v = Verbose Simple Checks - OK/Warnings/Danger
     b = black & white mode           w = view AIX wait processes
     f = Fast Response Cache Accelerator for IBM HTTP web server

     --- controls ---
     + and - = double or half the screen refresh time
     q = quit                      space = refresh screen now
     . = Minimum Mode =display only busy disks and processes
     0 = reset peak counts to zero (peak = ">")
```

Recording NMON information for import into spread sheets

A great benefit **nmon** provides is the ability to collect data over time to a file and then just import the file with a spread sheet program.

To collect **nmon** data in comma-separated format for easy spread sheet import, do the following:

1. Run **nmon** with the -f flag. See **nmon -h** for the details, but as an example, to run **nmon** for an hour capturing data snapshots every 30 seconds use:

   ```
   nmon -f -s 30 -c 120
   ```

2. This will create the output file in the current directory called:

   ```
   <hostname>_date_time.nmon
   ```

3. Before this file can be loaded into a spreadsheet, it needs to be sorted. On AIX, follow this example:

   ```
   sort -A <hostname>_date_time.nmon > xxx.csv
   ```

To load this into a spreadsheet, check the spreadsheet documentation for loading comma-separated data files. Many spreadsheets accept this data as just one of the possible files to load or provide an import function to do this.

Many spreadsheets have fixed numbers of columns and rows. We suggest you collect a maximum of 300 snapshots to avoid hitting these issues.

When you are capturing data to a file, the **nmon** tool disconnects from the shell to ensure that it continues running even if you log out. This means that **nmon** can appear to crash, but it is still running in the background.

NMON options for ESS performance monitoring

For ESS-attached hosts, there are four features of **nmon** that you will definitely want to take advantage of:

- View adapter I/O performance including Fibre Channel adapters.
- View real-time vpath performance.
- View real-time rank performance.
- Record data for easy import into spreadsheets.

I/O adapter performance

Example 6-18 displays the ability of **nmon** to show I/O performance based on system adapters. Notice the output shows two SCSI controllers and two Fibre Channel adapters. The I/O load is balanced across each Fibre Channel adapter as we would expect if SDD is functioning properly. At the time these **nmon** stats were taken, system garmo-aix was performing a backup, so the workload shows heavy sequential reads performing at 101 MB/sec.

Example 6-18 NMON example of adapter stats

```
nmon v6f               Hostname=garmo-aix          Refresh=2.0secs  19:57.15

Adapter I/O      read       write        xfers              Adapter Type
10-60             2.0       0.0 kB/s      0.5             Wide/Fast-20 SCSI I/O Controller
30-68             0.0       0.0 kB/s      0.0             Wide/Fast-20 SCSI I/O Controller
20-58-01      50679.7       0.0 kB/s      461.5           FC SCSI I/O Controller Protocol Device
30-70-01      51022.9       0.0 kB/s      464.0           FC SCSI I/O Controller Protocol Device

TOTALS       101704.7      0.0 kB/s      925.9    TOTAL=101704.7
```

vpath iostats

The next example of **nmon** shows I/O activity based on vpaths. Notice this system has 199 vpaths however, so we lose some effectiveness here. If you have a large number of vpaths, you will want to list I/O stats based on ESS ranks, which we discuss in the next section.

We also discuss creating a script to measure iostats based on ranks in 6.8, “Viewing iostats based on ranks - ess_iostat script” on page 196.

Example 6-19 NMON vpath output

```
nmon ‘v’ option
ESS I/O        AvgBusy         read-KB/s      write-KB/s  xfers/s        Total vpaths=199
vpath0           2.0%            536.5          0.0           4.4
vpath1           0.2%            189.4          0.0           1.5
vpath2           0.2%            126.2          0.0           1.0
vpath3           4.2%           1262.4          0.0           9.9
```

Rank iostats

NMON version 8 has a feature called *disk grouping*. First you need to create a file that maps hdisks to nicknames. For example, you could create a map file like that shown in Example 6-20.

Example 6-20 NMON disk-group mapping file

```
vi /tmp/vg-maps

rootvg hdisk0 hdisk1
testvg hdisk2 hdisk3 hdisk4
oraclevg hdisk5 hdisk6 hdisk7 hdisk8 hdisk9 hdisk10
```

Then start **nmon** with the -g flag to point to the map file:

```
nmon -g /tmp/vg-maps
```

When **nmon** starts, press the G key to view stats for your disk groups. An example of the disk-group performance output from **nmon** is shown in Example 6-21.

Example 6-21 NMON disk group output

```
nmon v8a [H for help]  Hostname=garmo-aix  Refresh=1.0secs  14:39.03
Disk Group I/O
Name          Disks AvgBusy Read|Write-KB/s  TotalMB/s   xfers/s   R:W-SizeKB
rootvg            2   0.0%       0.0|0.0          0.0       0.0   NaNQ
testvg            3   0.0%       0.0|0.0          0.0       0.0   NaNQ
oraclevg          6   0.0%       0.0|0.0          0.0       0.0   NaNQ
        TOTALS   11   0.0%       0.0|0.0          0.0       0.0/s
```

Notice that:

- **nmon** reports real-time iostats for the different disk groups.
- In this case, the disk groups we created are for volume groups.
- You can create logical groupings of hdisk for any kind of group you like.
- You can make multiple disk-group map files and start **nmon -g <map-file>** to report on different groups.

To enable **nmon** to report iostats based on ranks, you can make a disk-group map file listing ranks with the associated hdisk members.

Use the ESS Utility **lssdd** to provide a list of ranks and hdisks for your system. For example, we created a disk-group file to measure iostats based on ranks for testvg. An example of the **lssdd** output for testvg is shown in Example 6-22.

Example 6-22 LSSDD output for testvg

```
garmo-aix:/ESS> lssdd | grep testvg
garmo-aix    testvg     vpath0   hdisk6   2V-08-01       10B21621 Y  R1-B3-H2-ZA 33.7  11   11   1100
garmo-aix    testvg     vpath0   hdisk14  2b-08-01       10B21621 Y  R1-B1-H2-ZA 33.7  11   11   1100
garmo-aix    testvg     vpath2   hdisk8   2V-08-01       00B21621 Y  R1-B3-H2-ZA 33.7  10   11   1000
garmo-aix    testvg     vpath2   hdisk16  2b-08-01       00B21621 Y  R1-B1-H2-ZA 33.7  10   11   1000
garmo-aix    testvg     vpath4   hdisk10  2V-08-01       30B21621 Y  R1-B3-H2-ZA 33.7  13   11   1300
garmo-aix    testvg     vpath4   hdisk18  2b-08-01       30B21621 Y  R1-B1-H2-ZA 33.7  13   11   1300
garmo-aix    testvg     vpath6   hdisk12  2V-08-01       20B21621 Y  R1-B3-H2-ZA 33.7  12   11   1200
garmo-aix    testvg     vpath6   hdisk20  2b-08-01       20B21621 Y  R1-B1-H2-ZA 33.7  12   11   1200
garmo-aix    testvg     vpath8   hdisk22  2V-08-01       50B21621 Y  R1-B3-H2-ZA 33.7  15   11   1500
garmo-aix    testvg     vpath8   hdisk30  2b-08-01       50B21621 Y  R1-B1-H2-ZA 33.7  15   11   1500
garmo-aix    testvg     vpath10  hdisk24  2V-08-01       40B21621 Y  R1-B3-H2-ZA 33.7  14   11   1400
garmo-aix    testvg     vpath10  hdisk32  2b-08-01       40B21621 Y  R1-B1-H2-ZA 33.7  14   11   1400
garmo-aix    testvg     vpath12  hdisk26  2V-08-01       70B21621 Y  R1-B3-H2-ZA 33.7  17   11   1700
garmo-aix    testvg     vpath12  hdisk34  2b-08-01       70B21621 Y  R1-B1-H2-ZA 33.7  17   11   1700
garmo-aix    testvg     vpath14  hdisk28  2V-08-01       60B21621 Y  R1-B3-H2-ZA 33.7  16   11   1600
garmo-aix    testvg     vpath14  hdisk36  2b-08-01       60B21621 Y  R1-B1-H2-ZA 33.7  16   11   1600
```

From the **lssdd** output, you can see which hdisks belong to each ESS rank; rank 1100 contains hdisk6 and hdisk14, for example. Using the **lssdd** output, we made a disk-group map for **nmon**, as shown in Example 6-23.

Example 6-23 NMON disk-group file for ESS RANKS

```
vi /tmp/disk-group
```

```
rank1100 hdisk6 hdisk14
rank1000 hdisk8 hdisk16
rank1200 hdisk12 hdisk20
rank1300 hdisk10 hdisk18
rank1500 hdisk22 hdisk30
rank1400 hdisk24 hdisk32
rank1700 hdisk26 hdisk34
rank1600 hdisk28 hdisk36
```

After making the disk-group file called /tmp/disk-group, we started **nmon** with:

```
nmon -g /tmp/disk-group
```

Then we and pressed G to view the performance for our disk (rank) groups. The **nmon** output is shown in Example 6-24.

Example 6-24 NMON stats for RANKS

```
nmon v8a [H for help]  Hostname=garmo-aix Refresh=2.0secs  19:32.08
Disk Group I/O
Name          Disks AvgBusy Read|Write-KB/s  TotalMB/s   xfers/s   R:W-SizeKB
rank1100          2  30.5%   15671.6|0.0         15.3     122.9  127.5
rank1000          2  30.5%   13750.8|0.0         13.4     109.9  125.1
rank1200          2  34.5%   15669.6|0.0         15.3     122.9  127.5
rank1300          2  29.5%   13688.9|0.0         13.4     107.4  127.4
rank1500          2  33.0%   15733.5|0.0         15.4     123.4  127.5
rank1400          2  37.7%   15733.5|0.0         15.4     122.9  128.0
rank1700          2  35.5%   15733.5|0.0         15.4     122.9  128.0
rank1600          2  39.2%   15669.6|0.0         15.3     122.9  127.5
        TOTALS   16  16.4%  112186.3|0.0        109.6     880.4/s
```

Notice that **nmon** output for Example 6-24 shows:

- The read/write speeds for each rank.
- Each rank averaged 15 MB/sec reads and no writes.
- We see 2 disk devices per rank as we would expect from the /tmp/disk-group map.
- Read speeds were 112 MB/sec.
- Transactions/second to the testvg were 880 tps.
- The KB/transaction size was about 128 KB/transaction.

6.5.3 FILEMON

The **filemon** command monitors a trace of file system and I/O system events, and reports performance statistics for files, virtual memory segments, logical volumes, and physical volumes. The **filemon** command is useful to those whose applications are believed to be disk-bound, and want to know where and why.

Monitoring disk I/O with the **filemon** command is usually done when there is a known performance issue with regards to the I/O. The **filemon** command will show the load on different disks, logical volumes, and files in great detail.

The **filemon** command resides in /usr/sbin and is part of the bos.perf.tools file set, which can be installed from the AIX base installation media.

Filemon syntax

The syntax of the **filemon** command is as follows:

```
filemon [-d ][-i Trace_File -n Gennames_File ][-o File ] [ -O Levels ] [-P ] [ -T n ] [
--u ][ --v ]
```

Flags:

- -i Trace_File

  Reads the I/O trace data from the specified Trace_File, instead of from the real-time trace process. The **filemon** report summarizes the I/O activity for the system and period represented by the trace file. The -n option must also be specified.

- -n Gennames_File

  Specifies a Gennames_File for offline trace processing. This file is created by running the **gennames** command and redirecting the output to a file as follows (the -i option must also be specified): **gennames >file**.

- -o File

  Writes the I/O activity report to the specified file instead of to the stdout file.

- -d

  Starts the **filemon** command, but defers tracing until the **trcon** command has been executed by the user. By default, tracing is started immediately.

- -T n

  Sets the kernel's trace buffer size to n bytes. The default size is 32,000 bytes. The buffer size can be increased to accommodate larger bursts of events (a typical event record size is 30 bytes).

- -P

  Pins monitor process in memory. The -P flag causes the **filemon** command's text and data pages to be pinned in memory for the duration of the monitoring period. This flag can be used to ensure that the real-time **filemon** process is not paged out when running in a memory constrained environment.

- -v

  Prints extra information in the report. The most significant effect of the -v flag is that all logical files and all segments that were accessed are included in the I/O activity report, instead of only the 20 most active files and segments.

- -O Levels

  Monitors only the specified file system levels. Valid level identifiers are:

  lf Logical file level

  vm Virtual memory level

  lv Logical volume level

  pv Physical volume level

  all Short for lf, vm, lv, and pv

  The vm, lv, and pv levels are implied by default.

- -u

  Reports on files that were opened prior to the start of the trace daemon. The process ID (PID) and the file descriptor (FD) are substituted for the file name.

Filemon measurements

To provide a more complete understanding of file system performance for an application, the **filemon** command monitors file and I/O activity at four levels:

- Logical file system

  The **filemon** command monitors logical I/O operations on logical files. The monitored operations include all read, write, open, and seek system calls, which may or may not result in actual physical I/O depending on whether the files are already buffered in memory. I/O statistics are kept on a per-file basis.

- Virtual memory system

  The **filemon** command monitors physical I/O operations (that is, paging) between segments and their images on disk. I/O statistics are kept on a per segment basis.

- Logical volumes

  The **filemon** command monitors I/O operations on logical volumes. I/O statistics are kept on a per-logical volume basis.

- Physical volumes

  The **filemon** command monitors I/O operations on physical volumes. At this level, physical resource utilizations are obtained. I/O statistics are kept on a per-physical volume basis.

Filemon examples

A simple way to use **filemon** is to run the command shown in Example 6-25, which will:

- Run **filemon** for 10 seconds and stop the trace.
- Store output in /tmp/fmon.out.

Example 6-25 Using FILEMON

```
#filemon -o /tmp/fmon.out ; sleep 10 ; trcstop
```

To produce some sample output for **filemon**, we ran a sequential write test in the background, and started a **filemon** trace, as shown in Example 6-26. We used the **lmktemp** command to create a 2 GB file full of nulls while filemon gathered I/O stats.

Example 6-26 filemon with a sequential write test

```
cd /SPREADSF
time lmktemp 2GBtest 2000000000 &
filemon -o /tmp/fmon.out ; sleep 10 ; trcstop
```

The output stored in /tmp/fmon.out from the example write test is quite large, so we use **awk** to see the sections we are interested in. In Example 6-27, we look the busiest logical volumes and find, as expected, that /dev/spreadlv (/SPREADFS) is the busiest logical volume. Notice that the **awk** command is used to look at only the logical volume section of the **filemon** output.

Notice that **filemon** shows the read and write activity to each logical volume over the performance interval as well as the transfer throughput. The average throughput for /SPREADFS was 44354.4 KB/sec or 44 MB/sec.

Example 6-27 FILEMON most active logical volumes report

```
# awk '/Most Active Logical Volumes/,/^$/' /tmp/fmon.out

Most Active Logical Volumes
------------------------------------------------------------------------
```

```
  util  #rblk  #wblk   KB/s  volume                  description
------------------------------------------------------------------------
  1.00      0 1039520 44354.4  /dev/spreadlv            /SPREADFS
  0.08      0     40    1.7  /dev/hd8                 jfslog
  0.05      0     16    0.7  /dev/testjfslog          jfslog
```

The fields in the Most Active Logical Volumes report of the **filemon** command are as follows:

util Utilization of the volume (fraction of time busy). The rows are sorted by this field, in decreasing order. The first number, 1.00, means 100 percent.

#rblk Number of 512-byte blocks read from the volume.

#wblk Number of 512-byte blocks written to the volume.

KB/sec Total transfer throughput in Kilobytes per second.

volume Name of volume.

description Contents of volume; either a file system name, or logical volume type (jfs2, paging, jfslog, jfs2log, boot, or sysdump). Also indicates if the file system is fragmented or compressed.

To view the physical disks involved in the write test, we used **awk** to list the Most Active Physical Volumes in the **filemon** output, as shown in Example 6-28.

Note: Notice that **filemon**, like **iostat**, also reports stats based on hdisk devices—which are paths to ESS luns.

Example 6-28 FILEMON most active physical volumes report

```
# awk '/Most Active Physical Volumes/,/^$/' /tmp/fmon.out

Most Active Physical Volumes
------------------------------------------------------------------------
  util  #rblk  #wblk   KB/s  volume                  description
------------------------------------------------------------------------
  0.73      0  69376 2960.1  /dev/hdisk6             IBM FC 2105F20
  0.72      0  70656 3014.8  /dev/hdisk14            IBM FC 2105F20
  0.66      0  72960 3113.1  /dev/hdisk30            IBM FC 2105F20
  0.64      0  68864 2938.3  /dev/hdisk22            IBM FC 2105F20
  0.48      0  83872 3578.7  /dev/hdisk36            IBM FC 2105F20
```

The fields in the Most Active Physical Volume report of the **filemon** command are interpreted as follows:

util Utilization of the volume (fraction of time busy). The rows are sorted by this field in decreasing order.

#rblk Number of 512-byte blocks read from the volume.

#wblk Number of 512-byte blocks written to the volume.

KB/sec Total volume throughput in Kilobytes per second.

volume Name of volume.

description Type of volume.

The **filemon** command is a very useful tool to determine where a host is spending I/O. More details on the **filemon** options and reports are available in the publication *AIX 5L Performance Tools Handbook*, SG24-6039, which can be downloaded from:

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/SG246039.html

6.5.4 LVMSTAT

A new performance monitoring tool was introduced in AIX 5L called **lvmstat**, which reports input and output statistics for logical partitions, logical volumes, and volume groups. The **lvmstat** command is useful in determining whether a physical volume is becoming a hindrance to performance by identifying the busiest physical partitions for a logical volume.

The **lvmstat** command generates reports that can be used to change logical volume configuration to better balance the input and output load between physical disks.

lvmstat resides in /usr/sbin and is part of the bos.rte.lvm file set, which is installed by default from the AIX 5L base installation media.

The syntax of the **lvmstat** command is as follows:

```
lvmstat {-l |-v }Name [-e |-d ][-F ][-C ][-c Count ][-s ][ Interval [ Iterations ]]
```

Flags:

-c Count Prints only the specified number of lines of statistics.

-C Causes the counters that keep track of the iocnt, Kb_read, and Kb_wrtn to be cleared for the specified logical volume or volume group.

-d Specifies that statistics collection should be disabled for the logical volume or volume group specified.

-e Specifies that statistics collection should be enabled for the logical volume or volume group specified.

-F Causes the statistics to be printed in colon-separated format.

-l Specifies the name of the stanza to list.

-s Suppresses the header from the subsequent reports when Interval is used.

-v Specifies that the Name specified is the name of the volume group.

Parameters:

Name Specifies the logical volume or volume group name to monitor.

Interval The interval parameter specifies the amount of time, in seconds, between each report. If Interval is used to run **lvmstat** more than once, no reports are printed if the statistics did not change since the last run. A single period is printed instead.

Enabling lvmstat sampling for volume groups

The **lvmstat** command generates reports that can be used to change logical volume configuration to better balance the input and output load between physical disks. By default, the statistics collection is not enabled. By using the -e flag you enable Logical Volume Device Driver (LVMDD) to collect the physical partition statistics for each specified logical volume or the logical volumes in the specified volume group. Enabling the statistics collection for a

volume group enables it for all the logical volumes in that volume group. On every I/O call done to the physical partition that belongs to an enabled logical volume, the I/O count for that partition is increased by LVMDD. All the data collection is done by the LVMDD and the **lvmstat** command reports on those statistics.

The first report section generated by **lvmstat** provides statistics concerning the time since the statistical collection was enabled. Each subsequent report section covers the time since the previous report. All statistics are reported each time **lvmstat** runs. The report consists of a header row, followed by a line of statistics for each logical partition or logical volume depending on the flags specified.

If the statistics collection has not been enabled for the volume group or logical volume you wish to monitor, **lvmstat** will report an error like:

```
#lvmstat -v rootvg
0516-1309 lvmstat:Statistics collection is not enabled for this logical device.
Use -e option to enable.
```

To enable statistics collection for all logical volumes in a volume group (in this case the rootvg volume group), use the -e option together with the -v <volume group> flag as the following example shows:

```
#lvmstat -v rootvg -e
```

When you do not need to continue collecting statistics with **lvmstat**, it should be disabled because it impacts the performance of the system. To disable statistics collection for all logical volumes in a volume group (in this case the rootvg volume group), use the -d option together with the -v <volume group> flag as the following example shows:

```
#lvmstat -v rootvg -d
```

This will disable the collection of statistics on all logical volume in the volume group.

Monitoring volume group I/O using lmstat

Once a volume group is enabled for **lvmstat** monitoring, like rootvg in this example, you would only need to run **lvmstat -v rootvg** to monitor all activity to rootvg. An example of the **lvmstat** output is shown in Example 6-29.

Example 6-29 LVMSTAT example

```
#lvmstat -v rootvg
Logical Volume iocnt Kb_read Kb_wrtn Kbps
lv05           682478 16      8579672 16.08
loglv00        0      0       0       0.00
datalv         0      0       0       0.00
lv07           0      0       0       0.00
lv06           0      0       0       0.00
lv04           0      0       0       0.00
lv03           0      0       0       0.00
```

Notice that **lv05** is busy performing writes.

The **lvmstat** tool has a lot of powerful options such as reporting on a specific logical volume, or only reporting busy logical volumes in a volume group. For more information on using the **lvmstat** command and other tuning commands in detail, check the publication *AIX 5L Performance Tools Handbook*, SG24-6039, which can be downloaded from:

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/SG246039.html

6.6 HP-UX specific I/O monitoring commands

HP has graphical tools to measure system performance. Some of these tools are:

- HP Perfview/Measureware
- GlancePlus

HP Perfview/Measurware is good for recording performance measurements and maintaining a baseline of system performance data to refer to. The HP Perfview/Measurware tool can show stats for each physical disk in graphical format and you can change the time scale easily to your liking.

6.7 Viewing iostats based on vpaths - vpath_iostat script

By using the LSSDD command, you can make a wrapper program to convert **iostat** or **sar** information to present information on disk I/O by vpaths instead of disk devices. Information on the **lssdd** utility is provided in “LSSDD output example” on page 157.

A vpath_iostat script for AIX is included in Appendix A, “UNIX shell scripts” on page 435, and can be modified for HP-UX or Sun Solaris.

The vpath_iostat script for AIX depends on the **lssdd** command and **iostat**. The script builds a map file to list hdisk devices and their associated vpaths and then converts **iostat** information from hdisks to vpaths.

To run the script:

1. Make sure the ESS utility **lssdd** is working properly, that is, all volume groups are using vpaths and not hdisk devices and reported correctly.
2. Enter `vpath_iostat` (Ctrl + C to break out) or `vpath_iostat <interval> <iteration>`.

   An example of the output vpath_iostat produces is shown in Example 6-30.

Example 6-30 vpath_iostat output

```
garmo-aix: Total VPATHS used:        8    16:16 Wed 26 Feb 2003   5 sec interval
garmo-aix Vpath:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix vpath0          12.698          63.0            201.5           0.0             63.5
garmo-aix vpath6          12.672          60.6            209.1           0.0             63.4
garmo-aix vpath14         11.238          59.8            187.9           0.0             56.2
garmo-aix vpath8          11.314          44.6            253.7           0.0             56.6
garmo-aix vpath2          6.963           44.2            157.5           0.0             34.8
garmo-aix vpath12         7.731           30.2            256.0           0.0             38.7
garmo-aix vpath4          3.840           29.4            130.6           0.0             19.2
garmo-aix vpath10         2.842           13.2            215.3           0.0             14.2
--------------------------------------------------------------------------------------------
garmo-aix   TOTAL READ:   0.00    MB      TOTAL WRITTEN:  346.49  MB
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   70.00   MB/sec
```

In this example, host garmo-aix has eight active vpaths. There is a sequential write occurring at 70 MB/sec. The load is fairly well distributed among several vpaths. We do not know, however, if these vpaths are on different arrays. To see output based on rank/arrays, use the ess_iostat script discussed in 6.8, “Viewing iostats based on ranks - ess_iostat script” on page 196.

6.8 Viewing iostats based on ranks - ess_iostat script

Measuring ESS I/O stats is a challenge for these reasons:

- **iostat** measures performance on disk devices that could be paths to the same LUN.
- The vpath_iostat script and NMON (for AIX) show vpath information, but vpaths could be on the same ranks in the ESS.
- HP-UX and Sun Solaris do not have tools (that we are aware of) that measure iostats based on vpaths or ranks.
- ESS Expert has lots of good information and can delve into details for the SSA adapters, ranks, volumes, cache, and clusters. But:

  ESS Expert is not made to be used for real-time measurements; the smallest time slice between stats is 5 minutes.

  The ESS Expert can show the busiest ranks, but does not easily tell which hosts are driving that I/O.

One solution is to use the ESS utility (ESSUTIL) commands like **lsess** and **lssdd** to transform iostats into stats based on ranks. A shell script called ess_iostat is included in Appendix A, "UNIX shell scripts" on page 435. The script is written for AIX, but can be modified for HP-UX or Sun Solaris. An example of the **ess_iostat** output is shown in Example 6-31.

The **ess_iostat** command syntax is:

```
ess_iostat [default interval is 5 seconds, with 1000 iterations ]
ess_iostat <interval in seconds> <number of iterations>
```

Example 6-31 ESS_IOSTAT output

```
# ess_iostat 5 1

garmo-aix: Total RANKS used:       12      20:01 Sun 16 Feb 2003   5 sec interval
garmo-aix  Ranks:       MBps             tps            KB/trans       MB_read        MB_wrtn
garmo-aix  1403         9.552           71.2            134.2            47.8            0.0
garmo-aix  1603         6.779           53.8            126.0            34.0            0.0
garmo-aix  1703         5.743           43.0            133.6            28.8            0.0
garmo-aix  1503         5.809           42.8            135.7            29.1            0.0
garmo-aix  1301         3.665           32.4            113.1            18.4            0.0
garmo-aix  1601         3.206           27.2            117.9            16.1            0.0
garmo-aix  1201         2.734           22.8            119.9            13.7            0.0
garmo-aix  1101         2.479           22.0            112.7            12.4            0.0
garmo-aix  1401         2.299           20.4            112.7            11.5            0.0
garmo-aix  1501         2.180           19.8            110.1            10.9            0.0
garmo-aix  1001         2.246           19.4            115.8            11.3            0.0
garmo-aix  1701         2.088           18.8            111.1            10.5            0.0
--------------------------------------------------------------------------------------------
garmo-aix    TOTAL READ:   430.88  MB         TOTAL WRITTEN:  0.06    MB
garmo-aix    READ SPEED:    86.18 MB/sec      WRITE SPEED:    0.01   MB/sec
```

Notice the output is similar to **iostat** and the output from vpath_iostat, but presents stats based on ranks. The output is sorted by transactions per second (TPS) and also includes the total amount of data written and read over the interval period. In this case the interval period was 5 seconds. The average read speed was 86 MB/sec and 12 ranks were involved. I/O is spread fairly evenly over all 12 ranks and there is a very low number of transactions per second per rank.

The way ess_iostat works is to:

1. Run **iostat <interval> <iteration>** to a file.
2. Use the **lsess** command to change every disk name to the associated rank.
3. Sum up all the iostats for each rank.

The ess_iostat script relies on the **lsess** command to build a map file listing disk devices and rank IDs.

The **lsess** output for AIX, Sun Solaris, and HP-UX is shown in Example 6-32.

Example 6-32 LSESS output for AIX, SUN Solaris, and HP-UX

```
AIX
Disk        Location       LUN SN       Type        Size    LSS    Vol    Rank   C/A   S    Connection
------      ------------   --------  ------------   ----    ---    ---    ----   ---   -   -----------
hdisk1      10-68-01[FC]   0O8FC106  IBM 2105-E20 16.0      10      8    1001    01   Y   R1-B2-H4-ZA

SUN
 Disk        Location                LUN SN   Type   Size  LSS Vol Rank C/A S Connection
------     ------------              -------- ------- ----- --- --- ---- --- - -----------
c1t0d0   /sbus@a,0/fcaw@2,0/sd@0,0   30412342 2105F20  12.0  19 004 1301 2/2 N R1-B2-H4-ZA

HP/UX
Disk        Location                LUN SN   Type   Size  LSS Vol Rank C/A S Connection
------     ------------              -------- ------- ----- --- --- ---- --- - -----------
c12t2d1  0/4/0/0.2.16.0.32.2.1 [FC]  011FC106 2105E20 008.0  16 017 1001 1/1 Y R1-B2-H4-ZA
```

Notice the positioning of the disk device and rank information in the **lsess** output:

- AIX: The hdisk number is in column 1 and the rank ID is in column 9.
- Sun Solaris: The disk name is in column 1 and the associated rank is in column 8.
- HP-UX: The disk name is in column 1 and the rank is in column 9.

The ess_iostat script for AIX runs **lsess | awk '{print $1 "\t" $9}' > $essfile** to extract just the hdisk (column 1) and rank (column 9) information to make a disk device-to-rank map with information like that shown in Example 6-33.

Example 6-33 Disk-to-rank mapping

```
Disk    Rank
------  ---
hdisk6  1100
hdisk7  1101
hdisk8  1000
hdisk9  1001
```

After **ess_iostat** builds the disk to rank map, it runs **iostat**. Then the script uses sed to change every disk name to the associated rank number, sums up the stats by ranks, and displays the output.

For HP-UX and Sun Solaris, find an **iostat** or **vmstat -d** command you would like to see based on ranks. Then modify the ess_iostat script to:

- Build the disk device-to-rank map properly from **lsess**.
- Run your preferred **iostat** or **vmstat -d** command.
- Replace all disk names with rank numbers.
- Sum up and display the results.

Rank stats for each host

After you have ess_iostat working, you could use it to collect information over a period of time to report stats based on ranks over any time period desired. You could start an ess_iostat collection from cron to start at midnight and collect ess_iostats every 15 minutes all day with:

```
ess_iostat 900 96
```

(15 minutes * 60seconds/min. = 900 seconds)

(24 hours 4 iterations/hour = 96 iterations)

To see rank performance for a given host, sum up the rank stats over the time period desired and divide by the number of seconds in the time period.

Entire ESS performance report from host perspective

The next step is to have all ESS-attached servers running ess_iostats periodically. By then combining the ess_iostats from all servers, you can sum up the I/O stats for each rank by server and see:

- Which ranks in the ESS are the busiest.
- Which servers are driving the I/O to each rank.
- The enterprise read/write ratio.
- The enterprise transaction rate.
- The average transaction size for each server.
- The read/write ratio for each server.

An example of using ess_iostat from multiple servers to create an enterprise view of ESS performance is shown in Figure 6-3. Notice that in this example, taken from 7 a.m. to 5 p.m., that:

- HostG is the busiest server, contributing to over half of the ESS workload.
- The read/write ratio for the whole enterprise is about 2:1 (499 GB read/244 GB written).
- Average I/Os/sec to the ESS were 4280 transactions per second.

SHARK IO ACTIVITY TUESDAY 7am-5pm

Host	MBps	tps	KB/trans	GB_read	GB_wrtn
hostA	1.255	52.3	27	8.448	36.743
hostB	6.748	547.6	9.2	206.693	36.219
hostC	1.638	126.6	9	28.393	30.568
hostD	0.044	5.4	5	0.668	0.9
hostE	1.992	332.9	4.2	17.759	53.942
hostF	0.857	89.6	6.5	18.19	12.665
hostG	5.335	2605.3	2.7	155.536	36.509
hostH	0.811	336.4	5.7	20.297	8.906
hostI	0.153	23.9	5	1.241	4.285
hostJ	0.177	19.7	3.7	3.44	2.94
hostK	1.578	132	7.8	37.878	18.92
hostL	0.099	9.1	9.9	1.157	2.4
TOTAL	20.687	4280.8	8	499	244

Figure 6-3 ESS enterprise view

The next part of the report displays the ranks in order of highest transaction rates, as shown in Figure 6-4.

SHARK	Ranks:	MBps	tps	KB/trans	GB_read	GB_wrtn
ESS1	**r1600**	**1.9**	**782.3**	**5.7**	**30**	**33**
ESS1	r1002	2.3	666.8	5.4	50	4
ESS1	r1500	1.9	615.0	9.8	47	4
ESS1	r1200	1.5	553.4	5.4	40	9
ESS1	r1202	1.5	213.5	5.4	13	11
ESS1	r1102	1.3	203.3	3.2	16	1
ESS1	r1502	1.0	129.2	7.5	6	25
ESS1	r1601	1.3	85.2	11.9	27	11
ESS1	r1301	1.3	85.0	13.1	36	4
ESS1	r1501	1.3	78.9	14.1	35	4
ESS1	r1401	1.2	75.4	9.7	25	4
ESS1	r1400	0.4	74.1	4.3	3	7
ESS1	r1700	0.6	66.8	7.6	6	10
ESS1	r1101	1.0	61.6	15.0	30	2
ESS1	r1001	0.8	52.1	12.1	17	7
ESS1	r1701	1.0	49.9	17.6	27	4
ESS1	r1702	1.5	46.7	9.1	7	31
ESS1	r1000	0.6	43.3	6.4	8	6
ESS1	r1201	0.5	35.4	10.2	11	1
ESS1	r1300	0.5	34.6	5.1	8	3
ESS1	r1100	0.3	29.4	8.6	4	4

Figure 6-4 ESS busiest ranks

From the results shown in Figure 6-4 you can tell that:

- The busiest rank is rank1600.
- About 5 ranks are handling over 60 percent of the workload.

The detailed view of the busiest rank, rank 1600, shows the hosts driving the I/O in Figure 6-5. hostG is driving rank number 1600 very hard, over 700 transactions per second. It would help performance (especially for hosts F, E, C, and D) to divide some of the workload from hostG to other ranks in the ESS.

RANK	r1600:					

Host	Rank	MBps	tps	KB/trans	MB_read	MB_wrtn
hostG	r1600	1.599	748.6	2	24252	30706
hostF	r1600	0.167	20.5	7.2	5380	166
hostE	r1600	0.086	12.4	6.4	595	2362
hostC	r1600	0.034	0.7	11.2	22	295
hostD	r1600	0.004	0.1	1.7	99	5
TOTAL:	r1600	1.89	782.3	5.7	30348	33534

Figure 6-5 Busiest rank example

Note: Generally, you should try to tune I/O if an ESS array is performing over 600 transactions per second.

The ess_iostat script for AIX is especially good at monitoring sequential I/O rates. Use the information to correlate with ESS Expert reports to help distribute activity among all ESS ranks by moving I/O to less busy ranks.

6.9 Measuring ESS sequential I/O speeds

For random I/O workload, the ESS uses its cache to handle almost 100 percent of random writes at cache speeds—unless the workload is very cache unfriendly.

For sequential I/O, however, it will be beneficial to implement striping at the OS level across multiple ranks when driving over 100 MB/sec read speeds for large files. The number of ranks to stripe across depends on the rank throughput you can achieve in your environment and your requirements for sequential I/O speeds. With multiple hosts sharing the same rank, the bandwidth of the array will be divided between multiple hosts.

In the next sections we will show you how to:

- Determine the sequential read speed an individual vpath (LUN) can provide in your environment.
- How to measure sequential read and write speeds for file systems.

6.9.1 Using DD command to test rank read speeds

To test the sequential read speed of a rank, you can run the command:

```
time dd if=/dev/vpathX of=/dev/null bs=128k count=781
```

That command will read 100 MB off of vpathX and report how long it takes in seconds. Take 100 and divide by the number of seconds to determine MB/sec read speed.

A test_disk_speeds script that can be used to measure read speeds for ranks on HP-UX, Sun Solaris, and AIX is also included in “TEST_DISK_SPEEDS” on page 446.

If you determine that the average read speed for your vpaths is 50 MB/sec, then you know you need to stripe across at least 4 different ranks to achieve 200 MB/sec sequential read speeds.

An example of the output from test_disk_speeds is shown in Example 6-34.

Example 6-34 test_disk_speeds output

```
# test_disk_speeds
vpath0 43.0 MB/sec  100 MB      bs=128k
```

Remember that the bandwidth of a rank is shared among servers using logical disks on that same rank. In the example above, the server was able to read at 43 MB/sec from vpath0. The array vpath0 resides on was also being shared by six other servers running transactions as well. You may see higher or lower read speeds for vpaths depending on the time of day, when transaction rates from multiple hosts to the same array varies.

6.9.2 Testing file system sequential write/read speeds

A simple way to test sequential write/read speeds for file systems is to time how long it takes to create a large sequential file and then how long it takes to copy the same file to /dev/null. After creating the file for the write test, you need to take care that the file is not still cached in host memory because that will invalidate the read test since data will come from RAM instead of disk.

An example is shown below for AIX:

- Sequential write speed:

```
cd /STRIPEDFS
time lmktemp 1GBtest 1000000000
real    0m7.68s
user    0m0.13s
sys     0m7.54s
```

  Divide 1000/7.68 seconds = 130 MB/sec write speed.

- Sequential read speed:

```
cd /
umount /STRIPEDFS [ flush file from memory ]
mount /STRIPEDFS
cd - [cd back to the previous directory /STRIPEDFS ]
time cp 1GBtestfile /dev/null
real    0m9.04s
user    0m0.32s
sys     0m4.39s
```

  Divide 1000/9.04 seconds = 110 MB/sec read speed.

For HP-UX, use the **prealloc** command instead of **lmktemp** for AIX to create large files. For Sun Solaris, use the **mkfile** command.

Note: Note that the **prealloc** command for HP-UX and the **lmktemp** command for AIX have a 2 GB size limitation. Those commands are not able to create a file greater than 2 GB in size. If you want a file larger than 2 GB for a sequential read test, concatenate a couple 2 GB files together.

6.10 Implementing striped file systems

In this section, we will discuss the specific steps on how to implement striped file system for UNIX hosts attached to the ESS. A detailed overview of striping is discussed in section 3.10, “Open systems striping” on page 72.

Attention: The ESS presents LUNs that are hardware striped across RAID arrays, so you do not want to stripe across LUNs from the same array at the OS level. If you do so, you will defeat the performance gains of the ESS RAID array by serializing I/O down single DDMs within an array.

Logical disks presented from the ESS are often referred to as volumes, logical volumes, or LUNs. This can cause confusion when dealing with LVM commands from the OS which, use the same term, *logical volumes*, but for a different meaning.

In this section we use *logical volumes* when referring to the LVM commands to map out storage on top of ESS *logical disks*.

6.10.1 Creating striped file systems

To stripe fileystems on HP-UX, Sun Solaris, or AIX servers attached to the ESS, follow these steps:

1. Create a volume group from logical disks on different ESS ranks on different SSA adapters:
   a. Use **lssdd** to determine which ranks a vpath belongs to, or which ranks an existing volume group is using. Select vpaths on different ranks.
   b. Create a map of your ESS similar to the example shown in 5.9.4, “Mapping ranks to ESS Specialist disk groups” on page 159.
   c. Test the sequential read speeds for ranks, as shown in 6.9.1, “Using DD command to test rank read speeds” on page 200, to determine how many ranks you need. Otherwise, use logical disks from 6+ ranks for 200+ MB/sec reads.
   d. Use logical disks from 10+ arrays for 400+ MB/sec reads
2. Use logical disks that are all the same size (8 or 16 GB LUNs work well).
3. Use logical disks that are a multiple of 64, 128, or 256 K.
4. Create a *striped logical volume* using OS LVM commands.

   Do not stripe less than 32 KB; preferably 64 K or higher
   – AIX: Use the **mklv** command with **-S** to stripe.
   – HP-UX: Use **lvcreate** with -S.
   – Sun Solaris: Use **format**.
5. For AIX, create a striped jfslog also, across 2 or 4 ranks so the jfslog does not become a bottleneck.
6. Create a file system on top of the logical volume if necessary.
   – AIX: Use the **crfs** command.
   – HP-UX: Use **mkfs**, **newfs**.
   – Sun Solaris: Use **mkfs**, **newfs**.
7. Tune the OS in order to:
   – Turn on sequential read-ahead.
   – Add memory buffers for file I/O.

6.10.2 Example of striping on an AIX host

In the following steps, we show in detail how to create a striped file system for an AIX server using ESS logical disks. In the example, we use eight logical disks from eight different ranks to create a volume group and striped logical volume.

Creating the volume group

The first step is to create a volume group. We created a volume group called testvg with the command:

```
mkvg4vp -s 64 -y testvg vpath0 vpath2 vpath4 vpath6 vpath8 vpath10 vpath12 vpath14
```

After the volume group was created, we used a script called vgmap to verify the vpaths and ranks used by testvg. The vgmap script is available in Appendix A, “UNIX shell scripts” on page 435. An example of the output from vgmap is shown in Example 6-35.

Example 6-35 VGMAP output

```
garmo-aix:/ESS> vgmap testvg
```

```
PV_NAME         RANK    PV STATE  TOTAL PPs   FREE PPs
testvg:
vpath0          1100    active      502         502
vpath2          1000    active      502         502
vpath4          1300    active      502         502
vpath6          1200    active      502         502
vpath8          1500    active      502         502
vpath10         1400    active      502         502
vpath12         1700    active      502         502
vpath14         1600    active      502         502
```

Notice that all 8 vpaths for testvg come from 8 different ESS ranks.

Creating a striped logical volume

To make a striped jfslog and logical volume called stripedlv we used the **mklv** commands:

- Create a striped JFSlog:

```
mklv -t jfslog -y testjfslog -S 128K testvg 4 vpath4 vpath8 vpath12 vpath0
logform /dev/testjfslog (format the JFSlog)
```

- Create a striped logical volume: /dev/stripedlv:

```
mklv -t jfs -y stripedlv -S 128K  testvg 32 vpath0 vpath2 vpath4 vpath6 vpath8
vpath10 vpath12 vpath14
```

Notice that:

- The striping is done when the logical volume is made, not the file system.
- The stripe size used in the example 128 K.
- We have selected 32 physical partitions = 32* 64 MB = 2048 MB = 2 GB logical volume.
- We are striping across 8 vpaths, so each vpath will get 32/8 = 4 physical partitions.

After the logical volume group is created, we used a script called lvmap, which is available in Appendix A, “UNIX shell scripts” on page 435, to verify the vpaths and ranks used by the logical volume /dev/stripedlv. An example of the output from lvmap is shown in Example 6-36.

Example 6-36 LVMAP output

```
garmo-aix: /ESS> lvmap stripedlv

LV_NAME   RANK          COPIES        IN BAND        DISTRIBUTION
stripedlv:
vpath2    1000          004:000:000   100%          000:004:000:000:000
vpath4    1300          004:000:000   100%          000:004:000:000:000
vpath10   1400          004:000:000   100%          000:004:000:000:000
vpath12   1700          004:000:000   100%          000:004:000:000:000
vpath8    1500          004:000:000   100%          000:004:000:000:000
vpath0    1100          004:000:000   100%          000:004:000:000:000
vpath6    1200          004:000:000   100%          000:004:000:000:000
vpath14   1600          004:000:000   100%          000:004:000:000:000
```

Notice that:

- Each vpath has 4 logical partitions from /dev/stripedlv.
- The logical volume is striped across 8 different ranks.

Creating the striped file system

After the logical volume is created, the next step is to create a file system. Use the **crfs** command (similar to the following) to create file systems, or use smit to Add a JFS file system on an existing logical volume. The smit fastpath is **smitty jfs**.

```
crfs -v jfs -a bf=true -d stripedlv -m /STRIPEDFS -A yes -p rw -t no
```

Next, mount the file system:

```
mount /STRIPEDFS
df -tk shows:
/dev/stripedlv     2097152     65876   2031276    4% /STRIPEDFS
```

lslv output for stripedlv

We have a 2 GB file system now on a striped logical volume /dev/stripedlv. The output from **lslv /stripedlv** is shown in Example 6-37.

Example 6-37 LSLV for striped logical volume /dev/stripedlv

```
garmo-aix:/ESS> lslv stripedlv
LOGICAL VOLUME:     stripedlv              VOLUME GROUP:   testvg
LV IDENTIFIER:      0021d6ea00004c00000000f39b6284e4.3 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        64 megabyte(s)
COPIES:             1                      SCHED POLICY:   striped
LPs:                32                     PPs:            32
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       maximum                RELOCATABLE:    no
INTRA-POLICY:       middle                 UPPER BOUND:    8
MOUNT POINT:        /STRIPEDFS             LABEL:          /STRIPEDFS
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes (superstrict)
STRIPE WIDTH:       8
STRIPE SIZE:        128K
```

Notice that the output in Example 6-37 shows:

- The logical volume stripedlv has a stripe size of 128 K
- The stripe width is 8 since we used 8 vpaths to create the logical volume
- The file system associated with /dev/stripedlv is called /STRIPEDFS

Creating a spread logical volume

To be thorough, we include the steps to create a /SPREADFS file system to compare the performance difference for sequential I/O between spreading and striping data. We also create a /SINGLEFS file system, which resides on a single vpath (rank).

Using the same volume group, we created a spread logical volume called /dev/spreadlv and a file system called /SPREADFS with:

```
mklv -t jfs -y spreadlv -e x  testvg 32 vpath0 vpath2 vpath4 vpath6 vpath8 vpath10
vpath12 vpath14
crfs -v jfs -a bf=true -d spreadlv -m /SPREADFS -A yes -p rw -t no
```

Notice that:

- The -e x option will *spread* the logical volume from one vpath to the next in chunks the size of the physical partition size.

- The physical partition size of 64 MB was defined when the volume group testvg was created.

The **lslv** output for spreadlv is shown in Example 6-38.

Example 6-38 LSLV spreadlv

```
garmo-aix:/ESS> lslv spreadlv
LOGICAL VOLUME:     spreadlv               VOLUME GROUP:   testvg
LV IDENTIFIER:      0021d6ea00004c00000000f39b6284e4.2 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        64 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                32                     PPs:            32
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       maximum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
MOUNT POINT:        /SPREADFS              LABEL:          /SPREADFS
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```

Notice that:

- The INTER-POLICY is maximum; this was set by the -e x option when creating the logical volume.
- The mount point is /SPREADFS.

Creating a single-vpath logical volume

To create a file system /SINGLEFS on top of a logical volume that resides on a single vpath, we used the commands:

```
mklv -t jfs -y singlevpathlv testvg 32 vpath0
crfs -v jfs -a bf=true -d singlevpathlv -m /SINGLEFS -A yes -p rw -t no
```

The **lslv** output for singlevpathlv is shown in Example 6-39 on page 205.

Example 6-39 LSLV output for SINGLEVPATHLV

```
garmo-aix:/SINGLEFS> lslv -l singlevpathlv
singlevpathlv:/SINGLEFS
PV                COPIES        IN BAND       DISTRIBUTION
vpath0            032:000:000   100%          000:032:000:000:000

garmo-aix:/SINGLEFS> lslv  singlevpathlv
LOGICAL VOLUME:     singlevpathlv          VOLUME GROUP:   testvg
LV IDENTIFIER:      0021d6ea00004c00000000f39b6284e4.4 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        64 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                32                     PPs:            32
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
MOUNT POINT:        /SINGLEFS              LABEL:          /SINGLEFS
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```

Notice that the **lslv** output for singlevpathlv shows:

- The logical volume singlevpathlv resides only on vpath0.
- The physical partition size is 64 MB, as defined when the volume group was created.
- There are 32 logical partitions (LPs) in the logical volume.
- The file system associated with /dev/singlevpathlv is called /SINGLEFS.

After creating the jfslog testjfslog and the file systems /STRIPEDFS, /SPREADFS, and /SINGLEFS, the output from **lsvg -l testvg** looks like that shown in Example 6-40.

Example 6-40 LSVG -l testvg output

```
garmo-aix:/ESS> lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
testjfslog          jfslog     4     4     4    open/syncd    N/A
stripedlv           jfs        32    32    8    open/syncd    /STRIPEDFS
spreadlv            jfs        32    32    8    open/syncd    /SPREADFS
singlevpathlv       jfs        32    32    1    open/syncd    /SINGLEFS=
```

Notice that the **lsvg -l testvg** output shows:

- singlevpathlv, spreadlv, and stripedlv all use the same number of logical partitions: 32.
- stripedlv and spreadlv use 8 physical volumes.
- singlevpathlv only uses 1 physical volume (vpath0).

The size of the three logical volumes is the same, but the logical layout is different.

- Logical volume singlevpathlv is spread sequentially on one vpath in 64 MB partitions.
- Logical volume spreadlv is spread across 8 vpaths in 64 MB chunks.
- Logical volume stripedlv is striped across 8 vpaths in 128 K chunks.

The resulting file systems /SINGLEFS, /SPREADFS, and /STRIPEDFS, are all the same size, as shown in the **df -tk** output of Example 6-41.

Example 6-41 DF comparing single-vpath, striped, and spread file systems

```
# df -tk
/dev/singlevpathlv     2097152     65876   2031276    4% /SINGLEFS
/dev/stripedlv     2097152     65876   2031276    4% /STRIPEDFS
/dev/spreadlv     2097152     65876   2031276    4% /SPREADFS
```

Sequential I/O tests for /SINGLEFS, /SPREADFS, /STRIPEDFS

Now we can test the sequential write and read performance of the striped, spread, and single-vpath file systems.

We ran tests for each of the 3 file systems in the following scenarios:

- No OS tuning:
  a. Sequential *write*
  b. Sequential *read*
- After OS tuning:
  a. Sequential *write*
  b. Sequential *read*

Sequential write tests before tuning OS

We ran sequential write tests comparing write speeds from the /SPREADFS, /STRIPEDFS, and /SINGLEFS file systems *before* making any OS tuning changes. In one window we kept **ess_iostat** running and in the other window ran the write tests:

```
cd /SPREADFS
time lmktemp 1GBtest &

cd /STRIPEDFS
time lmktemp 1GBtest &

cd /SINGLEFS
time lmktemp 1GBtest &
```

The results of the sequential write tests are shown in Example 6-42. Notice in the output that:

- /SINGLEFS showed write speeds of 36 MB/sec to a single rank, but the real time to complete the write was only 8.3 seconds = 1000/8.3 = 120 MB/sec due to the ability of the ESS to complete many writes at cache speeds.
- The cache on the ESS made writes appear to the host to run at 120 MB/sec even though the destage from cache to ESS disks took longer. In all 3 tests, the 1 GB write test completed in about 8 seconds.
- Writes to /SPREADFS only used one, plus a little bit of a second, rank at a time. That is because the 64 MB logical partition size is too large to make all 8 disks, /dev/spreadlv is located on active at the same time.
- /STRIPEDFS has a destage rate of 69 MB/sec from ESS cache to the ESS disks. Almost twice as fast as /SINGLEFS.

Example 6-42 Sequential WRITE tests before OS tuning

```
/SINGLEFS
garmo-aix: Total RANKS used:        1     12:15 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            36.608          175.0           209.2           0.0             36.6
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   36.61   MB/sec
real    0m8.30s


/SPREADFS
garmo-aix: Total RANKS used:        2     12:29 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1700            45.952          311.0           147.8           0.0             46.0
garmo-aix 1400            17.152          41.0            418.3           0.0             17.2
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   63.19   MB/sec
real    0m8.50s


/STRIPEDFS
garmo-aix: Total RANKS used:        8     12:31 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            9.088           70.0            129.8           0.0             9.1
garmo-aix 1200            9.088           70.0            129.8           0.0             9.1
garmo-aix 1400            8.960           70.0            128.0           0.0             9.0
garmo-aix 1500            8.960           70.0            128.0           0.0             9.0
garmo-aix 1600            9.088           70.0            129.8           0.0             9.1
garmo-aix 1700            8.960           70.0            128.0           0.0             9.0
garmo-aix 1000            7.804           62.0            125.9           0.0             7.8
```

```
garmo-aix 1300            7.680           62.0            123.9           0.0             7.7
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   69.63   MB/sec
real    0m7.71s
```

Sequential read tests before tuning OS

Next, we ran tests to compare the read speeds between /SPREADFS, /STRIPEDFS, and /SINGLEFS file systems before making any OS tuning changes. In one window we kept **ess_iostat** running and in the other window ran the tests:

```
cd /SPREADFS
time cp 1GBtest /dev/null

cd /STRIPEDFS
time cp 1GBtest /dev/null

cd /SINGLEFS
time cp 1GBtest /dev/null
```

Note: We unmounted and re-mounted the file system between the write and read tests to make sure no file data was cached in system memory.

The read test results are shown in Example 6-43. Notice that:

- All three file systems had the same performance, 48 MB/sec.
- /SINGLEFS activity sent 1500 transactions/second to a single rank 1100 (vpath0).
- /SPREADFS used one (and a tiny bit of a second) rank at a time. /SPREADFS sent 1400 transactions per second to rank 1100.
- /STRIPEDFS used 8 ranks with the load evenly spread across all 8 ranks.
- The transaction size was 32 KB/transactions for all three file systems, which is what we would expect with RAID-5 arrays on the ESS.

Example 6-43 Sequential read tests without read ahead turned on

```
/SINGLEFS
garmo-aix: Total RANKS used:        1     11:57 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            48.000          1500.0          32.0            48.0            0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   48.00   MB/sec   WRITE SPEED:   0.00    MB/sec


/SPREADFS
garmo-aix: Total RANKS used:        2     13:05 Wed 26 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            45.376          1423.0          31.9            45.4            0.0
garmo-aix 1500            2.432           74.0            32.9            2.4             0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   47.81   MB/sec   WRITE SPEED:   0.00    MB/sec


/STRIPEDFS
garmo-aix: Total RANKS used:        8     13:07 Wed 26 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1600            6.144           194.0           31.7            6.1             0.0
garmo-aix 1500            6.144           193.0           31.8            6.1             0.0
garmo-aix 1700            6.144           193.0           31.8            6.1             0.0
```

```
garmo-aix 1100            6.144          192.0          32.0           6.1            0.0
garmo-aix 1200            6.144          192.0          32.0           6.1            0.0
garmo-aix 1400            6.144          190.0          32.3           6.1            0.0
garmo-aix 1000            5.312          168.0          31.6           5.3            0.0
garmo-aix 1300            5.248          164.0          32.0           5.2            0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   47.42   MB/sec   WRITE SPEED:   0.00    MB/sec

real    0m20.00sec for all 3 tests
1000/20 = 50MB/sec sequential READ speeds
```

Notice that without tuning the AIX OS, sequential read performance is the same for all three file systems; about 50 MB/sec.

Turning on read ahead and adding pbufs (physical buffers)

Next, we tuned the AIX operating system with the **vmtune** command (the **vmtune** command is discussed later in 6.11.1, “AIX OS tuning for sequential I/O” on page 212) to allow the system to detect sequential I/O and start reading more data into memory as soon as it detects sequential I/O patterns. We also increased the number of **pbufs** (physical memory buffers). A detailed explanation of these options is available in 6.11.1, “AIX OS tuning for sequential I/O” on page 212.

The **vmtune** values before we tuned the OS are shown in Example 6-44. Notice that:

- maxpgahead = 8
- maxfree =128
- numfsbufs = 93 (number of file system buffers)

Example 6-44 VMTUNE values before tuning

```
vmtune:  current values:
  -p       -P        -r          -R         -f       -F       -N        -W
minperm  maxperm  minpgahead maxpgahead  minfree  maxfree  pd_npages maxrandwrt
 153010   306020       2          8        120      128      65536       50

  -M      -w      -k      -c        -b         -B          -u        -l    -d
maxpin npswarn npskill numclust numfsbufs hd_pbuf_cnt lvm_bufcnt lrubucket defps
2516569    2304     576       1      93        704          9      131072     1
```

To increase read-ahead and allow for more memory buffers dedicated to file system I/O, we ran the **vmtune** command:

```
vmtune -R 256 -F 384 -b 400
```

This increased maxpgeahead to 256, maxfree to 384, and numfsbufs to 400. The system under test was running AIX 5L with 12 GB of memory.

Sequential write tests after tuning OS

The results of the write tests after tuning the OS to detect sequential I/O are shown in Example 6-45.

Example 6-45 Sequential write speeds after vmtune changes

```
/SINGLEFS
garmo-aix: Total RANKS used:        1     12:35 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            34.560          151.0           228.9           0.0             34.6
-------------------------------------------------------------------------------------------
```

```
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   34.56   MB/sec
real    0m7.26s

/SPREADFS
garmo-aix: Total RANKS used:        4     12:36 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1500            34.048          156.0           218.3           0.0             34.0
garmo-aix 1100            8.192           99.0            82.7            0.0             8.2
garmo-aix 1700            29.696          92.0            322.8           0.0             29.7
garmo-aix 1400            0.256           0.0             0.0             0.0             0.3
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   72.19   MB/sec
real    0m7.72s

/STRIPEDFS
garmo-aix: Total RANKS used:        8     12:46 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            10.244          63.0            162.6           0.0             10.2
garmo-aix 1200            10.624          62.0            171.4           0.0             10.6
garmo-aix 1300            7.808           59.0            132.3           0.0             7.8
garmo-aix 1700            8.576           57.0            150.5           0.0             8.6
garmo-aix 1500            8.316           53.0            156.9           0.0             8.3
garmo-aix 1000            8.192           52.0            157.5           0.0             8.2
garmo-aix 1400            8.064           51.0            158.1           0.0             8.1
garmo-aix 1600            8.704           50.0            174.1           0.0             8.7
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   70.53   MB/sec
real    0m7.71s
```

Notice that:

- Write speeds after making the vmtune change did not increase significantly from the previous write tests done before without the vmtune changes.
- The ESS still reported the writes as complete to the host in about 8 seconds which is equivalent to 125 MB/sec sequential write speeds (1000 MB/8sec = 125 MB/sec).

Sequential read tests after tuning OS

After making the vmtune changes, the sequential read performance for all three file systems, /SINGLEFS, /SPREADFS, and /STRIPEDFS, increased substantially, as shown in Example 6-46.

Example 6-46 Sequential read speeds after vmtune changes

```
/SINGLEFS
garmo-aix: Total RANKS used:        1     12:39 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1100            95.232          388.0           245.4           95.2            0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   95.23   MB/sec   WRITE SPEED:   0.00    MB/sec
real    0m10.31s

/SPREADFS
garmo-aix: Total RANKS used:        3     12:40 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1700            63.488          248.0           256.0           63.5            0.0
garmo-aix 1400            23.296          86.0            270.9           23.3            0.0
garmo-aix 1500            8.720           38.0            229.5           8.7             0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   95.50   MB/sec   WRITE SPEED:   0.00    MB/sec
```

```
real    0m10.31s

/STRIPEDFS
garmo-aix: Total RANKS used:        8     12:50 Fri 28 Feb 2003   1 sec interval
garmo-aix Ranks:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix 1400            16.128          128.0           126.0           16.1            0.0
garmo-aix 1500            16.256          126.0           129.0           16.3            0.0
garmo-aix 1200            16.256          125.0           130.0           16.3            0.0
garmo-aix 1100            16.256          124.0           131.1           16.3            0.0
garmo-aix 1600            16.128          123.0           131.1           16.1            0.0
garmo-aix 1700            16.132          123.0           131.2           16.1            0.0
garmo-aix 1000            14.336          111.0           129.2           14.3            0.0
garmo-aix 1300            14.208          111.0           128.0           14.2            0.0
-------------------------------------------------------------------------------------------
garmo-aix   READ SPEED:   125.70  MB/sec   WRITE SPEED:   0.06    MB/sec
real    0m8.27s
```

/SINGLEFS and /SPREADFS saw read speeds twice as fast as before making vmtune changes. /STRIPEDFS read speeds increased over 2.5 times, increasing from 47 MB/sec to 125 MB/sec, which is about the limit of two 1 Gb Fibre Channel adapters.

Notice the stats in Example 6-46 show that:

- /SINGLEFS
  - Is pushing a single rank at 95 MB/sec.
  - The transaction size increased from 32 KB/transaction to about 256 KB/transaction compared to the read tests without vmtune changes.
- /SPREADFS
  - Is using 2+ ranks now that sequential read ahead is turned on; however, one rank is performing most of the I/O load at a time.
  - The transaction size increased from 32 KB/transaction to about 256 KB/transaction compared to the read tests without vmtune changes.
- /STRIPEDFS
  - Is using all 8 ranks with the load evenly balanced across ranks.
  - The transaction size is about 128 KB/transaction.
  - The load on each rank is only about 16 MB/sec.
  - We are maxing out the limit of the I/O drawer and Fibre Channel adapters on the host. The ESS can deliver higher read speeds if we add more Fibre Channel adapters to our host, or go to 2 Gb adapters.

Sequential I/O testing summary

Sequential I/O performance can be improved greatly by turning on *read-ahead* and allocating more memory buffers for I/O by using the **vmtune** command. By striping file systems across multiple ranks in the ESS, you can make sequential reads so fast that you will max out the PCI bus or Fiber Channel adapters on a host system.

The host we used to test had two 1Gb Fibre Channel cards, which maxed out at 126 MB/sec sequential read speeds. The ESS in our example can deliver over 500 MB/sec (1.8 TB/hour) sequential read speeds if you have a need for such a high I/O rate (and have a host or tape library that can sustain that rate).

A summary of the test results is shown in Table 6-3.

Table 6-3 Sequential I/O test summary for /SINGLEFS, /SPREADFS, and /STRIPEDFS

Sequential I/O test	/SINGLEFS	/SPREADFS (64 MB partitions)	/STRIPEDFS (8 ranks)
Write - no vmtune	36 MB/sec ***	63 MB/sec ***	69 MB/sec***
Read - no vmtune	48 MB/sec	48 MB/sec	48 MB/sec
Write - vmtune	36 MB/sec ***	72 MB/s ***	70 MB/sec ***
Read - vmtune	95 MB/sec	96 MB/sec	126 MB/sec
(***) The write speeds show destaging speeds. The ESS reported write at about 120 MB/sec due to caching.			

Notice that the read speed of a simple file system (not striped or spread) using only a single rank can be quite fast with **vmtune** changes. However, keep in mind that these tests were done on an ESS that was not under significant load from multiple hosts. If a rank contains logical disks assigned to multiple hosts, then it is unlikely that a single host will see the 95 MB/sec sequential read speed from that rank.

To calculate GB/hour from MB/sec, just multiply by 3.6. For example:

```
1 MB/sec * 3600sec/hour * 1 GB/1000 MB = 3.6 GB/hour
126 MB/sec * 3.6 = 453 GB/hour
```

Spreading or striping file systems increases read and write speeds. Even though the cache on the ESS allows many writes to complete at cache speeds, in a cache unfriendly environment, the faster destages can complete from ESS cache to disks the better. In that case you will want to take advantage of the higher write speeds spread and striped file systems provide. You will also want to investigate other **vmtune** tuning options like **write behind**.

Important: By understanding your I/O requirements, you can decide what type of file systems to create. Striping is not always the best solution. There are some trade-offs, as discussed in 3.10.5, “Striping logical volumes - Trade-offs” on page 77.

Striping can help sequential and random I/O if implemented properly. If a host performs a random workload during the day (OLTP, Oracle, etc.) and a sequential workload at night (batch, offline backups), you may want to make **vmtune** changes, depending on the workload to maximize memory usage. You can increase or decrease the maxpgahead setting with the **vmtune** command, and AIX can dynamically make the change take effect.

6.11 Operating system tuning for sequential I/O

The following sections explain how to configure AIX, HP-UX, and Sun Solaris for high sequential I/O performance. When using striping to read at high speeds, the default installation settings of these operating systems do not have adequate memory buffers allocated to handle the increased I/O load that striping across ESS ranks delivers.

6.11.1 AIX OS tuning for sequential I/O

To modify the I/O tuning parameters of AIX, use the **vmtune** command. For sequential I/O you will definitely need to increase the maxpgahead setting. You will also likely need to increase the number of buffers available for I/O transactions by increasing the numfsbufs setting with **vmtune**.

The **vmtune** options and logical constructs are explained below.

vmtune

The **vmtune** command is found in /usr/samples/kernel and is included with the bos.adt.samples file set.

numfsbufs

The numfsbufs (-b) vmtune value specifies the number of file system buffer structures. This value must be greater than 0 (zero). If there are insufficient free buffer structures, the VMM will put the process on a the wait list before starting I/O. To determine if the value of numfsbufs is too low, use the command **vmtune -a**.

Monitor the fsbufwaitcount value displayed. This value is increased each time an I/O operation has to wait for a file system buffer structure. If the fsbufwaitcnt value increases under normal work load conditions, then the value of numfsbufs should be increased using the vmtune -b option.

Note: When the **numfsbufs** value is changed, it is necessary to unmount and mount the file system again for the changes to take affect.

PBUF - Physical disk buffers

The Logical Volume Manager (LVM) uses a construct called a *pbuf* to control a pending disk I/O. A single pbuf is used for each I/O request, regardless of the number of pages involved. AIX creates extra pbufs when a new physical volume is added to the system. When striping is used, you need more pbufs because one I/O operation causes I/O operations to more disks and, therefore, more pbufs. When striping and mirroring is used, even more pbufs are required.

Running out of pbufs reduces performance considerably because the I/O process is suspended until pbufs are available again. Increasing the number of pbufs is done with the **vmtune** command, however, pbufs are pinned so that allocating many pbufs will increase the use of memory.

A special note should be given to adjusting the number of physical buffers on systems with many disks attached or available with the **vmtune** command. The number of physical buffers (pbufs) per active disk should be twice the queue depth of the disk or 32, whatever is greater. The default maximum number of pbufs should not exceed a total of 65536.

lvm_bufcnt

The lvm_bufcnt (-u) value specifies the number of LVM buffers for raw I/O. This value can range from 1 to 64 and has a default of 9. Extremely large volumes of I/O are required to cause a bottleneck at the LVM layer. The number of *uphysio* buffers can be increased to overcome this bottleneck. Each uphysio buffer is 128 KB. If I/O operations are larger than 128 KB * 9, then a value larger than the default value of nine should be used. The pd_npages (-N) value determines number of pages that should be deleted in one chunk from real memory when a file is deleted (that is, the pages are deleted in a single VMM critical section with interrupts disabled to INTPAGER). By default, all pages of a file can be removed from memory in one critical section if the file was deleted from disk. To ensure fast response time for real-time applications, this value can be reduced so that only a smaller chunk of pages is deleted before returning from the critical section. When the numfsbufs value is changed, it is necessary to unmount and mount the file system again for the changes to take affect.

Setting read ahead with maxpageahed

The following example provides suggestions about **vmtune** and logical volume striping. Sequential and random accesses benefit form disk striping. The following technique for configuring striped disks is recommended.

1. Spread the logical volume across as many physical volumes as possible.
2. Use as many adapters as possible for the physical volumes.
3. Create a separate volume group for striped logical volumes.
4. Do not mix striped and non-striped logical volumes in the same physical volume.
5. All physical volumes should be the same size within a set of striped logical volumes.
6. Set the stripe unit size to 64 KB or higher.
7. Set the value of minpgahead to 2.
8. Set the value of maxpgahead to 16 times the number of disks.

An example **vmtune** command that increases maxgahead, maxfree, and the numfsbufs settings is:

```
vmtune -R 48 -F 168 -b 100
```

You will want to run **vmstat -a**, as mentioned, to check if the number of the numfsbufs setting is sufficient. Make changes in small increments. Having more pbufs available than you need will waste system memory.

Note: The **vmtune** settings are dependent on platform type, AIX level, and amount of memory, so a **vmtune** setting on one AIX system will not work on all others. You can decrease the maxpgahead value without rebooting, but some other **vmtune** options cannot be decreased without rebooting. Be careful with the **vmtune** command.

Ensure that the difference between maxfree and minfree is equal to or exceeds the value of maxpgahead.

AIX file system caching - minperm and maxperm

The AIX operating system will leave pages that have been read or written to in memory. If these file pages are requested again, then this saves an I/O operation. The minperm and maxperm values control the level of this file system caching. The thresholds set by maxperm and minperm can be considered as the following:

- If the percentage of file pages in memory exceeds maxperm, only file pages are taken by the page replacement algorithm.
- If the percentage of file pages in memory is less than minperm, both file pages and computational pages are taken by the page replacement algorithm.
- If the percentage of file pages in memory is in the range between minperm and maxperm, the page replacement algorithm steals only the file pages, unless the number of file repages is higher than the number of computational repages.

Computational pages can be defined as working storage segments and program text segments. File pages are defined as all other page types, usually persistent and client pages. In some instances, the application may cache pages itself. Therefore there is no need for the file system to cache pages as well. In this case, the values of minperm and maxperm can be set low.

To reduce minperm and maxperm to 5 and 15 percent, respectively, you would run:

```
vmtune -p 5 -P 15
```

vmtune - Using recommendations

Do not attempt to use an incorrect version of the **vmtune** command on an operating system. Invoking the incorrect version of the **vmtune** command can result in the operating system failing. The functionality of the **vmtune** command also varies between versions of the operating system.

Setting the value for minfree too high can result in excessive paging because premature stealing of pages occurs to satisfy the required size of the memory free list. Always ensure that the difference between the maxfree value and the minfree value is equal to or greater than the maxpgahead value.

On SMP systems the value of the maxfree and minfree as displayed by vmtune are the sum of the maxfree and minfree values for all of the memory pools. It is recommended that the **vmstat** command be used to determine the correct value for minfree.

When changing the value of the **maxpin** value, ensure that there is always at least 4 MB of memory available for the kernel.

Remember that **vmtune** changes do not survive a reboot. To make **vmtune** changes permanent, add an entry to the inittab file like:

```
vmtune:2:once:/usr/samples/kernel/vmtune -R 48 -F 168 -b 100 -p 5 -P 10 # VMTUNE
```

More information on the **vmtune** command and other AIX tuning parameters can be found in the publication *AIX 5L Performance Tools Handbook*, SG24-6039, which can be downloaded from:

http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246039.pdf

6.11.2 HP-UX OS tuning for sequential I/O

For sequential I/O, HP-UX needs to turn on *read ahead* kernel options. A list of tunable kernel parameters for HP-UX Release 11i can be found at:

http://docs.hp.com/hpux/onlinedocs/TKP-90202/TKP-90202.htmlbufpages

There are two different file system types for HP-UX, VxFS, and HFS. VxFS is preferred for performance reasons.

VxFS read-ahead options

The kernel options for enabling read ahead for VxFS file systems are the following.

vxfs_max_ra_kbytes Maximum amount of read-ahead data, in KB, that the kernel may have outstanding for a single VxFS file system

vxfs_ra_per_disk Maximum amount of VxFS file system read-ahead per disk, in KB

vx_fancyra_enabled Enable or disable VXFS file system read-ahead

HFS read-ahead options

The kernel options for enabling read-ahead for HFS file systems are the following:

hfs_max_ra_blocks The maximum number of read-ahead blocks that the kernel may have outstanding for a single HFS file system.

hfs_max_revra_blocks The maximum number of reverse read-ahead blocks that the kernel may have outstanding for a single HFS file system.

hfs_ra_per_disk The amount of HFS file system read-ahead per disk drive, in KB.

hp_hfs_mtra_enabled Enable or disable HFS multi-threaded read-ahead. No manpage.

Tip: Tuning the read ahead options varies from system to system depending on the platform and amount of memory installed. Experiment with different values, making small changes at a time.

Dynamic buffer cache

Another important options for HP-UX and I/O performance is dbc_min_pct and dbc_max_pct. These two kernel parameters, dbc_min_pct and dbc_max_pct, control the lower and upper limit, respectively, as a percentage of system memory that can be allocated for buffer cache.

How many pages of memory are allocated for buffer cache use at any given time is determined by system needs, but the two parameters ensure that allocated memory never drops below dbc_min_pct and cannot exceed dbc_max_pct percent of total system memory.

The default value for dbc_max_pct is 50 percent, which is usually overkill. If you want to use a dynamic buffer cache, set the dbc_max_pct value to 25 percent. If you have 4 GB of memory or more, start with an even smaller value.

With a large buffer cache the system is likely to have to pageout or shrink the buffer cache to meet application memory needs, which causes I/Os to paging space. You want to avoid that from happening and set memory buffers to favor applications over cached files.

6.11.3 Sun Solaris OS tuning for sequential I/O

Solaris 2.6, Solaris 7, and Solaris 8 require patches to ensure that the host and ESS function correctly. See the following Web site for the most current list of Solaris-SPARC patches and the Solaris-x86 patch for Solaris 2.6, Solaris 7, and Solaris 8.

```
http://java.sun.com/j2se/1.3/install-solaris-patches.html
```

Check the following guide for an updated list of the updates Sun Solaris needs for different attachment types to the ESS: *IBM TotalStorage Enterprise Storage Server Host System Attachment Guide,* SC26-7446. This guide can be downloaded from:

```
http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105
```

Sun Solaris settings for high sequential I/O

Sun Solaris has some kernel settings that should be tuned for sequential I/O and also has settings that should be set depending on which type of Fibre Channel adapter the host is using.

maxphys

This parameter specifies the maximum number of bytes that you can transfer for each SCSI transaction. The default value is 126976 (124 KB). If the I/O block size that you requested exceeds the default value, the request is broken into more than one request. The value should be tuned for the application requirements. For maximum bandwidth, set the maxphys parameter by adding the following line to the /etc/system file:

```
set maxphys=1048576 (1 MB)
```

Attention: Do not set the value for maxphys greater than 1048576 (1 MB). Doing so can cause the system to hang.

vxio:vol_maxio

If you are use the Veritas volume manager on the ESS LUNs, you must set the VxVM maximum I/O size parameter (vol_maxio) to match the maxphys parameter. When you set the maxphys parameter to 1048576 and you use the Veritas Volume Manager on your ESS LUNs, set the maxphys parameter like in the following sentence:

```
set vxio:vol_maxio=2048
```

Sun Fibre Channel settings

The following sections contain the procedures to set the Sun host system parameters for optimum performance on the ESS with the Fibre Channel adapters:

1. Type `cd/etc` to change to the /etc subdirectory.
2. Back up the system file in the subdirectory.
3. Edit the system file and set the following parameters for servers with configurations that use JNI, Emulex, or Qlogic Fibre Channel adapters:
   - sd_max_throttle (JNI only)

     This sd_max_throttle parameter specifies the maximum number of commands that the sd driver can queue to the host adapter driver.

     Note: Use this setting for JNI Fibre Channel adapters only.

     The default value is 256, but you must set the parameter to a value less than or equal to a maximum queue depth for each LUN connected. Determine the value by using the following formula:

     ```
     256 ÷ (LUNs per adapter)
     ```

     Where LUNs per adapter is the largest number of LUNs assigned to a single adapter.

     To set the sd_max_throttle parameter for the ESS LUNs in this example, you would add the following line to the /etc/system file:

     ```
     set sd:sd_max_throttle=5
     ```

   The following settings should be set for all Fibre Channel adapter types (JNI, Emulex, or Qlogic).

   - sd_io_time

     This parameter specifies the time-out value for disk operations. Add the following line to the /etc/system file to set the sd_io_time parameter for the ESS LUNs:

     ```
     set sd:sd_io_time=0x78
     ```

   - sd_retry_count

     This parameter specifies the retry count for disk operations. Add the following line to the /etc/system file to set the sd_retry_count parameter for the ESS LUNs:

     ```
     set sd:sd_retry_count=5
     ```

SUN Solaris resources

More information on SUN Solaris commands and tuning options is available from the following web sites:

- http://sunsolve.sun.com/handbook_pub/
- http://www.sun.com/bigadmin/collections/performance.html
- http://www.context-switch.com/reference/exscripts/perform/diskstat

7

Open system servers - Linux for xSeries™

This chapter discusses the monitoring and tuning tools and techniques that can be used with Linux systems in order to optimize throughput and performance when attaching the ESS.

In this chapter we also discuss the supported distributions of Linux when using the ESS Model 800, as well as the tools that can be helpful for the monitoring and tuning activity:

- uptime
- dmesg
- top
- iostat
- vmstat
- sar, isag
- GKrellM
- KDE System Guard
- LVM
- Bonnie

7.1 Supported Linux distributions

For Intel-based servers attaching the ESS, currently there are two supported Linux distributions:

- Red Hat Linux 7.2
- SuSE Linux 7.3

The following SuSE Linux and Redhat levels are no longer recommended with the ESS:

- Red Hat Linux 7.1 running kernel level 2.4.9.
- Red Hat Linux 7.2 running kernel level 2.4.9.31
- Red Hat Linux Advanced Server 2.1 running kernel level 2.4.9-e.8
- SuSE Linux 7.2 running kernel level 2.4.9
- SuSE Linux 7.3 running kernel level k_smp-2.4.16-22
- SuSE Linux Enterprise Server (SLES) 7 running kernel level 2.4.18-134

If problems are encountered with installed versions, you may be required to update your Linux configuration to a higher supported level before problem determination can take place.

For further clarification and the most current information on ESS-supported Linux distributions and kernel support compatible with the ESS, you can refer to the Web site:

`http://www.storage.ibm.com/disk/ess/supserver.htm`

Once there click the link for the PDF file, **ESS interoperability matrix**.

7.2 Introduction to Linux O/S components

It is important to understand the makeup of Linux and how the different components relate and play together in the overall performance of the system.

7.2.1 Understanding and tuning virtual memory

To get the most performance out of a Linux server, it is important to understand how Linux manages memory resources. It uses an *always full* concept of memory managing, which means that the system fills up the whole memory with data (such as applications, kernel, cache). When the server boots, the first thing it does is divide the memory info different pieces. The memory is divided into three main components:

1. Kernel space

   Kernel space is where the actual kernel code is loaded, and where memory is allocated for kernel-level operations. Kernel operations include scheduling, process management, signaling, device I/O, paging, controlling of the underlying hardware and swapping, and the core operations that other programs rely on to be taken care of.

2. User space

   User space is where all the application code (for example, database, e-mail, and Web server code, user shell login or Xwindows) is loaded. In the user space, the memory is again divided into chunks. Every process has its own allocated memory space. No other process can access that data. This makes the operating system more stable since each process is using its own protected part of the memory.

3. Buffer space

The rest of the memory is used as buffer space for caching. Every time an I/O operation (for example, disk, network) is performed, the data is transferred first to memory for caching. All DMA and busmaster transfers are also performed through the buffer space.

It is the kernel's job to manage all of these different memory spaces. When, for example, an application is started, the kernel must transfer all the data from the hard disk to the buffer space. After that, it must free some memory in the user space to load the application. Since the user space will be divided into different chunks, it must sometimes rearrange certain processes to get a big enough chunk for the application it is trying to load. When it has finished and there is a contiguous section of memory available, it will load the application code to the user space. This is why most of the Linux machines appear to load applications more slowly than Windows boxes.

Version 2.4 of the Linux kernel introduced some major changes in memory management over earlier kernel versions:

- Page *aging* was reintroduced in 2.4, but a few small changes have been made to avoid some problems with previous implementations. For example, a counter is maintained for each physical page and used to determine whether or not to keep the page in memory. Each time the memory is scanned for pages that should be evicted, the counter is increased. Each time the page is requested, the counter is decreased. So over time, the pages that are less often used get a higher counter and those that are more often used get a lower counter. If there is a request to evict pages, the higher a page's counter is, the greater chance there is of that page being cleared.
- Page *flushing* has been optimized to avoid performance losses when writing to the ESS Model 800 disks. Writing out unneeded pages can dramatically decrease the performance of a server, because of the extra disk seeks performed. A better solution is to delay the writes and wait for another page flushing so that the disk seeks can be optimized and the seek time minimized.

7.2.2 Understanding and tuning the swap partition

During installation, Linux creates a swap partition. The size of the swap partition should be at least equal to the amount of RAM installed in the server, but we recommend that you make the partition double the size of the RAM.

If there is insufficient memory installed in a server, it will begin paging the least used data from memory to the swap partitions on the disks. A general rule is that the swap partitions should be on the fastest drives available. If the server has more than one array, it is always a good idea to spread the swap partitions over all of the arrays. This will generally improve the performance of the server.

Furthermore, there is a way to *parallelize* swap file read/writes. It is possible to give each swap partition a priority setting in the /etc/fstab file. If you open the /etc/fstab file, you might see something like in Example 7-1.

Example 7-1 /etc/fstab file

```
/dev/sda2 swap swap sw 0 0
/dev/sdb2 swap swap sw 0 0
/dev/sdc2 swap swap sw 0 0
/dev/sdd2 swap swap sw 0 0
```

Under normal circumstances, Linux would use the swap partition /dev/sda2 first, then /dev/sdb2, and so on, until it had allocated enough swapping space. This means that perhaps only the first partition, /dev/sda2, will be used if there is no need for a large swap space.

Spreading the data over all available swap partitions will improve performance, because all read/write requests will be performed simultaneously to all selected partitions. If you change the file, as in Example 7-2, you will assign a higher priority level to the first three partitions.

Example 7-2 /etc/fstab file, modified

```
/dev/sda2 swap swap sw,pri=3 0 0
/dev/sdb2 swap swap sw,pri=3 0 0
/dev/sdc2 swap swap sw,pri=3 0 0
/dev/sdd2 swap swap sw,pri=1 0 0
```

Swap partitions are used from the highest priority to the lowest (where 32767 is the highest and 0 the lowest). Giving the same priority to the first three disks causes the data to be written to all three disks; the system does not wait until the first swap partition is full before it starts to write on the next partition. The system uses the first three partitions in parallel and performance generally improves.

The fourth partition is used if the first three are completely filled up and there is still additional space needed for swapping. It is also possible to give all partitions the same priority to stripe the data over all partitions, but if one drive is slower than the others (/dev/sdd2 in Example 7-2), performance would decrease.

If the server is running out of swap space and there is additional hard disk space left, it is possible to create additional swap partitions with FDISK. However, if you cannot create a new partition, you can create a swap file instead. There are two disadvantages to locating a swap file outside a dedicated swap partition.

- The performance of swap files in a data partition is slower than on a swap partition.
- If the swap file gets damaged, the data on the whole partition may be lost.

For these reasons, we recommended that you not place the swap file on a data partition. In the following example, we will create a 512 MB swap file with a block size of 2 KB:

1. Start by creating a directory for the swap file:

   ```
   mkdir /swap
   ```

2. Create the swap file:

   ```
   dd if=/dev/zero of=/swap/swapfile bs=2048 count=262144
   ```

   This command creates a file called /swap/swapfile with a block size of 2 KB. The size will be 512 MB (2048*262144=512 MB). The size is determined using the bs and count parameters of dd, so the command could have also been:

   ```
   dd if=/dev/zero of=/swap/swapfile bs=1M count=512
   ```

3. Initialize the swap file:

   ```
   mkswap /swap/swapfile 262144
   ```

4. Synchronize the file:

   ```
   sync
   ```

5. Configure Linux to use the swap file:

   ```
   swapon /swap/swapfile
   ```

6. If the swap file is no longer needed, you can instruct the system to stop using the swap file and then delete the file:

   ```
   swapoff /swap/swapfile
   rm /swap/swapfile
   ```

It is also possible to use a swap file permanently. The information needs to be put into the /etc/fstab file, which would look as illustrated in Example 7-2 on page 222.

Example 7-3 /etc/fstab file

```
/swap/swapfile none swap sw 0 0
```

7.2.3 Understanding and tuning the daemons

A *daemon* is comparable to a *service* in Windows 2000. Daemons provide different services. For example:

- The httpd daemon is a Web server.
- The sendmail daemon is a mail server.

There are daemons running on every server that are probably not needed. Disabling these daemons frees memory and decreases the number of processes the CPU has to handle.

linuxconf, chkconfig, and serviceconf are tools that make it easy, among other things, to disable and enable daemons. If linuxconf is not found on your system it is available from:

`http://www.solucorp.qc.ca/linuxconf/`

Figure 7-1 shows how to disable daemons on Red Hat Linux, with linuxconf.

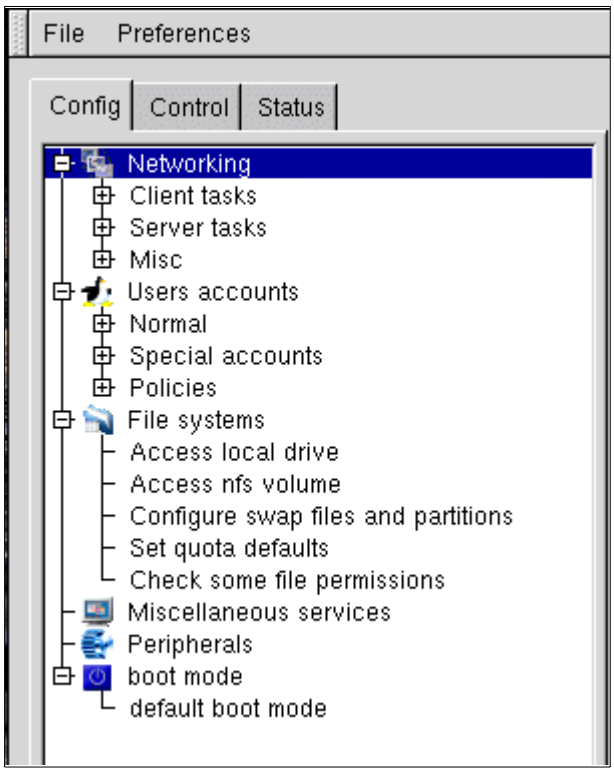


Figure 7-1 linuxconf screen

chkconfig is a text type tool run from the command line. Example 7-4 shows output with the chkconfig command line.

Example 7-4 chkconfig

```
host system:~# chkconfig --list
keytable        0:off   1:on    2:on    3:on    4:on    5:on    6:off
```

```
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
kdcrotate       0:off   1:off   2:off   3:off   4:off   5:off   6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
gpm             0:off   1:off   2:on    3:on    4:on    5:on    6:off
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
ipchains        0:off   1:off   2:on    3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
anacron         0:off   1:off   2:on    3:on    4:on    5:on    6:off
lpd             0:off   1:off   2:on    3:on    4:on    5:on    6:off
xfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:on    4:off   5:on    6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
nscd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
identd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
radvd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
rwhod           0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
rhnsd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypbind          0:off   1:off   2:off   3:on    4:on    5:on    6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
rstatd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
rusersd         0:off   1:off   2:off   3:off   4:off   5:off   6:off
rwalld          0:off   1:off   2:off   3:off   4:off   5:off   6:off
vncserver       0:off   1:off   2:off   3:off   4:off   5:off   6:off
yppasswdd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypserv          0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypxfrd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
smb             0:off   1:off   2:off   3:off   4:off   5:off   6:off
bcm5820         0:off   1:off   2:off   3:off   4:off   5:off   6:off
httpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
squid           0:off   1:off   2:off   3:off   4:off   5:off   6:off
tux             0:off   1:off   2:off   3:off   4:off   5:off   6:off
webmin          0:off   1:off   2:on    3:on    4:off   5:on    6:off
xinetd based services:
        chargen-udp:    off
        chargen:        off
        daytime-udp:    off
        daytime:        off
        echo-udp:       on
        echo:   on
        time-udp:       off
        time:   off
        sgi_fam:        on
        finger: off
        rexec:  off
        rlogin: off
        rsh:    off
        ntalk:  off
        talk:   off
```

```
        telnet: on
        wu-ftpd:        on
host system:~#
```

Change a value (and check if the setting changed). For example, to turn off the **sshd** daemon, type in the following command from the host system:

```
chkconfig --level 5 sshd off
```

Check to see if it has changed by typing in the following command:

```
chkconfig --list sshd
```

Example 7-5 is the output of what was just changed.

Example 7-5 chkconfig --list sshd

```
sshd            0:off   1:off   2:on    3:on    4:on    5:off   6:off
```

Now turn the **sshd** daemon back on by typing in the following command:

```
chkconfig --level 5 sshd on
```

And now check to see if it has changed by typing in the following command:

```
chkconfig --list sshd
```

Example 7-6 is the output of what was just changed.

Example 7-6 chkconfig --list sshd

```
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:of
```

You can get online help using (**man chkconfig**).

Also you can use **serviceconfig** to disable unnecessary daemons, as illustrated in Figure 7-2 on page 226.

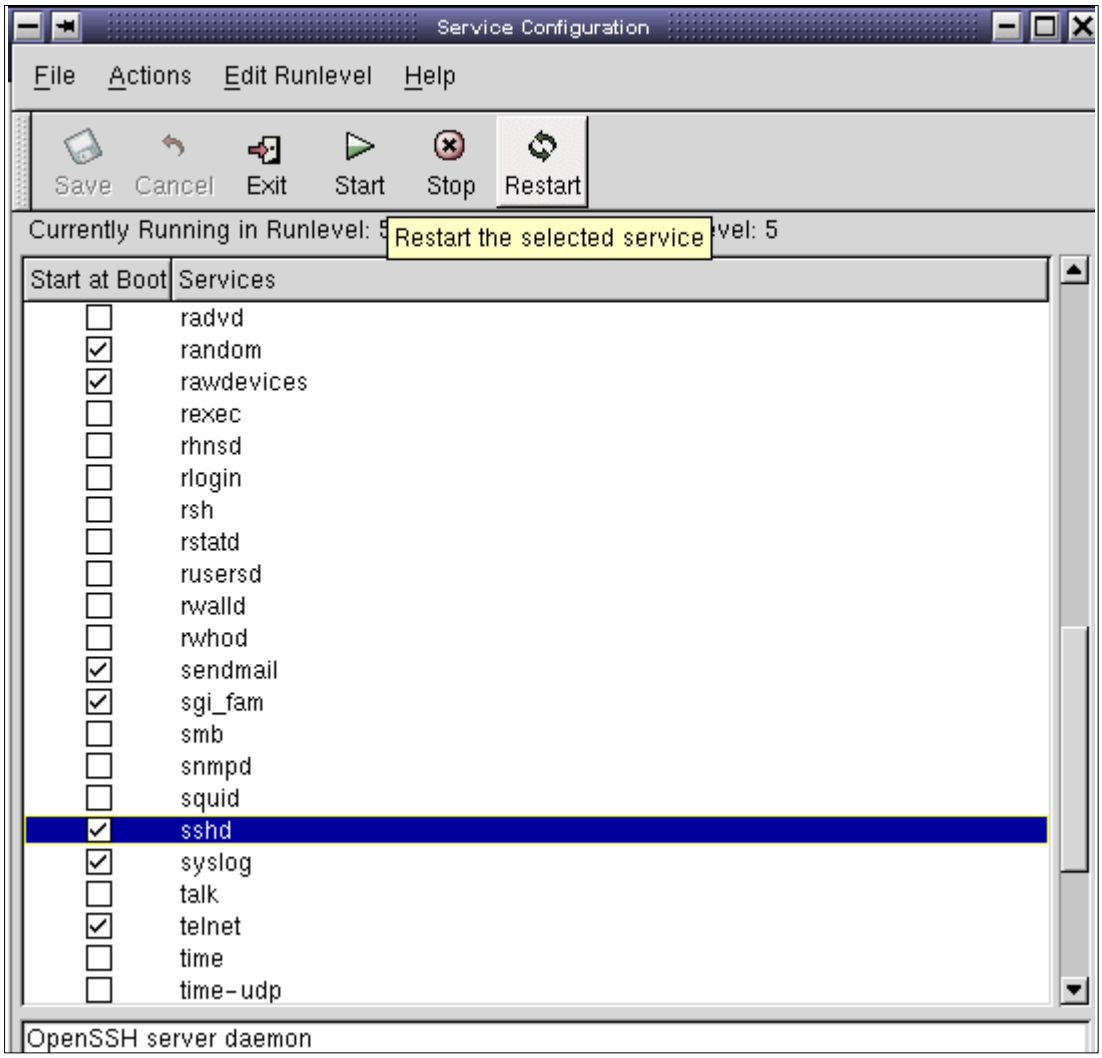


Figure 7-2 serviceconfig screen

If you do not have the ability to run linuxconf, serviceconfig, or chkconfig, or you do not want to use them, it is also possible to disable or enable daemons from the command line. In the following example, we will show how to stop the sendmail daemon. First log on as root and enter the following command:

```
/etc/init.d/sendmail stop
```

Every daemon can be started and stopped in the same way. Some also provide further functions such as restart, status, etc.

If you do not want the daemon to start the next time the machine boots, you will need to change the contents of the various run level directories.

1. Determine which run level the machine is running with the command **`runlevel`**.

   This will print the previous and current run level (for example, N 3 means that there was no previous run level (N) and that the current run level is 3).

2. To switch between run levels, use the **`init`** command. For example, to switch to run level 5, enter the command **`init 5`**.

Following, we provide a short description of the different run levels used in Linux:

- 0 - Halt (do not set initdefault to this or the server will immediately shut down after finishing the boot process).
- 1 - Single user mode.
- 2 - Multi-user, without NFS (the same as 3, if you do not have networking).
- 3 - Full multi-user mode non-graphical.
- 4 - Unused.
- 5 - Full multi-user mode graphical X11.
- 6 - Reboot (do not set initdefault to this or the server machine will continuously reboot at startup).

3. To prevent a daemon from starting, you will need to rename the appropriate file in the \etc directory structure. For example, to disable the sendmail daemon in run level 3 at startup, enter the command:

   ```
   rename /etc/rc3.d/S80sendmail /etc/rc3.d/K80sendmail
   ```

   or

   ```
   mv /etc/rc3.d/S80sendmail /etc/rc3.d/K80sendmail
   ```

   Daemons with an S at the beginning of the symbolic link name will be started; those starting with a K will not be started in that specific run level. In our example, the sendmail daemon will not be started on the next reboot. Please note that you must select the correct run level to change this.

7.2.4 Tuning the GUI

Whenever possible, do not run the GUI on a Linux server. Normally, there is no need for a GUI on a Linux server. All administration tasks can be achieved by the command line or through a Web interface. There are several different useful Web-based tools (for example, webmin, Linuxconf, SWAT).

If there is really a need for a GUI, then stop it whenever possible: Log out from the Xsession (if not booted in run level 5) or switch to run level 3 if you have booted to run level 5 by using the **`init 3`** command. If you want to restart the Xserver, use **`startx`** from a command prompt.

If you want to always boot into run level 3 you can change the run level to 3 in the /etc/inittab file.

7.2.5 Compiling the kernel

Compiling the kernel is not absolutely necessary to improve the performance of the server, but we do recommend that you configure your Linux server to have the latest ESS supported kernel and drivers. There are always new improvements being made available, which not only fix bugs, but also improve the performance of the Linux machine.

Before you begin, you will need to know what hardware is installed in the server. You can obtain a list by typing in the command **`lspci`**.The most important things to know are:

- CPU type
- Amount of memory installed
- SCSI adapter
- RAID controller
- Fibre Channel adapter
- Network adapter

- Video adapter

The more information you have about the hardware used, the more easily the Linux kernel can be configured.

This procedure can be tricky at some steps, so we refer you to a complete discussion of how to compile the kernel in the IBM Redpaper, *Running the Linux 2.4 Kernel on IBM eServer @*server *xSeries Servers*, REDP0121, available from:

```
http://www.redbooks.ibm.com
```

Select **Redpapers** from the left navigation bar and do a search using the redpaper form number REDP0121.

7.2.6 Understanding and tuning the file systems

There are many different file systems available for Linux today. We will cover ext2, the most commonly used file system, and ext3, a journaling file system.

ext2

ext2 is the most widely used file system in the Linux community. It provides the standard UNIX file semantics and advanced features. It is robust and offers excellent performance. The ext2 standard features include:

- Support for standard UNIX file types (regular files, directories, device special files, and symbolic links)
- Up to 4 TB of volume size
- Support for long file names (up to 255 characters)

The ext2 kernel code contains many performance optimizations, which improve I/O speed when accessing data on a disk. One of the optimizations is a read ahead algorithm. When a block is read, the kernel code automatically requests the follow-on blocks. In this way, it ensures that the next block is already in the buffer cache and available for further processing. Read aheads improve the performance most when you have sequential read requests on your server (video, audio streaming).

In addition, ext2 contains many allocation optimizations. Block groups are used to store related inodes and data together. The kernel always tries to allocate data blocks for a file in the same group as its inode. This results in fewer disk head seeks performed when the kernel reads an inode and its data blocks.

One problem with ext2 is that if an unexpected power failure or an unclean shutdown occurs, the file system may be in an inconsistent state. Therefore, an e2fsck is forced on the next reboot of the system, which may or may not recover the file system from its inconsistent state. Journaling file systems like ext3 greatly reduces the chance of getting an inconsistent file system.

Since you cannot change the stripe size on the disks of the ESS, to achieve optimal performance, your O/S software stripe size should be changed to be a multiple of your file system block size or slightly larger. The actual file system block size for /dev/sda1 can be found with the following command:

```
dumpe2fs -h /dev/sda1 |grep -F "Block size"
```

Which produces output shown in Example 7-7 on page 229.

Example 7-7 Determining file system block size from the dumpe2fs command

```
dumpe2fs 1.23,15-Aug-2001 for EXT2 FS 0.5b,95/08/09
Block size: 1024
```

The block size cannot be changed when the partition is already formatted, so you have to decide which block size you will use when formatting the partition. So, if you create a new ext2 partition on /dev/sda5 with a block size of 4096 bytes/block, the command will be:

```
mke2fs -b 4096 /dev/sda5
```

For more information about ext2, refer to:

http://e2fsprogs.sourceforge.net/ext2.html

ext3

ext3 is the updated version of the ext2 file system. It has many new features and enhancements compared to the previous ext2. Its main advantages are:

- Availability: ext3 always writes data in a consistent way to the disks. So in case of an unclean shutdown (unexpected power failure, system crash), the server does not need to check the consistency of the data on a ext3 volume.

  The time spent to recover the journal is about one second (depending on the hardware used). On an ext2 volume, the e2fsck performed after a unclean shutdown may take hours, depending on the size of the volume and number of files.

- Data integrity: You can choose the type and level of protection of your data. You can choose to keep the file system consistent, but allow for damage to data on the file system in case of unclean system shutdown. This can improve performance under some, but not all, circumstances.

  Alternatively, you can choose to ensure that the data is consistent with the state of the file system. This second choice is the safer choice and is the default.

- Speed: There are three different journaling modes available to optimize speed:
  - data=writeback: This limits the data integrity guarantees, allowing old data to show up in files after a crash. However, under some circumstances, this will increase the performance of your disks.
  - data=ordered (default mode): This guarantees that data is consistent with the file system.
  - data=journal: This requires a larger journal for a reasonable speed. It takes longer to recover in case of an unclean shutdown, but is sometimes faster for certain database operations.

  To change the mode, add one of the following lines to the mount options for that volume in /etc/fstab:

  ```
  data=writeback
  data=ordered
  data=journal
  ```

For more information on ext3, see:

http://www.symonds.net/~rajesh/howto/ext3/index.html

If maximum performance is needed, use ext2 since it has generally less overhead than any journaling file system. But keep in mind that your data may be inconsistent in the event of a power failure or an unclean shutdown.

7.2.7 Tuning TCP window size

You will most likely want to modify the TCP window size and use window scaling if your server is connected to a network with high latency such as the Internet. You can either modify the parameters on a running system by modifying values in /proc/sys/net/core/ and /proc/sys/net/ipv4/ or modify the parameters permanently by changing values in the Linux kernel sources and compiling your own kernel. We describe both methods in this section.

Testing performed with FTP transmissions has shown that with scalable window support enabled and the TCP window size set to an appropriate level (depending on the network), network throughput improves 100–500 percent on WAN links. There is less impact on performance in LAN environments, but you may still want to experiment with these parameters.

The default setting of 64 KB for most Linux configurations is fine for most LANs, but too low for Internet connections. Set this to a value between 256 KB for T1 lines or lower, and 2 to 4 MB for T3, OC-3, or even faster connections.

To determine the optimal buffer size for your environment, you can use the following formula:

```
buffer size =2 *bandwidth *delay
```

Where bandwidth is the bandwidth of the slowest connection between the server and the client.

Changing the source and recompiling the kernel for TCP

Go to include/linux/skbuff.h header file in your Linux source directory and edit the values for the maximum send and receive windows.

```
#define SK_WMEM_MAX 262140
#define SK_RMEM_MAX 262140
```

Changing values on a running system

Change the maximum parameters to an appropriate value, depending on your connection speed.

For a Linux kernel 2.4.x system, add the following lines to /etc/rc.d/rc.local:

```
echo "4096 65536 4194304">/proc/sys/net/ipv4/tcp_rmem
echo "4096 65536 4194304">/proc/sys/net/ipv4/tcp_wmem
```

The three values describe the minimum, default, and maximum window sizes used by TCP. The Linux kernel 2.4.x actually does a good job of adjusting the window size automatically, depending on network conditions. You simply need to specify appropriate minimum and maximum values.

7.3 Linux monitoring tools

In this section we discuss the tools available for the supported Linux distributions, that will aid you in the performance monitoring and tuning activities of your Linux system I/O when using the ESS.

7.3.1 uptime

The **uptime** command can be used to see how long the server has been running, how many *logged on* users there are, and gives a quick overview of what *average load* the server has.

The system *load average* is displayed for the last one, five, and fifteen minute intervals. The load average is not a percentage, but instead the number of processes in queue waiting to be processed. If processes that request CPU time are blocked (which means the CPU has no time for processing them), the load average will increase. On the other hand, if the each process gets immediate access for CPU time and there are CPU cycles lost, the load will decrease.

The optimal value of the load would be 1, which means each process gets immediate access to the CPU and there are no CPU cycles lost. The typical loads can vary from system to system: For a uniprocessor workstation, 1–2 might be acceptable, whereas you will probably see values of 8–10 on multiprocessor servers.

You can use **uptime** to narrow down a problem to your server or the network. If, for example, a network application is running poorly, run **uptime** and you will see if the system load is high or not (see Example 7-8). If not, the problem may more likely be related to your network than to your server.

Example 7-8 Sample output of uptime

```
7:22pm up 2:31, 3 users, load average:1.12, 1.04, 0.77
```

For more information on **uptime**, see the online help (uptime) or the man pages on **uptime**.

Note: You can also use w, who, or finger instead of **uptime**. They also provide information on who is currently logged onto the machine and what the user is doing.

7.3.2 dmesg

With **dmesg**, you can determine what hardware is installed in your server. During every boot, Linux checks your hardware and logs this information. You can view these logs using **dmesg**. You can see what CPU, ESS Model 800 disk subsystem, network adapters, and amount of memory is installed. Example 7-9 illustrates the output of the **dmesg** command.

Example 7-9 Sample output of dmesg

```
Linux version 2.4.7-10 (bhcompile@stripples.devel.redhat.com)(gcc version 2.96
20000731 (Red Hat Linux 7.1 2.96-98))#1 Thu Sep 6 17:27:27 EDT 2001
...
Initializing CPU#0
Detected 448.957 MHz processor.
...
Memory:252964k/262128k available (1269k kernel code,6844k reserved,90k data,
220k init,0k highmem)
...
CPU:Before vendor init,caps:0183fbff 00000000 00000000,vendor =0
CPU:L1 I cache:16K,L1 D cache:16K
CPU:L2 cache:512K
...
CPU:Intel Pentium II (Deschutes)stepping 02
...
SCSI subsystem driver Revision:1.00
(scsi0)<Adaptec AIC-7895 Ultra SCSI host adapter>found at PCI 0/6/1
(scsi0)Wide Channel B,SCSI ID=7,32/255 SCBs
(scsi0)Downloading sequencer code...383 instructions downloaded
(scsi1)<Adaptec AIC-7895 Ultra SCSI host adapter>found at PCI 0/6/0
(scsi1)Wide Channel A,SCSI ID=7,32/255 SCBs
(scsi1)Downloading sequencer code...383 instructions downloaded
scsi0 :Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI)5.2.4/5.2.0
```

```
<Adaptec AIC-7895 Ultra SCSI host adapter>
scsi1 :Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI)5.2.4/5.2.0
<Adaptec AIC-7895 Ultra SCSI host adapter>
scsi2 :IBM PCI ServeRAID 4.72.00 <ServeRAID 3H>
Vendor:IBM Model:SERVERAID Rev:1.0
Type:Direct-Access ANSI SCSI revision:01
Vendor:IBM Model:SERVERAID Rev:1.0
Type:Processor ANSI SCSI revision:01
Vendor:SDR Model:GEM200 Rev:2
Type:Processor ANSI SCSI revision:02
Attached scsi disk sda at scsi2,channel 0,id 0,lun 0
SCSI device sda:53313536 512-byte hdwr sectors (27297 MB)
...
pcnet32_probe_pci:found device 0x001022.0x002000
ioaddr=0x002180 resource_flags=0x000101
eth%d:PCnet/FAST+79C972 at 0x2180,00 04 ac b8 a0 6e
tx_start_pt(0x0c00):~220 bytes,BCR18(9a61):BurstWrEn BurstRdEn NoUFlow
SRAMSIZE=0x1700,SRAM_BND=0x0800,
pcnet32:pcnet32_private lp=ce324000 lp_dma_addr=0xe324000 assigned IRQ 10.
pcnet32.c:v1.25kf 26.9.1999 tsbogend@alpha.franken.de
```

For more information on **dmesg** see the online help (man dmesg).

7.3.3 top

The **top** command shows you actual processor activity. By default, it displays the most CPU-intensive tasks of the server and updates the list every five seconds. You can sort the processes by PID (numerically), age (newest first), and resident memory usage and time (time the process has occupied the CPU since startup). Example 7-10 shows a sample of the output of the **top** command.

Example 7-10 Sample output of top

```
7:25pm up 2:34, 3 users, load average:1.02, 1.06, 0.83
40 processes:38 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:4.2%user, 18.9%system, 0.0%nice, 76.6%idle
Mem:255572K av, 239200K used, 16372K free, 0K shrd, 19308K buff
Swap:1048120K av, 0K used, 1048120K free 87736K cached

PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
12795 root 20 0 10592 2404 924 R 98.9 0.9 22:11 jre
13125 root 10 0 1028 1024 832 R 0.9 0.4 0:00 top
1 root 8 0 524 524 456 S 0.0 0.2 0:04 init
2 root 9 0 0 0 0 SW 0.0 0.0 0:00 keventd
3 root 19 19 0 0 0 SWN 0.0 0.0 0:00 ksoftirqd_CPU0
4 root 9 0 0 0 0 SW 0.0 0.0 0:03 kswapd
5 root 9 0 0 0 0 SW 0.0 0.0 0:00 kreclaimd
6 root 9 0 0 0 0 SW 0.0 0.0 0:00 bdflush
7 root 9 0 0 0 0 SW 0.0 0.0 0:00 kupdated
8 root -1 -20 0 0 0 SW<0.0 0.0 0:00 mdrecoveryd
15 root 9 0 0 0 0 SW 0.0 0.0 0:00 scsi_eh_2
18 root 9 0 0 0 0 SW 0.0 0.0 0:08 kjournald
93 root 9 0 0 0 0 SW 0.0 0.0 0:00 khubd
185 root 9 0 0 0 0 SW 0.0 0.0 0:00 kjournald
628 root 9 0 620 620 524 S 0.0 0.2 0:00 syslogd
633 root 9 0 1100 1100 448 S 0.0 0.4 0:00 klogd
653 rpc 9 0 592 592 504 S 0.0 0.2 0:00 portmap
681 rpcuser 9 0 764 764 664 S 0.0 0.2 0:00 rpc.statd
```

You can further modify the processes using **renice** to give a new priority to each process. If a process hangs or occupies too much CPU, you can kill the process. Of course you can also use the standard commands **renice** or **kill** to perform these steps, but with **top** you have one interface to perform all these tasks.

The columns in the output are as follows:

PID	Process identification.
USER	Name of the user who owns (maybe started) the process.
PRI	Priority of the process (see Process priority and nice levels, for details).
NI	Niceness level (that is, if the process tries to be nice by adjusting the priority by the number given, see below for details).
SIZE	Amount of memory (code+data+stack) in KB in use by the process.
RSS	Amount of physical RAM used, in KB.
SHARE	Amount of memory shared with other processes, in KB.
STAT	State of the process: S=sleeping, R=running, T=stopped or traced, D=interruptible sleep, Z=zombie. Zombie processes are discussed further in “Zombie processes” on page 234.
%CPU	Share of the CPU usage (since last screen update).
%MEM	Share of physical memory.
TIME	Total CPU time used by the process (since it was started).
COMMAND	Command line used to start the task (including parameters.

For more information on **top**, see the online help (**man top**).

Process priority and nice levels

Process priority is a number that determines how much CPU time a process gets. The kernel adjusts this number up and down as needed. The nice value is a limit on the priority. The priority number is not allowed to go below the nice value (a lower nice value is a more favored priority).

Note: It may not always be possible to change the priority of a process via the nice level. If a process is running too slowly, you can assign more CPU to it by giving it a lower nice level. Of course, this means that all other programs have fewer processor cycles and will run more slowly.

Linux supports nice levels from 19 (lowest or least nice—gets more CPU) to -20 (highest or nicest). Without an option the default value is 10. To change the nice level of a program to a negative number (which makes it less nice to other processes and therefore may increase priority), it is necessary to log on as root.

To start the program xyz with a nice level of -5, issue the command:

```
nice -n 5 xyz
```

To change the nice level of a program already running, issue the command:

```
renice -10 pid
```

Where pid is the process identification of the process. The process will decrease its nice level to -10.

Zombie processes

When a process has already terminated through receiving a signal to do so, it normally takes some time until it has finished all tasks (closing open files, etc.) before ending itself. In that normally very short time frame, the process is a *zombie*.

After the process has finished all these shutdown tasks, it reports to the parent process that it is about to terminate. Sometimes a zombie process is unable to terminate itself, in which case, you will see processes with a status of Z (zombie).

It is not possible to kill such a process with the **kill** command, because it is already considered "dead". If you cannot get rid of a zombie, you can kill the parent process and then the zombie disappears as well. However, if the parent process is the init process, you should not kill the process. The init process is a very important process and therefore a reboot may be needed to get rid of the zombie process.

7.3.4 iostat

The **iostat** command lets you see average CPU times since the system was started, in a way similar to **uptime**. In addition, however, **iostat** creates a report about the activities of the ESS Model 800 disk subsystem on the server. The report is split in CPU utilization and device utilization, where device utilization means the disk subsystem. Example 7-11 illustrates a sample output of the **iostat** command.

Example 7-11 Sample output of iostat

```
Linux 2.4.7-10 (nf5000)11/07/2001
avg-cpu:%user %nice %sys %idle
5.27 0.03 27.26 67.45

Device:tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dev3-0 2.03 223.78 0.00 2365914 0
dev8-0 12.17 125.81 351.54 1330060 3716564
```

The CPU utilization report has four sections:

- %user: Shows the percentage of CPU utilization that occurred while executing at the user level (applications).
- %nice: Shows the percentage of CPU utilization that occurred while executing at the user level with nice priority (priority and nice levels are described in "Process priority and nice levels" on page 233).
- %sys: Shows the percentage of CPU utilization that occurred while executing at the system level (kernel).
- %idle: Shows the percentage of time the CPU was idle.

The device utilization report is split in the following sections:

- Device: The name of the block device
- tps: The number of transfers per second (I/O requests per second) to the device. Multiple single I/O requests can be combined in a transfer request; because of that a transfer request can have different sizes.
- Blk_read/s, Blk_wrtn/s: Blocks read and written per second indicates data read/written from/to the device in seconds. Blocks may also have different sizes. Typical sizes are 1024, 2048, or 4048 bytes, depending on the partition size. For example, the block size of /dev/sda1 can be found with:

```
dumpe2fs -h /dev/sda1 |grep -F "Block size"
```

Which will give an output similar to:

```
dumpe2fs 1.23,15-Aug-2001 for EXT2 FS 0.5b,95/08/09
Block size:1024
```

- Blk_read, Blk_wrtn: Blocks read/written indicates the total number of blocks read/written since boot.

For more information on **iostat** see the online help (**man iostat**).

7.3.5 vmstat

vmstat provides information about processes, memory, paging, block I/O, traps, and CPU activity. Example 7-12 shows a sample **vmstat** output.

Example 7-12 Output of vmstat

```
procs memory swap io system cpu

r b w swpd free buff cache si so bi bo in cs us sy id
2 0 0 0 39256 19820 159460 0 0 200 100 133 141 31 5 64
```

The columns in the output are as follows:

- procs
  - r: The number of processes waiting for runtime.
  - b: The number of processes in un-interruptable sleep.
  - w: The number of processes swapped out but otherwise runnable. This field is calculated, but Linux never desperation swaps.
- memory
  - swpd: The amount of virtual memory used (KB)
  - free: The amount of idle memory (KB)
  - buff: The amount of memory used as buffers (KB)
- swap
  - si: Amount of memory swapped in from disk (KB/s)
  - so: Amount of memory swapped to disk (KB/s)
- IO
  - bi: Blocks sent to a block device (blocks/s)
  - bo: Blocks received from a block device (blocks/s)
- system
  - in: The number of interrupts per second, including the clock
  - cs: The number of context switches per second
- cpu (these are percentages of total CPU time):
  - us: User time
  - sy: System time
  - id: Idle time

7.3.6 sar

The **sar** command, which is included in the sysstat package, uses the standard system activity daily data file to generate a report.

To install the sysstat package, log in as root and mount the CD-ROM containing the package. The do the following steps:

```
cd /mnt/cdrom/RedHat/RPMS
```

Or

```
mount -t iso9660 /dev/cdrom /mnt
rpm -Uivh sysstat sysstat-3.3.5-3.i386.rpm
```

The system has to be configured to grab the information and log it; therefore, a **cron** job must be set up. Add the following lines to the /etc/crontab. Example 7-13. illustrates an example of automatic log reporting with *cron*.

Example 7-13 Example of automatic log reporting with cron

```
....
#8am-7pm activity reports every 10 minutes during weekdays.
0 8-18 **1-5 /usr/lib/sa/sa1 600 6 &
#7pm-8am activity reports every an hour during weekdays.
0 19-7 **1-5 /usr/lib/sa/sa1 &
#Activity reports every an hour on Saturday and Sunday.
0 ***0,6 /usr/lib/sa/sa1 &
#Daily summary prepared at 19:05
5 19 ***/usr/lib/sa/sa2 -A &
....
```

You get a detailed overview of your CPU utilization (%user, %nice, %system, %idle), memory paging, network I/O and transfer statistics, process creation activity, activity for block devices, and interrupts/second over time.

These are the main values that are displayed if you use **sar -A** (the -A is equivalent to **-bBcdqrRuvwWy -I SUM -I PROC -n FULL -U ALL**, which selects the most relevant counters of the system):

kbmemfree	Free memory in KB
kbmenmuse	Used memory in KB (without memory used by the kernel)
%memused	Percentage of used memory
kbmemshrd	Amount of shared memory by the system (always 0 with kernel 2.4)
kbbuffers	Memory used for buffers by kernel in KB
kbcached	Memory used for caching by kernel in KB
kbswpfree	Free swap space in KB
kbswpused	Used swap space in KB
%swpused	Percentage of used swap space
intr/s	Interrupts per second

Example 7-14 shows a sample of the output of the **sar -A** command.

Example 7-14 Example sar -A

```
Linux 2.4.7-10 (nf5000)11/07/2001

05:00:01 PM proc/s
05:10:00 PM 13.16
05:20:00 PM 0.14
05:30:00 PM 0.05
05:40:00 PM 0.05
```

```
05:50:01 PM 0.05
06:00:01 PM 0.05
06:10:01 PM 0.07
06:20:01 PM 0.05
06:30:00 PM 0.05
06:40:00 PM 0.05
06:50:00 PM 0.05
07:00:00 PM 0.05
07:09:59 PM 0.41
07:20:00 PM 0.12
07:30:00 PM 0.41
07:40:00 PM 0.49
07:50:00 PM 0.33
08:00:00 PM 0.26
08:10:02 PM 0.20
Average:0.85
05:00:01 PM cswch/s
...
Average:130.10
05:00:01 PM CPU %user %nice %system %idle
...
Average:all 5.67 0.03 33.46 60.84
05:00:01 PM INTR intr/s
...
Average:sum 114.41
05:00:01 PM pgpgin/s pgpgout/s activepg inadtypg inaclnpg inatarpg
...
Average:123.07 128.17 6011 17141 438 422
05:00:01 PM pswpin/s pswpout/s
...
Average:0.00 0.00
05:00:01 PM tps rtps wtps bread/s bwrtn/s
...
Average:10.89 5.46 5.43 246.15 256.34
05:00:01 PM frmpg/s shmpg/s bufpg/s campg/s
...
Average:0.01 0.00 -0.02 -1.84
05:00:01 PM kbmemfree kbmemused %memused kbmemshrd kbbuffers kbcached kbswpfree kbswpused
%swpused
...
Average:20006 235566 92.17 0 44418 49939 1048120 0
0.00
05:00:01 PM CPU i000/s i001/s i002/s i008/s i010/s i011/s i012/s i014/s i015/s
...
Average:0 100.00 0.90 0.00 0.00 0.00 9.68 0.67 3.16 0.00
05:00:01 PM dentunusd file-sz %file-sz inode-sz super-sz %super-sz dquot-sz %dquot-sz
rtsig-sz %rtsig-sz
...
Average:176307 491 5.99 178411 11 4.30 0 0.00
1 0.00
05:00:01 PM IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcst/s
...
Average:lo 0.01 0.01 0.35 0.35 0.00 0.00 0.00
Average:eth0 0.00 0.00 0.00 0.00 0.00 0.00 0.00
05:00:01 PM IFACE rxerr/s txerr/s coll/s rxdrop/s txdrop/s txcarr/s rxfram/s
rxfifo/s txfifo/s
...
Average:lo 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00
Average:eth0 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
0.00 0.00
05:00:01 PM totsck tcpsck udpsck rawsck ip-frag
...
Average:77 6 3 0 0
05:00:01 PM runq-sz plist-sz ldavg-1 ldavg-5
...
Average:4 57 0.64 0.67
05:00:01 PM DEV tps blks/s
...
Average:dev3-0 0.87 96.40
Average:dev8-0 10.01 406.08
```

7.3.7 isag

The output of **sar** is straight text and can be very time consuming to process. Instead, the **isag** command (Interactive System Activity Grapher) can show the data gathered by **sar** in a graphical format (see Figure 7-3).

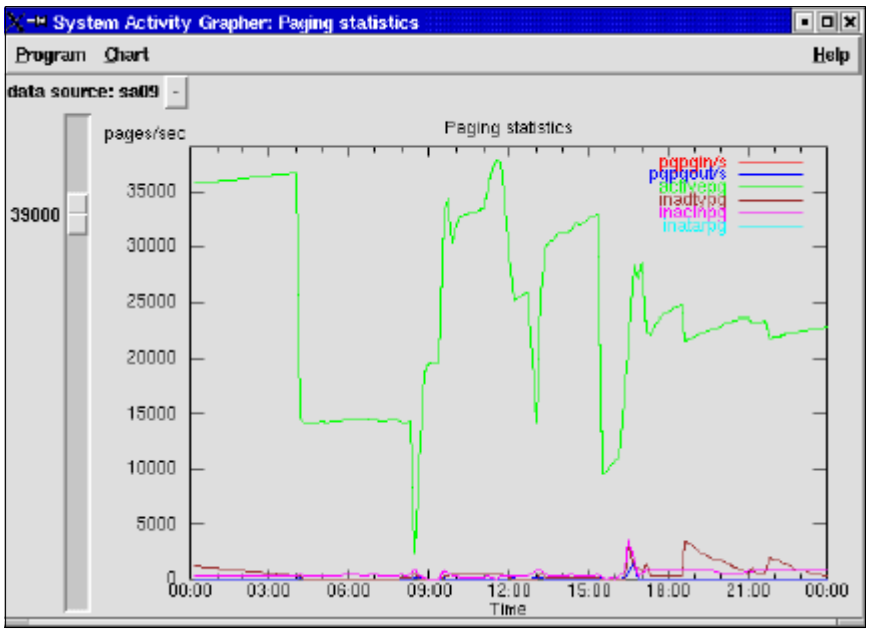


Figure 7-3 Paging statistics

When you start **isag** you must first select a data source. Click the **-** button to the right of `data source`. A menu will appear showing the different data sources available. The data sources are named sa01, sa02, sa03, etc., each standing for a day of the month when recorded (for example, sa11 would mean the log file recorded on the 11th day of the current month). However, only the last nine days are available for analysis.

The slider on the left of the window (see Figure 7-3) is used to adjust the vertical scale of the graph. By default, **isag** will display the paging statistics, but you can change the view by clicking **Chart** and then choosing the data you are interested in:

- I/O transfer rate
- Paging statistics
- Process creation
- Run queue
- Memory and swap
- Memory activities

- CPU utilization
- node status
- System switching
- System swapping

Paging statistics

Paging statistics has the following counters:

pgpgin/s	Total number of blocks the system paged in from disk per second
pgpgout/s	Total number of blocks the system paged out to disk per second
activepg	Number of active (recently touched) pages in memory
inadtypg	Number of inactive dirty (modified or potentially modified) pages in memory
inaclnpg	Number of inactive clean (not modified) pages in memory

Note: `isag` keeps data for only one week. After one week, the collected data for the seventh day will be deleted. This might not be enough to do a proper bottleneck analysis or to make a trend analysis of the server.

I/O transfer rate

I/O transfer rate has the following counters:

rtps	Total number of read requests issued to physical disk.
wtps	Total number of write requests issued to physical disk.
bread/s	Total amount of data read from the drive in blocks per second. A block is of indeterminate size.
bwrtn/s	Total amount of data written to the drive in blocks per second.

Figure 7-4 on page 240 illustrates a sample I/O transfer rate graphic report.

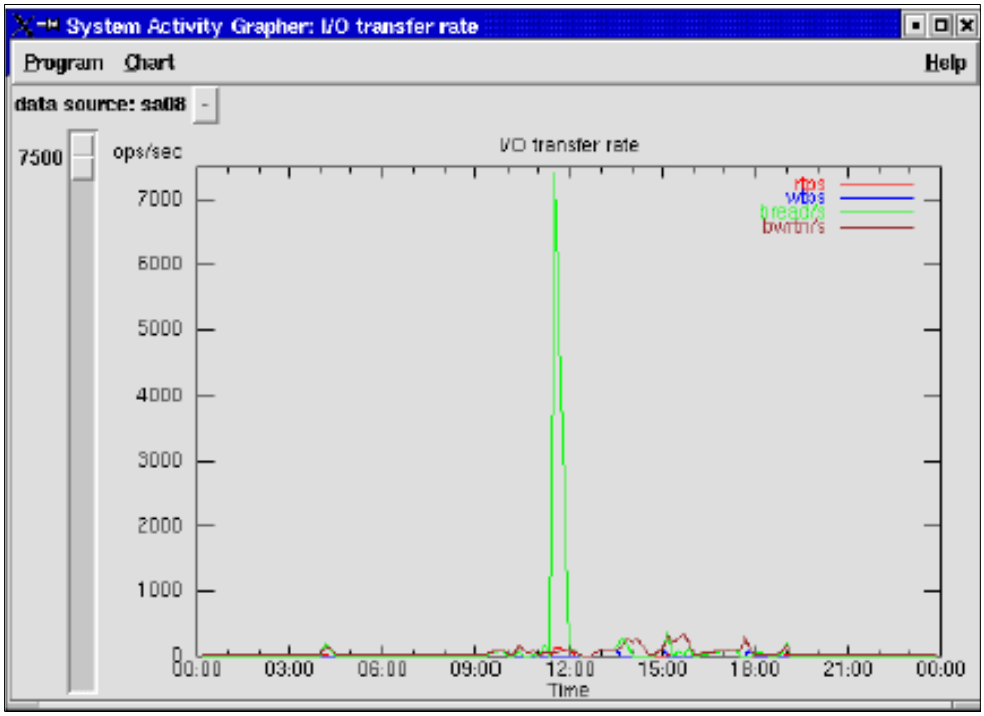


Figure 7-4 I/O transfer rate report

Run Queue

Run Queue has the following counters:

runq-sz	Run queue length (number of processes waiting for runtime)
plist-sz	Number of processes in the process list
ldavg-1	System load average for the last minute
ldavg-5	System load average for the last five minutes

Figure 7-5 on page 241 illustrates a sample Run Queue graphic report.

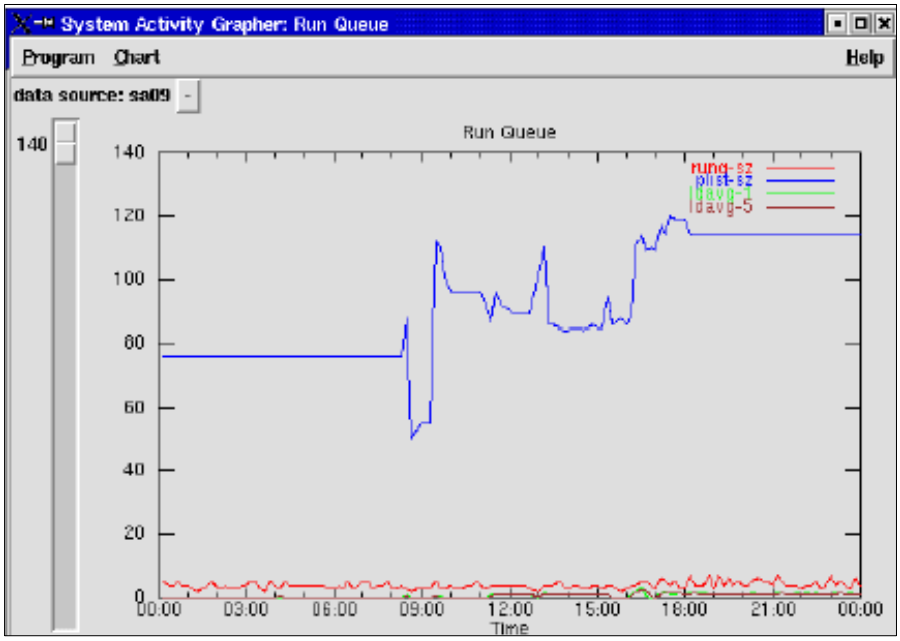


Figure 7-5 Run Queue report

Memory and Swap

Memory and Swap has the following counters:

kbmemfree	Amount of free memory in KB
kbmemused	Amount of used memory in KB (without memory used by the kernel)
kbmemshrd	Amount of memory shared by the system in KB (always 0 with kernel 2.4)
kbbuffers	Amount of memory used as buffers
kbcached	Amount of memory used for caching
kbswpfree	Amount of free swap space
kbswpused	Amount of used swap space

Figure 7-6 on page 242 illustrates a sample Memory and Swap graphic report.

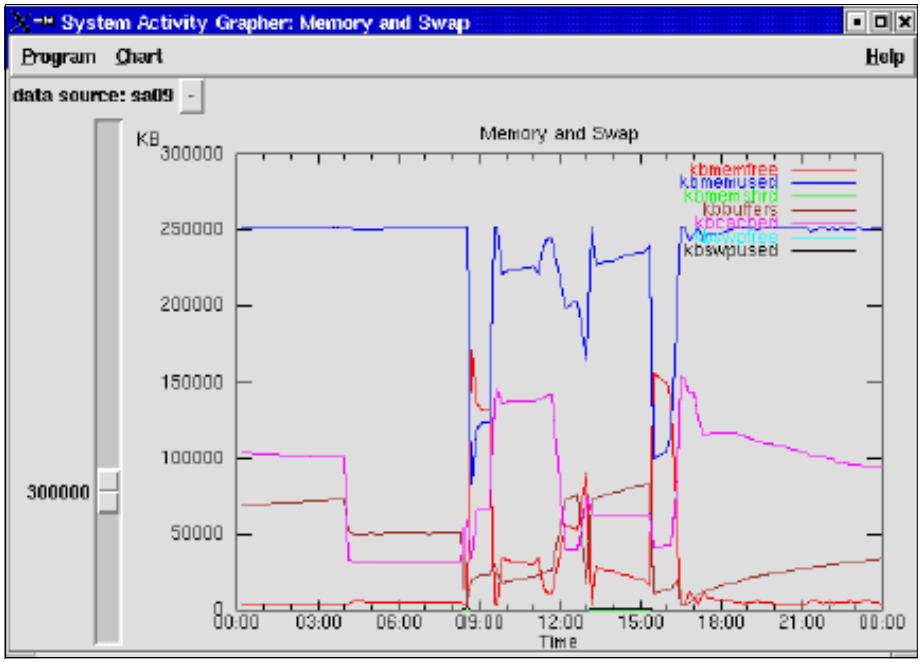


Figure 7-6 Memory and Swap report

Memory Activities

Memory Activities has the following counters:

frmpg/s Number of memory pages freed by the system per second. (A negative value represents the number of pages allocated by the system.)

shmpg/s Number of additional memory pages shared by the system per second. A negative value means fewer pages shared by the system.

bufpg/s Number of additional memory pages used as buffers by the system per second. A negative value means fewer pages used as buffers by the system.

camp/s Number of additional memory pages cached by the system per second. A negative value means fewer pages in the cache.

Figure 7-7 on page 243 illustrates a sample Memory Activities graphic report.

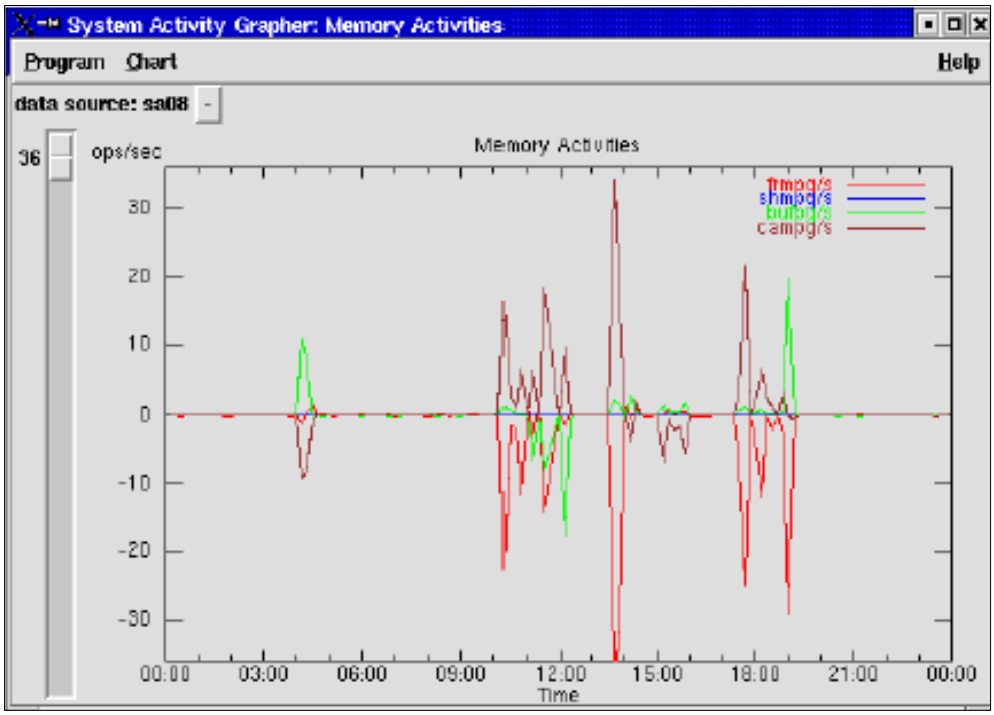


Figure 7-7 Memory Activities report

CPU Utilization

CPU Utilization has the following counters:

%user Percentage of CPU utilization that occurred while executing at the user level application)

%nice Percentage of CPU utilization that occurred while executing at the user level with nice priority

%system Percentage of CPU utilization that occurred while executing at the system level (kernel)

Figure 7-8 on page 244 illustrates a sample CPU Utilization graphic report.

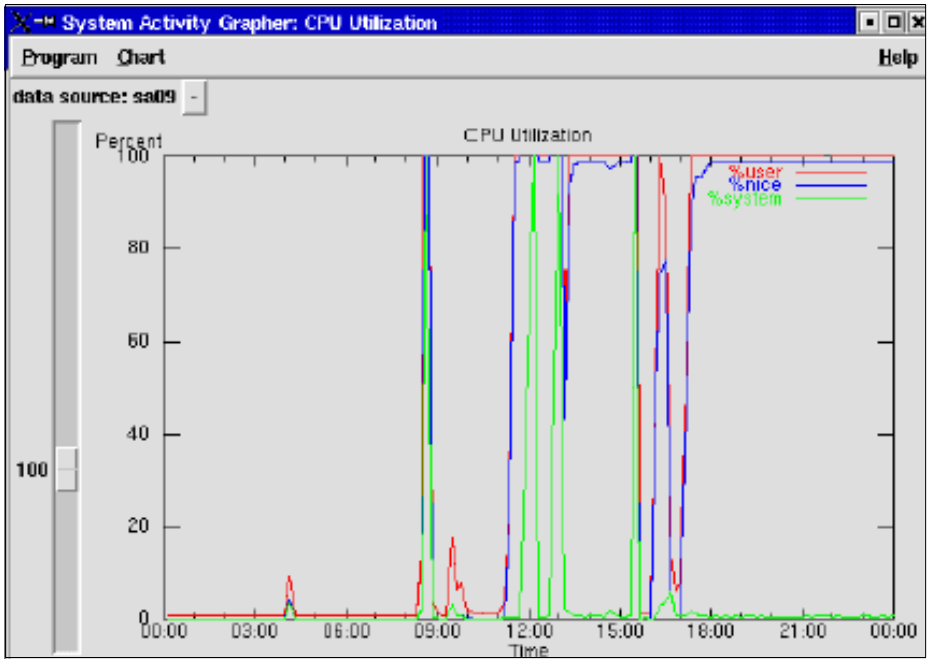


Figure 7-8 CPU Utilization

System swapping

System swapping uses the following counters:

pswpin/s Total number of swap pages the system brought in per second

pswpout/s Total number of swap pages the system brought out per second

For more information on **sar** and **isag** see the man pages (**sar**, **man isag**).

Figure 7-9 illustrates a sample system swapping graphic report.

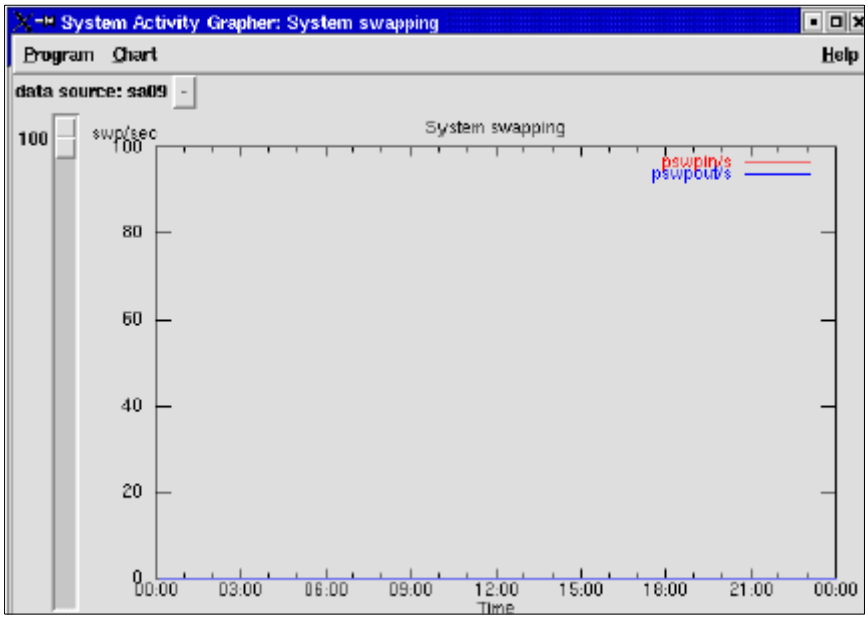


Figure 7-9 System swapping

7.3.8 GKrellM

GKrellM is one of the many tools available that can be used to get the actual system status. Other than the **sar** and **isag** commands, you can use GKrellM to get an idea of what your system is doing at a specific point in time.

Note: GKrellM is an Xwindow tool. Running X may impact your performance analysis.

GKrellM has the following counters:

- SMP CPU monitor that can chart individual CPUs
- Process monitor with a chart for load and a display of number of current processes and users
- Disk monitor that can chart individual disks or a composite disk
- Net interface monitors with charts for all routed net interfaces
- Memory and swap space usage meters and a swap page in/out chart

Of course, you can get all these things with multiple monitors, but the one big advantage of GKrellM is that it only takes up one process for monitoring your system. Further, the charts have an autoscaling feature, but you can also use fixed scaling modes. Figure 7-10 shows the output of GKrellM.

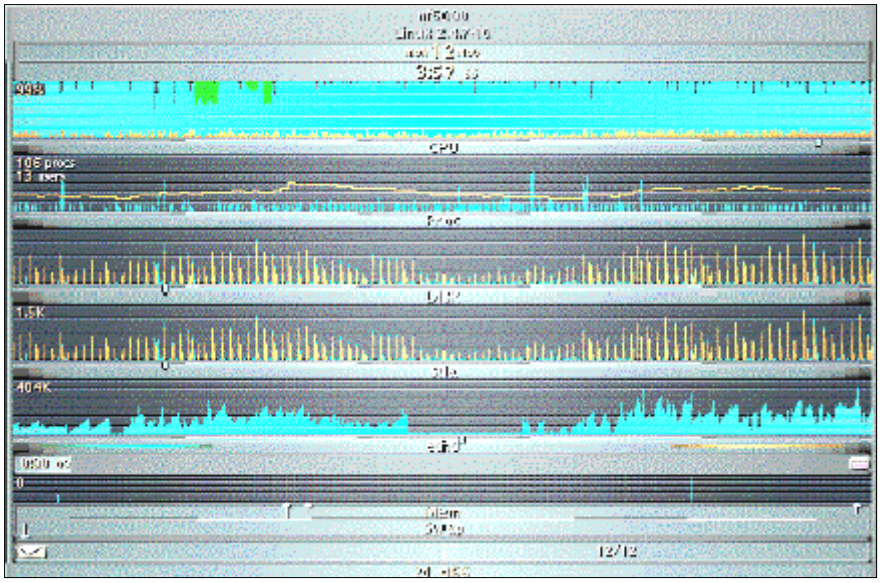


Figure 7-10 GKrellM

For more information on GKrellM see:

http://web.wt.net/~billw/gkrellm/gkrellm.html

7.3.9 KDE System Guard

KDE System Guard (KSysguard) is the KDE task manager and performance monitor. It features a client server architecture that allows monitoring of local as well as remote hosts. Figure 7-11 on page 246 shows the KDE System Guard default window.

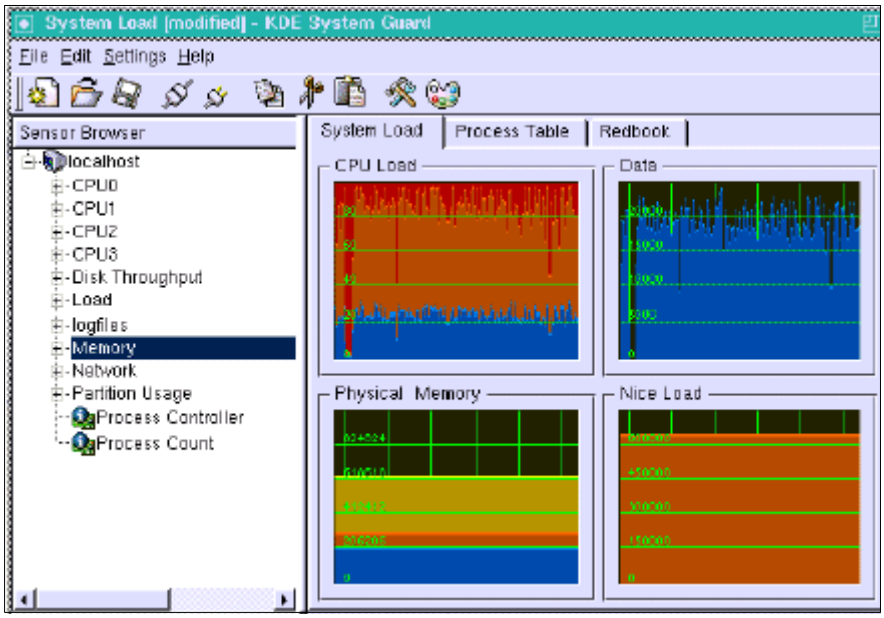


Figure 7-11 KDE System Guard default window

The graphical front end uses sensors to retrieve the information it displays. A sensor can return simple values or more complex information such as tables. For each type of information, one or more displays are provided. Displays are organized in worksheets that can be saved and loaded independently from each other.

Note: KSysguard is an Xwindow tool. Running X may impact your performance analysis.

The KSysguard main window (see Figure 7-11) consists of a menu bar, an optional tool bar and status bar, the sensor browser, and the work space. When first started, you see your local machine listed as localhost in the sensor browser and two pages in the work space area. This is the default setup.

The sensor browser displays the registered hosts and their sensors in a tree form, and includes the type of data. Each sensor monitors a certain system value. All of the displayed sensors can be dragged and dropped in the work space. There are two options:

- You can delete and replace sensors in the actual work space.
- You can create a new worksheet and drop new sensors meeting your needs.

KSysguard is part of the KDE project and information and updates can be obtained at:

`http://www.kde.org`

7.4 Logical Volume Manager for Linux (LVM)

Embedded into the Linux kernel, Sistina's Logical Volume Manager software is the standard LVM distribution used by tens of thousands of end users worldwide. LVM is a major building block in Linux because it enables robust, enterprise-level disk volume management by grouping arbitrary physical disks into virtual disk volumes. In addition, LVM increases availability and performance by:

- Providing online addition and removal of physical devices
- Dynamic disk volume re-sizing

- Striping
- Allows the IT managers the ability to administer entire storage configurations without interrupting access to end-user or application data

7.4.1 Implementation

Let us see where to get the tool and what O/S platforms are on the LVM supported list.

Downloading the LVM tool

LVM for Linux can be downloaded free of charge at the following Web site:

http://www.sistina.com/products_lvm_download.htm

Note: Because LVM is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing, the copyright holders and/or other parties provide the program “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair, or correction.

In order to use the LVM you will need to make sure that the kernel supports the tool. This will require another kernel build and re-compilation. Again we refer you to the IBM redpaper *Running the Linux 2.4 Kernel on IBM eServer xSeries Servers*, REDP0121, available from:

http://www.redbooks.ibm.com

Select **Redpapers** from the left navigation bar and do a search using the redpaper form number REDP0121, which contains detailed steps to add the LVM to the source kernel.

Supported platforms

There is a wide range of kernels where LVM is available on. In Linux 2.4, LVM will be fully integrated. From kernel 2.3.47 and onwards, LVM is in the process of being merged into the main kernel.

- Red Hat 7.3 and 8.0
- Red Hat 2.1 Advanced Server
- SuSE 7.3 and 8.0
- SuSE Linux Enterprise Server 7
- Standard Linux kernel 2.4.x

The standard Linux kernel 2.4 will contain everything you need. It is expected that most distributions will release with LVM included as a module. If you need to compile, just tick off the LVM option when selecting your block devices.

For a more complete discussion on how to use the LVM tool, visit the Web site:

http://tldp.org/HOWTO/LVM-HOWTO/index.html

7.4.2 Performance Management

The LVM tool can aid you in the performance monitoring and tuning tasks.

Data striping

For performance reasons, it is possible to spread data in a *stripe* over multiple disks. For example, Figure 7-12 illustrates an example where block 1 is on Physical Volume 0 (PV 1), and block 2 is on PV 1, while block 3 is on PV 2. You can also stripe over more than 3 disks.

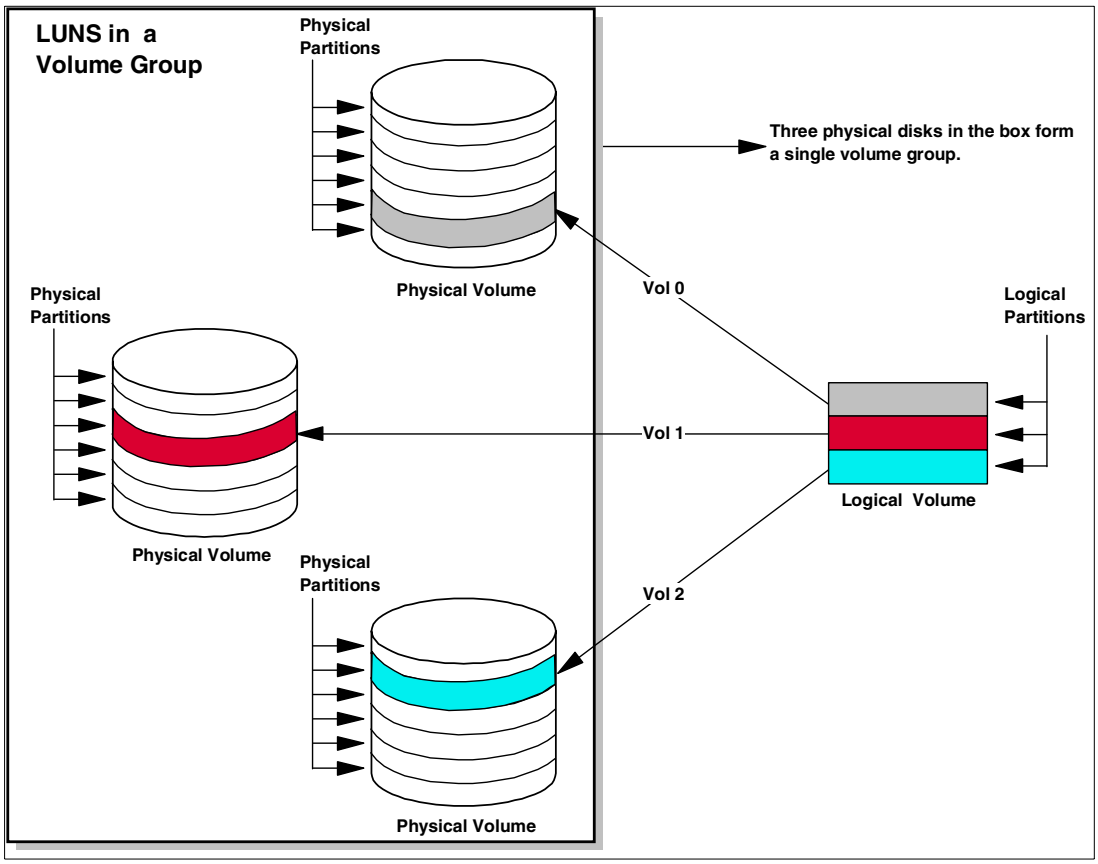


Figure 7-12 Striped volume set

This arrangement means that your have more disk bandwidth available. It also means that more **spindles** are involved.

Besides increasing performance, it is also possible to have your data in copies on multiple disks. This is called mirroring. Currently, LVM does not support this natively, but there are ways to achieve this. The ESS Model 800 is capable of creating RAID-10 volumes that can be imported as simple disks to the O/S. At this point the disks to the O/S appear as ordinary disks with no special features. All RAID functions are handled on the ESS Model 800.

Striping can be implemented on the RAID-5 disks to increase performance, as well as on RAID-10 disks.

Note: Remember that you should stripe across LUNs on different eight-packs to gain performance. Otherwise, if you stripe on the same eight-pack, then you could worsen the performance. For detailed discussion refer to 3.10, “Open systems striping” on page 72.

Benefits

Disk performance is influenced by three things, at least.

1. The most obvious is the speed at which data on a disk can be read or written sequentially. This is the limiting factor when reading or writing a large file on a SCSI/IDE bus with only a single disk on it.
2. Then there is the bandwidth available to the disk. If you have 7 disks on a SCSI bus, this may well be less than the writing speed of your disk itself. If you spend enough money, you can prevent this bottleneck from being a problem.
3. Then there is the latency. As the saying goes, latency is always bad news. And even worse, you cannot spend more money to get lower latency. Most disks these days appear to have a latency somewhere around 7 msec. Then there is the SCSI latency, which used to be something like 25 msec. What does this mean? The combined latency would be around 30 msec in a typical case. You can therefore perform only around 33 disk operations per second. If you want to be able to do many thousands of queries per second, and you don't have a massive cache, you are very much out of luck.

The disk rotation time for a 10,000 rpm drive is 6 msec, resulting in a 3 msec average latency. For the 15,000 rpm drive, the disk rotation time is 4 msec, resulting in a 2 msec average latency.

With the ESS Model 800 RAID implementation, you have multiple disks, or 'spindles', working in parallel, thus you can have multiple commands being performed concurrently, which nicely circumvents your latency problem. This is what striping does. With the RAID functionality in the ESS Model 800 and the proper SAN connectivity for max throughput, even sequential reading and writing may go faster. This helps the load level at the rank array level in the ESS.

LVM cannot tell that two PVs are on the same physical disk, so if you create a striped LV then the stripes could be on different partitions on the same disk, resulting in a decrease in performance rather than an increase.

Tip: The performance gain may well be negative if you stripe over 2 partitions of the same physical disk. Striping with two disks on a single IDE bus also appears useless. The way to prevent this is to stripe at the Linux O/S LVM level using LUNs from different arrays in the ESS Model 800.

Considerations

Striping without further measures raises your fault chance, on a per bit basis. If any of your disks die, your entire Logical Volume is gone. If you just concatenate data, only part of your file system is gone.

Do not create stripe sizes less than 32 KB for logical drives made up of arrays containing 18.2 GB drives on the ESS Model 800. For logical drives made up of arrays containing 36.4 or higher capacity drives, do not create a stripe size less than 64 KB. If many applications are addressing the same array, then software striping at the O/S level will not help much for host performance improvement, but it also will not hurt the array performance.

LVM native striping

Specifying stripe configuration is done when creating the Logical Volume with **`lvcreate`**. There are two relevant parameters. For example, if you wanted to create a 64 Kbyte stripe over 2 volumes, you would enter the following command:

```
# lvcreate -n stripedlv -i 2 -I 64 mygroup -L 20M
```

With -i we tell LVM how many physical volumes it should use to stripe across. Striping is not really done on a bit-by-bit basis, but on blocks. With -i we can specify the granulation in kilobytes. Note that this must be a power of 2, and that the coarsest granulation is 128 Kbyte.

7.4.3 Hardware RAID

A striped RAID-5 or RAID-10 LUN is created on the ESS Model 800 before it is imported to the Linux O/S. When Linux boots on ESS Model 800 with this configuration it will only *see* this disk as a single disk. This means, as far as LVM is concerned, that there is just one disk in the machine and it is to be used as such. If one of the disks fails in the ESS Model 800, LVM will not even know. When the IBM representative replaces the disk (even on the fly), LVM will not know about that either, and the controller will resync the mirrored array and all will be well. This is where most users take a step back and ask: Then what good does LVM do for me with this RAID controler? The easy answer is that in most cases, after you define a logical drive in the ESS Model 800, you cannot add more disks to that drive later. So if you miscalculate the space requirements, or you simply need more space, you cannot add a new disk or set of disks into a pre-existing O/S level software stripe-set. This means that you must create or assign through the ESS Specialist a new RAID LUN in the ESS, and then with LVM you can simply extend the LVM logical volume so that it seamlessly spans both LUNs on the host platform.

7.5 Swapping

While swapping (writing modified pages out to the system swap space) is a normal part of a Red Hat and SuSE Linux system's operation, it is possible for a system to experience too much swapping. The reason to be wary of excessive swapping is that the following situation can easily occur, over and over again: Pages from a process are swapped; the process becomes runnable and attempts to access a swapped page; the page is faulted back into memory; a short time later, the page is swapped out again.

If this sequence of events is widespread, it is known as *thrashing* and is normally indicative of insufficient RAM for the present workload. Thrashing is extremely detrimental to system performance, as the CPU and I/O loads that can be generated in such a situation can quickly outweigh the load imposed by system's real work. In extreme cases, the system may actually do no useful work, spending all its resources on moving pages to and from memory.

7.6 Virtual memory

While virtual memory makes it possible for computers to more easily handle larger and more complex applications, as with any powerful tool, it comes at a price. The price in this case is one of overhead: An application that is 100 percent memory-resident will run faster than one residing in virtual memory.

However, this is no reason to throw up one's hands and give up. The benefits of virtual memory are too great to do that. And, with a bit of tuning, good performance is possible. The thing that must be done is to look at the system resources that are impacted by heavy use of the virtual memory subsystem.

Worst case scenario

Let us consider what system resources are used by extremely heavy page fault and swapping activity:

- RAM: It stands to reason that available RAM will be low (otherwise there would be no need to page fault or swap).
- Disk: While disk space would not be impacted, I/O bandwidth would be.
- CPU: The CPU will be expending cycles doing the necessary processing to support memory management and setting up the necessary I/O operations for paging and swapping.

The interrelated nature of these loads makes it easy to see how resource shortages can lead to severe performance problems. All it takes is:

- A system with too little RAM
- Heavy page fault activity
- A system running near its limit in terms of CPU or disk I/O

At this point, the system will be thrashing, with performance rapidly decreasing.

Best case scenario

At best, virtual memory will present a minimal additional load to a well-configured system:

- RAM: Sufficient RAM for all working sets with enough left over to handle any page faults.
- Disk: Because of the limited page fault activity, disk I/O bandwidth would be minimally impacted.
- CPU: The majority of CPU cycles will be dedicated to actually running applications, instead of memory management.

From this, the overall point to keep in mind is that the performance impact of virtual memory is minimal when it is used as little as possible.

The primary determinant of good virtual memory subsystem performance is having enough RAM. Next in line (but much lower in relative importance) are sufficient disk I/O and CPU capacity. However, these resources do little to help the virtual memory subsystem performance (although they obviously can play a major role in overall system performance).

Note: A reasonably active system will always experience some page faults, if for no other reason than because a newly-launched application will experience page faults as it is brought into memory.

7.7 Bonnie

Bonnie is a performance measurement written by Tim Bray. For a more complete description and documentation on Bonnie go to the following Web site:

`http://www.coker.com.au/bonnie++/readme.html`

Bonnie performs a series of tests on a file of known size. If the size is not specified, Bonnie uses 100 Mb, but that probably is not enough for a big modern server. Bonnie works with 64-bit pointers if you have them.

For each test, Bonnie reports the bytes processed per elapsed second, per CPU second, and the % CPU usage (user and system).

7.7.1 Benchmarks

Bonnie does the following benchmarks:

- char output with put() / putc_unlocked()

  The result is the performance a program will see that uses putc() to write single characters. On most systems, the speed for this is limited by the overhead of the library calls into the libc, not by the underlying device. The _unlocked version (used if bonnie is called with -u) may be considerably faster, as it involves less overhead.

- char input with getc() / getc_unlocked()

  The result is the performance a program will see that uses getc() to read single characters. The same comments apply as to putc().

- Block output with write()

  This is the speed with which your program can output data to the underlying file system and device writing blocks to a file with write(). As writes are buffered on most systems, you will see numbers that are much higher than the actual speed of your device, unless you sync() after the writes (option -y) or use a considerably larger size for your test file than your OS will buffer. For Linux, this is almost all your main memory.

  If called with the -o_direct option, this operation (and the ones described in the following two paragraphs) is done with the O_DIRECT flag set, which results in direct DMA from your hardware to userspace, thus avoiding CPU overhead copying buffers around. This will prevent buffering, and gives a much better estimate of real hardware speed, also for small test sizes.

- Block input with read()

  This is the speed with which you can read blocks of data from a file with read(). The same comment as for block output regarding your OS doing buffering for you applies, with the exception that using -y does not help to get realistic numbers for reading. You would need to flush the buffers of the underlying block device, but this turns out to not be trivial,as you first have to find out the block device. It would be a Linux-only feature anyway.

- Block in/out rewrite

  Bonnie does a read(), changes a few bytes, write()s the data back and reread()s it. This is a pattern that occurs on some database applications. Its result tells you how your operating root (/) file system can handle such access patterns.

- Seeks

  Multiple processes do random lseek(). The idea of using multiple processes is to always have outstanding lseek() requests, so the device (disk) stays busy. Seek time is an indication of how good your OS can order seeks and how fast your hardware actually can do random accesses.

7.7.2 Downloading

For downloading Bonnie, go to the following Web site:

`http://www.textuality.com/bonnie/download.html`

Once there you find the sources of bonnie-1.4, selecting:

- Source tar ball, gzipped
- Source tar ball, bzipped2
- Source RPM (SuSE Linux)
- i386 binary RPM (SuSE Linux 7.1+)
- Alpha binary RPM (SuSE Linux 7.1+)

Installation and compilation should be straightforward. For Linux, your easiest option is to use rpm --rebuild on the source RPM. If you use Linux (preferably SuSE Linux) with a i386 machine, you can even use the binary RPM.

7.8 Bonnie++

Bonnie++ is a benchmark suite that is aimed at performing a number of simple tests of hard drive and file system performance. Then you can decide which test is important and decide how to compare different systems after running it.

The main program tests database type access to a single file (or a set of files if you wish to test more than 1 G of storage), and it tests creation, reading, and deleting of small files that can simulate the usage of programs such as Squid, INN, or Maildir format e-mail.

The ZCAV program tests the performance of different zones of a hard drive. It does not write any data (so you can use it on full file systems). It can show why comparing the speed of Windows at the start of a hard drive to Linux at the end of the hard drive (typical dual-boot scenario) is not a valid comparison.

Bonnie++ was based on the code for Bonnie by Tim Bray. Go to the following Web site for a summary of the differences between Bonnie 1.0 and Bonnie++ 1.0.

http://www.coker.com.au/bonnie++/

The original author (Tim Bray) has also put a description of bonnie on his pages.

Implementing Linux with IBM Disk Storage, SG-24-6261, has examples of outputs with the monitoring tool Bonnie. Please refer to this publication for a more complete discussion on Linux implementation and the ESS.

7.9 Disk bottlenecks

As a general rule for Linux performance analysis, the more observable gains with system performance can be achieved by properly tuning and sizing the memory subsystem, disk subsystem, and network subsystem, in that order.

The disk subsystem can be the most important aspect of I/O performance, but problems can be hidden by other factors, such as the lack of memory. Finding disk bottlenecks is easier than finding processor and memory bottlenecks, because the performance degradation is readily apparent.

The disk subsystem's speed affects the overall performance of the file server in the following ways:

- It usually improves the minimum sustained transaction rate.
- It may only slightly affect performance under light loads because most requests are serviced directly from the disk cache. In this case, network transfer time is a relatively large component and disk transfer times are hidden by disk cache performance.
- As the server disk performance improves, increased network adapter and CPU performance is required to support greater disk I/O transaction rates.

When the I/O subsystem is well tuned and performing efficiently, more throughput and transactions per second can be done by the system as users and workload increase (see Figure 7-13 on page 254).

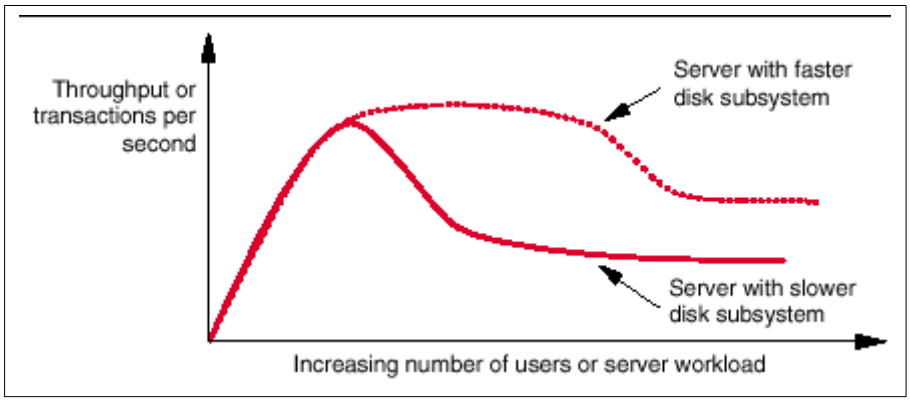


Figure 7-13 Effect of tuning the I/O subsystem

The I/O operations per second counter in the tools so far discussed in this chapter can be used to determine if the server has disk bottlenecks. Collect logged data over a period of time and then analyze the collected data to find if a trend can be detected, which will point to a future disk bottleneck.

After verifying that the disk subsystem is causing a system bottleneck, a number of solutions are possible. These solutions include the following:

- Consider using faster disks. Allocating your application's data on the 15 Krpm disk drive ranks will deliver better performance as compared to the 10 Krpm disk drive ranks.
- Eventually change the RAID implementation if this is relevant to the server's I/O workload characteristics. For example, going to RAID-10 if the activity is heavy random writes may show observable gains.
- Add more arrays. This will allow you to spread the data across multiple physical disks and thus improve performance for both reads and writes. Also, use hardware RAID instead of the software implementation provided by Linux. If hardware RAID is being used, the RAID level is hidden from the operating system.
- Add more RAM. Adding memory will increase system memory disk cache, which in effect improves disk response times.

Finally, if the previous actions do not provide the desired application performance, then offload processing to another host system in the network (either users, applications, or services).

8

Open system servers - Intel based

In this chapter we discuss performance considerations for attaching supported Intel-based open systems to the IBM TotalStorage Enterprise Storage Server Model 800. We discuss the use of specific tools for monitoring and tuning host performance issues. Also the most common disk performance bottlenecks are discussed, as well as how to mitigate these problems.

The most current list of Intel-based servers that can attach to the ESS can be found in the following Web site:

http://www.storage.ibm.com/disk/ess/supserver_summary_open.html

8.1 Host system performance

Gaining disk throughput and performance for any individual host will be affected by the connectivity between the host and the disk subsystem. But the health and tuning of the whole system plays an important part and should also be optimized in order to gain the best disk performance.

Tuning all the components in a system is demanding and will require you to not only take a benchmark before you change anything, but to take periodic measurements as you go. Nonetheless, this more comprehensive activity pays off with an optimized system performance.

The various system components that can affect the disk performance and are discussed in this chapter, are:

- Priorities between foreground and background processes
- Virtual memory
- System cache
- File system layout and management

There is a recommended publication that can help you when tuning the whole system: *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287.

Performance monitoring tools

The tools that can be used for monitoring and tuning the whole system performance discussed in this chapter are:

- Windows 2000 Performance console (Windows 2000 and Windows NT)
- Task Manager (Windows 2000 and Windows NT)
- NT Disk Administrator and Windows 2000 Disk Management
- IOMeter (Windows NT)
- NetWare Remote Manager (Novel Netware)
- Monitor (Novel Netware)
- VTune (Novel Netware)

8.2 Tuning Windows 2000 and NT systems

Windows 2000 and Windows NT are largely self-tuning, so even leaving the system defaults is reasonable from a performance perspective. There are, however, some things you can adjust to get the most out of your system.

The following list provides additional steps that can be taken to provide better disk performance on the host:

- Modify the priorities between foreground and background processes.
- Applications that are CPU and memory intensive should be scheduled during after-hours operation. Examples of these applications are virus scanners, backup software, and disk fragmentation utilities. These type of applications should be scheduled to run when the server is not being utilized.
- Allocate virtual memory pro-actively.
- Specify the server type to determine how system cache is allocated and used.
- Disable unnecessary services.

It is also helpful to understand the file system structure. Microsoft has also produced a document comparing the performance of Windows 2000 with Windows NT. This document name is *Windows 2000 Performance: An Overview*, and is available from:

http://www.microsoft.com/windows2000/server/evaluation/performance/overview.asp

8.2.1 Foreground and background priorities

Windows 2000 preemptively prioritizes process threads that the CPU has to attend to. Preemptive multitasking is a methodology whereby the execution of a process is halted and another is started, at the discretion of the operating system.

This setting lets you choose how processor resources are shared between the foreground process and the background processes. Typically, for a server, you do not want the foreground process to have more CPU cycles allocated to it.

To change this:

1. Open **Control Panel**.
2. Open **System**.
3. Select the **Advanced** tab.
4. Click the **Performance Option** button and the window shown in Figure 8-1 will pop up.

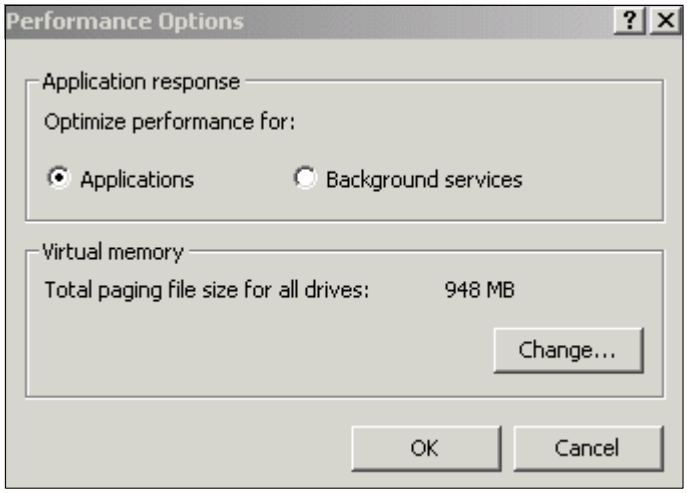


Figure 8-1 Performance options in Windows 2000

Under Application response, you can choose one of two settings to optimize performance:

- **Applications** if more processor resources are given to the foreground process than the background processes
- **Background services** if all programs receive equal amounts of processor resources

The underlying effect of application performance boost control significantly differs between various versions of Windows. In Windows 2000, the application boost GUI settings have the following counterpart in the Registry:

```
HKEY_LOCAL_MACHINE \System \CurrentControlSet \Control \PriorityControl \
Win32PrioritySeparation
```

The Win32PrioritySeparation Registry values in Windows 2000 Server are:

- 38 for applications performance optimization
- 24 for background services performance optimization

It is a good tuning practice to disable any prioritization of foreground applications if services serving users are running in the background. You should set this window according to you needs. By clicking the applications setting, the process priority is given to the foreground applications, and by clicking the background setting, the priority is given to background processes.

We strongly recommend that you use only the GUI interface for these settings in order to always get valid, appropriate, operating system revision-specific, and optimal values in the registry.

8.2.2 Virtual memory

Memory paging occurs when memory resources required by the server exceed the physical amount of memory installed in the server. Memory can be accessed at over 1 GB/s, whereas a single disk drive provides data at about 2–3 MB/s. Servers accessing information from disk run much more slowly than when the information can be accessed directly from memory.

Windows 2000, as most other server operating systems, employs *virtual memory* techniques that allow applications to address greater amounts of memory than what is physically available. Memory pressure occurs when the demand for physical memory exceeds the amount of installed memory, causing the operating system to *page* excess memory onto a disk drive.

Paging is the process whereby blocks of data are swapped from the physical memory to a file on the hard disk. The *paging file* is PAGEFILE.SYS. The combination of the paging file and the physical memory is known as *virtual memor*y.

You can control the size of the paging file and this can improve performance if you specify the minimum value to be what the server normally allocates during the peak time of the day. This ensures that no processing resources are lost to the allocation and segmentation of the paging space.

To configure the page file size:

1. Open the

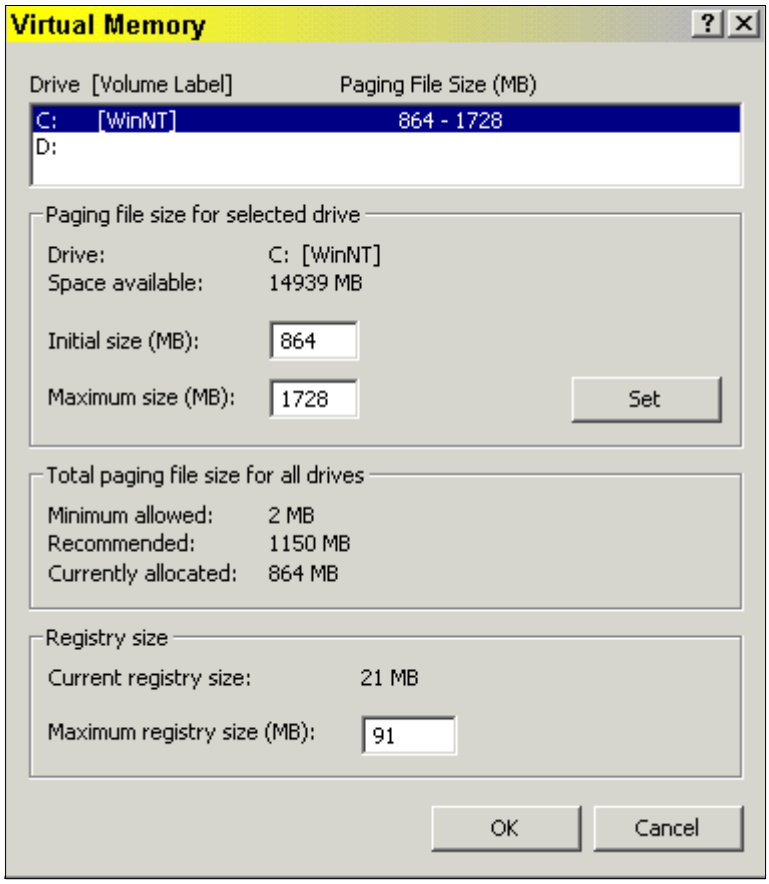


Figure 8-2 Virtual memory settings

You can set the initial size and the maximum size of the paging file for every drive. For a file server, set the minimum to the recommended value, as shown in the window. For other server applications the recommendation varies. For a discussion of recommended values refer to the publication *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287.

The size of the registry does not affect performance.

In a production environment with well-written server applications, hard page faults should not constantly occur. If there is any sustained paging, check the available bytes in the Task Manager.

- If Available Bytes is less than 20 percent of installed RAM, then add more RAM.
- If Available Bytes is much greater than 20 percent of total installed RAM, then the application cannot make use of additional RAM, so the only solution is to optimize the page device.

Consistent paging should be avoided because when the server is paging memory to disk, every user's performance will be adversely affected.

Configuring page file size for maximum performance gain

The operating system can do the work of dynamically increasing the page file size, but this is a relatively expensive operation and it will be done in a low memory situation, when additional work overhead is least wanted. Another reason to set the page file size to be big enough to prevent Windows from having to increase it, is to prevent fragmentation within the page file.

Creating the whole page file in one step reduces the possibility of having a partitioned page file and therefore improves system performance.

The page file size should be set to at least the size of the physical memory size, to allow for a full memory dump in the event of a system crash. In addition, having a page file size smaller than the current RAM size will affect performance of the server. Our recommendation is to set the memory page file size to twice the size of the RAM for a maximum performance gain. The only drawback of having a big page file is the restriction in space available for files on the hard drives. Since the host will be using ESS disks this should not be a concern.

The best way to create a contiguous static page file is to follow this procedure:

1. Remove the current page file from your server by clearing the Initial and Maximum size values in the Virtual Memory settings window, then click the **Set** button (refer to Figure 8-2 on page 259).
2. Reboot the machine and click **OK**; ignore the warning message about the page file.
3. Defragment the disk you want to create the page file on. This step should give you enough continuous space to avoid partitioning of your new page file.
4. Create a new static page file by setting the Initial and Maximum size with the same value. If possible, use twice the size of your RAM.
5. Reboot the server.

The above procedure will leave you with a contiguous static page file.

8.2.3 Windows paging optimization

Most Performance Monitor memory counters in Windows 2000 refer to virtual memory; they do not provide much information about physical memory usage. However, it is possible to detect memory paging by using the Performance console to examine the Memory:Pages/sec counter. Server performance degrades when this counter is consistently non-zero. In general, any sustained paging is detrimental to server performance. The Performance console counter Memory: Available Bytes refers to physical installed memory.

Many well-designed applications can use all installed memory to improve performance and reduce paging. However, some applications have limitations on the amount of memory that can be used (for example, Lotus Notes® only works with up to 2 GB of RAM). Installing additional memory with such applications yields no performance increase. When this is the case, it is important to maximize the efficiency of the paging drive to improve performance.

The example in Figure 8-3 on page 261 shows about 44 pages/second for the second period (the thick red line) and 35 pages/second for the third period. This translates to poor performance. The application should be evaluated to see if it can use additional memory. If so, memory should be added in 25 percent increments until paging is reduced to zero, as shown in the first event period.

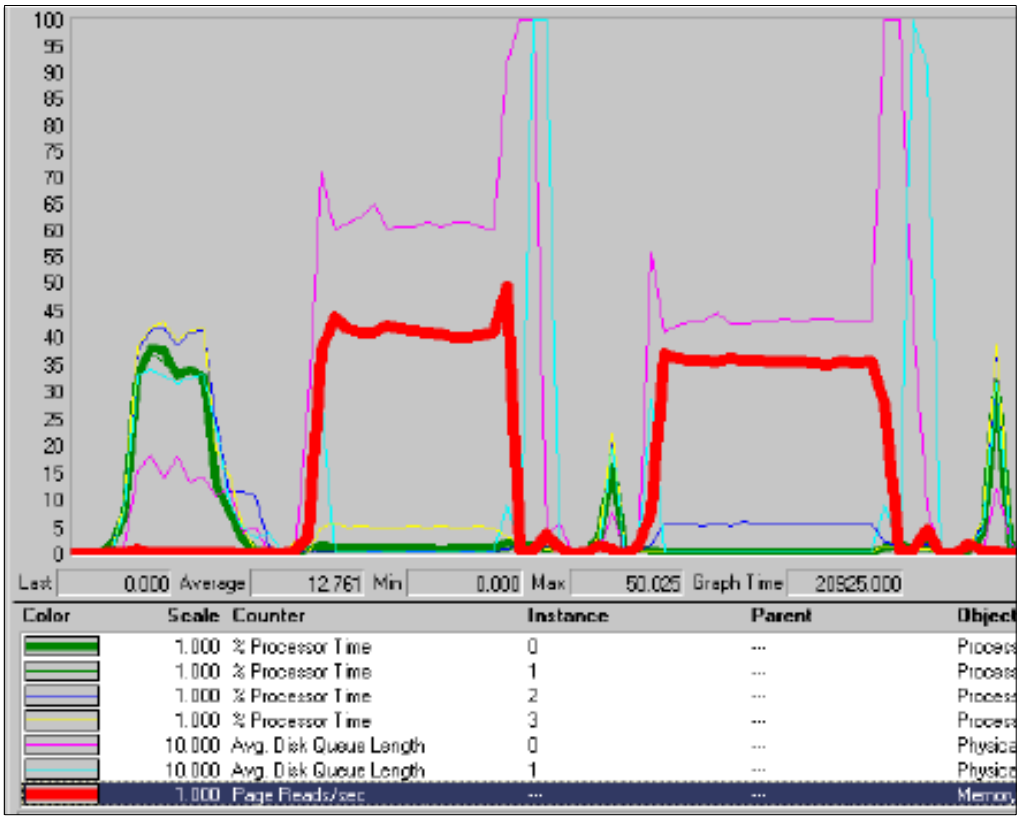


Figure 8-3 Detecting memory paging

For cases in which paging occurs, the page drive should be configured for optimal I/O performance.

Paging on a server local drive

For optimized performance, you may consider defining the paging on a server local drive instead of the ESS. In this case paging should be configured to be on a separate physical drive. Having data that is being accessed on the same drive as the paging drive can reduce performance, especially when multiple logical drives are configured on one physical drive. This causes long seek operations and slows performance.

If the page files and active data must reside on the same physical device, place them on the same logical drive. This will keep the page file and data files physically close together, and will improve performance by reducing the time spent seeking between the two logical drives. Of course, you can ignore this issue if no I/O access is made to the data drive during normal operation.

Ideally, the paging device will be a separate physical drive. In fact, performance can be improved if the paging drive can be formatted specifically for paging. In general, physical paging will occur in I/O sizes of up to 64 KB, so using a larger stripe size for the paging drive can improve performance.

However, in most cases, the paging drive's stripe size must be the same as the data drive's, and a compromise stripe size must be selected. 32 KB is probably the best choice because most paging I/O will be less than 64 KB, and most data I/O will also be in this range. The only way to know the exact stripe size is to identify the amount of paging I/O that is occurring, and determine the average bytes/transfer to the paging drive. Then, select the stripe size that is the best fit.

Note: Do not create stripe sizes less than 32 KB for logical drives made up of arrays containing 18.2 GB drives on the ESS Model 800. For logical drives made arrays containing 36.4 or higher capacity drives, do not create a stripe size of less than 64 KB. This helps the load level at the rank array in the ESS. If many applications are addressing the same array, then software striping at the O/S level will not help much, but it also will not hurt the array performance.

8.2.4 System cache tuning

In Windows 2000, you can optimize server performance by tuning the system cache.

The system cache is a dynamic memory pool used to store recently accessed data for all cacheable peripheral devices, which includes data transfers between hard drives, networks cards, and networks. The Windows Virtual Memory Manager copies data to and from the system cache as though it were an array in memory.

There is a specific configuration option that directly relates to a server's network performance. You can configure how the server allocates memory to local applications versus network connections, which affects both disk and network performance. The most important memory allocation controlled by this setting is that given to the system cache. It controls how a server prioritizes its memory allocations and thread priorities for network services versus local applications and the size given to the system cache.

1. Click **Start -> Settings -> Network and Dial-Up Connection**.
2. Select any of your local area connections (it does not matter which one).
3. Click **File -> Properties**.
4. Select **File and Print Sharing for Microsoft Networks**.
5. Click the **Properties** button. The window shown in Figure 8-4 will pop up.

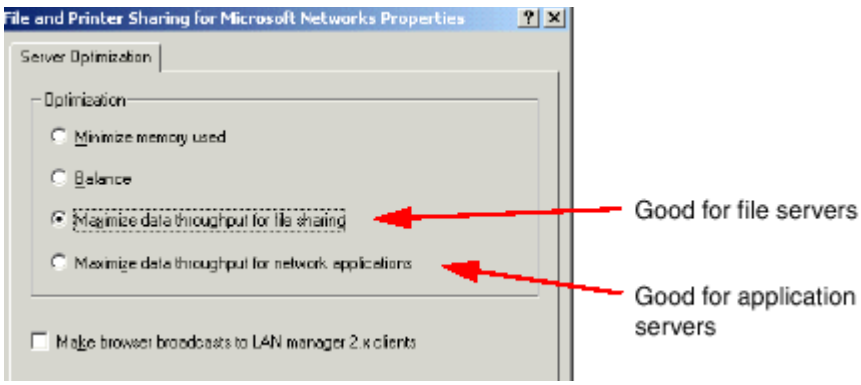


Figure 8-4 Configuring the system cache in Windows 2000

Note: This setting affects all LAN connections, so which LAN connection you choose in the above steps is not important. If you are not using this system as a file system server then you will not be able to modify the cache priorities here.

You can specify how much memory is reserved for the operating system indirectly by specifying how much memory (relatively) to give to server processes.

You have four choices:

1. Minimize memory used.

This choice will minimize the memory used for disk cache and maximize the memory available for the operating system. However, on file servers, the resulting performance would not be desirable. Therefore, only use this choice for workstations.

2. Balance

   This choice will attempt to balance the use of real memory between the operating system and the disk cache. This is a good choice for a non-dedicated server that is also used as a workstation.

3. Maximize throughput for file sharing

   This is the default setting. This instructs Windows to give the system cache a higher priority for memory allocation than the working sets of applications. It will yield the best performance in the file server environment but will require sufficient physical memory; otherwise, a significant amount of swapping to the paging file will occur.

4. Maximize throughput for network application

   This choice is the recommended setting for machines used as application servers and database servers.

8.2.5 Disabling unnecessary services

There are services running on your server that may be unnecessary. To view these services running in Windows 2000, right-click **My Computer** and select **Manage**. A window similar to that shown in Figure 8-5 will appear. Select **Services** in the left pane of this window and all the services will appear in the right pane.

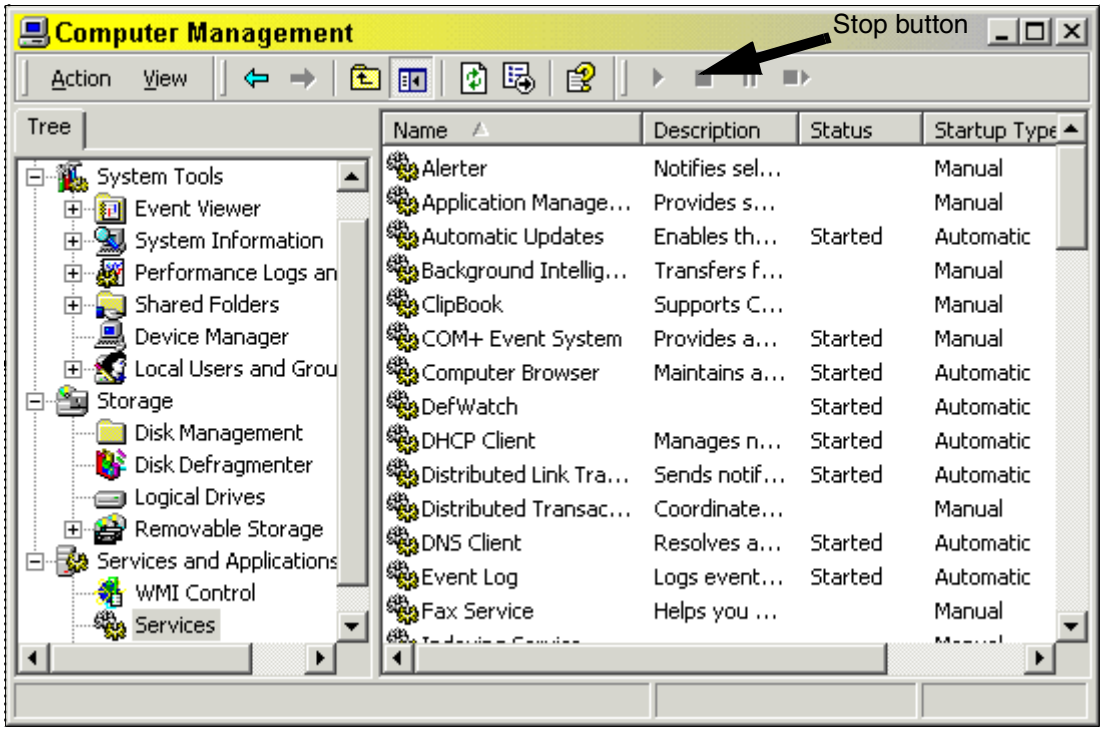


Figure 8-5 Windows 2000 Services window

You should stop services that are not needed to free additional memory to those that need it most, such as the operating system and user applications. To do this, select a service from the service list and click the **Stop** button.

Also examine the startup values of the installed services. Right-click the service and select **Properties**. Select **Disabled** if you do not want this service to run at all on server startup, or **Manual** if you want to start a service only at the time you need to use it.

You can also stop unneeded applications and processes using the Task Manager. Unneeded applications and processes are those that you do not need running at the moment, for example, when an application is launched at startup that does system maintenance, such as disk scanning and de-fragmentation.

To open Task Manager, press the Ctrl + Shift + Esc keys. From the Applications tab, select the unneeded application then click the **End Task** button. You can also do this from the Processes tab by selecting the unneeded process then clicking the **End Process** button, as illustrated in Figure 8-6.

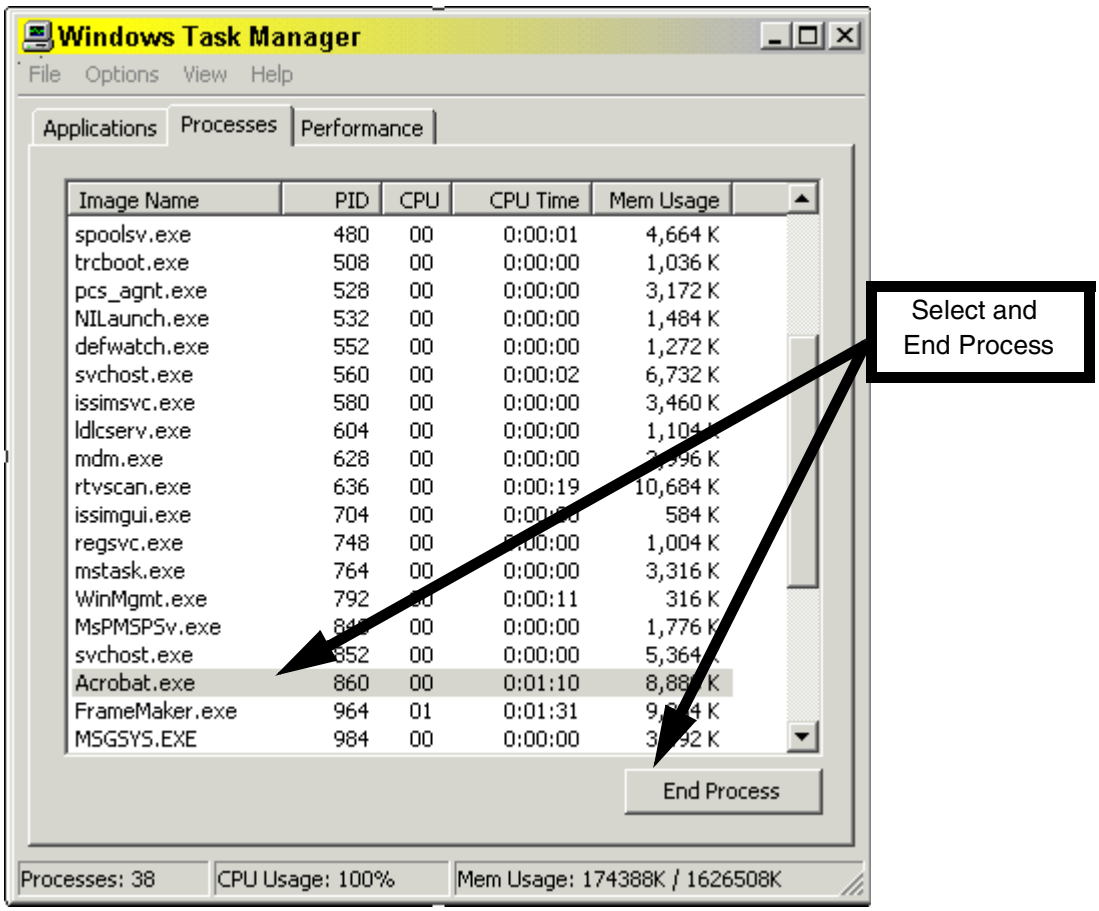


Figure 8-6 Task Manager

A process is considered unnecessary when it has nothing to do with your current server function. It could have been invoked from the registry by some application that was not correctly un-installed, for example.

8.2.6 File system overview

There are three disk file systems used in Windows 2000 systems:

FAT The File Allocation Table (FAT) file system is the original file system introduced by Microsoft. FAT was designed for small disks and a simple directory structure. The maximum drive size using FAT is 2 GB.

FAT and FAT32 do not have any file or folder security. Therefore the server does not have to check the permissions on an individual file when it is accessed by users. For performance reasons, it is recommended that FAT be used on drives that will be less than 400 MB.

FAT32 FAT32 is a newer, improved version of FAT. FAT32 allows you to format a disk that is greater than 2 GB. It also uses smaller clusters than the FAT. This saves disk drive space on large drives by being able to store data more efficiently.

NTFS NTFS uses a B-tree structure. This type of structure improves performance by minimizing the number of times the disk is accessed, which makes it faster than the FAT.

Many factors can influence the speed of the file systems:

- If the disk is fragmented, then NTFS will most likely be faster than FAT and FAT32.
- The file structure.
- The size of the files.
- The number of files in a directory.

The basic rule of thumb is that FAT or FAT32 should be used for logical volumes less than 400 MB and NTFS should be used to for partitions larger than 400 MB.

FAT overview

The file allocation table (FAT) is by far the most simplistic of the file systems supported by Windows NT. The FAT file system is characterized by the file allocation table, which resides at the very “top” of the volume. To protect the volume, two copies of the FAT are kept in case one becomes damaged. In addition, the FAT tables and the root directory are stored in a fixed locations so that the system's boot files can be correctly located.

A disk formatted with FAT is allocated in clusters whose size are determined by the size of the volume. When a file is created, an entry is created in the directory and the first cluster number containing data is established. This entry in the FAT table either indicates that this is the last cluster of the file, or points to the next cluster.

The FAT file system uses the cluster sizes shown in Table 8-1. These sizes are the same under Microsoft Windows NT, Microsoft MS-DOS, Microsoft Windows 95, 2000, and any other operating system that supports FAT.

Table 8-1 FAT file system cluster sizes

Drive size (logical volume)	FAT type	Sectors per cluster	Cluster size
0 MB - 15 MB	12-bit	8	4 k
16 MB - 31 MB	16-bit	1	512 bytes
32 MB - 63 MB	16-bit	2	1 K
64 MB - 127 MB	16bit	4	2 K
128 MB - 255 MB	16-bit	8	4 K
256 MB - 511 MB	16-bit	16	8 K
512 MB - 1023 MB	16-bit	32	16 K
1024 MB - 2048 MB	16-bit	64	32 K

Drive size (logical volume)	FAT type	Sectors per cluster	Cluster size
2048 MB - 4096 MB	16-bit	128	64 K
*4096 MB - 8192 MB	16-bit	256	128 K
*8192 MB - 16384 MB	16-bit	512	256 K
(*) Note: These implementations are Windows NT specific and are provided in order to support > 4 GB FAT partitions using 128 k or 256 k clusters; the drives must use > 512 byte sectors.			

Updating the FAT table is very important as well as time consuming. If the FAT table is not regularly updated, it can lead to data loss. It is time consuming because the disk read heads must be repositioned to the drive's logical track zero each time the FAT table is updated.

There is no organization to the FAT directory structure, and files are given the first open location on the drive. In addition, FAT only supports read-only, hidden, system, and archive file attributes.

NTFS overview

NTFS is a journaling file system that enables fast file recovery. Journaling file systems are based on the transaction processing concepts found in database theory. Internally, it more resembles a relational database than a traditional file system. It is comparable in function to the Veritas file system found on some UNIX implementations.

NTFS was designed to provide recoverability, security, and fault tolerance through data redundancy. In addition, support was built into NTFS for large files and disks, Unicode-based names, bad-cluster remapping, multiple data streams, general indexing of file attributes, and POSIX. All of these contribute to making NTFS a robust file system.

Windows NT uses the following default cluster sizes for NTFS, as shown in Table 8-2, where the value for number of sectors assumes a standard, 512 byte sector. On systems with sectors that are not 512 bytes, the number of sectors per cluster may change, but the cluster size remains fixed.

Table 8-2 NTFS default cluster sizes

Drive size	Cluster size	Number of sectors
512 MB or less	512 bytes	1
513 MB to 1024 MB (1 GB)	1 K	2
1025 MB to 2048 MB (2 GB)	2 K	4
2049 MB to 4096 MB (4 GB)	4 K	8
4097 MB to 8192 MB (8 GB)	8 K	16
8193 MB to 16384 MB (16 GB)	16 K	32
16385 MB to 32768 MB (32 GB)	32 K	64
Greater than 32768 MB (32 GB)	64 K	128

These values are only used if an allocation unit size is not specified at format time, using the /A:<size> switch with the format command.

Disabling short name generation

By disabling the short name generation on a NTFS partition, you can increase the directory enumeration performance. The increase in performance is mainly seen when a directory contains a large amount of files and directories. Before disabling short name generation, make sure that there is no DOS or 16-bit application running on the server that requires 8.3 file names, nor are there any users accessing the files on the server via 16-bit applications.

To disable the generation of 8.3 short names, edit the following registry parameter:

```
HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Control \FileSystem
\NtfsDisable8dot3NameCreation
```

Change its value from 0 to 1.

Reliability

To ensure reliability of NTFS, three major areas were addressed: Recoverability, removal of fatal single sector failures, and hot fixing.

The recoverability designed into NTFS is such that a user should never have to run any sort of disk repair utility on an NTFS partition. This is because NTFS uses a journaled log to keep track of transactions made against the file system. When a CHKDSK is performed on a FAT file system, the consistency of pointers within the directory, allocation, and file tables is being checked. Under NTFS, because a log of transactions against these components is maintained, CHKDSK need only roll back transactions to the last commit point in order to recover consistency within the file system.

Under FAT, if a sector that is the location of one of the file system's special objects fails, then a single sector failure will occur. NTFS avoids this in two ways. First, by not using special objects on the disk and tracking and protecting all objects that are on the disk. Second, under NTFS, multiple copies (the number depends on the volume size) of the Master File Table are kept.

Similar to OS/2® versions of HPFS, NTFS supports hot fixing. NTFS will attempt to move the data in a damaged cluster to a new location in a fashion that is transparent to the user. The damaged cluster is then marked as unusable. Unfortunately, it is possible depending on what damage has occurred that the moved data may be unusable.

Added functionality

NTFS fully supports the Windows NT security model and supports multiple data streams. No longer is a data file a single stream of data. Additionally, under NTFS, a user can add his or her own user-defined attributes to a file.

Removing limitations

First, NTFS has greatly increased the size of files and volumes so that they can now be up to 2^64 bytes (16 exabytes). NTFS has also returned to the FAT concept of clusters in order to avoid the HPFS problem of a fixed sector size. This was done because Windows NT is a portable operating system and different disk technology is likely to be encountered at some point. Therefore, 512 bytes per sector was viewed as having a large possibility of not always being a good fit for the allocation. This was accomplished by allowing the cluster to be defined as multiples of the hardware's natural allocation size. Finally, in NTFS all file names are Unicode based, and 8.3 file names are kept along with long file names.

NTFS and FAT performance and recoverability considerations

The NTFS is generally faster than FAT for disk reads and for fast recovery in case of a system failure. However, NTFS performs transaction logging on writes, resulting in a slightly slower write performance than FAT.

Under Windows NT a FAT is a careful-write file system. The FAT's careful-write file system only allows writes one at a time and alters its volume information after each write. This is a very secure form of writing. It is, however, also a very slow process. In order to improve performance on FAT you can opt to utilize the lazy-write file system feature, which uses the systems memory cache. This means that all writes are performed to this cache and the file system intelligently waits for the appropriate time to perform all the writes to disk.

This system gives the user faster access to the file system and prevents holdups due to slower disk access. It is also possible, if the same file is being modified more than once, that it may never actually be written to disk until the modifications are finished within the cache. Of course, this can also lead to lost data if the system crashes and unwritten modifications are still held in the cache.

NTFS provides the speed of a lazy-write file system along with additional recovery features. Each write request to an NTFS partition generates both redo and undo information in the transaction log. In the recovery process this log can assure that after only a few moments after a reboot that the file system's integrity is back to one hundred percent without the need of running a utility such as CHKDSK, which requires the scanning of an entire volume. The overhead associated with this recoverable file system is less than the type used by the careful-write file system.

Choosing a file system depends on your particular environment. Some of the factors for choosing a file system include MSDOS compatibility, file level and file system security, performance, and recoverability. In general, NTFS is best for use on logical volumes of about 400 MB or more. This is because performance does not degrade under NTFS, as it does under FAT, with larger volume sizes.

Users seeking highly scalable solutions will use software and hardware solutions in combination. For example, NTFS uses 64-bit addresses and file offsets. This allows for theoretically immense file and volume sizes. Today, there are external limitations on volume and file sizes imposed by the logical disk manager's disk partitioning system and by the underlying hardware. However, NTFS will continue to scale as these limitations are broken down.

8.2.7 Disk partitioning

Disk management in Windows 2000 and Windows NT is based on the principals of logically partitioning the available disk resources. There are two different types of partitions that can be allocated. The *primary* partition, which is usually used for the operating system; and the *extended* partitions for the additional logical drives.

Windows 2000 and NT, like many other operating systems, have many different disk configuration options. These include mirror sets, volume sets, and stripe sets.

All of the disk configuration in Windows NT is done by using the Disk Administrator. The Disk Administrator interface (see Figure 8-7 on page 269) displays a graphical representation of your physical and logical hard drives and CD-ROM drives. You can see at a glance the different partitions and their sizes, the volume names, the file systems in use, the drive letter assignments, and the amount of free space that is available for creating new partitions.

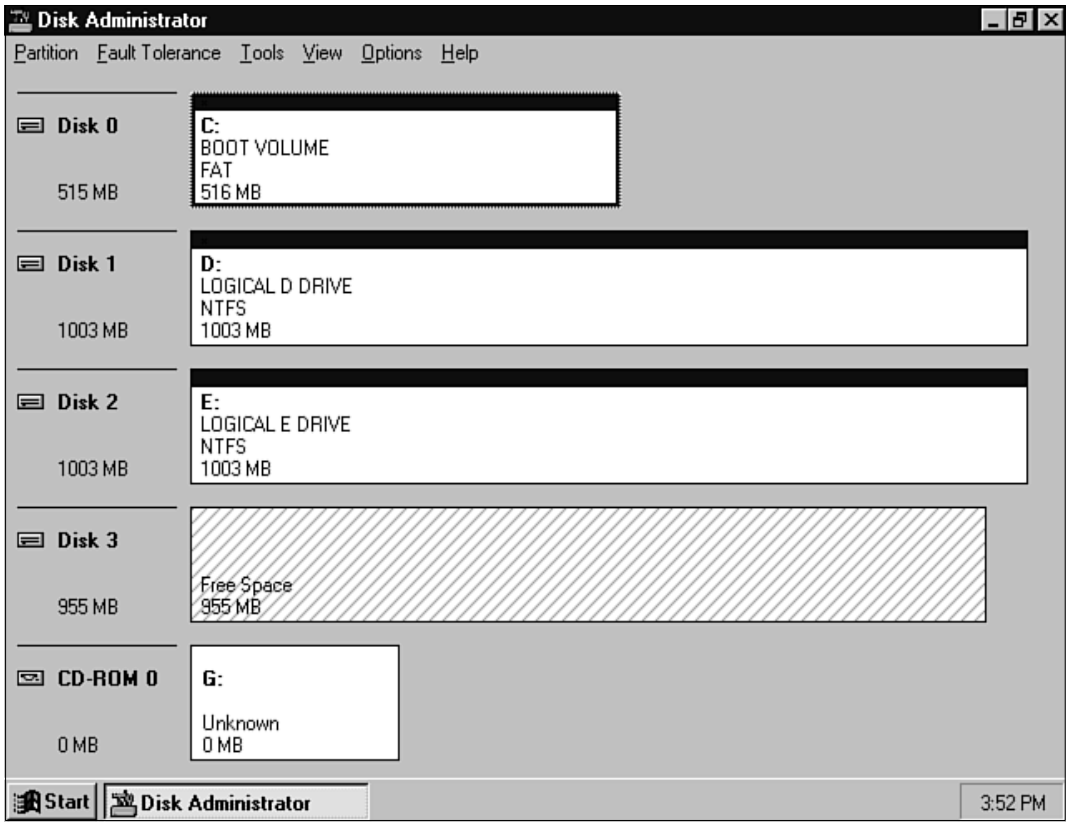


Figure 8-7 Windows NT Disk Administrator

The Disk Administrator gives you a graphical way to look at the partitioning schemes that you have implemented on these drives. It makes it that much easier to work with these partitions, or un-partitioned space, when being able to view the entire drive or set of drives. In addition to the bar chart view above, there is a more tabular view that contains further details about volume type and fault tolerance.

Mirror sets

With disk mirroring, partitions on two drives store identical information so that one is the mirror of the other. All data written to the partition on the primary disk is also written to the mirror, or secondary, partition. If one disk fails, the system is able to use the data from the other disk. The following facts apply to disk mirroring under Windows NT.

Mirrors are file system independent. Any partition using a file system that Windows NT recognizes or that is blank can be used to create a mirror. In addition, the mirrors are not dependent on disk geometry. The only requirement is that the free disk space used to place the mirror on must be equal to or greater than the size of the primary partition. Mirroring is not restricted to a partition of identical geometry (size, number of heads, cylinders, tracks, sectors, and so on), nor is it restricted to a drive of the same type (IDE, ESDI, SCSI, and so on).

Please note that only the Windows NT Server installation that created the mirror set will normally recognize it. However, Windows NT and other installations of Windows NT Server can recognize a mirror set created by the Windows NT Server by restoring disk configuration information (saved by the Disk Administrator on the originating server).

The fault tolerance driver makes the loss of one partition in a mirror set invisible. You will be able to read from and write to the remaining partition as if the mirror set were healthy.

However, if only one partition of a mirror set is functioning, then it is no longer fault tolerant. Loss of the remaining partition will result in an unrecoverable loss of all data in the mirror set.

Currently the ESS does not support having a Windows NT boot partition.

Volume sets

Both Windows NT Server and Windows NT Workstation can create and use volume sets. Volume sets more effectively use memory and drive letters under Windows NT by combining free disk space from one to 32 disks into a single volume with a single drive letter.

Volume sets provide no fault tolerance; if even one area of disk space in the set is lost, then all access to the data is lost. This may be the case if an adapter in the host system fails and the volume set is spread across multiple adapters. In all cases where volume sets are used, especially if they are across multiple SCSI host adapters, the IBM's Subsystem Device Driver (SDD) product should be used to provide the additional path availability.

Volume sets are transparent to the user. When a volume set is created, all areas of free space are assigned the same drive letter. Volume sets are the only Windows NT disk partition management option that allows more than one area of disk space in the set to reside on the same physical hard disk. Volume sets are the only Windows NT disk partition management option that allows the individual areas of disk space making up the volume to be of different sizes.

Volume sets must be created from free disk space—they cannot be used with existing partitions. To create a volume set, first select free space on one to 32 disks, then select **Create Volume Set** from the Partition menu in Disk Administrator. You must then shut down and restart the computer. When Windows NT restarts, Autochk.exe will run the equivalent of chkdsk /f on the entire volume set and the volume set will be created or extended. You can then format it for a file system.

Note: The chkdsk procedure must complete before the volume set will be accessible and may take many hours, depending on the volume size, directory, and file structure.

Stripe sets

While a stripe set without parity is a cost-efficient way to increase disk performance, as with volume sets, there is not fault tolerance, and the loss of one part of the stripe set leads to the loss of the entire stripe set. Again SDD can be used to provide additional availability and counteract this type of problem.

Because the ESS logical volume numbers (LUNs) are already striped with parity for RAID-5 and have fault tolerance for RAID-10, no parity striping should be used on the system software striping. Both RAID-10 and RAID-5 LUNs can be striped with a finer granularity for faster performance.

Do not create stripe sizes less than 32 KB for logical drives made up of arrays containing 18.2 GB drives on the ESS Model 800. For logical drives made up of arrays containing 36.4 GB or higher capacity drives, do not create a stripe size of less than 64 KB. This helps the load level at the rank array in the ESS. If many applications are addressing the same array, then software striping at the O/S level will not help much, but it also will not hurt the array performance.

ESS logical volume sizes

Using the ESS Specialist, you are able to create various sizes of logical volume that can be presented to the operating system. The maximum size can be up to the maximum capacity of

the disk array, depending on the size of physical disks that you have inside your ESS; however, the minimum disk size is 0.1 GB.

Since Windows 2000 and NT has some drive letter limitations, where possible, it is more efficient to create larger logical volumes. However, you can create smaller logical volumes, and combine them into one logical partition by utilizing some of the features provided by Windows 2000 and NT.

8.3 Tools for Windows 2000 and NT

In this chapter we discuss the available tools for the Windows 2000 and Windows NT users, how to implement them, how they can help you with the disk performance activities, and we show some examples of how to use them and get support.

The following tools are discussed in the following sections:

- Windows 2000 Performance console (Windows 2000 and Windows NT)
- Task Manager (Windows 2000 and Windows NT)
- IOMeter (Windows NT)

The NT Disk Administrator tool is also available for the Window users. This tool was discussed in 8.2.7, “Disk partitioning” on page 268.

8.4 Windows 2000 and NT Performance console

The Performance console is one of the most valuable monitoring tools available to Windows 2000 administrators. It is commonly used to monitor server performance and to isolate bottlenecks. The tool provides real-time information about server subsystem performance. The data collection interval can be adjusted based on your requirements.

The logging feature of the Performance console makes it possible to store, append, chart, export, and analyze data captured over time. Products such as SQL Server and Exchange provide additional monitors that allow the Performance console to extend its usefulness beyond the operating system level.

The Performance console includes two tools:

- System Monitor
- Performance Logs and Alerts

The main Performance console window is shown in Figure 8-8 on page 272.

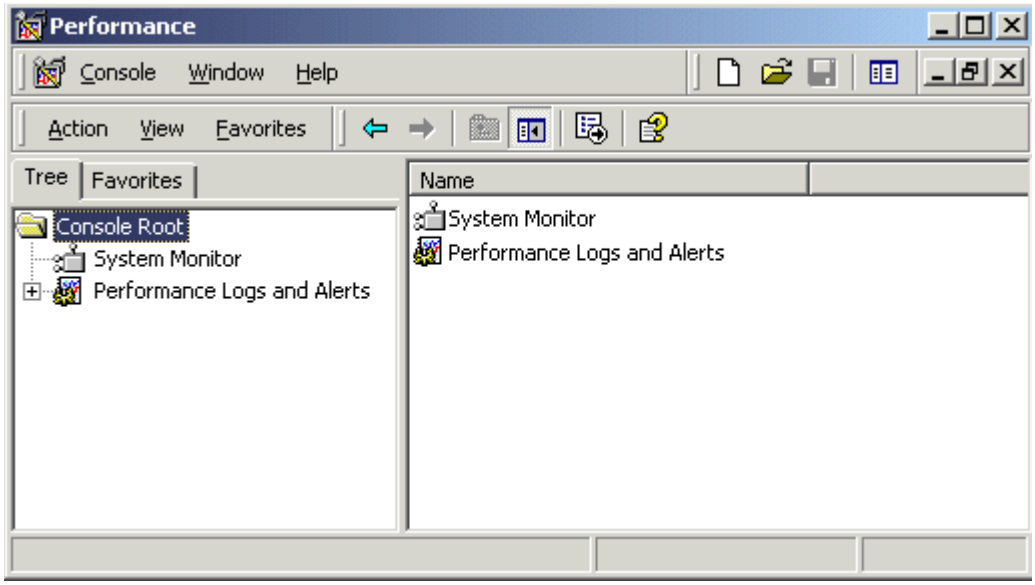


Figure 8-8 Main Performance console window

The Performance console is a snap-in for Microsoft Management Console (MMC). The Performance console is used to access the System Monitor and Performance Logs and Alerts tools.

The Performance console can be opened by clicking **Start -> Programs -> Administrative Tools -> Performance** or by typing `PERFMON` on the command line.

If there is no Administrative Tools folder within Programs, you can display it as follows:

1. Right-click a blank area of the task bar and click **Properties**.
2. Click **Advanced**.
3. Select the option **Display Administrative Tools** and click **OK**.

In Figure 8-9 on page 273 we see the System Monitor. The System Monitor can be used to view real-time or logged data of objects and counters. Performance Logs and Alerts can be used to log objects and counters, and create alerts.

Displaying the real-time data of objects and counters is sometimes not enough to identify server performance. Logged data can provide a better understanding of the server performance.

Alerts can be configured to notify the user or write the condition to the system event log based on thresholds.

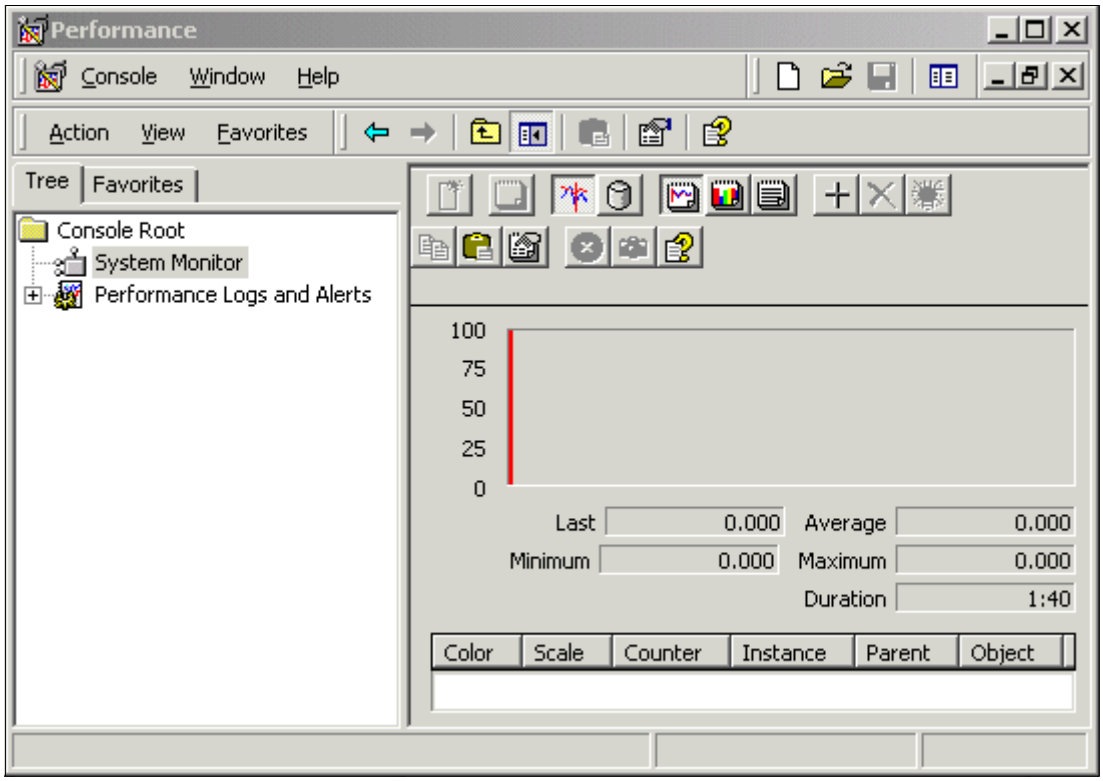


Figure 8-9 The Performance console: System Monitor

8.4.1 Key objects and counters

Performance Monitor provides the ability to monitor many aspects of your system; however, for our discussion we are interested mainly in I/O-related performance. Table 8-3 describes the different I/O-related statistics that can be reported on by Performance Monitor. Please note that some of the statistics require the disk counters that are provided by **`diskperf`**.

The key objects in Windows 2000 and NT are:

- Memory
- Processor
- Disk
- Network

Tuning these key objects will greatly improve the performance of disk I/O.

Table 8-3 Performance monitoring objects

Object	Counter	Description
Physical disk	Percent disk time	The percentage of time that a disk is busy. The general rule is that the total percent disk time for all logical disks should be less than 85 percent.

Object	Counter	Description
Physical disks	Average disk queue length	The number of requests for disk access. The general rule of thumb is that the total average disk queue length should be less than or equal to three. It may be important to note the actual number of spindles in a hardware RAID set and multiply the number of spindles by the average disk queue length.
Logical disks	Avg. disk sec/transfer	
Logical disks	Disk bytes/sec	
Logical disks	Current disk queue length	The current number of requests for access to the logical disk device.
Cache	Copy read hits	The percentage of requests found in the cache. Based on an analysis of the user environment the goal was to emulate approximately an 80 percent cache hit rate.
Memory	Cache bytes	The amount of the cache memory currently being used by the system. The maximum amount of RAM that the system can use for caching is 512 MB.
Memory	Cache bytes peak	Maximum number of bytes used by the cache at any given time.
Memory	Pages per second	The number of pages read from or written to disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory and must be retrieved from disk. A high level of paging activity can be acceptable (pages per sec > 500), but if it is associated with low available bytes a problem may exist.

Object	Counter	Description
Memory	Pool non-paged bytes	The number of bytes in the non-paged pool. An area of system memory (physical memory used by the operating system) for objects that cannot be written to disk, but must remain in physical memory as long as they are allocated. The system may be overloaded when this value is greater than 120 M or the sum of paged and non-paged pools is 256 MB. This counter displays the latest observed value only if it is not an average.
Memory	Pool paged bytes	The number of bytes in the paged pool. An area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. The system may be overloaded with a SAM size or a large number of user sessions when this value is greater than 156 MB or the sum of paged and non-paged pools is 256 MB. This counter displays the last observed value only if it is not an average.

8.4.2 Performance console output information

There are several situations in which you would use the Performance console:

- The Performance console can be run during activity-intensive periods to get a real picture of server performance. For example, you may want to measure your Windows NT domain controller performance early morning when user logins storm the server. The Performance console can effectively be used to monitor networking-related counters such as NetBEUI, NWLink, TCP/IP, network utilization, and others. This is important if you are monitoring how these protocols are affecting your network subsystem performance.
- The Performance console can send alerts when predefined threshold levels are reached. This is useful especially when you want to perform actions as soon as your pre-set threshold conditions are met.
- With the Performance console, you can view special counters provided by Microsoft BackOffice applications. For example, when you install Exchange Server, it installs additional object counters of its own. You can then monitor and analyze these counters and relate them to your Exchange Server's performance.

There are three ways to view the real-time or logged data counters:

- Chart

  This view displays performance counters in response to real-time changes or processes logged data to build a performance graph.

- Histogram

  This view displays bar graphics for performance counters in response to real-time changes or logged performance data. It is useful for displaying peak values of the counters.

- Report

  This view displays only numeric values of objects or counters. It can be used for displaying real-time activity or displaying logged data results. It is useful for displaying many counters.

8.4.3 Performance Logs and Alerts

The Performance Logs and Alerts window, shown in Figure 8-10, lets you collect performance data manually or automatically from local or remote systems. Saved data can be displayed in System Monitor or data can be exported to a spreadsheet or database.

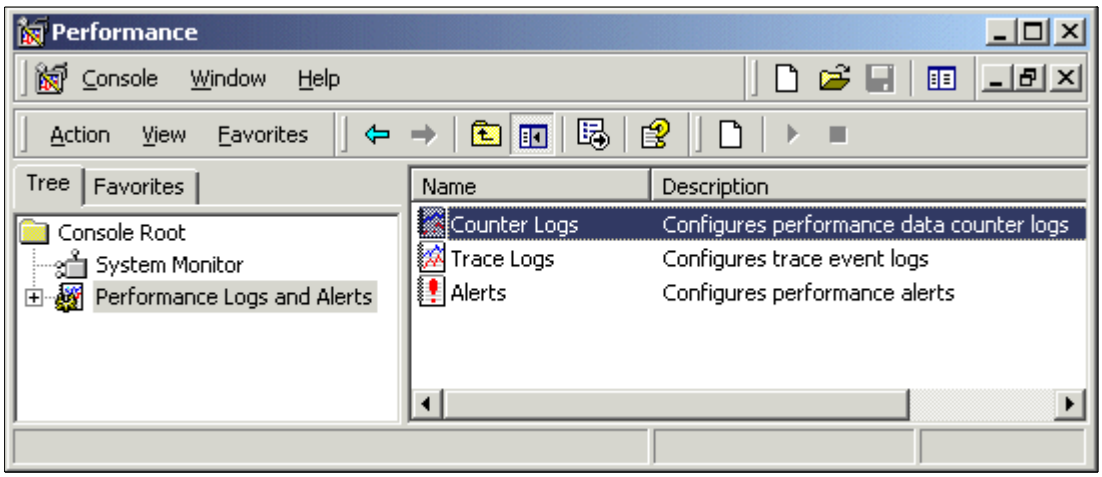


Figure 8-10 Performance Logs and Alerts

Performance Logs and Alerts provides the following functions:

- Counter logs

  This function lets you create a log file with specific objects and counters and their instances. Log files can be saved in different formats (file name + file number, or file name + file creation date) for use in System Monitor or for exporting to database or spreadsheet applications. You can schedule the logging of data, or the counter log can be started manually using program shortcuts. Counter log settings can also be saved in HTML format for use in a browser either locally or remotely via TCP/IP.

- Trace logs

  This function lets you create trace logs that contain trace data provider objects. Trace logs differ from counter logs in that they measure data continuously rather than at specific intervals. You can log operating system or application activity using event providers. There are two kinds of providers: System and non-system providers.

  For system providers, the following events are provided by the Windows 2000 kernel trace provider:

  – Process creations/deletions
  – Thread creations/deletions
  – Disk input/output
  – Network TCP/IP

- Page faults
- File details

Non-system providers can be application events. You can also code provider events. Any application or service installed in your system may support even providers. For example, in Active Directory, NetLogon and Local Security Authority are non-system event providers.

- Alerts

This function lets you track objects and counters to ensure they are within a specified range. If the counter's value is under or over the specified value, an alert is issued. Actions from an alert include:

- Send the alert to another machine.
- Log the alert in the application event log.
- Start a new counter log.
- Run a command from the command line.
- Alerts can be started and stopped automatically or manually.

8.4.4 Monitoring disk counters

In Windows 2000, there are two kinds of disk counters: Physical disk counters and logical disk counters. Physical counters are used to monitor single disks and hardware RAID arrays, and are enabled by default. Logical drive counters are used to monitor software RAID arrays, and are disabled by default.

Use the DISKPERF command to enable or disable these counters. Enter `DISKPERF -?` for help with this command.

Note: Physical drive counters should be used if the system is using hardware RAID, such as the ESS Model 800.

8.4.5 Monitoring disk performance

Windows NT by default does not collect any disk performance statistics because it uses system resources and obviously provides a performance impact on the system. On average the disk counters increase disk access time by approximately 0.5 percent.

To enable the disk counters you must be a member of the Administrator group.

1. Open the command prompt.
2. Do one of the following to enable the disk counters:
   - Type `DISKPERF -YE` for disks in RAID arrays.
   - For non fault-tolerant disks, type `DISKPERF -Y`.
3. Restart the computer.

To disable the collection of disk counters, follow the same procedure, replacing step 2 with: Type `DISKPERF -N`.

Note: To see if the disk counters are enabled on your computer, type `DISKPERF` without parameters at the command prompt.

In order to analyze the figures collected by enabling the disk counters we can use Performance Monitor. This is explained further in 8.4.7, "Performance reports" on page 280.

8.4.6 Disk bottlenecks

Windows 2000 retrieves programs and the data these programs use from disk. The disk subsystem can be the most important aspect of I/O performance, but problems can be hidden by other factors, such as lack of memory.

Performance console disk counters are available with either the LogicalDisk or PhysicalDisk objects:

- For non-ESS RAID disks, LogicalDisk monitors the operating system partitions of physical drives. It is useful to determine which partition is causing the disk activity, possibly indicating the application or service that is generating the requests. PhysicalDisk monitors the individual hard disk drives, and is useful for monitoring disk drives as a whole.
- For the ESS (all disks are RAID disks), LogicalDisk monitors the operating system partitions (if any), while PhysicalDisk monitors the logical drives created from the ESS Model 800 RAID arrays.

Finding disk bottlenecks is easier than finding processor and memory bottlenecks because the performance degradation is readily apparent.

Tip: When attempting to analyze disk performance bottlenecks, you should always use physical disk counters.

Activating disk performance counters

In Windows 2000, physical disk counters are enabled by default, but logical disk counters are disabled by default. If you use RAID software, you will need to enable logical disk counters using the command DISKPERF -YV.

Keeping these settings on all the time draws about 2–3 percent CPU. But if your CPU is not a bottleneck, this is irrelevant and can be ignored. Type `DISKPERF -?` for more help with the DISKPERF command, and just DISKPERF (no parameters) to get the current status.

Finding the bottlenecks

You can use the processor object counters (described in Table 8-4) in the Performance console to help you determine if you have disk bottlenecks. Then examine the indications of disk bottlenecks based on the object counter readings. Afterwards, you should perform appropriate actions to respond to the situation.

Table 8-4 Performance counters for detecting disk bottlenecks

Counter	Description
Physical Disk: Avg. Disk Queue Length	This is the average number of both read and write requests that were queued for the selected disk during the sample interval. If this value is consistently over 2 times the number of disks in the array (for example, 8 for a RAID-10, 4-disk array), it indicates that the operating system is waiting for the controlling hardware to write information to it. Consider a disk subsystem upgrade.
Physical Disk: Avg. Disk Bytes/Transfer	This is the average number of bytes transferred to or from the disk during write or read operations.The larger the transfer size, the more efficiently the system is running.
Physical Disk: Avg. Disk sec/Transfer	This is the time to complete a disk I/O. For optimal performance, this should be less then 15–18 ms. In general, this counter can be high when insufficient numbers of disks, slow disks, poor physical disk layout, or disk fragmentation occurs.

Counter	Description
Memory: Pages/sec	This is the number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. High value indicates disk activity due to insufficient memory. Add more RAM to your server. The product of this counter and Physical Disk: Avg. Disk sec/Transfer is an approximation of the amount of disk time spent on paging file activity during the sampling period. If it exceeds 0.1 (10 percent) then you may have excessive paging.

Tip: Do not use the % Disk Time physical disk counter. This is the percentage of elapsed time that the selected disk drive is busy servicing read or write requests. The counter is only useful with IDE drives, which, unlike SCSI disks, can only perform one I/O operation at a time. % Disk Time is derived by assuming the disk is 100 percent busy when it is processing an I/O and 0 percent busy when it is not. The counter is a running average of the 100 percent versus 0 percent count (binary).

Figure 8-11 shows a sample chart setting for finding disk bottlenecks.

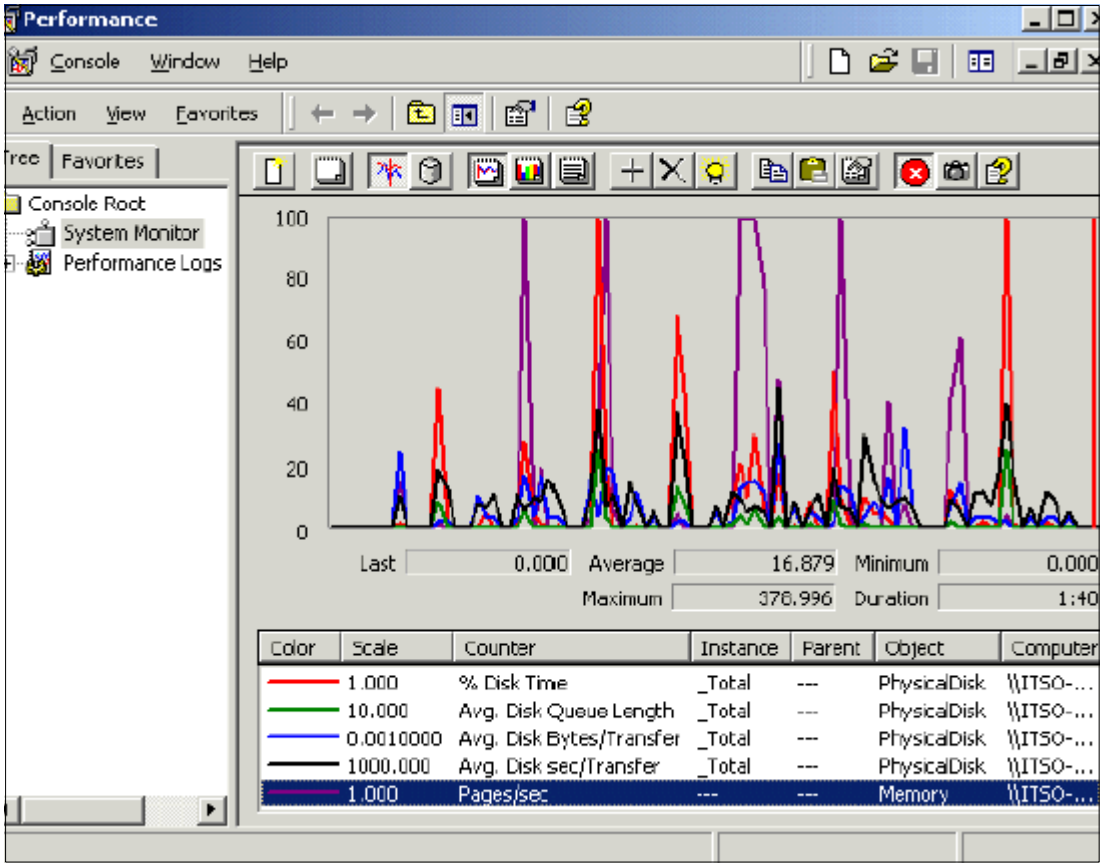


Figure 8-11 Chart setting for finding disk bottlenecks

Removing disk bottlenecks

Once a disk bottleneck has been detected, some of the actions that can be taken are the following:

- Use faster disks: 15,000 rpm arrays instead of 10,000 rpm arrays on the ESS Model 800. This will mean migrating the data to another LSS when incorporating the new disks.
- Offload processing to another system in the network (either users, applications, or services).
- Add more RAM. Adding memory will increase system memory disk cache, which in effect improves disk response times.
- Spread the I/O activity across the ESS arrays (logs, page file, etc.).
- Since the ESS arrays can be RAID-5 and RAID-10, swap RAID arrays from one to the other and take a new set of measurements. For example, if the databases workload activity is mostly sequential write operations then using RAID-5 can improve the performance. The following list could be followed.
  - Sequential writes work fine on RAID-5.
  - High I/O sequential reads work better on RAID-10.
  - High I/O random writes work better on RAID-10.
  - High I/O random reads work better on RAID-10.
  - Low I/O random read/writes can work as well on RAID-5 as on RAID-10.
- Correct the host O/S stripe size used to match the I/O transfer size.
- Use more ESS ranks.
- Use more adapter cards on the host to increase the I/O bandwidth your application is using.

8.4.7 Performance reports

The Windows NT Performance Monitor tool allows you to monitor the performance of your system by either displaying information on a real-time basis, or by collecting the data into a log file. The following sections explain how this can be set up.

Monitoring disk performance in real time

Monitoring disk activity in real-time will permit you to view current disk activity on local or remote disk drives. Only the current, and not the historical, level of disk activity is shown in the chart. If you wish to determine whether excessive disk activity on your system is slowing performance, log the disk activity of the desired disk(s) to a file, over a period of time that represents typical use of your network. View the logged data in a chart to see if disk activity is affecting system performance.

1. Enable the disk counters on the computer you wish to monitor, and restart the computer. To see if the disk counters are enabled on your computer, type `DISKPERF` without parameters at the command prompt.
2. Open Performance Monitor.
3. Create a chart for the following Performance Monitor counters:
   - Logical Disk: Avg. Disk sec/Transfer
   - Logical Disk: Disk Bytes/sec
   - Logical Disk: Current Disk Queue Length

Monitoring disk performance from collected data

Disk activity is best monitored using a log, since real-time monitoring provides a view of only the current disk activity, not the historical disk usage over an extended period of time. The information in the log can be viewed and evaluated in a chart at a later time.

1. Make sure disk counters are enabled. To see if the disk counters are enabled on your computer, type `DISKPERF` without parameters.
2. Open Performance Monitor.
3. Select the **View** menu, and select **Log**.
4. Select the **Edit** menu, and select **Add to Log**. The Add to Log dialog box appears.
5. (Optional) To log the disk performance of a remote computer:
   a. Click the **Browse** button. The Select Computer dialog box appears
   b. Select the desired remote computer, and click **OK**
6. From the Objects list box, select **Logical Disk**, and click **Add**. Please note that the Logical Disk object contains a number of different items that the system monitors.
7. Click **Done**.
8. Select the **Options** menu, and select **Log**. The Log Options dialog box appears.
9. Create the log file:
   a. Select the drive on which the log file will be saved from the Save in drop-down list box, and select the desired folder location.
   b. Type the name of the log file in the File name text box (use the file extension.LOG with the name of the log file).
10. From the Update Time group, select the log interval:
    a. To automatically log data, select the **Periodic Update** radio button, and type the desired time interval in the Interval (seconds) box.
    b. To manually update the log, select the **Manual Update** radio button.
11. Click **Start Log**.
12. To end logging disk activity, stop the log:
    a. Select the **Options** menu, and select **Log**. The Log Options dialog box appears
    b. Click **Stop Log**

Once you have collected the data, you then have to configure Performance Monitor so that the data can be viewed.

1. Select the **Options** menu, and select **Data From**. The Data From dialog box appears.
2. Choose the **Log File** radio button, and browse for the log file that you have created.

The log containing the performance data is now open. There are two ways that the data can be viewed, either as a chart or a text-based report. To choose the method of report you want to see, click the **View** menu and then select either **Chart** or **Report**.

Once you have chosen how you want to view the report, you need to add to the report or chart the specific items of interest to you. The selection is done by clicking the '**+**' icon, or selecting from the Edit menu, **Add to chart/report**.

A new dialog box is displayed containing the options available for the original objects that you chose to monitor. Select the relevant items you want to view and click **Add**, and they will appear in the chart/report.

8.5 Task Manager

In addition to the Performance console, Windows 2000 also includes Task Manager, a small utility that allows you to view the status of processes and applications and gives you some real-time information about memory usage.

The Windows 2000 Resource Kit also contains INTFILTR, which is an interrupt binding tool that allows you to bind device interrupts to specific processors on SMP servers. This is a useful technique for maximizing performance, scaling, and partitioning of large servers. It can provide a network performance increase of up to 20 percent.

8.5.1 Starting Task Manager

You can run Task Manager using any one of the following three methods:

- Right-click a blank area of the task bar and select **Task Manager**.
- Press Ctrl + Alt + Delete and click the **Task Manager** button.
- Press Ctrl + Shift + Esc.

Figure 8-12 shows that Task Manager has three views: Applications, Processes, and Performance. The latter two are of interest to us in this discussion.

Processes tab

In this view (see Figure 8-12) you can see the resources being consumed by each of the processes currently running. You can click the column headings to change the sort order which will be based on that column.

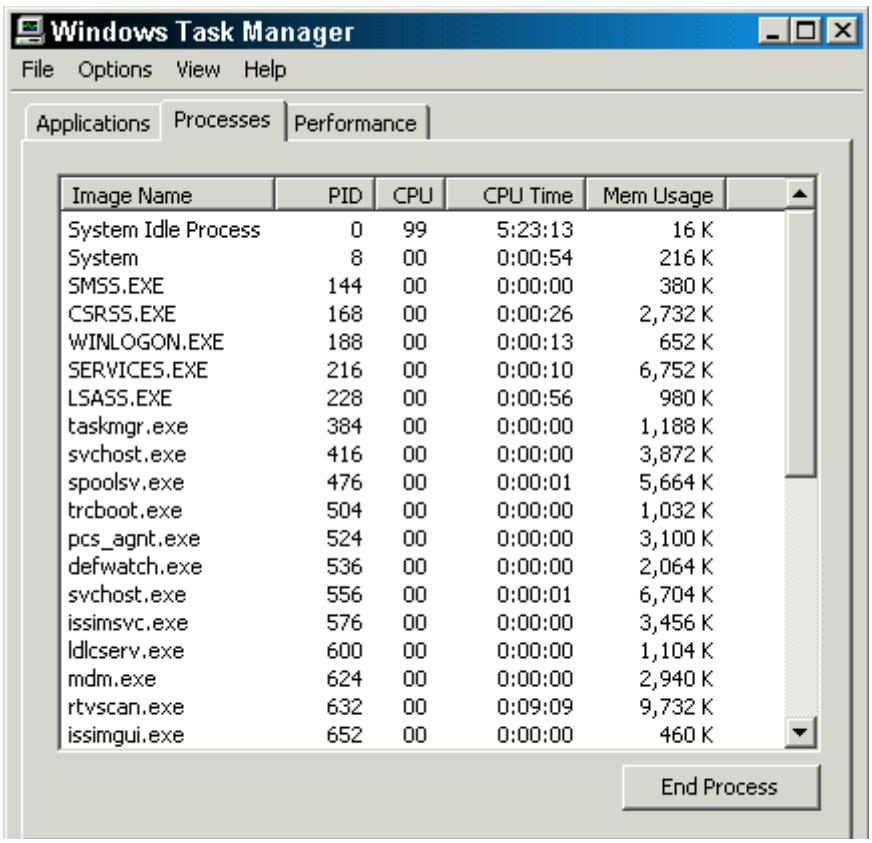


Figure 8-12 Windows Task Manager - Processes tab

Click **View -> Select Columns**. This displays the window shown in Figure 8-13, from which you can select additional data to be displayed for each process.

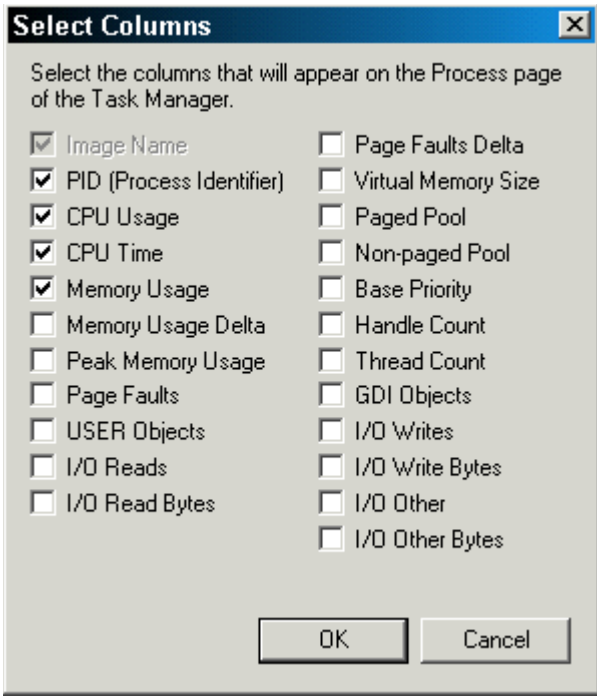


Figure 8-13 Select columns for the Processes view

Table 8-5 shows the columns available in the Windows 2000 operating system that are related to disk I/O.

Table 8-5 Task Manager - Disk-related columns

Column	Description
Paged Pool	The paged pool (user memory) usage of each process. The paged pool is virtual memory available to be paged to disk. It includes all of the user memory and a portion of the system memory.
Non-Paged Pool	The amount of memory reserved as system memory and not pageable for this process.
Base Priority	The process's base priority level (low/normal/high). You can change the process's base priority by right-clicking it and selecting Set Priority. This remains in effect until the process stops.
I/O Reads	The number of read input/output (file, network, and disk device) operations generated by the process.
I/O Read Bytes	The number of bytes read in input/output (file, network, and disk device) operations generated by the process.
I/O Writes	The number of write input/output operations (file, network, and disk device) generated by the process.
I/O Write Bytes	The number of bytes written in input/output operations (file, network, and device) generated by the process.
I/O Other	The number of input/output operations generated by the process that are neither reads nor writes (for example, a control type of operation).

Column	Description
I/O Other Bytes	The number of bytes transferred in input/output operations generated by the process that are neither reads nor writes (for example, a control type of operation).

Performance tab

The Performance view shows you performance indicators, as shown in Figure 8-14.

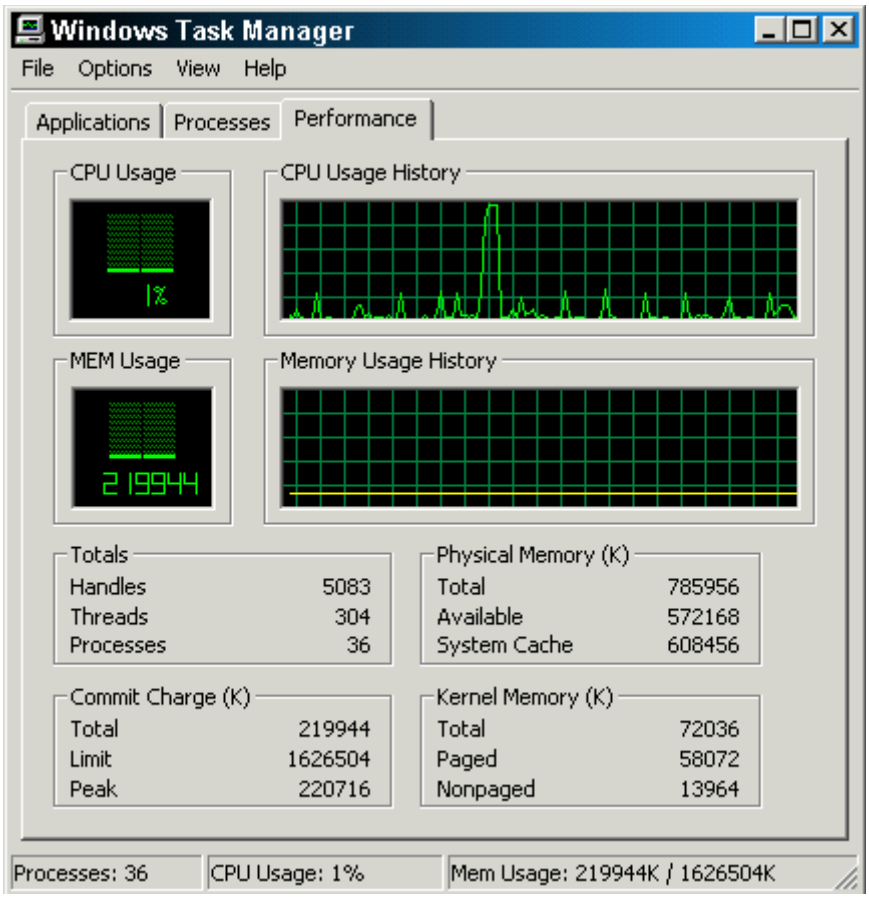


Figure 8-14 Task Manager - Performance view

The charts show you the CPU and memory usage of the system as a whole. The bar charts on the left show the instantaneous values, and the line graphs on the right show the history since Task Manager was started.

The four sets of numbers under the charts are as follows:

- Totals
  - Handles: Current total handles of the system
  - Threads: Current total threads of the system
  - Processes: Current total processes of the system
- Physical Memory (K)
  - Total: Total RAM installed (in KB)
  - Available: Total RAM available to processes (in KB)
  - File Cache: Total RAM released to the file cache on demand (in KB)
- Commit Charge (K)

  - Total: Total amount of virtual memory in use by all processes (in KB)
  - Limit: Total amount of virtual memory (in KB) that can be committed to all processes without adjusting the size of the paging file
  - Peak: Maximum virtual memory used in the session (in KB)
- Kernel Memory (K)
  - Total: Sum of paged and non-paged kernel memory (in KB)
  - Paged: Size of paged pool allocated to the operating system (in KB)
  - Non-paged: Size of non-paged pool allocated to the operating system (in KB)

Refer to the Windows 2000 Resource Kit for more information on this topic.

8.6 Iometer

Iometer is an I/O subsystem measurement and characterization tool for single and clustered NT systems. Iometer is owned by Intel Corporation. For information about Iometer go to:

`http://developer.intel.com/design/servers/devtools/iometer/`

Iometer is both a workload generator (it performs I/O operations in order to stress the system) and a measurement tool (it examines and records the performance of its I/O operations and their impact on the system). It can be configured to emulate the disk or network I/O load of any program or benchmark, or can be used to generate entirely synthetic I/O loads. It can generate and measure loads on single or multiple (networked) systems.

Iometer can be used for measurement and characterization of:

- Performance of disk and network controllers
- Bandwidth and latency capabilities of buses
- Network throughput to attached drives
- Shared bus performance
- System-level hard drive performance
- System-level network performance

Iometer consists of two programs, Iometer and Dynamo.

Iometer is the controlling program. Using Iometer's graphical user interface, you configure the workload, set operating parameters, and start and stop tests. Iometer tells Dynamo what to do, collects the resulting data, and summarizes the results in output files. Only one copy of Iometer should be running at a time. It is typically run on the server machine.

Dynamo is the workload generator. It has no user interface. At Iometer's command, Dynamo performs I/O operations and records performance information, then returns the data to Iometer. There can be more than one copy of Dynamo running at a time. Typically one copy runs on the server machine and one additional copy runs on each client machine. Dynamo is multi threaded. Each copy can simulate the workload of multiple client programs. Each running copy of Dynamo is called a manager. Each thread within a copy of Dynamo is called a worker.

8.7 Performance configuration options

In many environments, Windows NT has been chosen as the standard operating system for the desktop users. Typically these workstations will have local applications installed that access shared resources such as file servers, application and database servers, mail servers, etc.

These types of environments rely heavily on the performance of the network and the performance of the server where the resource resides. For the discussion in this section, we assume that the network has enough bandwidth to cater for the demand placed on it. Remember that there are various different registry entries associated with network tuning. We do not talk about them here, but they can help to directly improve a server's throughput.

8.8 General considerations for Windows servers

In this section we discuss the general considerations for improving the disk-based I/O throughput of either a file server or database server. First there are some general guidelines that can be followed:

- Microsoft does not recommend that large dedicated file servers or database servers be configured as backup domain controllers (BDC). This is due to the overhead associated with the netlogon service.
- Carefully evaluate the installed Windows NT services to determine whether they are needed for your environment or can be provided for by another server. Consider stopping, manually starting, or disabling the following services: Alerter, Clipbook Server, Computer Browser, Messenger, Network DDE, OLE Schedule, and spooler.
- Set up the Windows NT tasking relative to your workload types. More often than not, many applications, such as SQL Server, run as background tasks and therefore setting the background and foreground tasks to run equally can be of benefit.
- You can improve the performance of file caching by optimizing the server service for the file and print server.
- Format the logical volumes with 64 KB allocations. Setting the allocation size to 64 KB improves the efficiency of the NTFS file system by reducing fragmentation of the file system and reducing the number of allocation units required for large file allocations. This is accomplished through the following command line entry: `format x: /A:64 K /fs:ntfs`.
- Setting the NTFS log file to 64 MB reduces the frequency of the NTFS log file expansion. Log file expansion is costly because it locks the volume for the duration of the log file expansion operation. This is accomplished through the following command line entry: `chkdsk x: /L:65536`.

8.9 Windows NT registry options

This section details changes that can be made to the Windows NT registry.

Warning: Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system.

For information about how to edit the registry, view the "Changing Keys and Values" Help topic in Registry Editor (Regedit.exe) or the "Add and Delete Information in the Registry" and "Edit Registry Data" Help topics in Regedt32.exe. Note that you should back up the registry before you edit it. If you are running Windows NT, you should also update your Emergency Repair Disk.

8.9.1 Speed up file system caching

If you have some extra RAM and an active file system, you can speed up file system activity by increasing the IoPageLockLimit from the default 512 K bytes to 4096 K bytes or more

(refer to Table 8-6). This entry is the maximum number of bytes that can be locked for I/O operations. When the value is 0, the system defaults to 512 K. The largest value is based on the amount of memory in your system.

Table 8-6 NT registry IoPageLockLimit settings

RAM	IoPageLockLimit
32 - 64	4096000 (4 MB)
128 - 156	16384000 (16 MB)
> 512	32768000 (32 MB)

Before making changes, get a baseline for your system by using performance monitor for a representative period of time. Example 8-1 should be used only as a guide. Make your changes in small increments and measure performance for another representative period after each change.

Example 8-1 Registry settings: Key

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management]
Value Name: IoPageLockLimit
Data Type: REG_DWORD ( Number of Bytes )
Data: 3e8000 ( hex )   4096000 (decimal)
```

8.9.2 Improve memory utilization of file system cache

This registry entry allows the system to improve memory utilization for the file system cache and allows for more files to be opened simultaneously on a large system. It can, however, utilize additional memory from the page pool.

The following steps can be used to add this registry key:

1. Start Registry Editor (Regedit.exe). Locate and select the following Registry subkey in the HKEY_LOCAL_MACHINE subtree:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory
   Management
   ```

2. From the Edit menu, click **Add Value**.
3. Enter `UnusedFileCache` as the value name and set the Data Type to REG_DWORD.
4. Click **OK** and enter a value of 0 or 5–40 as the data.
   - 0: Default behavior similar to Windows NT 4.0 Service Pack 3.
   - 5–40: Trim unused file cache based on pool usage.
     - The value being set represents the percent of pool that we allow to be consumed by unused segments, where 5 is most aggressive (for example, it increases the size of the cache the least), and 40 is least aggressive (for example, it lets the cache grow to the largest before trimming).
     - Testing performed by Microsoft found that this registry change has positive benefits in that it also increases the performance of some applications such as IIS. It works best when set to 15–20. Do not choose a value greater than 20 without extensive stress testing.
5. Click **OK**, quit Registry Editor, and then shut down and restart the computer.

8.10 Subsystem Device Driver (SDD)

The IBM Subsystem Device Driver (SDD) provides path failover/failback processing for the Windows server attaching the IBM TotalStorage Enterprise Storage Server Model 800.

It also provides I/O load balancing. For each I/O request, SDD dynamically selects one of the available paths to balance the load across all possible paths.

To receive the benefits of path balancing, ensure that the disk drive subsystem is configured so that there are multiple paths to each LUN. Doing this not only will enable performance benefits from the SDD path balancing, but also prevent loss of access to data in the event of a path failure.

The Subsystem Device Driver is discussed in further detail in 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149.

8.11 Novell NetWare monitoring tools

By default, NetWare 6 parameters are already optimized and are dynamically managed by the system itself (see 8.15, “NetWare dynamically configured parameters” on page 300).

The user can optionally change the parameters using the following tools:

- NetWare Remote Manager (NRM)
- Monitor
- VTune

All of these tools have graphical interfaces. NRM and Monitor were developed by Novell, while VTune uses a performance optimization package available from Intel.

The following sections describe these three different tools, which may be used in conjunction with each other to monitor performance metrics and adjust parameters when tuning a NetWare 6 server. They may also be used as diagnostics aids when troubleshooting server problems.

NetWare 5.1 and 6 users are able to use the information discussed in this chapter since, with the exception of NRM, the descriptions in this chapter apply to both 5.x and 6 versions of NetWare. In the case of NRM, only the version available in NetWare 6 is covered since it covers a much richer function set than its predecessor.

Note: As we discuss the features found in these tools, we recommend that you have access to a system that is running them so that you can quickly gain familiarity with their operation.

8.12 NetWare Remote Manager

NetWare Remote Manager (NRM) allows administrators to monitor a server’s health, change its configuration, or perform diagnostic and debugging tasks. It provides most of the functionality of Monitor, along with some functionality from other utilities available at the server console.

The first version of this tool was made available in NetWare 5.1 under the name NetWare Management Portal.

Using a standard Web browser, NRM can be used to manage NetWare servers from any workstation running Netscape 4.5 or later, or Internet Explorer 5 or later. The following list highlights the major tasks that can be performed with NRM:

- You can monitor and manage a server's health by:
  - Monitoring the health status of one or more servers
  - Building a group of servers, allowing them to be monitored together
  - Accessing NDS eDirectory health and troubleshooting tools
  - Tracking health statistics values
- Modify a server's configuration by:
  - Managing disk partitions
  - Viewing information about all hardware adapters, hardware resources, and processor data
  - Uploading and replacing NLM programs, LAN drivers, or disk drivers
  - Monitoring system disk space and memory resources
  - Accessing files on volumes and DOS partitions
  - Managing server connections
  - Configuring Set parameters
  - Setting schedules for running console commands
  - Shutting down, restarting, or resetting a server
- Troubleshoot server problems by:
  - Viewing recorded logs for server health parameters
  - Finding CPU hogs
  - Finding high memory users
  - Tracing abend sources
  - Locating server process hogs
  - Identifying disk space hogs
  - Viewing open file users

As you can see by the above list there are many areas you can focus on with this tool. We are going to focus on disk issues and performance.

8.12.1 Accessing NRM

NRM can be used either as a console tool or from a remote workstation using a Web browser. The console version of NRM is Java-based, and is demanding on both memory and processor cycles.

In contrast, the only additional overhead caused by monitoring a server remotely is the increase in network traffic. If network access is not a problem, we strongly recommend running the tool on a remote workstation rather than the server.

Local access

NetWare Remote Manager is available in the Utilities menu when running the Java-based X Server Graphical Console locally on the server. By default, the Graphical Console is always installed and started when booting the server.

If the Graphical Console is not running for some reason, it can be started by entering the command **`startx`** at the console window.

To start NRM, click **Novell -> Utilities -> NetWare Remote Manager** (see Figure 8-15 on page 290).

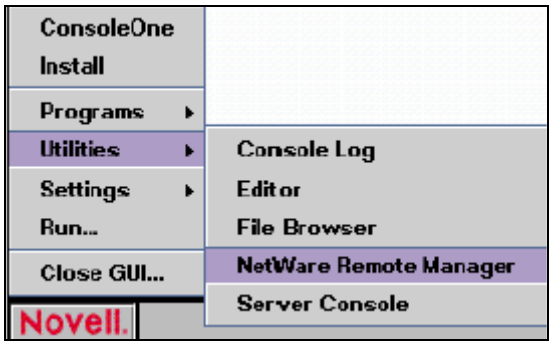


Figure 8-15 Accessing NetWare Remote Manager from the server Graphical Console

Remote access

Using a Web browser, connect to your NetWare server using either its IP address or host name. You will see the Welcome page, as shown in Figure 8-16. Click **Open Remote Manager** from the left-hand pane.

Tip: You can also access NRM directly using TCP/IP port 8008. After login, the connection is redirected to 8009, which is a secure port.

Figure 8-16 Starting NetWare Remote Manager from the NetWare Welcome page

Note: The remotely executed version of NRM offers the same interface and capabilities as the local version. In the following examples in this section, the screen captures have been taken from a browser running on a workstation, but all descriptions and options described apply to both versions of NRM.

NRM main page

The NetWare Remote Manager's main page is shown in Figure 8-17.

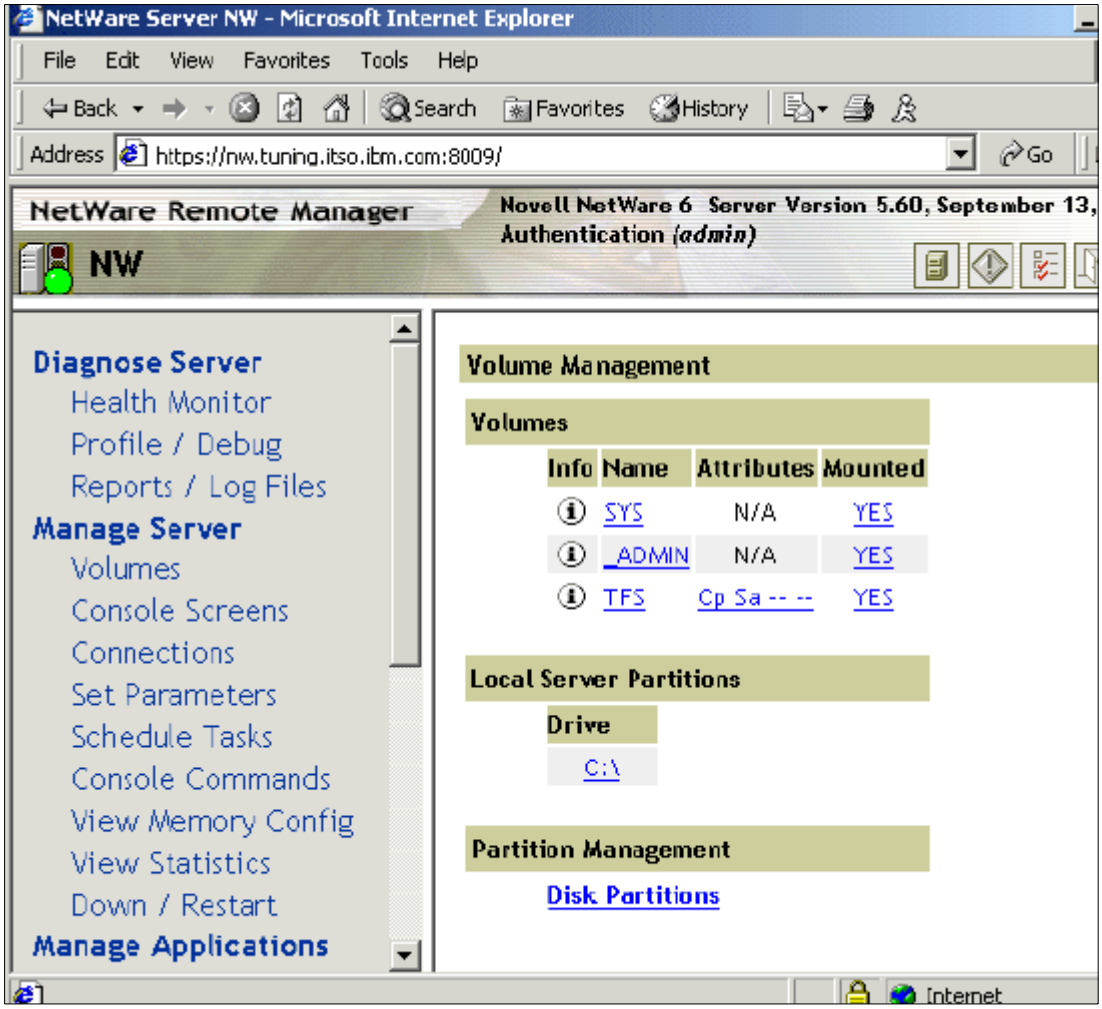


Figure 8-17 NetWare Remote Manager default window

As can be seen in from the example in Figure 8-17, NRM windows are divided into three frames:

- The header frame (top)

  The header frame contains general information about the server and quick links to access the Volumes, Health Monitor, and Configuration pages; it also contains an Exit link to close the browser window.

- The navigation frame (left)

  The navigation frame lists general tasks that can be performed using NRM. In addition, it displays links to specific pages for performing those tasks. Some of the links in the navigation frame change depending on the NLM programs that are loaded on the server.

- The main content frame (right)

  This frame displays information related to the link selected in the header frame or the navigation frame.

8.12.2 Volumes link

The Volume link is the default window displayed when you log into NRM (see Figure 8-17 on page 291). This page may also be accessed by clicking the **Volumes** link either in the header frame or under Manage Server in the navigation frame.

The Volume Management page provides a list of a server's volumes, access to its DOS partitions and other mounted drives, and the ability to perform partition management tasks. It allows you to track disk usage and have access to partition information for all drives.

Clicking the **Disk Partitions** link loads a page displaying the current configuration for the storage devices attached to the Novell NetWare server. You can change the configuration by clicking the different links available (see Figure 8-18).

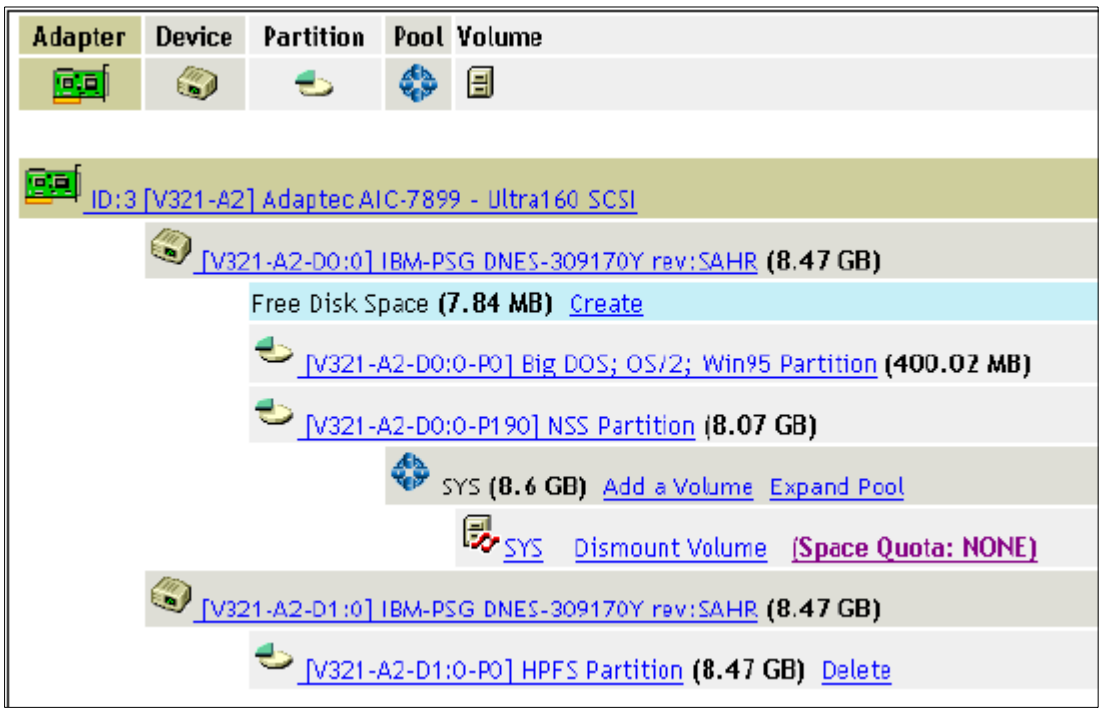


Figure 8-18 NetWare server disk partition information

The system administrator can create or expand a partition or a pool and mirror a partition. It is important to remember that changing file system options may affect the volume's performance. We have included a section about optimizing the Novell Storages Services file system for performance (see 8.16, "Novell Storage Services file system" on page 301).

A graphical view of the disk is available from the Health Monitor page by clicking the **Available Disk Space** link in the Description column.

8.12.3 Disk/LAN adapters link

Click the **Disk/LAN Adapters** link to display a view of all the storage and network adapters that are installed in the server, including information about the slot numbers in which each adapter is located (see Figure 8-19).

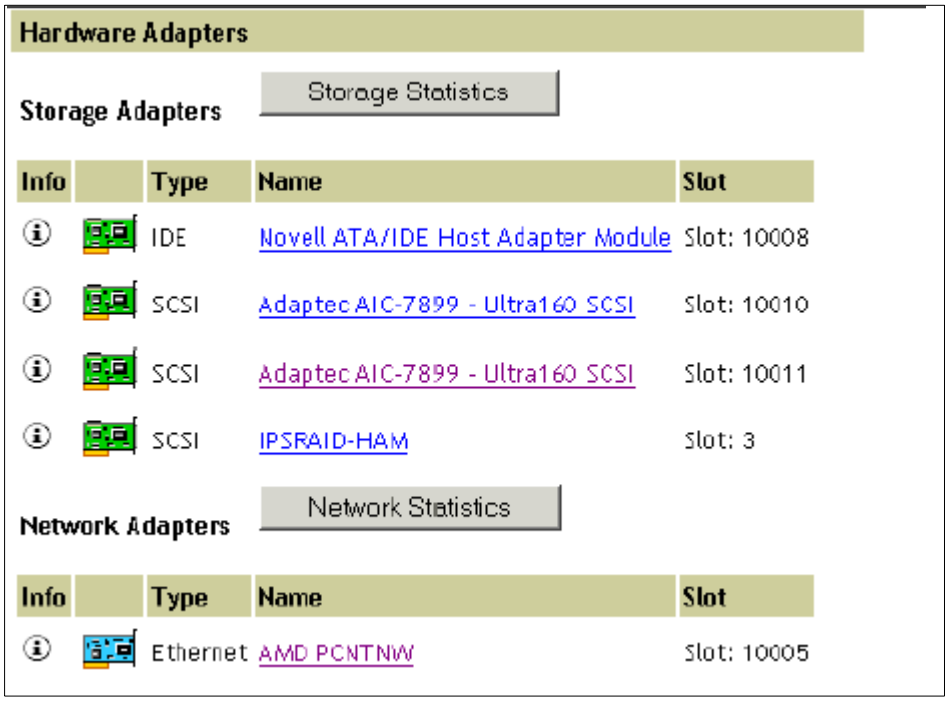


Figure 8-19 NetWare system Hardware Adapters

This page also has the Storage Statistics and Network Statistics buttons.

You can click the information symbol or the adapter icon to access data about an adapter (if you see a *not* symbol, the driver is not active). The device name, device type, capacity, Target ID for SCSI devices, Logical Unit Number, and the object identification are displayed, as shown in Figure 8-20.

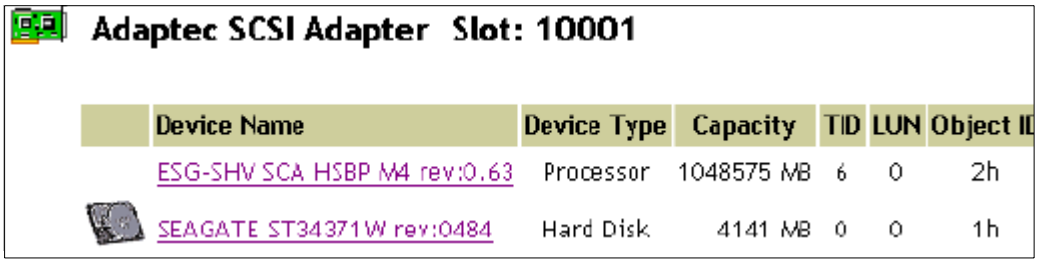


Figure 8-20 Viewing information about a particular storage adapter

Clicking the device name (for a disk) displays the partition information associated with this device, as shown in Figure 8-21 on page 294.

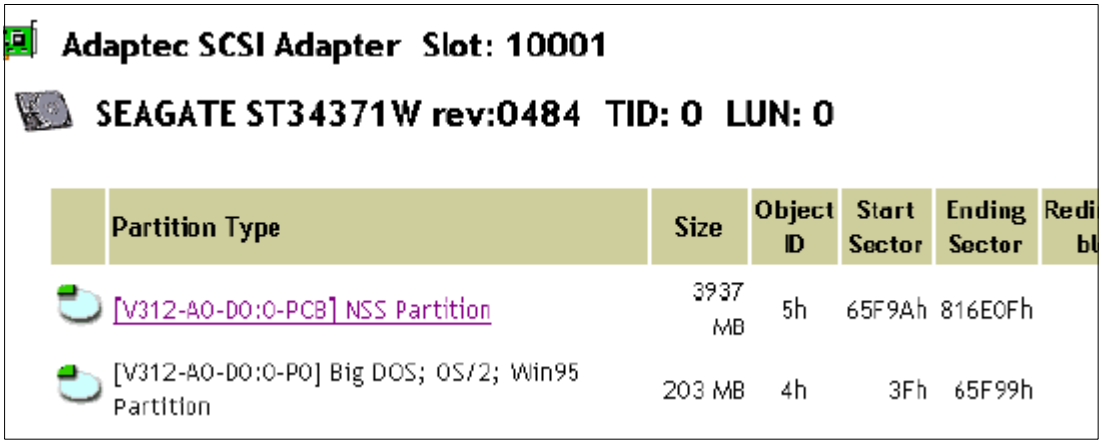


Figure 8-21 Details of a storage device with NSS and DOS partitions

Here you find information such as partition types and sizes, object IDs, starting and ending sectors, number of redirected blocks, which blocks are being used, and whether or not this partition is mirrored. Selecting the partition names shows the volume segments.

Clicking a network adapter icon displays the board number, the frame type, and the protocols used.

NRM is an all-in-one tool, and because of that it contains much information and many options, which may be overwhelming at first. If you are managing one or many NetWare 6 servers, it is worth taking the time to learn how to use this effective tool.

8.13 Monitor

Monitor is a tool you can access using the server console; it is easy to use and provides access to a lot of information. It is particularly useful if you want to access information quickly or, perhaps, monitor a parameter without the overhead of accessing the server through a Web browser or a relatively slow Java-based application.

The Monitor program, as you can see in Figure 8-22 on page 295, has a full-screen, character-based interface.

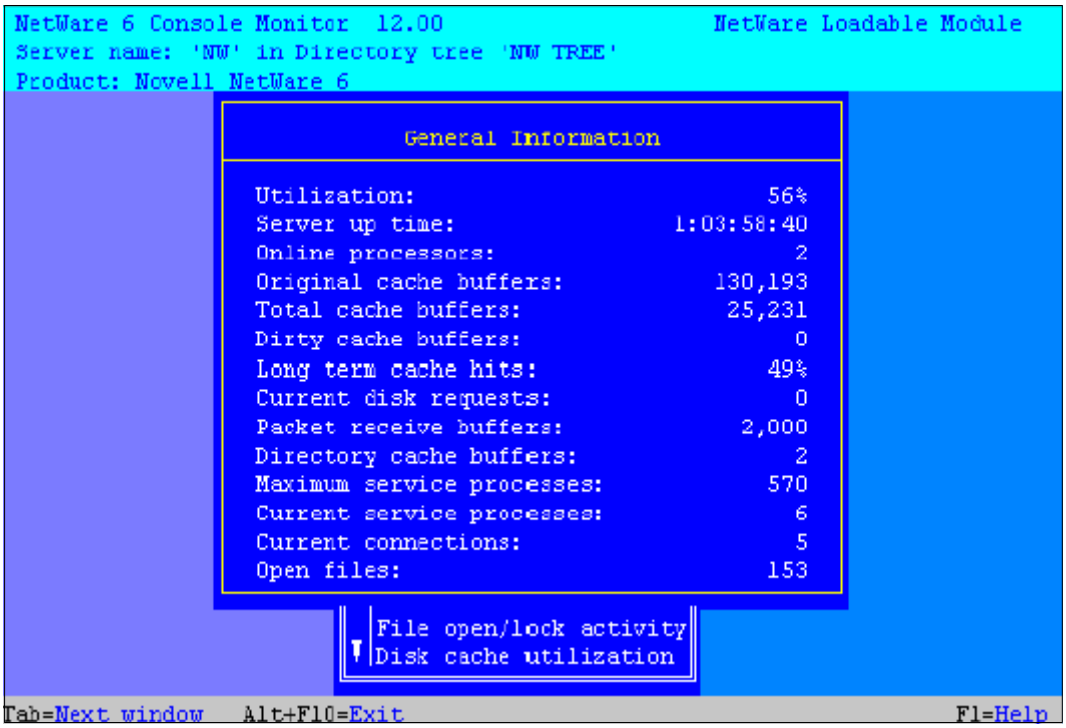


Figure 8-22 Monitor main screen

To load Monitor at the System Console prompt, enter the command MONITOR.

If Monitor is already running, the screen will switch to the open Monitor. The program starts by displaying the status of general parameters, as shown in Figure 8-22.

To exit Monitor, use either of the following procedures:

- Repeatedly press Esc until the Exit confirmation box appears. Then press Enter to return to the System Console prompt.
- To bring up the Exit confirmation box immediately, press Alt + F10. Then press Enter to return to the System Console prompt.

There are essentially two areas in the Monitor screen (see Figure 8-22). One displays the requested information. The other one, in the bottom part of the window, where other information is accessed by scrolling up and down. A nice feature of this tool is that the information window is resized to display all of its information (assuming that it fits in one screen).

Using Monitor

With the following few keys, you can navigate your way around Monitor:

- Tab key: Switch between the Information and the Options windows.
- Up and down arrow keys: Move up and down inside a window.
- Enter: Select the highlighted option.
- Esc: Return to the previous list or, if at the top level, display an exit dialog.

General Information window

The volume of information and statistics available for a NetWare server can be overwhelming. For quick access to the most important performance indicators that can be used to track,

diagnose, and resolve server problems, you can display the General Information window (illustrated in Figure 8-22 on page 295).

The following list describes the available performance indicators:

- Utilization

  This is the average of the server's total processing capacity used during the last one second, expressed as a percentage. The remainder of the time is spent in an idle loop. This value represents the average utilization of all running processors on a multi-processor server.

- Server up time

  This is the amount of time that has elapsed since the server was last started. If your server has Auto Restart enabled, you can use this field to determine whether your server has abended and restarted. You can also use this information to detect power failures or other instances of server failure.

- Online Processors

  This is the number of enabled and online processors. The Platform Support Module (PSM), loaded from STARTUP.NCF, enables NetWare to use secondary processors in a multiprocessing platform. The default is to autostart all processors.

- Total Cache Buffers

  This entry tells you how many cache buffers are currently available for file caching. The number decreases as you load NLM programs. File caching has a dramatic impact on server performance, so you want this number to be as high as possible.

- Dirty Cache Buffers

  This is the number of buffers that contain data that has been changed but not yet written to disk. If you are using the Traditional File System (TFS), an increase in this number may indicate that there is a bottleneck that should be addressed.

- Long Term Cache Hits

  This number shows the cumulative percentage of requests for disk blocks that were already in cache for TFS. When the requested data is already in memory, disk reads do not need to be performed, which is good for your server's performance. Use this percentage to assess overall disk cache utilization if you are using the traditional file system. If this value falls and server performance degrades, you should assess whether increasing the amount of physical memory in the machine would improve the situation.

- Current Disk Requests

  This indicates how many read requests are pending. Increases in this measurement may indicate a disk subsystem bottleneck.

- Packet Receive Buffers

  This measurement tells you how many buffers are allocated for holding client requests while the server processes them. The server allocates more packet receive buffers as needed (within minimum and maximum values). Buffer size depends on the network adapter. This number is the result of the interaction of three configurable parameters. You should monitor this number and take corrective action if it reaches its maximum value.

- Maximum Service Processes

  This indicates how many task handlers the server can allocate to service client requests. As the number of client requests increases, the server creates more service processes. When no more service processes can be allocated, server performance is adversely affected. You should monitor the current number of service processes (see the next parameter) to ensure that it does not reach this maximum value.

- Current Service Processes

  This measurement tells you how many task handlers are currently being used to service client requests. You should use this number to determine whether the server has enough free task handlers to service client requests. If the number of current service processes reaches the maximum (see the previous parameter), the value of the Maximum Service Processes (Communications category) parameter must be increased.

- Current Connections

  This shows you the number of active connections, whether licensed or unlicensed, authenticated or not-logged-in.

- Open Files

  This shows you the number of files currently being accessed by the server and by other clients. Certain files are always open, such as the hidden files that support eDirectory.

8.14 VTune

VTune is an NLM that generates trace files for analysis by the Intel tool of the same name. The Intel VTune product is a tool developed by the CPU manufacturer to perform processor-based profiling.

The VTune NLM can be downloaded from:

`http://developer.novell.com/support/sample/tids/topt2/topt2.htm`

You can obtain a 30-day trial version of the latest VTune client software from Intel at:

`http://developer.intel.com/software/products/eval/`

This site also offers a slide show to help you to learn to use the tool.

VTune allows you to track things such as where the processor is spending its time, where misaligned memory references occur, or where branch mis-predictions happen. VTune is a utility that runs under WIN95, NT, and Windows 2000.

Novell has developed the VTUNE NLM to act as a data collector application over a user-defined interval. To generate a trace file, simply load the VTune NLM by entering the command at the console prompt: `[LOAD] VTUNE`.

You then select the event to be profiled. Next, select the processor to profile; you can choose one in particular or all of them. Finally, you need to select the sampling and file parameters (see Figure 8-23).

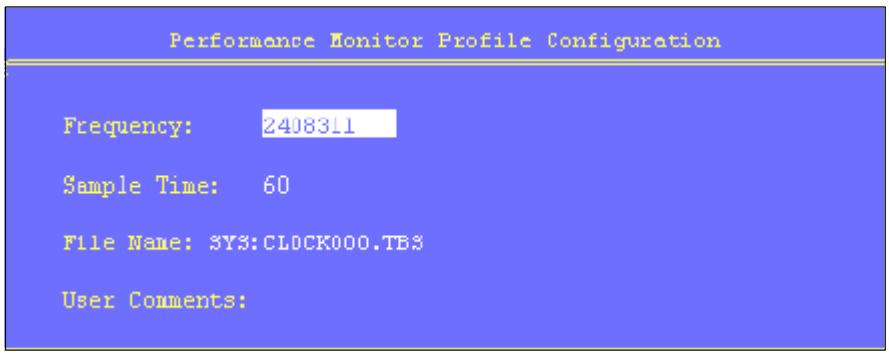


Figure 8-23 VTune NLM profiling configuration

A brief description of each field follows:

- Frequency

  This is the number of events the processor is to count before generating a Non Maskable Interrupt (NMI). A high frequency means fewer NMIs and thus fewer samples per second, and vice versa. The value displayed in Figure 8-23 on page 297 is based on the CPU clock speed and is calculated to deliver around 400 NMIs per second.

  Note: Do not make the frequency too small. Doing so may hang the system, since too many NMIs will be generated for the CPU to be able to perform useful work at the same time.

- Sample Time

  This is the period of time during which the profiling tool will collect samples. The default value is 60 seconds.

- File Name

  Type the target file path or use the default, which is the root on the SYS volume. VTune is not very flexible regarding file names; only the last three digits of the output file name should be changed, using only numeric values. The name should always have eight characters.

- User Comments

  Enter information here that will identify the sample set. This is useful if you have multiple sample files to manage. Press the Esc key to start the sampling session.

After generating a trace file, copy it to a location where the Intel VTune client utility running on Windows can access it, or map a drive to its location on your SYS volume. Then, in the VTune client, select **File -> Import** to load the trace file. A graphical display of the trace data by processor and NLM is then displayed (see Figure 8-24 on page 299).

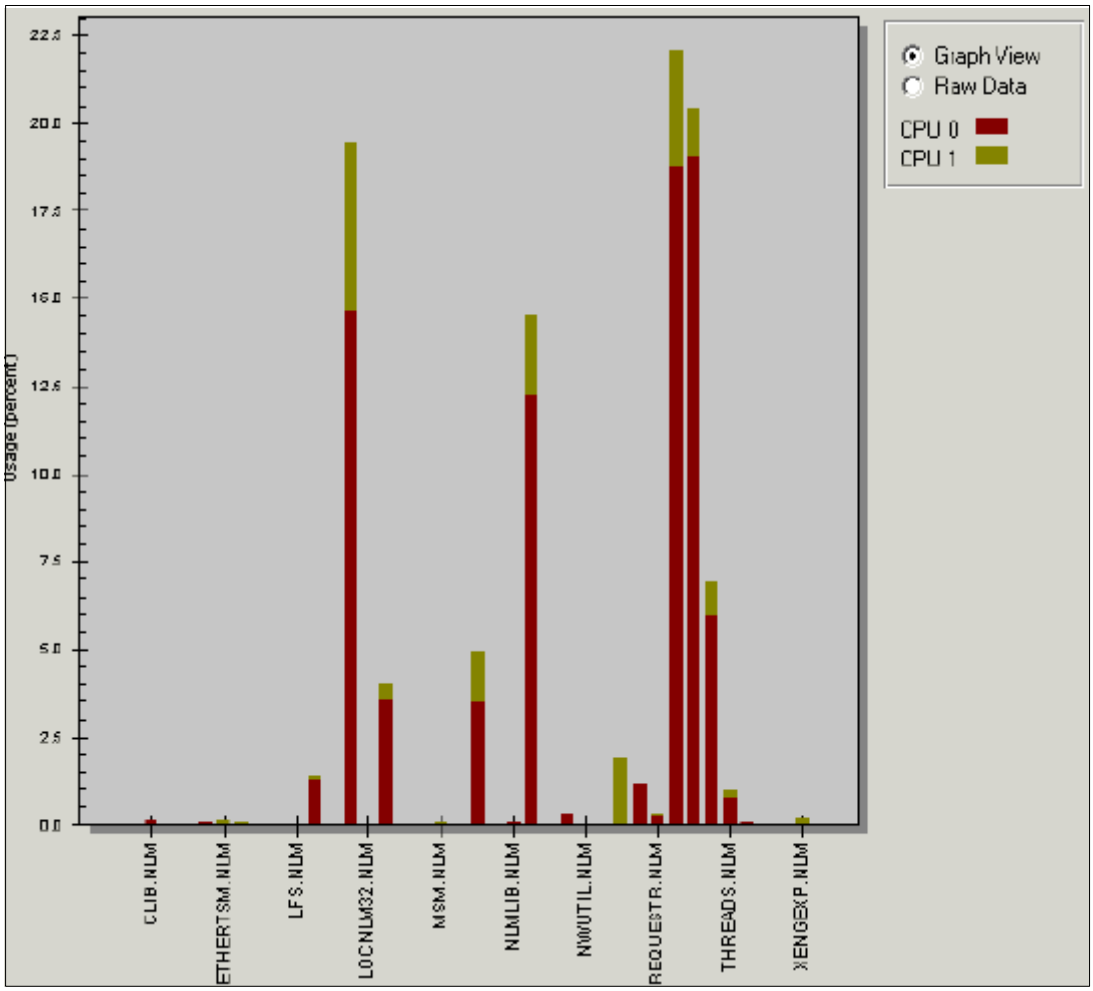


Figure 8-24 VTune client displaying where the server is spending time

Each vertical bar in Figure 8-24 corresponds to an NLM's percentage usage of all processors. Individual processor utilization is color coded. Selecting the **Raw Data** radio button in the top-right corner displays the information in tabular form. This information can be easily exported and used in a spreadsheet to build your own graphs and reports.

Double-clicking one of the bars in Figure 8-24 displays a graph similar to that shown in Figure 8-25 on page 300.

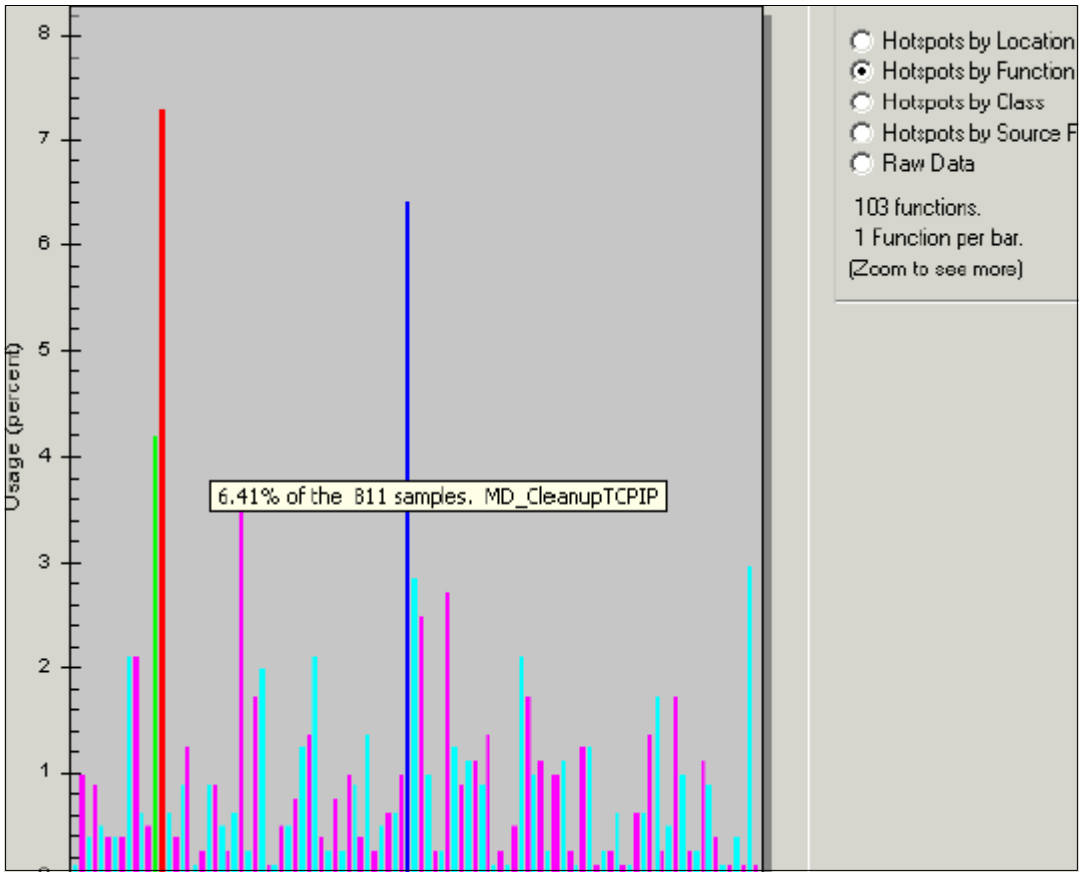


Figure 8-25 VTune displaying hot spots per location inside a program

In the example illustrated in Figure 8-25, we have selected NSHTTPD NLM. This graph displays where, within the NLM, the processor is spending its time (during the sampling period). It is the equivalent of the CPU profiling for a NLM in NetWare Remote Manager. The radio buttons at the top right of the graph allow the user to choose a number of different views.

This tool is useful to record server behavior over time under real-life conditions. For example, by taking samples at the same time each week, at different times of the day, or before and after making changes, you can quickly recognize trends, identify workload peaks, and avoid performance problems.

8.15 NetWare dynamically configured parameters

Typically, new resources are not immediately allocated when requests are received. The operating system waits a specified amount of time to see if existing resources become available to service the requests. If resources become available, no new resources are allocated. If they do not become available within the time limit, then new resources are allocated.

The time limit ensures that sudden, infrequent peaks of server activity do not result in permanently allocating resources that will not be needed again.

For example, when the server is started, all free memory is assigned to file caching. However, as demand increases for other resources (such as directory cache buffers), the number of available file cache buffers decreases.

Before attempting to tune a new server's subsystem, let the server operate at normal capacity for a week or two to allow NetWare to self-tune. After the server settles in to a predictable pattern of daily use, you can begin to fine-tune.

The following parameters are dynamically configured by the operating system:

- Directory cache buffers
- File locks
- Kernel processes
- Kernel semaphores
- Load balancing for multiple processors
- Maximum number of open files
- Memory for NLM programs
- Router/server advertising
- Routing buffers
- Service processes
- Packet receive buffers
- File cache buffers

To ensure that these parameters take explicit values, you can add commands for each change to the STARTUP.NCF or AUTOEXEC.NCF file. Each time the server is rebooted, the parameters will use the explicit value.

In addition to the parameters configured by NetWare, you can adjust the value of many server parameters to optimize the server for your network.

8.16 Novell Storage Services file system

In NetWare 6, the Novell Storage Services (NSS) is loaded on the server by default, but cache memory is not allocated to it unless it is actually running. NSS uses memory differently than the traditional file system (TFS), and you must use specific parameters for tuning your server when using NSS.

To view statistics for cache buffers used by NSS, enter the following command at the System Console prompt: **`nss /cachestats`**. The sample output is shown in Example 8-2.

Example 8-2 Results from a nss /cachestats command

```
NW:nss /cachestats
*****Buffer Cache Statistics *****
Num cache buffers: 512
Num hash buckets: 131072
Min OS free cache buffers: 256
Num cache pages allocated: 34686
Cache hit percentage: 80
Cache hits: 116317
Cache miss: 28375
Percent of buckets used: 7%
Max entries in a bucket: 4
Total entries: 11773
```

The main parameters you should look for in the output of the **nss** command are:

Cache hit percentage This gives you an idea of how your cache is performing. A percentage over 75 is good.

Percent of bucket used This parameter give you an idea of the percentage of the total cache currently in use.

For heavy access on an NSS logical volume, NSS might need configuration adjustments to resolve performance issues. NSS has its own caching and set of parameters. The following are SET parameters that you can use to tune the performance of NSS. They are all located under the Novell Storages Services category.

- NSS Cache Balance Percent

  This parameter controls how much memory is available for NSS to use for caching and is the percentage of memory NSS will use for its own caching system. A high cache balance percentage will impede the performance of TFS and other applications. A low cache balance will impede the performance of NSS. The goal here is to use most of the memory left by applications as cache for NSS. Running your server under normal load and monitoring the memory usage should give you a good idea of memory availability that could be used by NSS.

  We recommend that you set the cache balance parameter to equal the percentage of the total disk space you allocate for NSS. The default is 60 percent. If you are using mostly NSS volumes, you can run at a higher value, such as 80 percent, which is the value used in the Novell performance lab to run NSS benchmark test.

- NSS Closed File Cache Size

  Dictates how many closed files are kept in cache. The recommended value is 50,000.

- NSS File Flush Timer and NSS Buffer Flush Timer

  Increasing these parameters from their defaults helps the caching up to a certain point, but too much causes a throttling problem with writing out the metadata. This is somewhat dependent on processor speed and disk speed, so you could try adjusting them for your system. If NSS performance is very poor, 75 might be too high and you could try 50. If performance is just below normal, you could try increasing the values a little. The default values are:

  – File Flush Timer = 5
  – Buffer Flush Timer = 1

- NSS Name Cache Size

  Each entry in the name cache uses 64 bytes of memory or more (depending on the size of the name). The number of entries you need depends on the number of files that you will be accessing frequently. The default is 2,111 entries. Systems with a lot of memory, accessing a lot of different files, should change this parameter to a higher value.

  For example, let us say that you are serving up Web pages and your statistics tell you that there are 5,000 different files that are being accessed frequently and 50,000 total files being accessed over a day. You might want to set your cache to around 10,000 to hold the frequently accessed files and to allow for other file accesses without bumping the frequently used files from the cache. This would use a minimum of 640,000 bytes of memory.

All of these parameters are available under the Novell Storage Services category in Monitor and NetWare Remote Manager, and they can also be changed using a single NSS console command. You can also add the different configuration parameters in the AUTOEXEC.NCF file.

For example:

```
nss /cachebalance=80 /fileflushtimer=75 /bufferflushtimer=75 /closedfilecachesize=50000
/nameCacheSize=10000
```

8.17 NetWare virtual memory

NetWare provides a virtual memory system that moves data out of memory and into a swap file on disk if the data is not used frequently. Thus, the virtual memory system ensures that RAM is used more efficiently. It lessens the likelihood that low memory conditions will cause a problem with the server.

NRM has several pages that can assist you in monitoring and managing the use of the Virtual Memory system in your server.

To view the overall performance of the Virtual Memory system in NetWare Remote Manager:

1. Click **Monitor Health** in the navigation frame.
2. Click **Virtual Memory Performance**.
3. Click the **Info** icon for each item listed in the Health Info table.

To view specific performance of the Virtual Memory system in NRM:

1. Click **View Memory Config** in the navigation frame.
2. Click **Virtual Memory Page**.
3. Click **Virtual Memory Cache Pool** to view the statistics for each type.
4. Click **Virtual Memory Statistics** to view multiple statistics.

For information on each type of statistic listed, see the online help.

8.17.1 Swap files

Data moved to disk by virtual memory is stored in any available swap file; it does not matter which volume the swap file is in. Generally, you should place swap files on the less busy volume, the one with the most available space, or dedicate a volume to it. The latter also ensures that swap file corruption does not also impact user data.

You can create one swap file per volume. When you dismount a volume, the swap file is deleted. To keep a swap file on that volume, you must create the swap file again when the volume is mounted. The exception is the swap file for volume SYS:, which is created by default. Keeping a swap file on volumes can be done by adding commands to the AUTOEXEC.NCF file, so the files are created each time the server is started.

The command to create a swap partition is:

```
SWAP ADD volume_name [parameter=value...]
```

Optional parameters are min, max, and min free. All values are in millions of bytes.

- MIN or MINIMUM = Minimum swap file size (default = 2).
- MAX or MAXIMUM = Maximum swap file size (default = Free volume space).
- MIN FREE or MINIMUM FREE = Minimum free space to be preserved on a volume outside the swap file; this controls the maximum size of the swap file on this volume (default = 5).

For example:

```
SWAP ADD VOL1 MIN=3,MAX=6,MIN FREE=2000
```

It is also possible to remove the swap partition from the SYS: volume; you will do that to ensure the swap file on another disk is used instead of the one on the SYS: volume.

The command to remove a swap partition is:

```
SWAP DEL SYS
```

This command could also be added to the AUTOEXEC.NCF file. If you are using protected address spaces, the Java Virtual Machine, or any other application that uses virtual memory, be sure to keep at least one swap file.

If the swap file is being used when it is deleted, then the swapped data is moved to another swap file. If there is no other swap file and there is not enough physical memory to move all the swapped data, an error message is displayed and the file is not deleted.

Entering `SWAP` on the Server Console displays the information on all the swap files.

8.18 Analyzing bottlenecks in NetWare

Performance bottlenecks usually occur in one of the following areas:

- Central processing unit (CPU)
- Memory
- Disk subsystem
- Networking subsystem

In general, NetWare systems run short of memory before any other system resource. Counter-examples to this might be database or application servers, which can put significant loads on the CPU.

Once a memory shortage is ruled out as the cause of a bottleneck, the disk and network subsystems are the next most likely suspects.

As a result, the greatest gains in NetWare system performance are usually achieved by properly tuning and sizing the memory subsystem, disk subsystem, and network subsystem, in that order.

8.18.1 Finding disk bottlenecks

The disk subsystem can be the most important aspect of I/O performance, but problems can be hidden by other factors, such as the lack of memory. Finding disk bottlenecks is easier than finding processor and memory bottlenecks because the performance degradation is readily apparent.

I/O operations per second counters in NRM can be used to determine if the server has a disk bottleneck. The best way to optimize a NetWare server using NRM is to let the server collect data over a period of time. Collected data can then be analyzed and a trend may become apparent, indicating a future disk bottleneck.

Using this type of predictive technique allows you to upgrade hardware before the bottleneck arises. Table 8-7 on page 305 describes the NRM counters most useful for isolating the ESS disk subsystem bottlenecks.

Table 8-7 NRM counters for detecting disk bottlenecks on NetWare

Counter	Description
Disk Requests (IOs/s)	This counter displays the average and maximum number of requests made to the disk during a sample interval. The greater the number of I/Os, the more the bus and the disks are taxed. Adding channels and moving files across multiple disks should help you increase the maximum number of I/Os/s that can be supported.
Disk Throughput (KB/s)	This is the average number of KB transferred to or from the disk during write or read operations.The larger the value, the more efficiently the system is running. You should also compare this value against the stripe size of the RAID array (if you are using hardware RAID). If the stripe size and the I/O transfer size are very different, the subsystem may not be performing optimally.

Figure 8-26 shows disk maximum and average I/Os per second during a one-month sampling period.

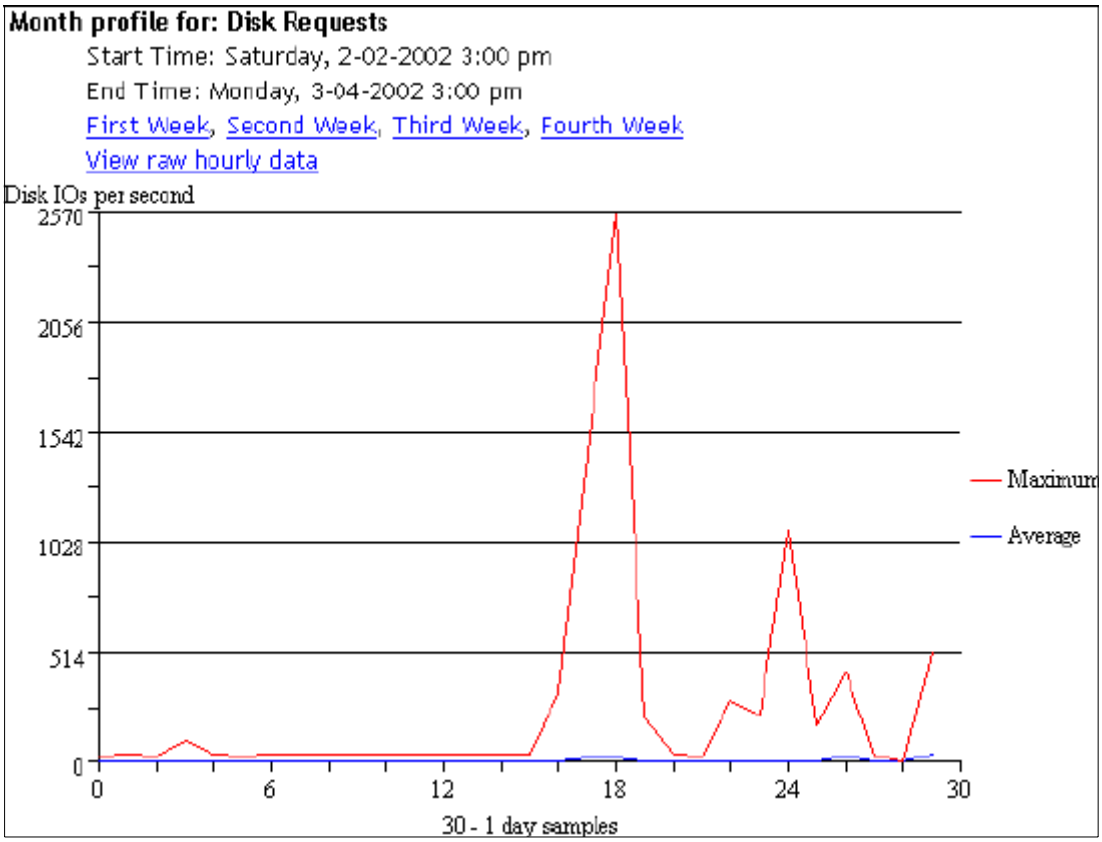


Figure 8-26 I/O operations per second

8.18.2 Performance tuning options

After analyzing the data, if the disk subsystem appears to be the cause of a bottleneck, then a number of solutions are possible. These solutions include the following:

- Use the arrays with faster disk speeds (rpm).
- Consider if a different RAID configuration (either RAID-5 or RAID-10) can make a significant difference.

- Use more arrays to hold the data.
- Offload processing to another system in the network (either users, applications, or services).
- Add more RAM. Adding memory will increase system memory disk cache, which in effect improves disk response times.

Figure 8-27 conceptually illustrates the gain that can be obtained when moving data to faster disks.

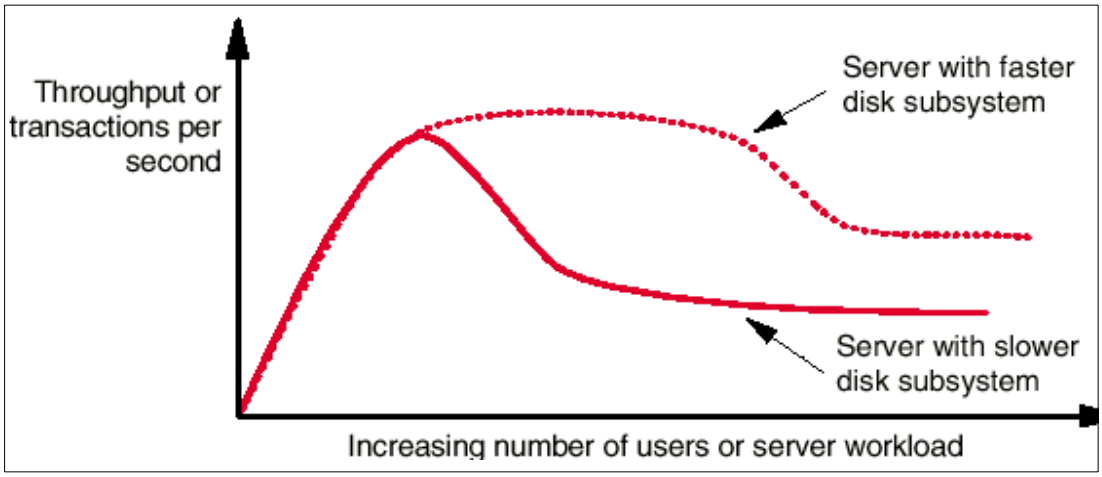


Figure 8-27 Effect of adding a faster disk or better RAID to fit the workload application server

Improving the disk subsystem speed affects the overall performance of the file server in the following ways:

- It usually improves the minimum sustained transaction rate.
- It may only slightly affect performance under light loads because most requests are serviced directly from the disk subsystem cache. In this case, network transfer time is a relatively large component and disk transfer times are hidden by disk cache performance.
- As the server disk performance improves, increased network adapter and CPU performance is required to support greater disk I/O transaction rates.

Accesses to the swap file used to hold paged out memory contents compete with other disk accesses. It is therefore a good idea to locate the swapping partition on its own disk, or at least on one that would otherwise see low activity.

9

zSeries servers

The IBM TotalStorage Enterprise Storage Server Model 800 delivers a multi-feature synergy with the zSeries servers specifically running the z/OS operating system, allowing an unprecedented breakthrough in performance for those environments.

In this chapter we describe these performance features and discuss some of the zSeries-specific factors that you need to consider when monitoring and tuning your ESS.

9.1 Overview

With its unique architecture, the IBM TotalStorage Enterprise Storage Server Model 800 provides the highest levels of performance across all the sever platforms it attaches.

Specifically for the zSeries servers, the IBM TotalStorage Enterprise Storage Server Model 800 also keeps from the previous ESS models the set of unique performance features that so well synergize with the host functions. The cooperation between these ESS unique features and the zSeries operating systems (mainly the z/OS operating system) makes the ESS performance even more outstanding in the zSeries environment.

These ESS features that have performance implications in the applications I/O activity are described in the following sections:

- Parallel Access Volumes
- Multiple Allegiance
- I/O Priority Queuing
- Large volumes
- Custom volumes
- FICON Host Adapters

In the following sections of this chapter we describe these ESS features and discuss how they can be used to boost the performance of your zSeries environment.

9.2 Parallel Access Volumes

Parallel Access Volume (PAV) is one of the original features that the IBM TotalStorage Enterprise Storage Server provides specifically for the z/OS users. Simply stated, PAV allows multiple concurrent I/Os to the same volume at the same time from applications running on the same z/OS system image. This concurrency helps zSeries applications better share the same logical volumes with reduced contention. The ability to send multiple concurrent I/O requests to the same volume nearly eliminates I/O queuing in the operating system, thus reducing I/O responses times.

Traditionally, access to highly active volumes has involved manual tuning, splitting data across multiple volumes, and more things in order to avoid those hot spots. With PAV and the z/OS Workload Manager, you can now almost forget about manual performance tuning. The Workload Manager is able to automatically tune your PAV configuration and adjust it to workload changes. The ESS in conjunction with z/OS has the ability to meet the highest performance requirements.

9.2.1 Response time components

Before explaining how PAV works and to better understand the gains it provides, let us first review the response time components of an I/O operation over an ESCON channel. WIth FICON attachment, the I/O operation has been enhanced and some of the response time components have been reduced or eliminated. These improvements are discussed later.

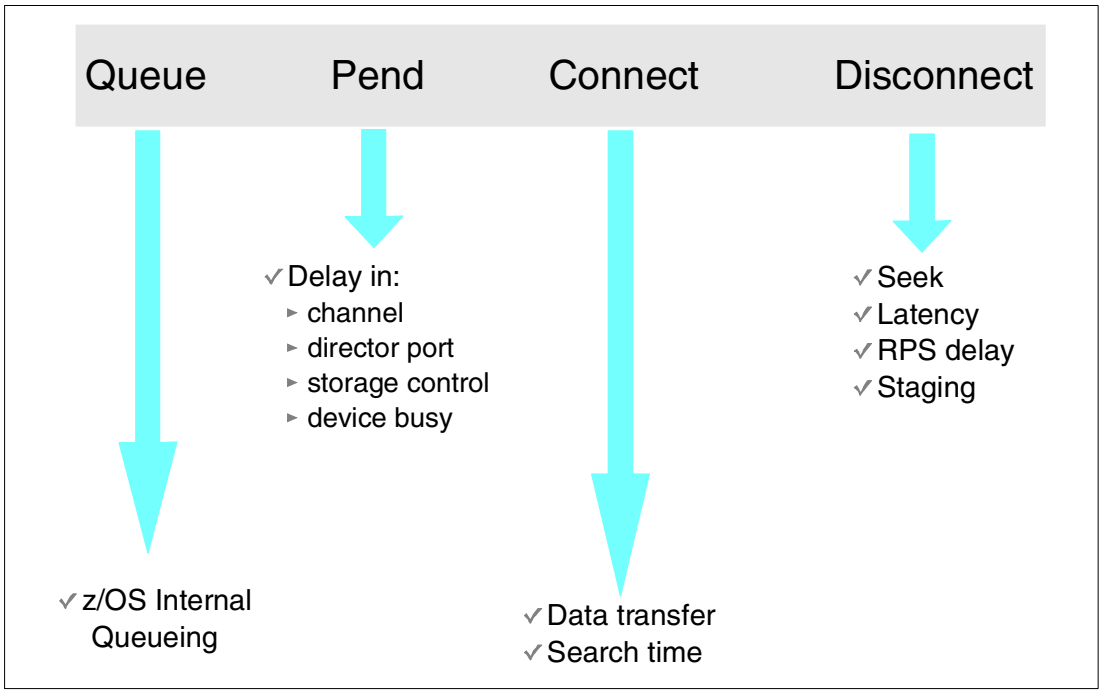


Figure 9-1 DASD response time components

Response time is a key measure of how well your I/O is performing, and a key indicator of what can be done to speed up the I/O. It is important to understand the components of response time, and what they mean. DASD response time is the sum of queue, pending, connect, and disconnect time. The following lists the response time components and what causes them:

Connect The part of the I/O during which data is actually transferred; protocol, search, and data transfer time.

Disconnect Time that an I/O request spends freed from the channel. This is the time that the I/O positions for the data that has been requested. It includes:

-SEEK and SET SECTOR (moving the device to the requested cylinder and track)

-Latency (waiting for the record to rotate under the head)

-Rotational Position Sensing (RPS) Rotational delay, as the device waits to reconnect to the channel

The term RPS delay traditionally means waiting for an extra disk rotation because a transfer could not occur when the data was under the head. Delays of this type are obsolete with the ESS.

Pending Time that the I/O is delayed in the path to the device. Pending time may be attributable to the channel, control unit, or wait for director (director port delay time), although it is often caused by shared DASD.

IOSQ time Represents the average time that an I/O waits because the device is already in use by another task on this system, signified by the device's UCBBUSY bit being on.

Reporting is based on one or both of the following measurements:

Service time Connect time plus disconnect time

Response time Connect time plus disconnect time plus pending time plus IOSQ time

We will discuss now how PAV can help you eliminate or reduce the ISOQ time component of the response time of your z/OS system.

9.2.2 PAV characteristics

The IBM TotalStorage Enterprise Storage Server Model 800 is a modern disk storage subsystem with large cache sizes and disk drives arranged in RAID arrays (ranks). I/O solved in cache is much faster than I/O solved on disk, as no mechanical parts (actuator) are involved. Plus I/Os can take place in parallel, even to the same volume. This is true for reads, and it is also possible for writes, as long as different extents on the volume are accessed.

The ESS emulates zSeries CKD volumes over RAID-5 and RAID-10 disk arrays. While the z/OS operating system continues to work with these logical DASD volumes as a unit, its tracks are spread over several physical disk drives. So parallel I/O from different applications to the same logical volume would also be possible for cache misses (when the I/Os have to go to the disk drives, involving mechanical movement of actuators) as long as the logical tracks are on different physical disk drives.

Traditionally the operating system does not attempt to start more than one I/O operation at a time to a logical device. If an application initiates an I/O to a device that is already busy, the I/O request will be queued by the I/O Supervisor (IOS) component of z/OS until the device completes the other I/O and becomes available. z/OS systems queue I/O activity on a Unit Control Block (UCB) that represents the logical device. When volumes are treated as a single resource and serially reused, high I/O activity could adversely affect performance. This could result in large IOS queuing (IOSQ) times, traditionally a major component of z/OS response time. This queue time component can be eliminated if the devices are defined with PAVs.

To take advantage of the ESS's capability to process multiple concurrent I/Os to a logical volume, the ESS and z/OS together support the Parallel Access Volume feature that allows parallel I/Os to a volume from one z/OS system image. PAV is an optional feature of the ESS.

The PAV implementation introduces the new concept of *alias device*, in addition to the conventional *base device*. The alias devices provide the mechanism for z/OS to initiate parallel I/Os to a volume.

Each device, base and alias alike, has a unique address in the ESS control unit image. The unit addresses range from x'00' to x'FF'. Each device has its own subchannel, UCB, and device address on the host. The base device's address is the actual unit address of the volume. There is one base address per volume.

An alias device does not have any physical space of its own associated with it. It simply maps to a base and allows a z/OS host to use several UCBs for the same logical volume instead of one UCB per logical volume. For example, base address 2C00 may have alias addresses 2CFD, 2CFE, and 2CFF associated with it. Each UCB supports one parallel I/O operation as before, but the three aliases in addition to the base now allow for four parallel I/O operations to the same volume.

The definition and exploitation of Parallel Access Volumes requires the operating system software to define and manage alias device addresses and subchannels. The z/OS operating system has this support and can issue multiple channel programs to a logical volume, allowing simultaneous access to the volume by multiple users or jobs. Reads can be satisfied

simultaneously, as well as writes to different domains. The domain of an I/O consists of the specified extents to which the I/O operation applies. Writes to the same domain still have to be serialized to maintain data integrity.

Both base and alias devices are defined on the ESS using the ESS Specialist. They must be accompanied by matching HCD device definitions on the zSeries server. In the HCD, you specify which device addresses are bases, and which aliases.

Important: The device definitions in the ESS and HCD must match.

A base device can be associated with aliases only from the same ESS logical subsystem (LSS). In other words, an alias can be assigned only to a base in the same LSS. The maximum number of devices per LSS is 256, including both bases and aliases. Theoretically, you could define one base and 255 aliases in an LSS. At the other extreme, you could define 256 base devices and no aliases. Typically, you will have a mix of both. The number of base devices you define depends on the amount of installed storage capacity in the LSS, and the size of the volumes you define. To provide enough concurrency and performance for the base devices, you then need to define and assign an adequate number of aliases to the bases.

Together the base and aliases are called *exposures*. We also use the terms parallel addresses, or PAV addresses (simply PAVs), when referring to a base and its aliases together. Thus the number of PAVs for a volume is one plus the number of aliases assigned to it.

9.2.3 Dynamic and static PAVs

If you have decided to take advantage of the performance benefits to be obtained from using PAVs then you need to consider what type of PAV support to implement and how many PAVs to define.

When initially defined, aliases get assigned to a certain base device. An alias can later be reassigned to another base while the devices remain online. Ideally, if you had a large number of aliases you could give each base enough aliases at the beginning so that the devices would never experience IOS queuing delays. More typically, you will not have that many aliases to work with; therefore, the alias assignments need to be modified to adapt to changes in workload that occurs during daily workload shifts, and in the long run. Since the number of devices per LSS is limited to 256, aliases can be a scarce resource that needs to be managed to maximize performance. For best I/O management, the aliases should be assigned to the base devices based on their needs. The busiest base devices at any given moment, or those handling the most important workload, should have more aliases than the inactive bases.

It will not always be easy to predict which volumes should have an alias address assigned, and how many. The workload intensity of volumes can change over time, sometimes so rapidly that manual tuning cannot keep up with the changes in volume activity. Particularly in cases when the number of aliases is limited, it is important that aliases can be assigned to the volumes most needing them.

You can reassign aliases manually using the ESS Specialist. However, this is a labor-intensive process and, to achieve optimum results, should be repeated whenever there are workload changes. The process of alias reassignment needs to be automatic and be sensitive to the importance and level of I/O activity on the device.

Instead of manually tuning your aliases, you can let z/OS Workload Manager (WLM) automatically manage the aliases according to your goals. The WLM cooperates with the IOS to allow for the automatic management of aliases. This function is called *dynamic alias management*. With dynamic alias management, WLM can automatically perform alias device reassignments from one base device to another to help work meet its goals and to minimize IOS queueing as workloads change.

WLM manages PAVs across all the members of a sysplex. When making decisions on alias reassignment, WLM considers I/O from all systems in the sysplex. By default, the function is turned off, and must be explicitly activated for the sysplex through an option in the WLM service definition, and through a device level option in HCD. Dynamic alias management requires your sysplex to run in WLM Goal mode.

Devices that are enabled for dynamic alias management are called *dynamic PAVs*. The alternative is *static PAVs*. With static PAVs, the association of aliases to a base device is predefined through the ESS Specialist. Static alias assignment can only be changed using the ESS Specialist. It is possible to define both dynamic and static PAVs on an LSS, if necessary.

9.2.4 Enabling dynamic PAV

You can enable or disable dynamic alias management on a device by device basis with the WLMPAV parameter in HCD. If WLMPAV=YES then the aliases of the device will be managed dynamically by WLM. If WLMPAV=NO, then the aliases of the device are static. The default for each device is YES.

Besides being enabled for each volume through the HCD, dynamic alias management must also be enabled globally in the sysplex. This is done by setting the Dynamic alias management service definition option to YES on the WLM Service Coefficient/Service Definition Options panel (Figure 9-2). The default is NO (dynamic alias management is disabled).

The I/O priority management option on the WLM Service Coefficient/Service Definition Options panel activates ESS I/O Priority Queuing (see 9.4, “I/O priority queuing” on page 325). At the same time, it also determines which of two dynamic alias management mechanisms will be used, when dynamic alias management is enabled. If the option is set to YES, the goal-based mechanism for dynamic alias management is used. If set to NO (the default), the efficiency-based mechanism is used. See 9.2.5, “WLM dynamic alias management” on page 313.

```
  Coefficients/Options  Notes  Options  Help
-------------------------------------------------------------
             Service Coefficients/Service Definition Options
Command ===>

Enter or change the Service Coefficients:

CPU  . . . . . . . . . . . . . _______ (0.0-99.9)
IOC  . . . . . . . . . . . . . _______ (0.0-99.9)
MSO  . . . . . . . . . . . . . _______ (0.0000-99.9999)
SRB  . . . . . . . . . . . . . _______ (0.0-99.9)

Enter or change the service definition options:

I/O priority management . . . . . . . . YES  (Yes or No)
Dynamic alias management  . . . . . . . YES  (Yes or No)
```

Figure 9-2 WLM service definition options for dynamic alias management

Dynamic aliases effectively form a pool from which WLM can choose aliases to assign to the base devices needing them most. This makes your configuration planning easier, as you do not have to plan in advance how many aliases each and every volume should have, but rather you only need to plan how many aliases in total there will be available in each LSS.

We recommend always using dynamic PAVs if possible. You should consider using static PAVs only in exception cases, such as:

- Your system is not running in WLM Goal mode. Note that z/OS V1R3 and later only run in Goal mode.
- You are sharing devices between sysplexes.

9.2.5 WLM dynamic alias management

When Dynamic alias management is enabled, the Workload Manager monitors the device performance and is able to dynamically reassign alias devices from one base to another if predefined goals for a workload are not met. The WLM instructs the IOS to reassign an alias.

WLM also keeps track of the devices utilized by the different workloads, accumulates this information over time, and broadcasts it to the other systems in the same sysplex. If WLM determines that any workload is not meeting its goal due to IOSQ time, WLM will attempt to find an alias device that can be reallocated to help this workload achieve its goal.

The use of dynamic PAVs requires that devices assigned as eligible for dynamic PAV management must not be shared by systems outside the sysplex. WLM does not consider I/O from systems outside a sysplex when reassigning aliases.

There are two different mechanisms that WLM uses to tune the alias assignment:

- The first mechanism is *goal based*. This logic attempts to give additional aliases to a PAV-enabled device that is experiencing IOS queue delays and is impacting a service class period that is missing its goal. To give additional aliases to the receiver device, a donor device must be found with a less important service class period. A bitmap is maintained with each PAV device that indicates the service classes using the device. The objective of a goal algorithm is to help a service class period meet its goal.
- The second mechanism is *efficiency based*. This algorithm moves aliases to high-contention PAV-enabled devices from low-contention PAV devices. This tuning is based on efficiency rather than directly helping a workload to meet its goal. The objective of an efficiency algorithm is to reduce overall queuing in the system.

Because adjusting the number of aliases for a PAV-enabled device affects any system using the device, a sysplex-wide view of performance data is important, and is needed by the adjustment algorithms. Each system in the sysplex broadcasts local performance data to the rest of the sysplex. By combining the data received from other systems with local data, WLM can build the sysplex view.

The efficiency algorithm uses base devices' IOS queue length (the average number of I/O requests in the IOS queue) to determine when aliases should be moved. Devices whose average IOS queue length exceeds 0.5 will be assigned more aliases to help the receiver devices. Devices whose average queue length is below 0.5 but above 0.05 will be assigned more aliases only if there are aliases available on idle or low-activity devices. (See APAR OW48647 for more information.)

If the goal mechanism is used, WLM will first take workload goals into consideration when moving aliases, then use volume queue lengths to determine which volumes within a service class should be assigned more aliases.

The efficiency algorithm scans the base devices once per minute to find devices that exceed thresholds and reassigns aliases if necessary. The goal algorithm scans devices every 10 seconds, but a device that received or donated an alias is not touched for the next minute. WLM only moves one alias at a time to or from a base device. With goal algorithm, multiple base devices of a service class can receive an alias during one interval.

Aliases of an offline device are considered *unbound*. WLM uses unbound aliases as the best donor devices. If you run with a device offline to some systems and online to others, you should make the device ineligible for WLM dynamic alias management in HCD.

Note: The thresholds used by WLM are subject to change. The values quoted here were valid at the time of writing our document, for systems with APAR OW48647 installed.

Figure 9-3 helps you relate queue length to volume utilization by showing how volume utilization corresponds to queue lengths of 0.5 and 0.05 at different PAV values. The figure is based on a theoretical calculation and may not exactly match your system behavior.

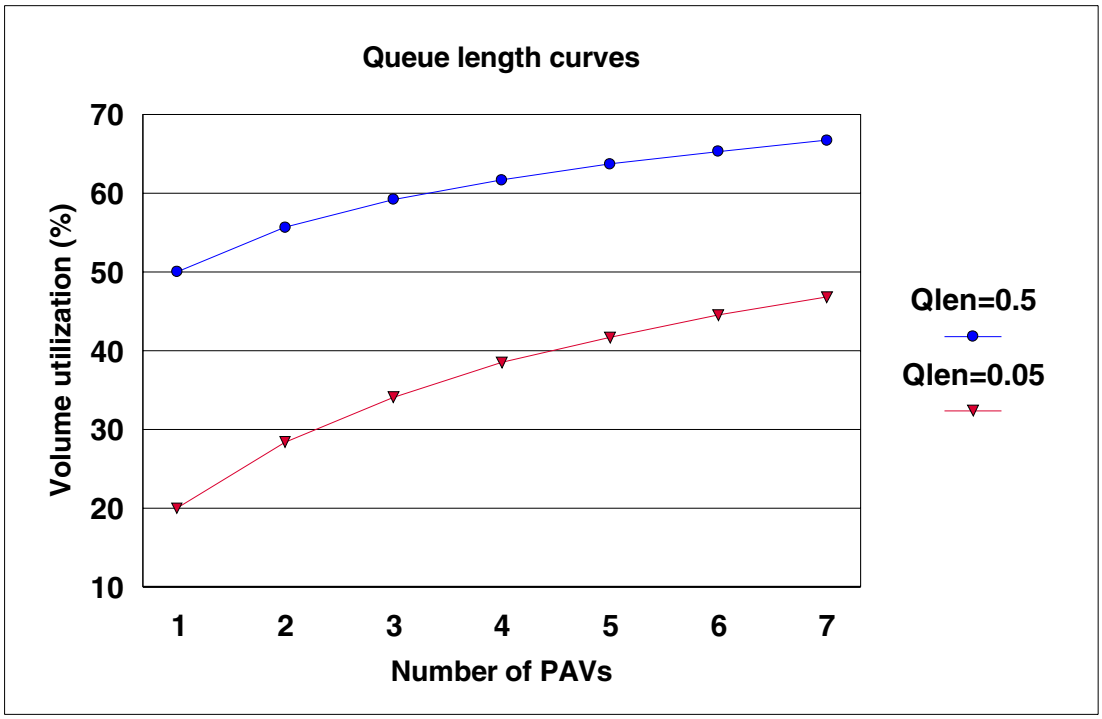


Figure 9-3 Utilization corresponding to queue lengths 0.5 and 0.05

For example, if the utilization of a device with two aliases (three PAVs) exceed 35 percent, then the efficiency algorithm will assign a new alias to it if an alias is available from idle or low-activity devices. If the utilization rises above 60 percent, an additional alias will be given to it even if it has to be taken from an active device.

Notice that queue length 0.5 corresponds to high, above 50 percent volume utilizations. The fact that WLM uses this high threshold should reduce unproductive and unnecessary movement of aliases between active devices.

Note that the graph refers to overall device utilization, whereas RMF reports device utilization as seen from one MVS™ system only. If you are sharing I/O across multiple systems, you will need to review RMF reports from each of the sharing systems in order to see the complete I/O activity to the device.

RMF reports the number of PAVs for each device in its Monitor/DASD Activity report and in its Monitor II and Monitor III Device reports. RMF also reports which devices had a change in the number of PAVs during the RMF interval.

9.2.6 PAV performance considerations

Our first recommendation is to always configure one alias for every base device as a minimum. There is a point when the benefits of adding more aliases starts to be less significative.

There is limited value in configuring more PAVs than you have ESCON channel paths to a system, as the number of ESCON channel paths becomes the limiting factor in data transfer. This effect will be seen more strongly if you are using EMIF shared channels, as sharing systems are competing for channel resources. The previous consideration does not apply for FICON channels, due to their ability to execute multiple channel programs from a single channel concurrently.

Although PAVs will increase I/O concurrency in your disk environment, they do not increase the capacity of the ESS to perform work. What they do is remove the limitations on throughput that are inherent to the operating system operation.

A good starting recommendation is:

- For 3390-3 and smaller devices, define one aliases per base.
- For 3390-9 devices, define three aliases per base.

Of course you will monitor your RMF data to see if any of your volumes still show IOSQ time contention that can be further reduced. See also 9.2.7, “PAV and large volumes” on page 316. If you have the resources available and your current configuration shows that IOSQ time is a major source of device delay, then assigning additional PAVs should provide performance benefits.

Dynamic PAV may require fewer PAVs in total to be defined, as WLM will be more effective in assigning resources where and when they are needed, than with a static configuration.

PAV provides the following advantages:

- Reduced potential for I/O hotspots.
- Reduced IOSQ times and improved throughput.
- Larger volumes can be defined, thus simplifying storage management.
- Easier data set management as the location of the data is less an issue.

The following are helpful recommendations when you are configuring PAVs.

- If you define three 3390-3 disks with one alias each, you will have six concurrent accessors for (approximately) 9 GB of data.
- If you define disks as 3390-9 and define three aliases, you will have four concurrent data accessors per (approximately) 9 GB. However, one 3390-9 with three aliases provides better performance than three 3390-3 volumes each with one alias, but with fewer total addresses.
- When you implement 3390-9 format volumes for the first time, physical data movement may not be an option and you must perform a logical data migration.
- You need to study your I/O intensity before making a decision, as the number of aliases required depends on the I/O activity.

Not all volumes need aliases. For example:

- FlashCopy target volumes, if used for backup purposes only, do not benefit from aliases since a volume will be used by only one application, the backup or dump program. FlashCopy itself does not use alias devices.
- PPRC secondary volumes do not use aliases because they are offline to the host. However, you should still define aliases in the secondary LSSs just in case you need to take the secondaries into production. PPRC itself does not use aliases.
- Small custom volumes dedicated to just one application or data set.

If you use dynamic alias management, you obviously do not need to plan which volumes to assign aliases. WLM will move the aliases around as needed.

9.2.7 PAV and large volumes

The general recommendation is to have one alias device for a 3390-3 volume and three aliases for a 3390-9 volume. With the ESS supporting large 3390 volumes of up to 32760 cylinders (approximately 27.8 GB), the recommendations need to be adjusted. Things we want to reconsider are:

- How many alias devices does a large volume require?
- How many large volumes can you define in an LSS with enough aliases for them?
- How should you split the 256 addresses of an LSS between the bases and aliases?
- Is it better to define fewer, larger volumes with more aliases, instead of several small volumes with fewer aliases? Or does it make any difference?

We will answer these questions both theoretically and by looking at performance measurements. The theory first is that by using queuing models[1] we can estimate how large volumes will be affected by IOS queuing. Table 9-1 shows what happens to IOS queuing if we migrate several small volumes into one larger volume with multiple aliases. The source and baseline for comparisons is a standard 2.8 GB size 3390-3 with no aliases. The table shows how much of the target volume's total response time, in percentage, will be queuing time. For each migration target two different PAV alternatives are shown. The figures are calculated for different source volume utilization levels. These are shown in the left-most column.

Table 9-1 Queuing percentages for different migration ratio/PAV combinations

Source vol Utilization (3390-3)	1:1 migration 2.8 GB target vol (3390-3)		3:1 migration 8.5 GB target vol (3390-9)		9.8:1 migration 27.8 GB target volume	
	PAV=1	PAV=2	PAV=3	PAV=4	PAV=5	PAV=6
10%	10	0	0	0	0	0
20%	20	1	1	0	2	0
30%	30	2	3	0	10	3
40%	40	4	7	1	33	11
50%	50	6	14	3	90	33
Unit of table values is percentage						

[1] A volume with multiple aliases becomes a general queuing system where a queue is being served by multiple independent servers. M/M/c queuing system formulas are applied here. The M/M/c queuing model assumes that arrival and service rate distributions in the system are exponential. The rates on a real system may deviate from the ideal distribution.

Three different migration targets are included in the table. First is a 1:1 migration. Column PAV=1 represents a 1:1 migration target with no aliases. This is actually the same as the source volume, a 3390-3 with no aliases, so the PAV=1 column shows the queuing percentage of the source. Column PAV=2 gives the queuing percentage for a 3390-3 volume with one alias. The next two columns represent a 3:1 migration where the target is a standard 8.5 GB size 3390-9 volume. For it, options of two aliases (PAV=3) and three aliases (PAV=4) are shown. The two right-most columns represent the largest supported volume, a 27.8 GB 3390 volume. It corresponds to a 9.8 to 1 migration ratio. For it, options of four aliases (PAV=5) and five aliases (PAV=6) are shown. For example, the 50 percent row in column PAV=3 tells us that if we migrate three 3390-3 volumes with an average utilization of 50 percent into one 3390-9 volume with two aliases, then 14 percent of the target volume's total response time will be queuing time.

By comparing columns PAV=1 and PAV=2 you can see the dramatic effect that adding just one alias to a base has. Queuing is reduced to a fraction of the original. This emphasizes the minimum recommendation of having at least one alias for each volume.

Column PAV=4 represents a 3390-9 volume with three aliases. This is the common recommendation for a 3390-9 volume. As you can see, it performs very well, with hardly any queuing at the shown utilization levels. By comparing columns PAV=1 and PAV=3 you can in fact see that a 3390-9 with just two aliases performs better than a base 3390-3.

A 27.8 GB large volume with four or five aliases (columns PAV=5 and PAV=6) performs better than the base volume below 40 percent to 50 percent utilization levels. At low utilization levels of up to about 20 percent they perform equally to the other combinations. (Keep in mind here that the utilizations we refer to are source volume utilizations.) Performance measurements (see 9.5.2, "Larger vs. smaller volumes performance examples" on page 328) actually show that less than four aliases on average can be adequate for large volumes even with high intensity workloads.

In the 9.8:1 migration we are effectively replacing nearly 10 addresses with just five or six addresses with the same total capacity, and the target still performs better at these utilization levels. So you get better performance with less addresses.

The model we use here assumes that the service time in the migrations remains constant. This is a fair assumption because the ESS can do simultaneous I/O operations on different extents of a logical volume. The results therefore can be used to compare IOS queuing between different logical volume size/PAV combinations. They should not be used to make assumptions on performance between a different number or different kinds of arrays (different HDDs or RAID configurations), because any change in the array configuration will affect the service time. The model does not take into account other system factors such as channel utilization or concurrent I/Os from sharing systems that affect the service time.

We presented the results in Table 9-1 on page 316 as queuing percentages, instead of as average utilizations, because average device utilization does not directly show how well volumes are performing unless they have the same number of aliases. A volume with multiple aliases can have less queuing at higher utilization levels than the same volume with less aliases at lower utilization levels. (Refer to Figure 9-3 on page 314.)

Another way to present the model results is to calculate the response time of the different target volumes. Figure 9-4 on page 318 shows the relative response time of the six combinations we have discussed. The source and reference again is a 3390-3 with no aliases (1:1 PAV=1).

The response times in the figure are relative, in that they do not represent any specific millisecond values, but rather show the response time relative to the response time at the 0 percent utilization mark.

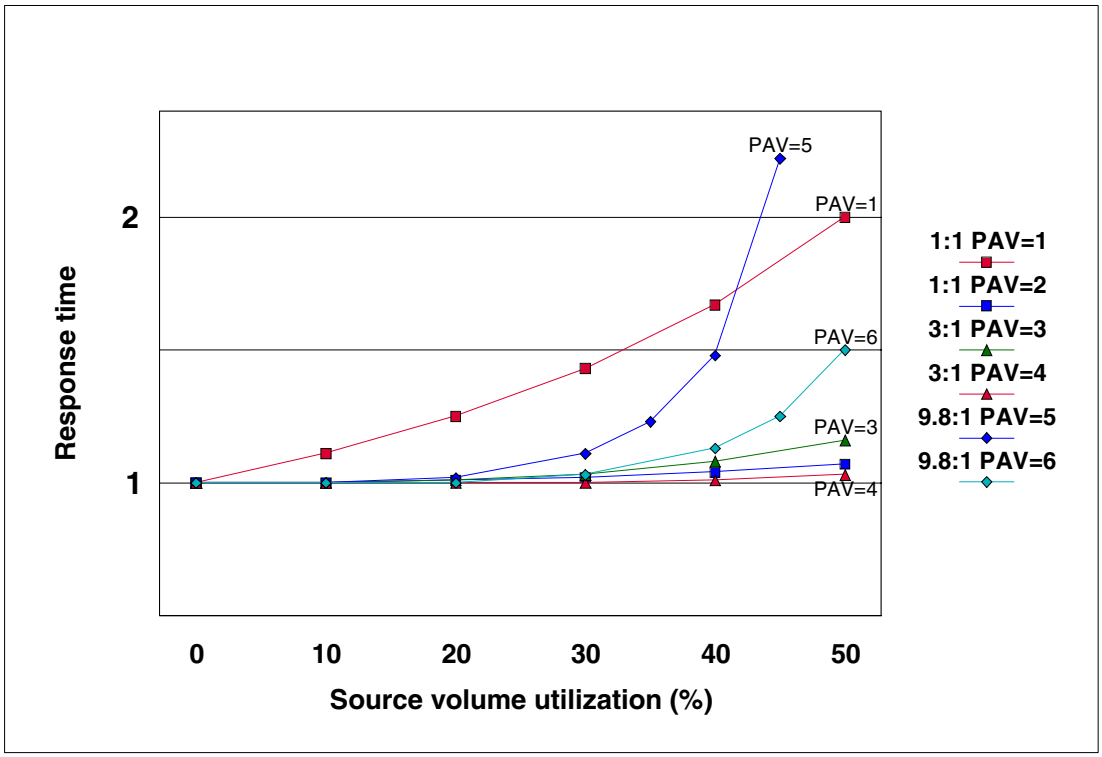


Figure 9-4 Response time of migration target volumes

The two lines at the bottom (PAV=2 and PAV=4) represent a 3390-3 with one alias, and a 3390-9 with three aliases, the common recommendations. They perform very well throughout with hardly any queuing. It is also obvious that adding more aliases to them will provide little benefit since the curves are practically flat already.

The large volumes with five and six parallel addresses (PAV=5 and PAV=6 lines) perform well up to about 30 percent and 40 percent utilization levels, respectively, although they begin to show signs of queuing between the 20 percent and 30 percent utilization marks, indicating that the volumes could start using more aliases at that point, if available.

In Figure 9-5 on page 319 you can see in more detail how many aliases are needed to keep queuing down. The figure shows for the 3:1 and 9.8:1 migration targets the number of PAVs that are required to keep the target volume's average queue length below 0.05 at different utilization levels. The 0.05 queue length is the threshold that WLM uses to determine that a volume should be given more aliases, if available. (Refer to 9.2.5, "WLM dynamic alias management" on page 313.)

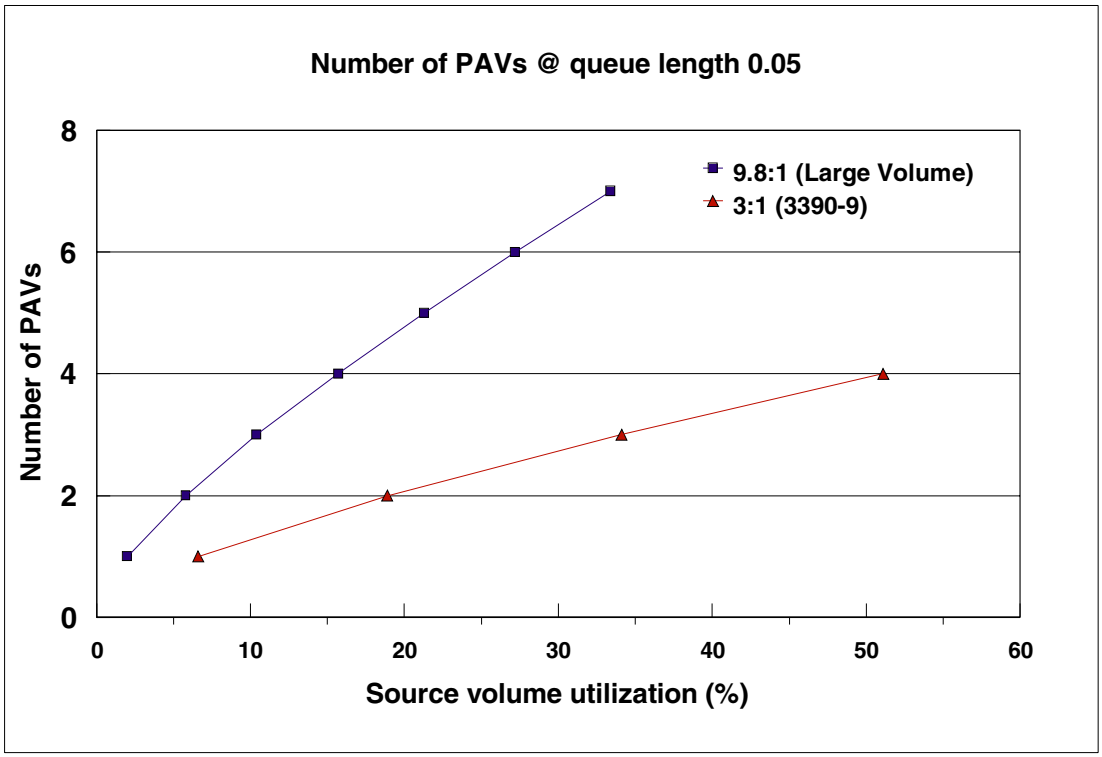


Figure 9-5 Number of PAVs required at different utilization levels

For example, if you migrate 3390-3 volumes whose average utilization is 10 percent into one large volume, the target volume would require two aliases (three PAVs) to keep its queue length down at 0.05. More than three aliases would be needed only if the average utilization of the source volumes exceeded approximately 16 percent.

Both PAV=5 and PAV=6 lines in Figure 9-4 on page 318 represent 27.8 GB volumes, which have less parallel addresses than the corresponding source volumes have in total. With 10 parallel addresses, a 27.8 GB volume would perform better at all utilization levels. This becomes apparent in Figure 9-6 on page 320, which shows the relative response times for migration combinations where each migration target has the same number of parallel addresses as the source volumes have together. Thus all the curves in the figure represent volumes with equal PAV per GB ratios.

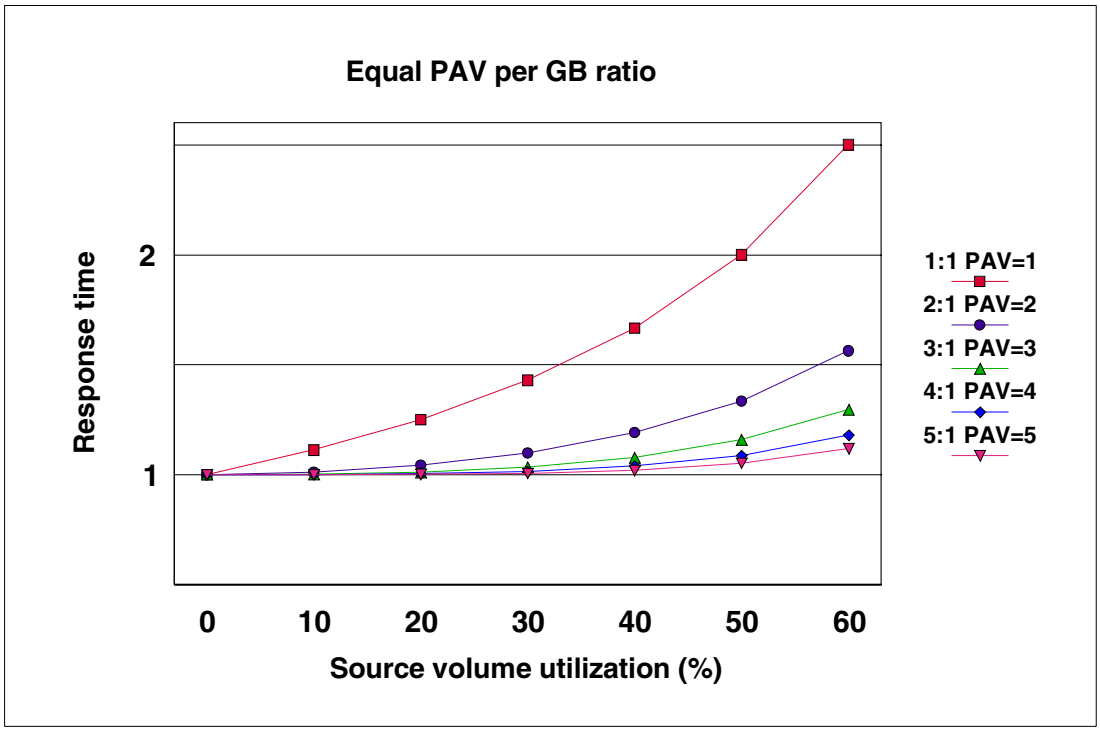


Figure 9-6 Response times at equal PAV per GB ratio

You can see that the larger the volume is the better the response time when PAV per GB ratios are equal. This applies to any number of PAVs, not just the ones shown in the figure. So, instead of defining nine 3390-3 logical volumes in an ESS array, you get lower IOSQ times by defining one large volume and using the extra addresses as aliases. The service times would be equal because for the ESS internal I/O performance it does not make a difference how an array is divided into logical volumes.

The fact that logical volume size has little impact on performance has also been demonstrated by performance measurements that compare a number of large volumes with an equal capacity of smaller logical volumes. Refer to 9.5.2, “Larger vs. smaller volumes performance examples” on page 328.

Conclusions

The overall conclusion from the discussion is that you can and should define large volumes on your ESS, and leave as many as possible of the 256 addresses in each LSS as PAV aliases. This will reduce overall queuing and increase the throughput of your system, besides making the storage administration more simpler. For the best result, combine this with WLM dynamic alias management, which will constantly monitor your volumes and dynamically reassign more aliases to the higher utilized volumes that need them most, in a continuous attempt to provide the best performance that can be achieved with the aliases that are available.

9.2.8 Available aliases are enough

We have so far seen that the best way to use the available 256 addresses of an LSS is to use large volumes. Now we look at how many aliases we can have, and if that is enough in large configurations.

The largest disk drive available for the ESS today is the 145.6 GB DDM which was introduced in late 2002. A maximum ESS configuration of 48 of such eight-packs will provide 2.8 TB of effective RAID-5 capacity per LSS (one 6+P and two 7+P ranks). You can define 97 maximum size large volumes in such an LSS. This leaves you with 159 aliases. Is this enough?

Figure 9-7 shows the distribution of bases and aliases on an LSS with up to three RAID-5 ranks for 72.8 GB and 145.6 GB DDMs configured full of 32760 cylinder volumes.

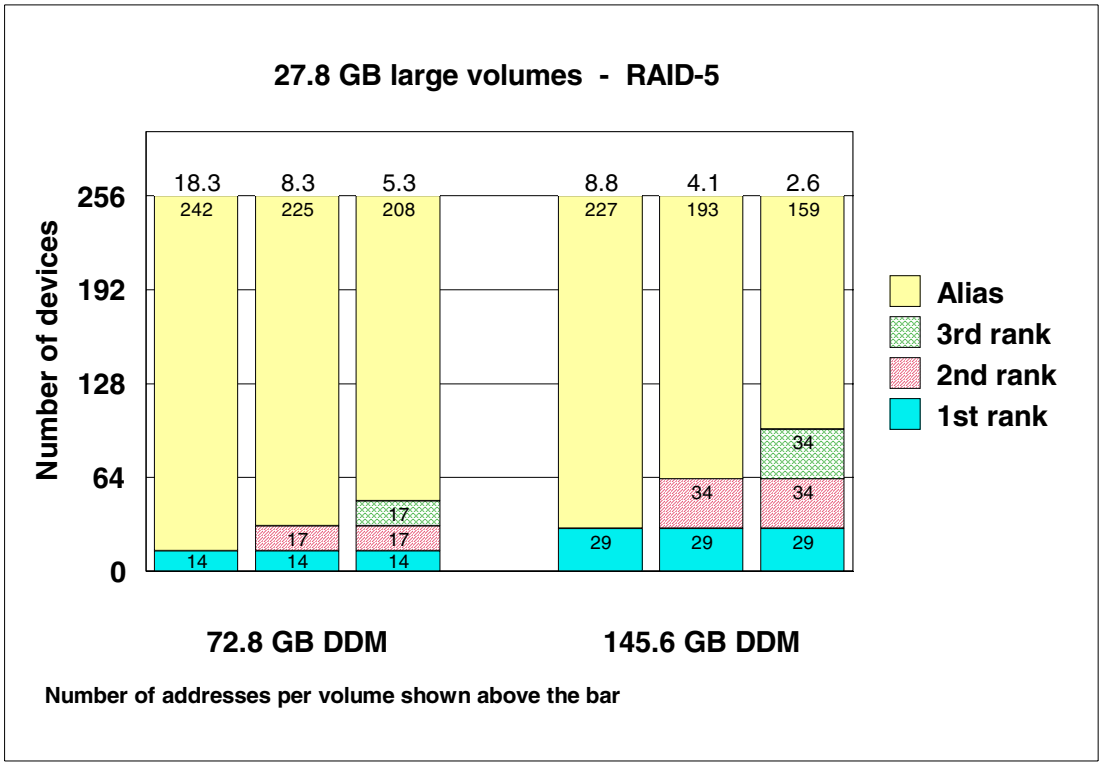


Figure 9-7 Number of PAVs per volume

The number above each bar shows the PAV ratio for the LSS, that is, the average number of PAV addresses (base + alias) per volume. In a configuration of three 145 GB disk ranks the PAV ratio is 256/97 = 2.6. Is this enough to keep IOSQ queuing down at a reasonable level? Performance measurements can help answer this question.

In the examples presented in 9.5.2, “Larger vs. smaller volumes performance examples” on page 328, you can see that large volumes needed from 3.2 up to 4.7 PAVs. This is more than you can have in a three rank, or even in a two rank LSS of 145 GB disks. However, the I/O rates in the measurement represent access densities of approximately 13 and 21 IO/sec/GB. That is more than you would expect to do on a large capacity ESS. Access densities closer to one IO/sec/GB are more typical today. The average access density for enterprise users in 2002 is estimated as 0.6 IO/sec/GB. At these lower, more typical access densities less PAVs are required, and the 4.1 and 2.6 addresses per volume that are available in two and three rank LSSs should be adequate.

Figure 9-8 on page 322 is from a series of performance measurements that compare different ESS disks drive configurations. It shows the average response time for equal capacities of 145 GB and 72 GB disk drives. The measurements were done using enough aliases per volume so that there is no IOSQ component in the measured response times. The 7000 IO/sec mark corresponds to one IO/sec/GB access density.

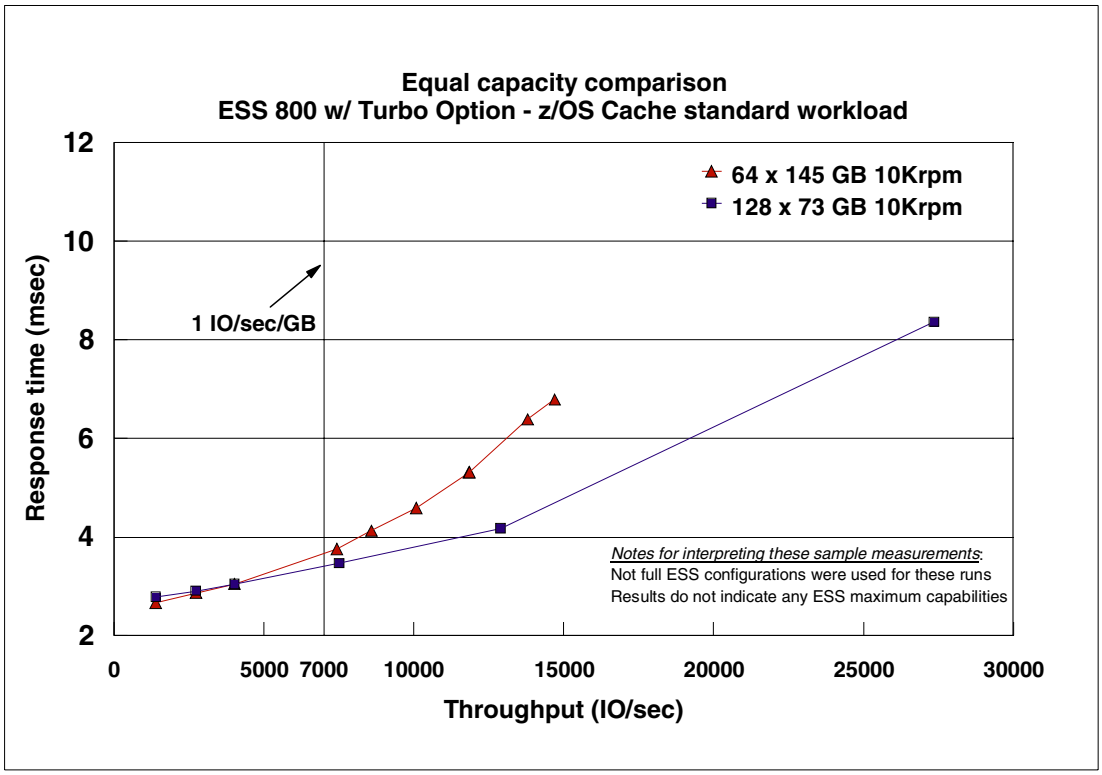


Figure 9-8 145 GB DDM vs. 72 GB DDM comparison

You can see that the 145 GB DDMs suit well for low access density workloads. At higher access densities the equal capacity 72 GB DDM configuration, with double the amount of disk arms, is a better choice. It provides better response times and can sustain higher throughput. Using 72 GB DDMs will also leave you with more aliases to better support the higher I/O rates (refer to Figure 9-7 on page 321).

The actual number of PAVs required in each environment depends on the I/O rate and device service time that can be achieved. This in turn depends on several application and hardware factors, such as read/write ratio, cache hit ratio, block size, DDM size and speed, cache size, RAID configuration, etc. The measurements presented in this chapter serve as examples, but indicate that two to three aliases per large volume should in most cases be enough.

In ESCON configurations the ESCON limit of max 1024 devices per channel may limit the number of devices you will configure per LCU to even less than the maximum 256 supported by the ESS. A typical number is 128, or even 64 per LCU, depending on the number of channels you have. This means that you can define less volumes if you want to achieve the same PAV ratio as with 256 addresses. As a result, you cannot install as much capacity in the LSS, or you must run lower intensity workloads that require less addresses.

Conclusions

As a conclusion we can say that there are enough aliases available for large volumes in 72 GB DDM configurations. And, even though the PAV ratio for 145 GB configurations at first seems small, we can say that it too is adequate for large volumes given the workload intensities that you would expect to run on a large 145 GB DDM configuration. This applies to large volumes only. Smaller volumes, even standard size 3390-9 volumes, are too small for the largest configurations available today (see 9.5, “Logical volume sizes” on page 326).

9.2.9 PAV performance measurements examples

Figure 9-9 shows the effect PAV has on read I/O rates on a single volume. In the measurement, multiple concurrent tasks read different extents of a single volume. The test workload had a 100 percent cache hit ratio so backend disk performance had no effect on the result.

We can see that four tasks can access the volume concurrently using the four PAV addresses. With more tasks than aliases, the I/O rate no longer improves. The data shows an ESCON channel limit of about 1000 operations per second for this workload.

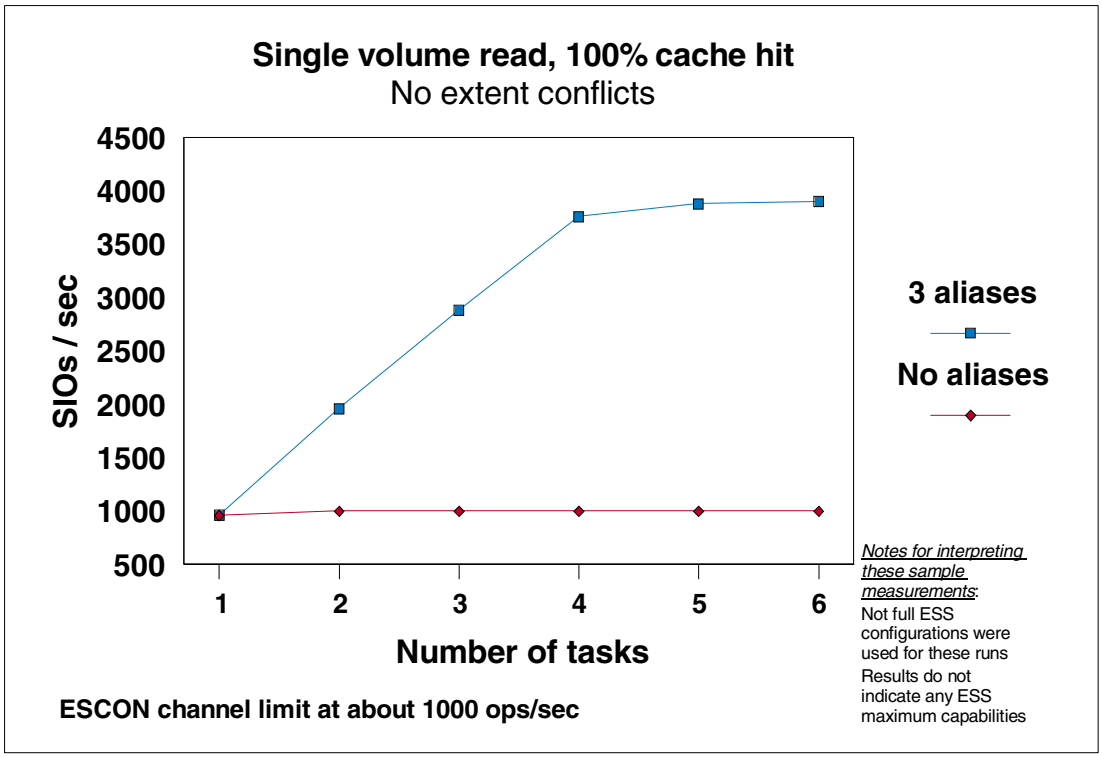


Figure 9-9 Effect of PAVs on single volume reads

Figure 9-10 on page 324 shows the effect of PAVs on simultaneous reads. The test setup used eight simultaneous read streams against a single data set on a volume. All streams had 100 percent cache hit ratios so backend disk performance does not play a role in the measurements.

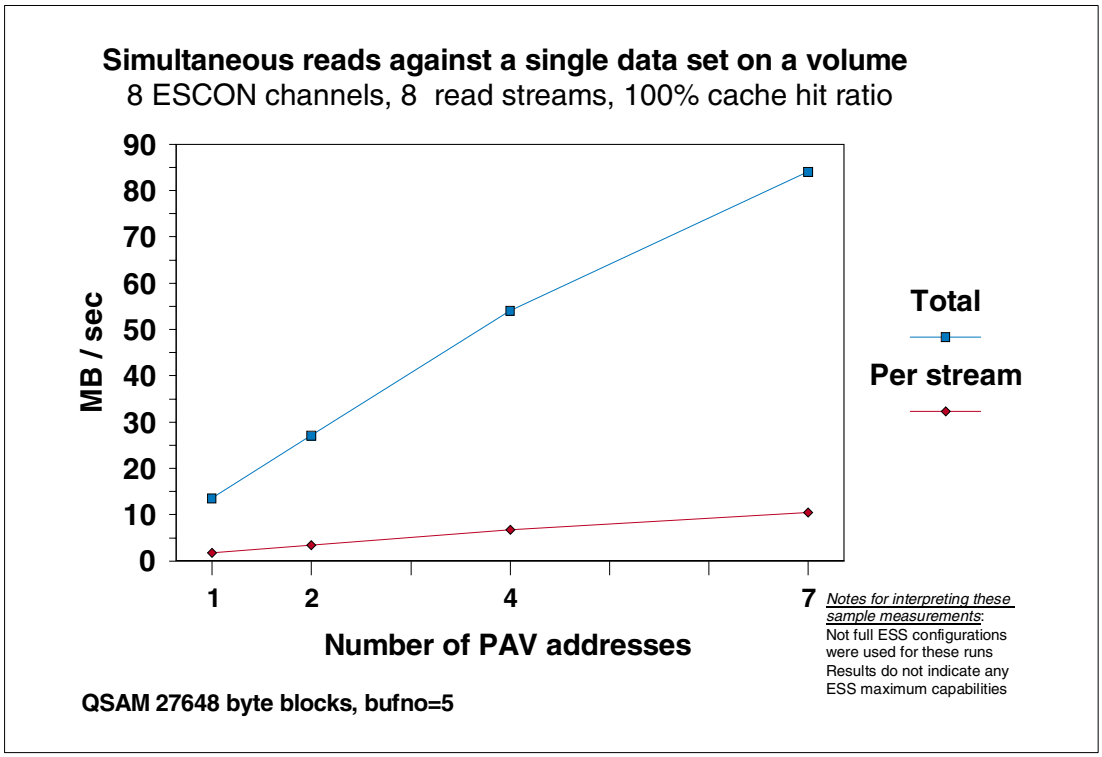


Figure 9-10 PAV effect on simultaneous reads

You can see that as the number of PAV addresses for the volume increases, the total data rate and also the per stream data rate increase almost proportionally. In reality, when cache hit ratios are less than 100 percent, the backend disk performance will at some point become a bottleneck, after which adding more aliases will provide little or no benefit.

9.3 Multiple Allegiance

Normally, if a zSeries host image (server or LPAR) does an I/O request to a device address for which the storage subsystem is already processing an I/O that came from another zSeries host image, then the storage subsystem will send back a *device busy* indication and the I/O has to be retried. This delays the new request and adds to processor and channel overhead (this delay is reported in the RMF Device Activity Report PEND time column).

The IBM TotalStorage Enterprise Storage Server accepts multiple parallel I/O requests from different hosts to the same device address, increasing parallelism and reducing channel overhead.

Previously to the ESS, a device had an implicit allegiance, that is, a relationship created in the disk control unit between the device and a channel path group when an I/O operation was accepted by the device. The allegiance caused the control unit to guarantee access (no busy status presented) to the device for the remainder of the channel program over the set of paths associated with the allegiance.

With Multiple Allegiance (MA), the requests are accepted by the ESS and all requests will be processed in parallel, unless there is a conflict when writing data to a particular extent of the CDK logical volume. Still, good application access patterns can improve the global parallelism by avoiding reserves, limiting the extent scope to a minimum, and setting an appropriate file mask, for example, if no write is intended.

In systems without Multiple Allegiance, all except the first I/O request to a shared volume were rejected, and the I/Os were queued in the zSeries channel subsystem, showing up as PEND time in the RMF reports.

Multiple Allegiance provides significant benefits for environments running a sysplex, or zSeries systems sharing access to data volumes. Multiple Allegiance and PAV can operate together to handle multiple requests from multiple hosts.

The ESS ability to run channel programs to the same device in parallel can dramatically reduce the IOSQ and the pending response time components in shared environments.

In particular, different workloads—for example, batch and online—running in parallel on different systems can have an unfavorable impact on each other. In such cases, ESS's Multiple Allegiance can dramatically improve the overall throughput. Figure 9-11 shows an example of a DB2 data mining application running in parallel with normal database access.

Performance Environment	Host 1 Online (4K read hits)	Host 2 Data Mining (32 record 4K read chains)
Max ops/sec Isolated	767 SIOs/SEC	55.1 SIOs/sec
Concurrent	59.3 SIOs/SEC	54.5 SIOs/sec
Concurrent with Multiple Allegiance	756 SIOs/SEC	54.5 SIOs/sec

Figure 9-11 Benefits of Multiple Allegiance for mixed workloads

The application running long CCW chains (Host 2) drastically slows down the online application (Host 1) when both applications try to access the same volumes concurrently (middle row in Figure 9-11), as compared to when they were operating isolated (first row in Figure 9-11). But with ESS's support for parallel I/Os, both applications can run concurrently with almost the same throughput as when they were running isolated, as the measurements in the last row of Figure 9-11 illustrate.

Multiple Allegiance is a hardware function. It requires no software tuning.

9.4 I/O priority queuing

The IBM TotalStorage Enterprise Storage Server can manage multiple channel programs concurrently, as long as the data accessed by one channel program is not altered by another channel program.

If I/Os cannot run in parallel, for example, due to extent conflicts, and must be serialized to ensure data consistency, the ESS will internally queue I/Os. Contrast this with the less efficient procedure used by traditional disk storage subsystems, which responded by posting a device busy status to the operating system, which then had to re-drive channel programs.

This subsystem I/O queuing capability provides significant benefits:

- Compared to the traditional approach, I/O queuing in the storage subsystem eliminates the overheads associated with posting status indicators and re-driving channel programs.
- Channel programs that cannot execute in parallel are processed in the order they are queued. A fast system cannot monopolize access to a device also accessed from a slower system. Each system gets a fair share.

The ESS can queue I/Os from different z/OS system images in a priority order. z/OS Workload Manager can make use of this and prioritize I/Os from one system against the others. You can activate I/O Priority Queuing in WLM Goal mode with the I/O priority management option in the WLM's Service Definition settings (see Figure 9-2 on page 312).

When a channel program with a higher priority comes in and is put in front of the queue of channel programs with lower priority, the priority of the low-priority programs is also increased. This prevents high-priority channel programs from dominating lower priority ones and gives each system a fair share.

9.5 Logical volume sizes

The ESS initially supported CKD logical volumes of any size from one cylinder up to 10017 cylinders. As it became possible for the user to flexibly select volume sizes, the term *custom volume* was introduced to denote a volume whose size did not match the size of the standard real devices such as the 3339 cylinders of the 3390 Model 3, or the 10017 cylinders of the 3390 Model 9.

The ESS also supports large volumes of up to 32760 cylinders, approximately 27.8 GB, for which there is no real counterpart. With the trend towards using more of these large volumes, we are seeing more and more ESS configurations where most, if not all, of the volumes are custom volumes in the sense that their size does not match that of a standard real device.

Despite the fact that both the smaller than and the larger than the standard size volumes are custom volumes, in general, the term *custom volume* is more generally used to refer to the smaller than the standard size volumes and the term *large volume* is more generally used to refer to the larger than the standard size volumes.

The ESS large volume support became available in late 2001. The enhancement was provided as a combination of ESS Licensed Internal Code (LIC) and system software changes on z/OS and z/VM™.

9.5.1 Selecting the volume size

A key factor to consider when planning the CKD volumes configuration and sizes is the 256 device limit per LSS. You need to define volumes with enough capacity so that you can use all your installed capacity with at most 256 devices. On ESCON-attached systems the number of devices can be even smaller, 128 or even 64, due to ESCON constraints. If using PAV, a part of the 256 addresses will be used for aliases.

When planning the configuration, you should also consider future growth. If you fully configure an LSS with 256 volumes, you will not be able to later add ranks in the LSS without redefining the existing volumes.

Figure 9-12 on page 327 shows the number of different size logical volumes (LVs) that can be defined on up to three RAID-5 ranks of different DDM sizes. LV in the figure refers to a large volume of 32760 cylinders.

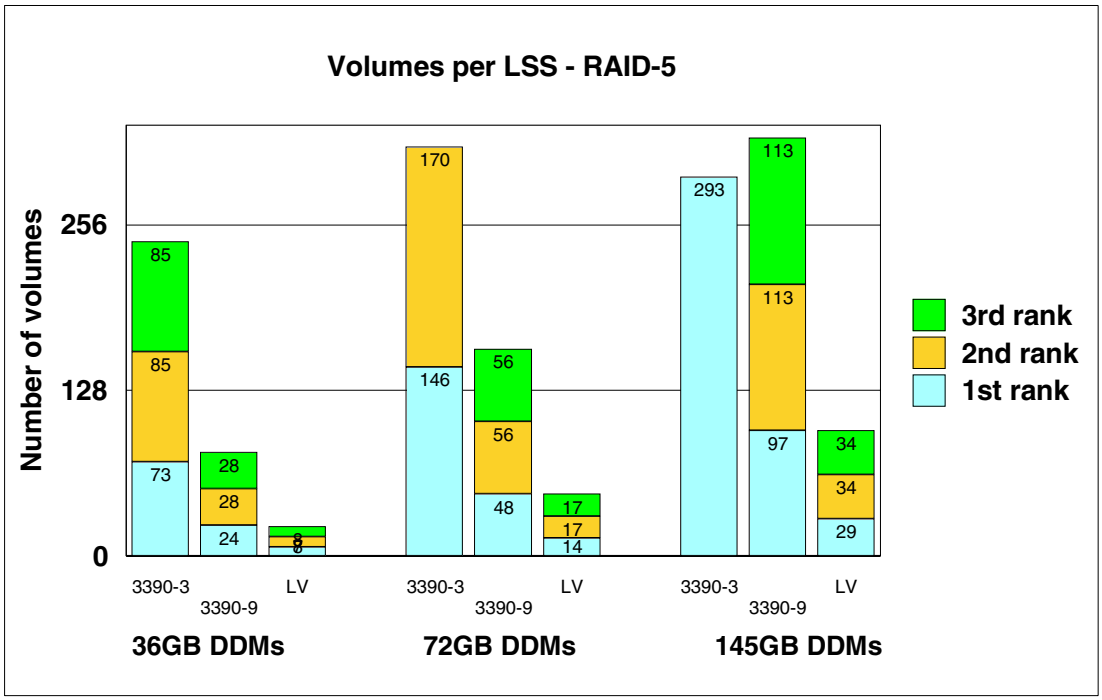


Figure 9-12 Logical volumes per LSS

You can see in Figure 9-12 that the standard 3390-3 volumes are too small for the larger DDM arrays. For example, you cannot use all capacity of one 145 GB array with the standard 3390-3 volumes. Even the standard 3390-9 is too small for an LSS with three 145 GB DDM ranks. If you have ordered an ESS with more than sixteen 72 GB disk eight-packs and configured them as RAID-5, this results in an effective capacity of 7.7 TB (see 2.6.3, “Disk eight-pack capacity” on page 24), with some of the LSSs having more than one rank. If you refer to Figure 9-12 you can see that the volumes that fit in those ranks cannot all be configured as the standard 3390-3, because you would be exceeding the 256 limit.

As ESS capacities grow larger, you need to configure larger logical volumes in order to use the available capacity while not exceeding the maximum 256 device addresses that the LSS provides. Large volumes relieve architectural constraints by allowing you to address more data within the existing 256 device per LCU limit.

The ESS does not serialize I/O on the basis of logical devices, so logical volume size does not affect ESS backend performance. Host operating systems, on the other hand, serialize I/Os against devices. As more data sets reside on a single volume, there will be greater I/O contention accessing the device. With large volumes it is important to try to minimize contention on the logical device level.

We have already seen in 9.2, “Parallel Access Volumes” on page 308, how PAV and large volumes together help minimize IOSQ times and provide the best overall performance for your ESS. As we have seen, PAV is a powerful function that allows you to reduce or eliminate contention on the ESS volumes and is especially valuable when you start using large volumes.

If your are not using PAVs, then decisions on logical volume sizing are effectively a trade-off between minimizing volume contention and maximizing contiguous available free space within the addressing limits. Small volume sizes help reduce contention on the volume. You can spread high-activity data sets on separate smaller custom volumes, or even give each high-activity data set its own small custom volume.

Using large custom volumes without PAV may cause contention problems unless data access rates are low. This applies, for example, to z/VM and guest systems running under it. z/VM supports large volumes, but since it does not support PAV for other than z/OS guests, you should be careful with large volumes.

9.5.2 Larger vs. smaller volumes performance examples

The performance of configurations using larger custom volumes as compared to an equal total capacity configuration of smaller volumes has been measured using various online and batch workloads. In this section we include some measurement examples that can help you evaluate which could be the performance implications of using larger volumes.

Random workload

The measurements for DB2 and IMS™ online transaction workloads in our measurements showed that there was only a slight difference in device response time between a six large custom volumes vs. a sixty smaller standard volumes configuration of equal capacity. The measurements for DB2 are shown in Figure 9-13. It should be noted that even when the device response time for a large volume configuration was higher, the online transaction response time could sometimes be lower due to the reduced system overhead of managing fewer volumes.

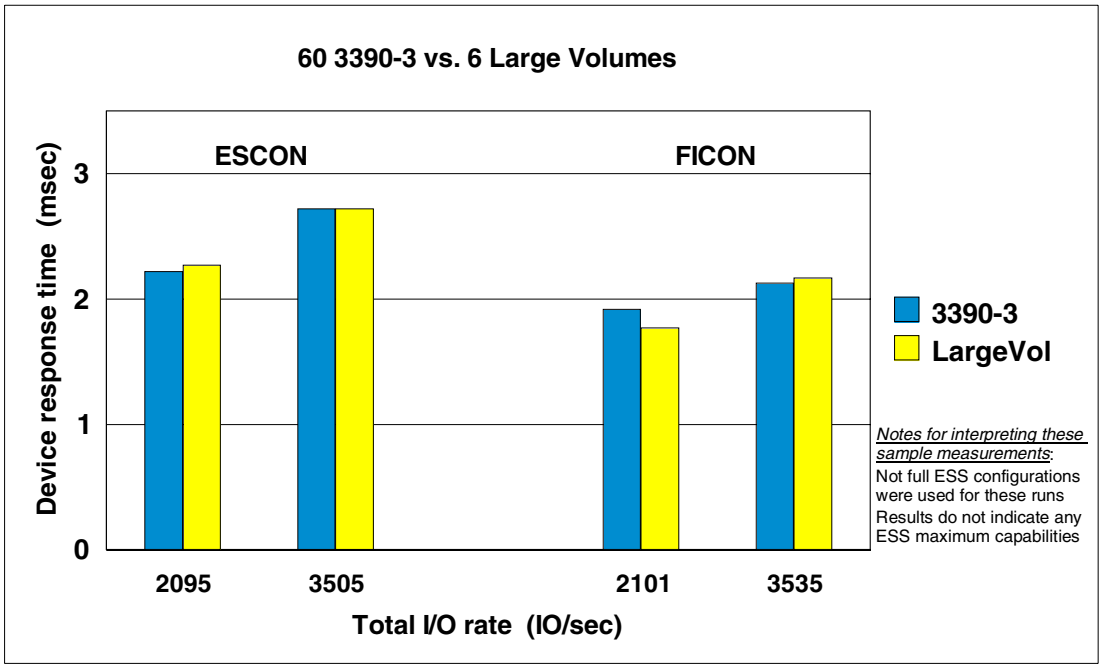


Figure 9-13 DB2 workload - Large volume vs. 3390-3 device response time comparison

The measurements were carried out so that all volumes were initially assigned zero or one PAV alias. WLM dynamic alias management then assigned additional dynamic aliases as needed. The number of aliases at the end of the test run reflects the number that was adequate to keep IOSQ down. For DB2, the alias assignment done by WLM resulted in an approximately 4:1 reduction in UCBs (see Figure 9-14 on page 329). The results for IMS were much the same.

The pool of unused aliases was in each case large enough so that WLM could assign new aliases when needed, without having to move them from one base to the other. The measured response times, therefore, include practically no IOSQ component.

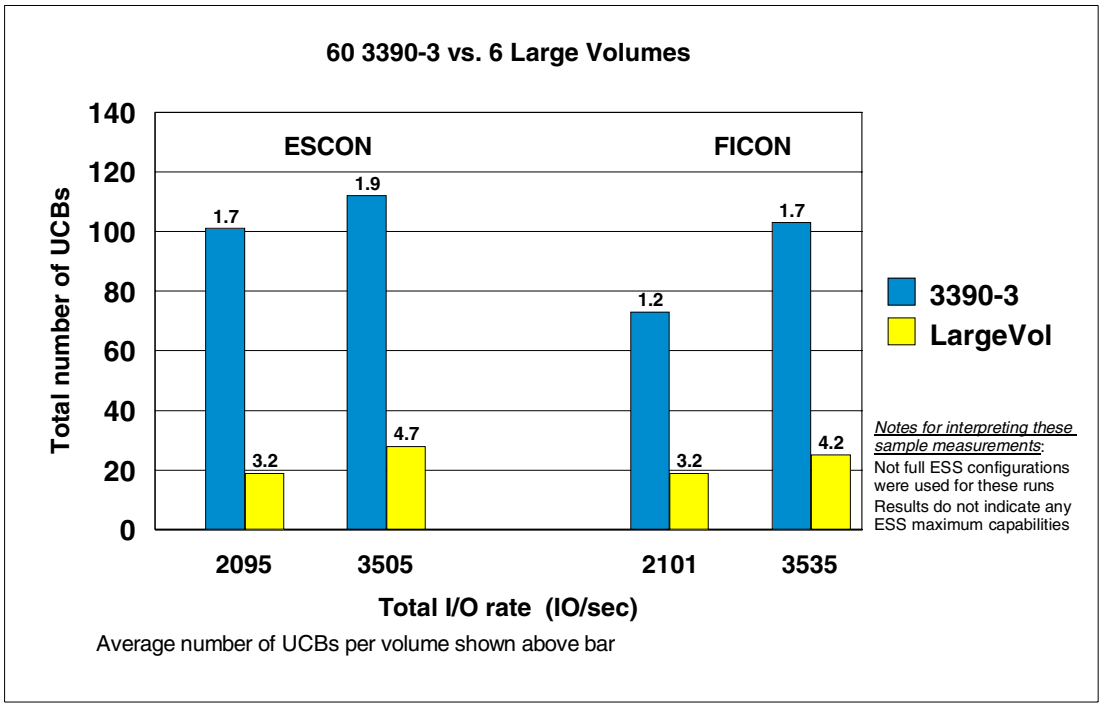


Figure 9-14 DB2 number of UCBs comparison

The number of parallel addresses per volume is shown above the bars in Figure 9-14.

It is worth noting that even at the high access densities in the measurement, less than five PAVs were required per volume. At the lower I/O rates in the figure only 3.2 PAVs were used. This again confirms that more than three aliases are rarely needed.

We can compare the benchmark results with the model results in 9.2.7, “PAV and large volumes” on page 316. Take, for example, the large volume measurement for ESCON in Figure 9-13 on page 328. The 3505 IO/sec rate with approximately 2.7 ms response time corresponds to about 16 percent average utilization for a 3390-3 base volume. In the benchmark the workload acquired 4.7 UCBs. By looking at Figure 9-5 on page 319 we see that, according to the model, approximately four PAVs would be needed for a large volume to keep the queue length below 0.05 at 16 percent utilization. This is somewhat smaller than the actual measured value.

Sequential workload

Figure 9-15 on page 330 shows elapsed time comparisons between nine 3390-3s precessed sequentially versus one large volume when a DFSMSdss full volume physical dump was taken, and when a DFSMSdss 2 MB sequential data set logical dump operation was done. The workloads were run on a 9672-XZ7 processor connected to an ESS with eight FICON channels. The volume(s) were dumped to a single 3590E tape with an A60 Control Unit with one FICON channel. No PAV aliases were assigned to any volumes for this test.

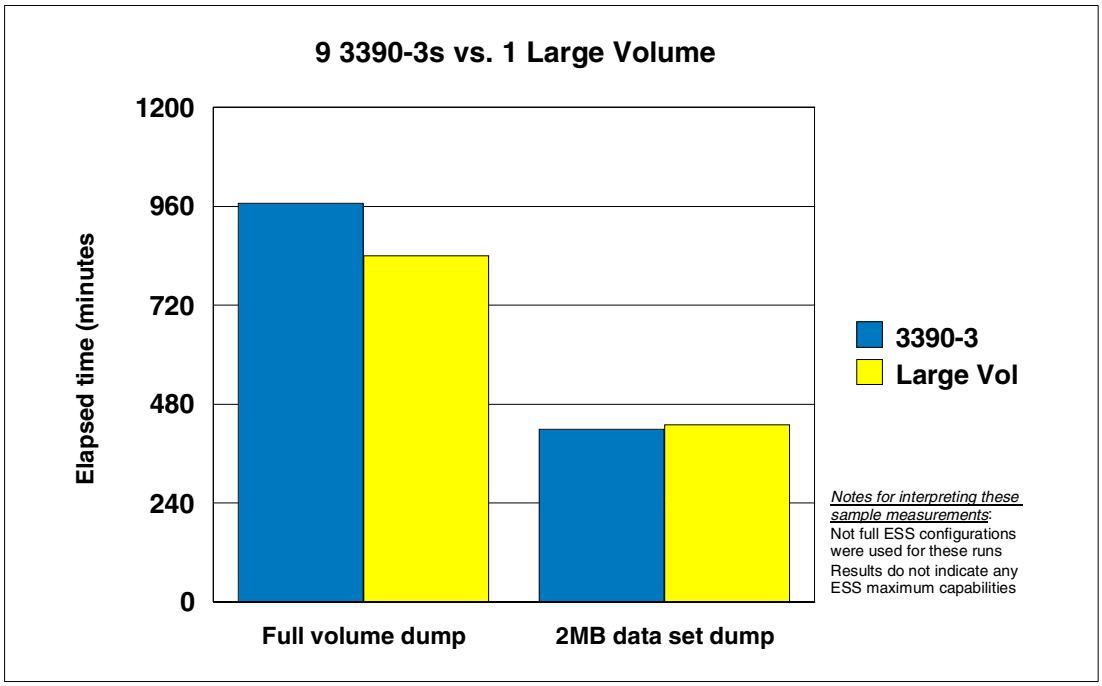


Figure 9-15 DFSMSdss dump to 3590E tape - Elapsed time comparison

The performance improvement in the large volume dump is due mainly to a decrease in overhead when processing one volume versus nine volumes. The results for the data set dumps can be considered equal.

9.5.3 Planning the volume sizes of your configuration

The great flexibility of selecting volume sizes may easily lead to an unwanted mixture of different sizes, if careful planning is not done ahead. For example, you can define fourteen 32760 cylinder large volumes on a 72 GB 6+P rank, but 30578 cylinders will be left over. You might be tempted to use a smaller volume size, which would allow you to define fifteen volumes on the rank. However, the size that is optimum for a 72 GB 6+P rank may not be the optimum for the ranks you install later.

From a simplified storage management perspective, we recommend that you select and use a uniform volume size for the majority of your volumes. With a uniform volumes size configuration in your ESS you do not have to keep track of what size each of your volumes is. Several functions, such as FlashCopy, PPRC, and full volume restores, require that the target volume cannot be smaller than the source. This simplifies and avoids mistakes in you storage administration activities.

Larger volumes

To avoid potential I/O bottlenecks when using large volumes you may also consider the following recommendations:

- Use of PAVs to reduce IOS queuing.

  Parallel Access Volume (PAV) is of key importance when using large volumes. PAV enables one z/OS system image to initiate multiple I/Os to a device concurrently. This keeps IOSQ times down even with many active data sets on the same volume. PAV is a practical “must” with large volumes. In particular, we recommend using dynamic PAVs.

- Eliminate unnecessary reserves by using WLM in GOAL mode.

As the volume sizes grow larger, more data and data sets will reside on a single CKD device address. Thus, the larger the volume, the greater the multi-system performance impact will be when serializing volumes with RESERVE processing. You need to exploit a GRS Star Configuration and convert all RESERVE's possible into system ENQ requests.

- Multiple allegiance will automatically reduce queuing on sharing systems.
- Some applications may use poorly designed channel programs that define the whole volume as their extent range, instead of just the actual cylinders where the I/O operates. This prevents concurrent write I/Os from other applications to the volume even when PAV is used. You need to identify such applications and allocate the data sets on volumes where they do not conflict with other applications. Custom volumes are an option here.

Other benefits of using large volumes can be briefly summarized as follows:

- Simplified storage administration
- Larger pools of free space thus reducing number of X37 abends and allocation failures
- Reduced number of multivolume data sets to manage

Smaller volumes

As we have already mentioned, the IBM TotalStorage Enterprise Storage Server is able to do several concurrent I/O operations to a logical volume thanks to its unique PAV and Multiple Allegiance functions. This drastically reduces or eliminates IOS queuing and pending times in the z/OS environments.

If you are not using PAVs you can still reduce or eliminate contention by defining small custom volumes to allocate specific data sets. You may have some data sets with high write rates and relatively small size that will benefit from being allocated on standard size 3390-3 volumes, or smaller custom volumes.

Although defining relatively small or tailored-to-fit custom volumes may be initially appealing if you are not using PAVs, smaller custom volumes have administrative costs both in the overhead of having to manage more devices and the cost of using extra UCBs. Every custom volume reduces the pool of aliases that is available for the majority of your volumes.

9.6 FICON

FICON provides several benefits as compared to ESCON, from the simplified system connectivity to the greater throughput that can be achieved when using FICON to attach the host to the IBM TotalStorage Enterprise Storage Server.

FICON allows you to significantly reduce the batch window processing time. Response time improvements may accrue particularly for data stored using larger block sizes. The data transfer portion of response time is greatly reduced because the data rate during transfer can be six times faster with 1 Gbit FICON than ESCON. This improvement leads to significant reductions in the connect time component of the response time. The larger the transfer, the greater the reduction as a percentage of the total I/O service time.

The pending time component of the response time, that is caused by director port busy, is totally eliminated because collisions in the director are eliminated with FICON architecture. For users whose ESCON directors are experiencing as much as 45–50 percent busy conditions, this will provide significant response time reduction.

FICON channels can process multiple concurrent data transfers, whereas ESCON channels process only one operation at a time. PAV and FICON working together allow multiple data

transfers to the same volume at the same time over the same channel, providing greater parallelism and greater bandwidth while simplifying configurations.

Another performance advantage delivered by FICON is that the ESS accepts multiple channel command words (CCWs) concurrently without waiting for completion of the previous CCW. This allows setup and execution of multiple CCWs from a single channel to happen concurrently. Moreover, I/O priority queueing is now handled at a "higher" point in the ESS system. Contention among multiple I/Os accessing the same data is now handled in the FICON host adapter, and queued according to the I/O priority indicated by the Workload Manager.

Significant performance advantages can be realized by users accessing the data remotely. FICON eliminates *data rate droop effect* for distances up to 100 km for both read and write operations by using enhanced data buffering and pacing schemes.

FICON thus extends the ESS's ability to deliver high bandwidth potential to the logical volumes needing it, when they need it. Older technologies are limited by the bandwidth of a single disk drive or a single ESCON channel, but FICON, RAID, and PAVs working together provide a high-speed pipe with multiplexed operation all the way down to your important data.

To illustrate the benefits associated with FICON, in this section we review examples of batch jobs doing QSAM and VSAM sequential processing, DB2 queries, and DSS volume dumps. These specific forms of sequential processing are all important in themselves, and also provide an indication of what level of performance improvement might be expected for sequential processing in general.

Note: The performance measurements in this section were obtained on an ESS Model F20 with 1 Gbit FICON adapters using (in most cases) a zSeries900 processor. The 2 Gbit FICON adapter with the ESS Model 800 would provide twice the bandwidth compared to the 1 Gbit adapter.

For additional information on FICON, see 5.4, “FICON” on page 132.

Benefits for QSAM/VSAM

QSAM and VSAM are arguably the most widely used access methods for general application processing. In the tests of QSAM and VSAM sequential performance, buffering was set so as to achieve transfer sizes per I/O of 2.5 tracks for QSAM, and 1 track for VSAM (5 buffers and 24 buffers, respectively, with half of the VSAM buffers being used for any given I/O), with neither the cache size nor the exact number and type of disk drives having a significant effect on measured ESS sequential throughputs.

Figure 9-16 on page 333 illustrates the improvements that can be obtained when using FICON. This example shows how FICON can dramatically reduce the elapsed time to run sequential jobs. For example, the elapsed time for reading a large QSAM data set is reduced by a factor of 33.7/13.9 = 2.4; the elapsed time for reading a large VSAM KSDS data set in key sequence is reduced by a factor of 19.8/10.7 = 1.9. Consider that for these examples the 1 Gbit FICON adapters were used, and not the most powerful 2 Gbit adapters. The block size was 27 KB for QSAM, 4 KB for VSAM. The transfer size per I/O was 2.5 tracks for QSAM, 1 track for VSAM.

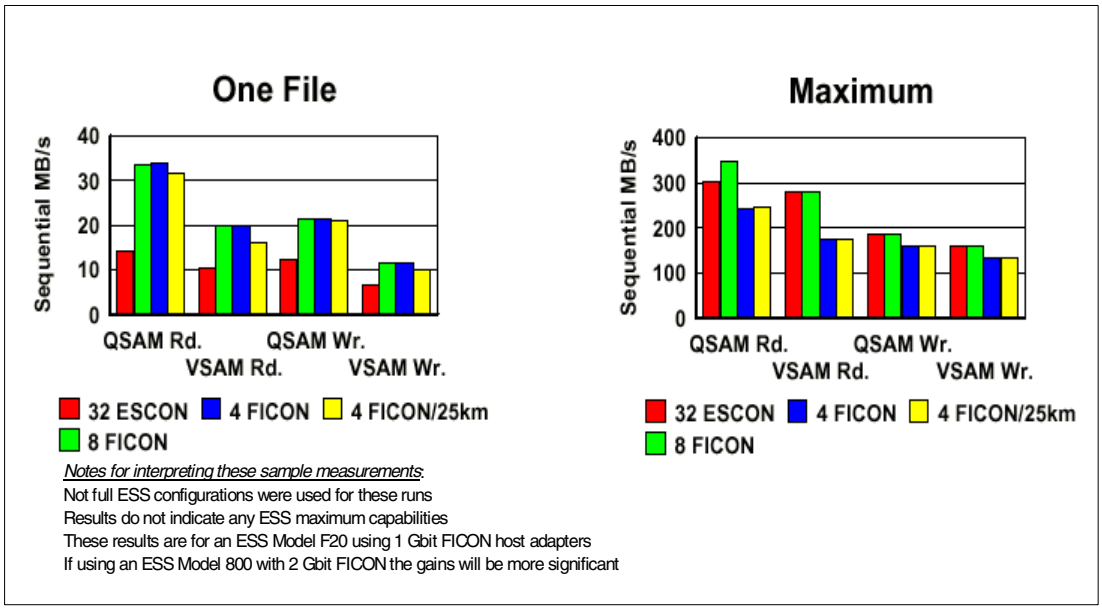


Figure 9-16 QSAM/VSAM sequential throughput - ESCON vs. FICON

Figure 9-16 also illustrates the total ESS sequential throughput capability when using either four or eight FICONs. These results are representative of the drastic gains that can be obtained when the sequential processing is done using FICON channels. Consider that the gains you see in Figure 9-16 are for an ESS model F20 using 1 Gbit FICON host adapters. These gains will be more significant if you are using an ESS Model 800 with 2 Gbit FICON host adapters.

Benefits for DB2

Large-scale DB2 queries, such as data warehouse processing, are among the specific types of batch work that can benefit dramatically from FICON. Two sets of measurements were done. These were obtained against a set of very large database scans. The scans were performed against both a non-partitioned database, as well as a database divided into 60 partitions.

Figure 9-17 on page 334 presents the data rates obtained during both the non-partitioned and the partitioned scans. As already recommended, the number of ESS FICON ports needed for bulk data movement should be estimated directly from the required data rates. The example in Figure 9-17 on page 334 provides the basis for such estimates in the case of large-scale DB2 queries.

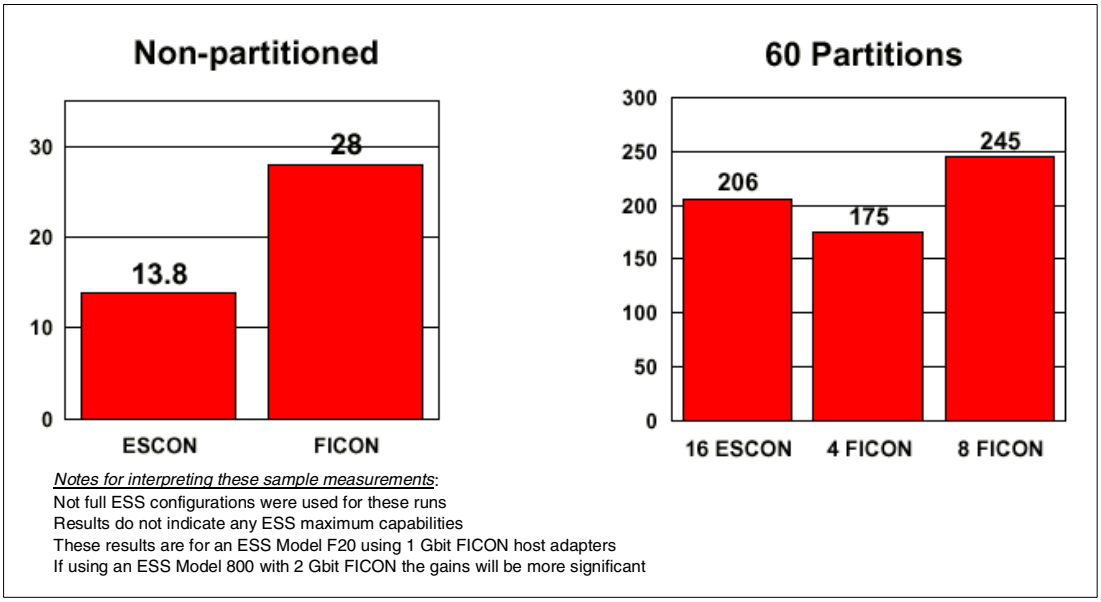


Figure 9-17 Throughput (MB/sec) for large-scale DB2 queries

It is important to note that the throughput for reading a non-partitioned database with FICON is double that with ESCON, and substantially exceeds the corresponding throughput for reading a single VSAM file (shown in Figure 9-16 on page 333). The faster FICON performance for DB2 queries, compared to simple VSAM, is due to the media manager's ability to exploit new channel commands, added to the ECKD protocol to optimize ESS sequential performance.

Consider that the gains you see in Figure 9-17 are for an ESS model F20 using 1 Gbit FICON host adapters. These gains will be more significant if you are using an ESS Model 800 with 2 Gbit FICON host adapters.

Benefits for DSS volume dumps

In many installations, the time required to perform volume dumps is a substantial part of the overall batch window. Figure 9-18 on page 335 presents the sample results for DSS volume dump jobs. Data rates are shown, both for dumping a single volume and for dumping four concurrent volumes. All tests were performed by copying data from an ESS Model F20 to 3590 tape. The tape channel configuration consisted of either four ESCON or two 1 Gbit FICON. The figure compares three cases:

- ESCON throughout the configuration
- ESCON for disk, FICON for tape
- FICON throughout the configuration

The improvement in throughput for a single volume dump (a factor of 2.6 times) is particularly striking. Throughput also improved by more than a factor of two for four concurrent dumps.

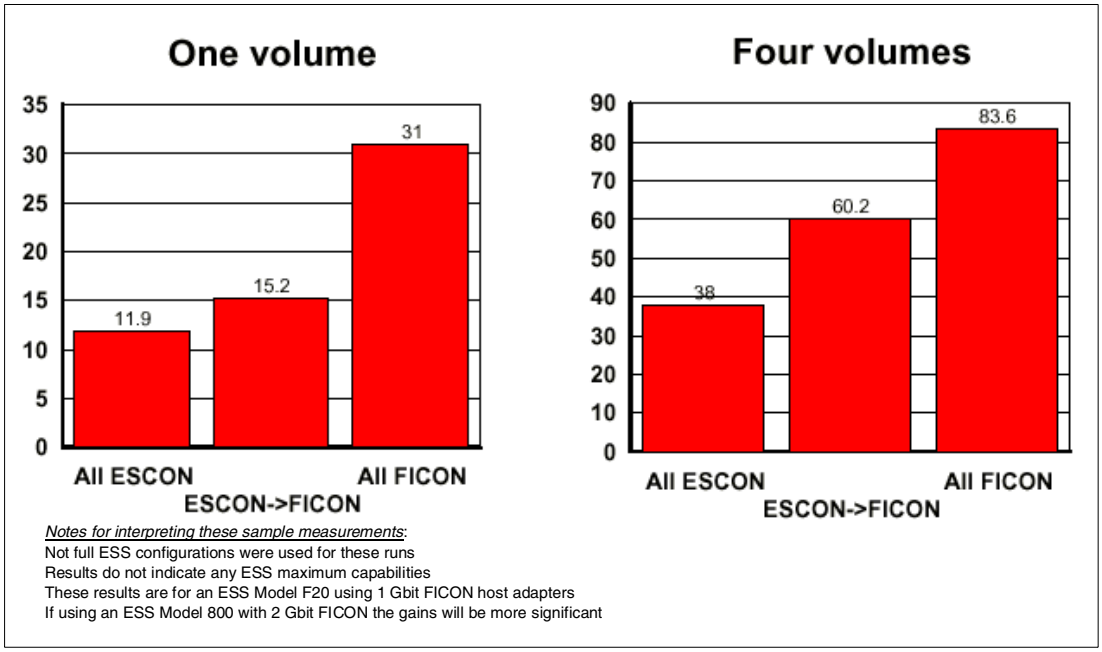


Figure 9-18 Volume dump data rates (MB/sec)

Consider that the gains you see in Figure 9-18 are for an ESS model F20 using 1 Gbit FICON host adapters. These gains will be more significant if you are using an ESS Model 800 with 2 Gbit FICON host adapters.

Database performance with FICON

Figure 9-19 on page 336 illustrates the ability of FICON to substantially improve database performance for some workloads. The results presented in the figure are based upon a benchmark workload, which executes a mix of DB2 transaction and query scripts. Due to the dramatically shorter time required to complete query requests, FICON delivers a significant improvement in the overall response time per I/O.

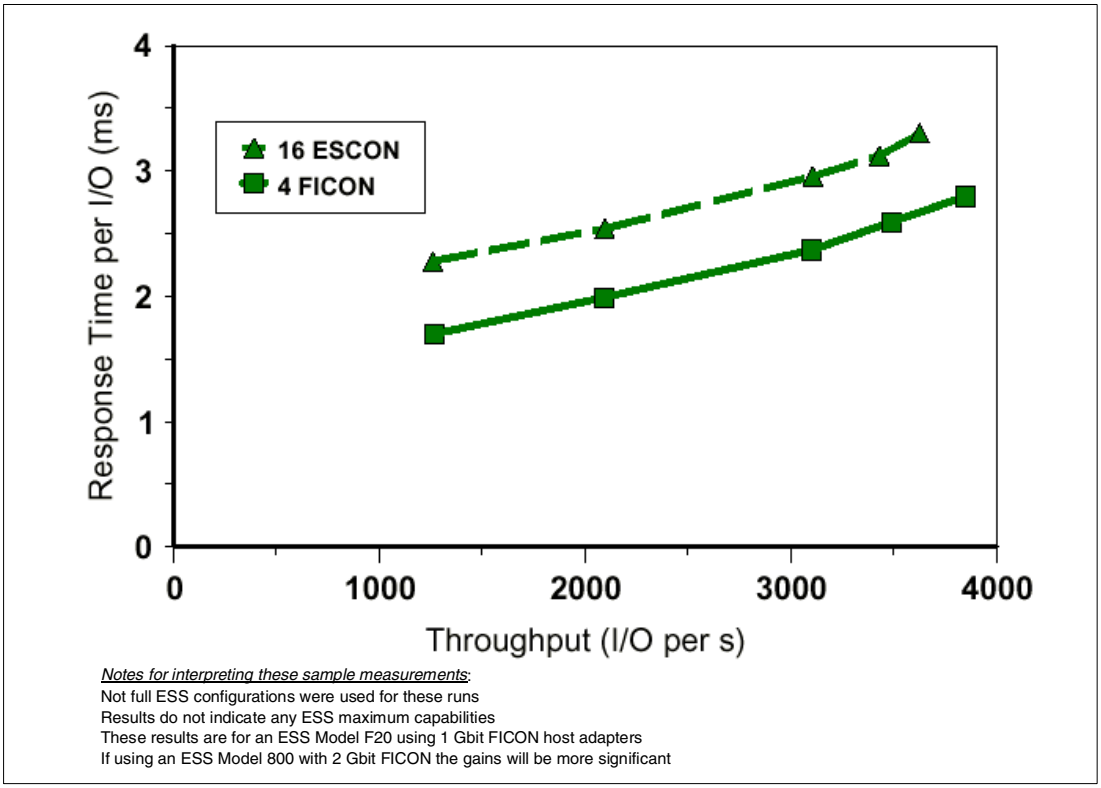


Figure 9-19 FICON DB2 benchmark

Consider that the gains you see in Figure 9-19 are for an ESS model F20 using 1 Gbit FICON host adapters. These gains will be more significant if you are using an ESS Model 800 with 2 Gbit FICON host adapters.

9.7 z/OS planning and configuration guidelines

This section discusses general configuration guidelines and recommendations for planning the ESS configuration. For a less generic and more detailed analysis that considers your particular environment, the Disk Magic and Sequential Sizer modeling tools are available to IBM personnel and business partners who can help you in the planning activities. Disk Magic can be used to help understand the performance effects of various configuration options, such as the number of ports and host adapters, disk drive capacity, number of disks, etc.

9.7.1 Channel configuration

Due to the number of possible configurations available, you need to observe some guidelines when configuring an ESS so that you do not make decisions that will limit the potential performance of the box. The following generic guidelines can be complemented with the information in Chapter 5, “Host attachment” on page 127:

- All control unit images are physically accessible over any installed ESCON or FICON physical paths. A FICON channel can address all 4096 devices in an ESS, while an ESCON channel can address only 1024 devices. This architecture limitation may cause you not to logically define all paths to all control unit images.
- Each control unit image can have from one to 256 devices. The ranges of supported device addresses may be non-contiguous. All defined devices, whether base devices or

alias devices, are allocated a UCB and count to the limit of 256 devices. To define your ESCON directors to z/OS as devices, we recommend that you choose channels that do not connect to ESS subsystems. Otherwise, you will only be able to define 255 ESS devices on one LCU.

- You need to remember that PAV and Multiple Allegiance have the effect of increasing the channel utilization by allowing I/Os that were previously serialized to process in parallel. You should plan to provide sufficient channel bandwidth to ensure that you receive the benefits from these functions.
- ESCON ports used for PPRC primary connections are not available for other uses. PPRC secondary ports may be used for host connections.
- For both PPRC and XRC you need to consider the amount of capacity that will be required to manage the additional movement of data that is required. You need to consider both the normal workload expected and the impact of re-synchronizing sessions after a disruption.

You need to consider both the number and location of channels that you connect to your ESS. The control unit images and channels can handle the interconnections listed in Table 9-2.

Table 9-2 FICON and ESCON connectivity comparison for ESS Model 800

	FICON	ESCON
Host adapter ports per ESS	16 (one per adapter)	32 (two per adapter)
Logical paths per host adapter port	256	64
Logical paths per LCU	256	256
Logical paths per ESS	4096	2048
Devices per channel	16384	1024

ESCON channels

A full ESCON configuration would use four eight-path groups; each path group would access one half (four) of the logical subsystems on a single ESS cluster. Each eight-path group would be connected to one ESCON host adapter per ESS bay (four bays per box). The ESS host adapter has two ports to accomplish this. The net result is that workload for a given path group is spread evenly across the ESS I/O bays. By having the channel path group access only one ESS cluster, read bandwidth is maximized when all channels are in use. Read bandwidth can be improved up to 15 percent when all channels are active and this cluster isolation strategy is employed. Keep in mind that the performance of any given path group in isolation is not affected by whether two LSSs on both clusters are actively in use.

The 32 ESCON channel configuration is difficult to implement for most users. Given no resource constraint, always configure at least 16 ESCON channels per z/OS system image to an ESS box. This rule applies to z/OS images sustaining substantial production work. If you cannot allocate 16 channels from a system image to the ESS then these rules cannot be strictly followed. For eight or fewer channels from a z/OS image, have all channels access all LSSs on the ESS in an n-way path group, where n is the number of channels available. You still want to spread the channels across as many host bays as possible. For 10 or more channels from an image, divide the channels into two sets and have each path group access LSSs on one cluster. Spread channels for each path group into all host adapter bays.

FICON channels

With ESS FICON support, the channel connectivity constraints have been relieved. The ESS FICON host adapter card is a single-port card, whereas the ESCON card has two ports. A full

FICON configuration would use two eight-path groups with each path group accessing LSSs on a single ESS cluster. Two FICON host adapters in each ESS bay should be used.

A 16 FICON channel configuration (like the 32 channel ESCON configuration) is also difficult for most users to implement. Fortunately, performance does not drop off markedly when using only eight FICON paths. With eight FICON channels the general performance recommendation is to use one eight-path group. This provides the best balance across the channel and host adapter resource. For workloads that demand the highest read bandwidth, then two four-path groups with ESS cluster isolation would provide a 15 percent read bandwidth improvement. In either case, the FICON channel paths should be spread evenly across the host adapter bays. If you have four FICON channels, use a single four-path group and use a ESS host adapter connection in each of the four ESS bays.

9.7.2 Balancing load for maximum throughput

Spreading I/O load for the performance-critical applications across the resources within an ESS subsystem (clusters, arrays, and device adapters) will maximize that application's throughput. Load balancing in general helps maximize the overall performance of the ESS subsystem. When attempting to balance load within an ESS, placement of application data is the determining factor. The following are the resources most important to balance, roughly in order of importance:

- Balance the activity among the RAID ranks. Use as many ranks as possible for the critical applications. Most performance bottlenecks occur because a few disks are saturated. Spreading an application across multiple ESS ranks allows for more parallelism.
- Balance the activity among the clusters. When selecting the ESS ranks for a critical application, spread them across separate clusters. Since each cluster has separate memory buses, cache, and NVS memory, this will maximize use of those resources.
- Balance the activity among the ESS device adapters. When selecting the ranks within a cluster for a critical application, spread them across separate SSA device adapters.
- Balance the activity among the host adapter bays. When selecting channels to assign to a given processor, spread them across the different adapter bays.

Note that load balancing between logical volumes within an ESS array does not significantly improve the I/O performance (although it may improve response times by eliminating some contention, for example, IOS queuing). All volumes on an ESS rank perform equally well (from the disk subsystem perspective).

Use the RMF RAID Rank Activity report to see how well the I/O load is balanced between the different RAID arrays. You can use the report to identify heavily loaded ranks that may be causing performance problems. The sample output in Example 9-1 shows measurements for an LSS with two ranks, 0000 and 0001.

Example 9-1 RMF RAID Rank Activity report

```
.                                   C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                                    PAGE    2
           OS/390                   SYSTEM ID FC01            DATE 02/18/2003           INTERVAL 14.59.526
           REL. 02.10.00            RPT VERSION 02.10.00      TIME 00.30.00

SUBSYSTEM  2105-01     CU-ID  B000     SSID 0400     CDATE  02/18/2003     CTIME 00.30.01      CINT  14.59
TYPE-MODEL 2105-F20
-------------------------------------------------------------------------------------------------------------------------------
                                                  RAID RANK ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------

ID    RAID     DA   HDD    -------- READ REQ -------     ------- WRITE REQ -------   ---------- HIGHEST UTILIZED VOLUMES ----------
      TYPE                 RATE  AVG MB  MB/S  RTIME     RATE  AVG MB  MB/S  RTIME

*ALL                14      94   0.037   3.5     17       31   0.023   0.7    101
0000  RAID-5   01     7      85   0.039   3.3     18       20   0.025   0.5    108   SAVRSM  GTS009  JPSSP9  DB2LO8  MCATP1  MBR398
```

9.7.3 Considerations for mixed workloads

When using the ESS, you may very likely combine data and workloads from several different kinds of independent servers onto a single ESS. Examples of mixed workloads include:

- z/OS and open
- Mission-critical and test

Sharing resources in an ESS has advantages from a storage administration and resource sharing perspective, but does have some implications for workload planning. Resource sharing has the benefit that a larger resource pool (for example, disk drives or cache) is available for critical applications. However, some care should be taken to ensure that uncontrolled or unpredictable applications do not interfere with mission-critical work.

If you have a workload that is truly mission-critical, you may want to consider isolating it from other workloads, particularly if those other workloads are less important (from a performance perspective), or very unpredictable in their demands. There are several ways to isolate the workloads:

- Place the data on separate ESS ranks. Note that z/OS and open system data will automatically be placed on separate arrays. This will reduce contention for use of disks.
- Place the data behind separate device adapters.
- Place the data behind separate ESS clusters. This will isolate use of memory buses, microprocessors, and cache resource. However, before doing that, make sure that a “half-ESS” provides sufficient performance to meet the needs of your important application. Note that Disk Magic provides a way to model the performance of a “half-ESS”. Consult your IBM representative for a Disk Magic analysis.

9.8 z/OS setup and usage guidelines

In this section we discuss some of the factors that need to be taken into account in an z/OS environment to ensure that your ESS provides optimal performance.

9.8.1 SMS considerations

From the SMS point of view, the ESS behaves like the control unit types that it emulates. There is no interaction between SMS and any of the ESS exclusive functions. You do need to consider the following SMS matters when defining your ESS.

Volume placement in storage groups

You need to decide how to use storage groups:

- Should you mix volumes from ESS and non-ESS subsystems?
- Should you mix volumes from multiple ESS subsystems?
- Should you mix volumes from multiple loops?

The decision you make about the mixing of volumes will probably be based on your current SMS configuration. However, there should be no reason to isolate single storage groups to loops or arrays. Potentially, the best performance will be achieved by spreading logical volumes as widely as possible. Since VOLSERs are defined to SMS, SMS will know only about base address and non-PAV volumes.

Storage class MSR values

The ESS uses its own internal cache management algorithms to maximize cache usage, as described in 2.5, “Cache and NVS” on page 17. In the past, smaller caches have been optimized by DFSMS/MVS® software in conjunction with a storage control function called Dynamic Cache Management Enhanced (DCME). This controlled cache usage in caches less than 1 GB in size using the Millisecond Response Time (MSR) value in the data class. There is no need to use DCME with the ESS, and it is not supported.

9.8.2 IDCAMS SETCACHE

The IDCAMS SETCACHE command allows the user to control caching on traditional cached storage controllers. However, on the ESS, caching, NVS, and DASD Fast Write (DFW) are turned on by default, and cannot be disabled. You will receive message `IDC31562I` with return code 12 indicating that a parameter is not available if you try to turn ESS caching off by any of the following SETCACHE commands:

- SETCACHE DFW OFF
- SETCACHE NVS OFF
- SETCACHE DEVICE OFF
- SETCACHE SUBSYSTEM OFF

9.8.3 IECIOSxx

MIH values are not supported for alias UCBs. Setting an MIH value will cause an error message (`IOS090I dev IS AN INCORRECT DEVICE`) to be generated.

MIH processing will only occur for base UCBs. The recommended MIH value for ESS volumes is 30 ms. The ESS devices are self-describing, so entries are not required for base addresses.

9.8.4 GTF

The z/OS Generalized Trace Facility (GTF) can be used to trace the flow of data and commands to specified devices. GTF should be run against base addresses and will automatically trace all activity to both the base and alias address.

9.8.5 S/390 device type

The ESS supports 3390-3, 3390-9, and 3390 with 3380 emulation device types. All device types are implemented the same way on the ESS (all are striped across several disks in a RAID array). The type of the logical volume defined has no impact on the performance that the ESS will deliver for that volume.

9.8.6 Extent reduction

As data sets are allocated and deleted, the free space on logical volumes gradually splits in small fragments over the volume. In order to rearrange the free fragments in larger, contiguous areas, volumes need to be regularly de-fragmented using tools such as the DFSMSdss DEFRAG command. The benefit is that new data sets can be allocated using fewer larger extents. On conventional volumes of previous disk subsystems this provided performance benefits, as the disk arm did not have to move between extents when processing a data set. For the same reason, tools like DFSMShsm™ are used to reduce the number of extents for existing data sets.

Due to large cache sizes, and the fact that data is striped on the ESS ranks, extent consolidation for the ESS logical volumes does not provide performance benefits. However, you should still run defrags to consolidate free space in order to avoid data set allocation failures because not enough space can be allocated using the maximum allowed number of extents.

9.9 Linux on zSeries

Native FICON connection over short-wave or long-wave fibre links is available for zSeries or S/390 servers running the Linux operating system. FICON extends the ESS's ability to deliver high bandwidth potential to the logical volumes needing it, when they need it. Older technologies are limited by the bandwidth of a single disk drive or a single ESCON channel, but FICON provides a high-speed pipe with multiplexed operation.

For further discussion on how to implement the ESS with Linux you can refer to *Implementing Linux with IBM Disk Storage*, SG24-6261.

9.10 ESS performance monitoring tools

The ESS is a high-performance disk storage solution developed using IBM's Seascape architecture. It takes advantage of IBM's leading technologies. Although the ESS provides unprecedented high performance, performance monitoring is still an important activity that allows you to fully exploit the ESS capabilities.

Two tools that will help you in monitoring and obtaining performance reports of your z/OS systems are:

- Resource Measurement Facility (RMF)
- IBM TotalStorage Expert

9.10.1 RMF

RMF provides performance information for the ESS and other disk subsystems for the z/OS users. RMF Device Activity reports account for all activity to a base and all its associated alias addresses. Activity on alias addresses is not reported separately, but RMF will report the number of PAV addresses (or in RMF terms, exposures) that have been used by a device, and whether the number of exposures has changed during the reporting interval.

RMF cache statistics are collected and reported by a logical control unit (LCU). So a fully configured ESS would produce 16 sets of cache data. One report only reflects the status of one logical subsystem. In other words, if you would like to check out the status of the whole cache, you have to check out all of the 16 reports.

With any RMF reports, the I/O information reported is from one z/OS system only. If you are sharing I/O across multiple systems, you will need to review RMF reports from each of the sharing systems in order to see the complete I/O activity to the LCUs and devices. This is not true for the Cache reports, however; since cache data is obtained from the control unit, it does include I/O activity from all sharing systems. Contention from a sharing system will generally be seen as increased pending (PEND) and disconnect (DISC) times in the Device Activity report.

9.10.2 IBM TotalStorage Expert

The IBM TotalStorage Expert provides detailed ESS performance and configuration information. Performance information provided includes:

- Disk array utilizations
- I/O rates
- Data transfer rates
- Cache hit rates

The ESS Expert component of the IBM TotalStorage Expert reports are based on the physical devices that comprise the ESS. ESS Expert should not be considered as a replacement to traditional monitoring and problem determination activities in an z/OS environment, but a very useful complement. Operating system based measurements are still required; however, using the ESS Expert allows you to see the activity of the whole system components, host, and disk subsystem. This is particularly more important when resources on the ESS are shared between different server types.

See 4.4, “IBM TotalStorage Expert” on page 104, for more information on the ESS Expert.

9.11 ESS performance monitoring for z/OS

In this section we discuss the basic tasks you will be undertaking when monitoring the performance of your ESS.

Before you can decide whether you have a performance problem you need to understand the normal workload of your system. This can only be done by regular monitoring. You need to define what is the acceptable performance for your ESS devices. Remember that different applications or disk pools may have differing performance requirements and that these requirements may differ during the day. The best way to determine what is good a response time (from the disk perspective) for your applications may be to measure what is actually happening in the ESS when the application is performing well and to use this as a reference benchmark.

Performance monitoring is the first step in the monitoring and tuning process. The second step is tuning, that is, when a problem exists, finding what caused it, and deciding what is the solution. RMF provides reports that enable you to monitor performance and potentially to discover the cause of performance problems. However, you should also consider other sources of information, for example, you should always check the syslog for messages or configuration changes that may have caused the problem.

Two key indicators for determining whether you have a performance problem are average response times and the number of I/O per second (throughput). RMF provides these figures on both a subsystem (LCU) basis and a device basis.

Figure 9-20 on page 343 illustrates one way of isolating the cause of a performance problem that you have either been informed of or have noticed during your monitoring. The steps illustrated in the figure should be used in conjunction with the RMF Direct Access Device Activity report as a starting point.

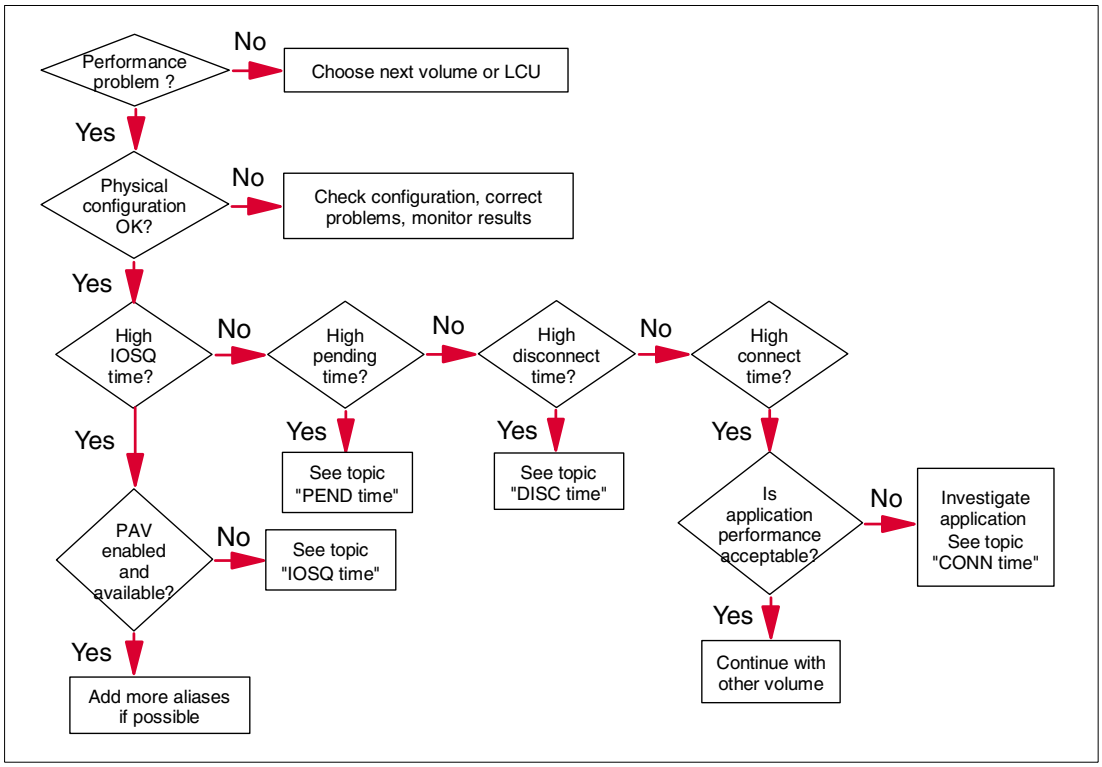


Figure 9-20 Steps in performance monitoring

In the following sections we discuss a sequence of events that you can follow when attempting to determine the cause of a performance problem. We also discuss some of the actions that should allow you to correct the problem. This section is not intended to be a comprehensive z/OS disk subsystem tuning guide, as each situation has its own peculiarities. We will concentrate on the issues that are unique to the ESS environment.

9.11.1 I/O operation sequence

The steps involved in processing an I/O operation (specifically in a z/OS environment) are described in Appendix B, “I/O terminology” on page 449. Understanding the process is useful both for tuning the ESS and for diagnosing potential performance problems.

In the I/O operation sequence there are numerous points where performance bottlenecks can be occurring:

- At the application level where poor application design can cause problems.
- Operating system problems may cause performance degradation, for example, a looping task may lock out other users, or a defect in an I/O driver may slow performance.
- Applications within the same z/OS may compete for access to the same data set. Sharing of data between users who have declared their intention to only read data is allowed; sharing between readers and writers is not. Non-compatible requests are queued and managed by Global Resource Serialization (GRS).
- Once access to a data set has been granted, z/OS has further access limited by allowing only one task to have I/O activity to a volume at any time. Other I/O operations from the same z/OS system image are placed in a queue. Queuing is based on the UCB and reported as IOSQ time. This potential bottleneck can be eliminated or greatly reduced by using PAV on the ESS.

- The storage subsystem will further limit concurrent access to a device for conflicting requests. Traditionally the channel subsystem (CSS) that requested access to a device was presented with a device busy status and the CSS had to re-drive the request. The ESS, on the other hand, will internally queue I/Os that cannot run in parallel, for example, due to extent conflicts. This delay is reported as pending (PEND) time. Multiple Allegiance on the ESS alleviates this potential performance bottleneck.

9.11.2 Where to start looking

If you are trying to determine why an application has poor response time, the first place to look is the response time on those volumes where these application's data is allocated. If these volumes do not show evidence of poor performance then you need to cast your net more widely. Alternatively, you can concentrate initially on those subsystems or volumes that show the worst response times.

You need to look not only at the response time of individual subsystems and volumes but also at the number of I/O operations that these are performing. A measure often used is to multiply the activity rate by the I/O response time to produce a number often called the I/O *intensity* or response time volume, measured in milliseconds per second. I/O intensity is a good measure of both how busy a subsystem or volume is and how important its response time is to your environment. An easy place to start is to pick the twenty devices or three subsystems with the top I/O intensity for your system and see if you can improve performance on these.

Once you have selected your targets you then need to check the average response time for these volumes. Is this value acceptable? If this value is acceptable, then start on the next volume. If not, then proceed as we discuss in the following points.

9.11.3 Check physical and logical components

Check that all the physical resources that the application volumes use are available. The following z/OS operator commands will help you:

- **D M=CONFIG(xx)**

  Are any errors reported in your current target configuration?

- **D U,,,dddd,1**

  Check that the device is online, not boxed or pending offline.

- **D M=DEV(dddd)**

  What is the state of the paths to the device? This command gives the operating system view of the paths.

- **DS P,dddd**

  Performs I/O to the device and checks to see whether the paths are physically available, and returns information about the state of subsystem resources such as cache and NVS.

- **F IHV,D D dddd ***

  What is the state of the ESCON links to the device? This command checks the current state of the paths through all switches that are in the path to the device down all channels.

- **D M=CHP(xx)**

  What is the state of paths to devices on this CHPID? Are there any unbound aliases on the LCUs that attach to these channels?

- **D M=CHP**

  What is the state of each CHPID to the device?

From the output of these commands look for error conditions, and correct them to see if the performance problem is eliminated. These commands will tend to highlight physical problems with the channels or paths to a device, although they may also highlight logical configuration errors. For more information about these commands and their interpretation, please refer to the publication *z/OS MVS System Commands*, SA22-7627.

Once you have checked that there are no problems with the physical resources that the volumes are using, you should check that the logical resources are also correct. Check the number of aliases assigned to the volume. The number of aliases is reported in the PAV column of the RMF Direct Access Device Activity report. If the number of aliases assigned to a base has been changed during the reporting interval, it is followed by an asterisk (*). Is the number of aliases bound to the base address adequate? Verify that the number reported by RMF is the number reported by the DEVSERV QPAVS command.

You can use the D M=CHP(xx) command to display the current status of base and alias device addresses; for example, the command in Example 9-2 shows base addresses, bound aliases, and unbound alias addresses. The most likely cause of an alias being unbound is that the device is defined to z/OS but has not been defined using ESS Specialist.

Example 9-2 DISPLAY MATRIX command output

```
DISPLAY M=CHP(C1)
 IEE174I 11.07.04 DISPLAY M 790
 CHPID C1:  TYPE=1A, DESC=FICON POINT TO POINT, ONLINE
 DEVICE STATUS FOR CHANNEL PATH C1
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
 C10 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
 C11 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
 C12 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
 C13 +  +  +  +  $@ $@ $@ $@ $@ $@ $@ $@ $@ $@ $@ $@
 C14 UL UL UL UL UL UL UL UL UL UL AL AL AL AL AL AL
 C15 AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL
 C16 AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL
 C17 AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL
 ************************ SYMBOL EXPLANATIONS ************************
 + ONLINE    @ PATH NOT VALIDATED   - OFFLINE    . DOES NOT EXIST
 * PHYSICALLY ONLINE   $ PATH NOT OPERATIONAL
 BX DEVICE IS BOXED           SN SUBCHANNEL NOT AVAILABLE
 DN DEVICE NOT AVAILABLE      PE SUBCHANNEL IN PERMANENT ERROR
 AL DEVICE IS AN ALIAS        UL DEVICE IS AN UNBOUND ALIAS
```

One potential problem you may see is alias addresses becoming boxed. You can use the DEVSERV QPAVS,xxxx,UNBOX command to resolve any boxed devices. PAV addresses may become boxed if changes are made using ESS Specialist while the devices are allocated on a z/OS system. For more information about the D M and DS QP commands when issued against disks allocated on an ESS, please see the publication *DFSMS/MVS Software Support for the IBM Enterprise Storage Server*, SC26-7318.

9.11.4 Analyze the response time components

Use the RMF Direct Access Device Activity report (see Example 9-3 on page 346) as a starting point in performance monitoring. Concentrate on the largest component of the response time (see 9.2.1, “Response time components” on page 308, for a discussion of the response time components). For selected volumes, drill down and identify the relative importance of each of the components of response time. Decide whether the effort required to reduce each of the components is worthwhile.

FICON brings some notable changes to I/O response times. Please refer to 9.11.7, “FICON RMF information” on page 349, for additional information on how to monitor FICON-attached ESS systems.

Example 9-3 on page 346 shows a sample RMF Direct Access Device Activity report.

Example 9-3 RMF Direct Access Device Activity report

```
.                              D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y
                                                                                                PAGE    1
           z/OS V1R2                SYSTEM ID XXXX            START 11/13/2001-09.00.00  INTERVAL 000.30.00
                                    RPT VERSION V1R2 RMF      END   11/13/2001-09.30.00  CYCLE 0.500 SECONDS

TOTAL SAMPLES =  3,600   IODF = 02   CR-DATE: 09/20/2001   CR-TIME: 15.48.19     ACT: POR

                                        DEVICE   AVG  AVG   AVG  AVG  AVG   AVG  AVG  AVG    %      %     %     AVG    %      %
STORAGE  DEV  DEVICE   VOLUME PAV  LCU  ACTIVITY RESP IOSQ  DPB  CUB  DB    PEND DISC CONN   DEV    DEV   DEV   NUMBER ANY    MT
 GROUP   NUM  TYPE     SERIAL          RATE     TIME TIME  DLY  DLY  DLY   TIME TIME TIME   CONN   UTIL  RESV  ALLOC  ALLOC  PEND

SYSTEEMI 9000 33909    SYS201   4  001C   4.887   1.5  0.0       0.0  0.0   0.3  0.2  1.0   0.12   0.14  0.0    501   100.0  0.0
YLEISET  9001 33909    PUB201   4  001C   0.661   1.3  0.0       0.0  0.0   0.3  0.3  0.7   0.01   0.02  0.0   13.7   100.0  0.0
YLEISET  9002 33909    PUB202   4  001C   0.206   1.5  0.0       0.0  0.0   0.3  0.5  0.7   0.00   0.01  0.0   31.7   100.0  0.0
YLEISET  9003 33909    PUB203   4  001C  11.675   1.8  0.0       0.0  0.0   0.3  0.8  0.7   0.21   0.44  0.0    3.0   100.0  0.0
YLEISET  9004 33909    PUB204   4  001C   6.689   1.1  0.0       0.0  0.0   0.3  0.3  0.5   0.08   0.13  0.0   20.9   100.0  0.0
YLEISET  9005 33909    PUB205   4  001C   0.132   2.4  0.0       0.0  0.0   0.3  1.3  0.8   0.00   0.01  0.0   13.0   100.0  0.0
YLEISET  9006 33909    PUB206   4  001C   0.066   4.3  0.0       0.0  0.0   0.3  2.3  1.7   0.00   0.01  0.0   18.9   100.0  0.0
TYO      9007 33909    WRK201   4  001C   0.001   0.8  0.0       0.0  0.0   0.4  0.0  0.4   0.00   0.00  0.0    0.0   100.0  0.0
YLEISET  9008 33909    PUB207   4  001C   0.359   7.6  0.0       0.0  0.0   0.4  3.4  3.8   0.03   0.06  0.0    7.1   100.0  0.0
YLEISET  9009 33909    PUB208   4  001C   0.004   3.3  0.0       0.0  0.0   0.3  2.7  0.4   0.00   0.00  0.0   25.9   100.0  0.0
YLEISET  900A 33909    PUB209   4  001C   0.214   2.8  0.0       0.0  0.0   0.3  2.0  0.5   0.00   0.01  0.0   17.3   100.0  0.0
YLEISET  900B 33909    PUB210   4  001C   0.045  13.0  0.0       0.0  0.0   0.3  5.5  7.2   0.01   0.01  0.0   19.0   100.0  0.0
YLEISET  900C 33909    PUB211   4  001C   0.960   1.0  0.0       0.0  0.0   0.3  0.2  0.5   0.01   0.02  0.0   11.1   100.0  0.0
YLEISET  900D 33909    PUB212   4  001C   1.319   2.6  0.0       0.0  0.0   0.3  1.5  0.8   0.03   0.07  0.0   16.1   100.0  0.0
```

Note: For the %DEV CONN and %DEV UTIL fields, the number presented in the RMF report is not the sum of the base + alias(es), but is the average of the activity on the base and alias(es). So this number cannot exceed 100 percent.

IOSQ time

This is the time measured when a request is being queued in the operating system.

If the problem is high IOSQ time then this can usually be solved by moving the data: Either separating active data sets or by moving active data to faster storage, for example, a coupling facility structure. On an ESS subsystem you have these additional alternatives:

- If PAV is not enabled for the device, enable it.
- If you are using static PAVs, assign more aliases to the device.
- If you are using dynamic PAV, then see if it is possible to define more aliases in the LSS. This may require HCD changes.
- Check to ensure that all PAVs that should be bound to the device are online and operational. You can use the DEVSERV QP and DS QP,xxxx,UNBOX commands to do this.

PEND time

Pending time represents the time an I/O request waits in the hardware.

A high pending (PEND) time suggests that the channel subsystem is having trouble initiating the I/O operation. There is a blockage somewhere in the path to the device. That might be due to:

- AVG DPB DELAY. Delay due to ESCON director ports being busy. This problem has been eliminated with FICON.
- AVG CUB DELAY. Delay due to the DASD control unit being busy, due to I/O from another sharing z/OS system. This problem has been eliminated with FICON.
- AVG DB DELAY. Delay due to the device being busy, due to another I/O. See “DB DLY” on page 347.
- Channel path wait. Whatever pending (PEND) time is not accounted for by the above three measures is due to delay for channel paths. This is the measure of channel delay

that matters—not channel busy. If you think your channels are too busy, track this component of response time for volumes that serve important work to see if it is really a problem.

If you think shared DASD causes the problem, look at the DASD Activity report from the other system, taken at the same time.

High pending times are usually caused by shared DASD contention (extent conflicts, reserves) or high channel path utilization. In the ESS, Multiple Allegiance reduces pending times. If problems with pending times exist you have the following options:

- Change the mix of data on the volume to reduce contention. If you can identify one data set that is contributing the most to the problem, this data set is then eligible to be moved to another volume.
- Check channel utilization. Changes have been made to CCWs to reduce channel overheads for the ESS. This will tend to lower channel utilization and increase throughput.

DISC time

If the major cause of delay is the disconnect time then you will need to do some further research to find the cause. The most probable cause of high disconnect time is having to wait while data is being staged from ESS disk array into cache. You need to check for the following conditions:

- High DISK to CACHE transfer rate. Check the ESS Expert Disk to Cache Transport report and the RMF Cache report. Use the drill-down functions of the ESS Expert to identify the logical volumes that are experiencing response time problems.
- High disk utilization. Check the ESS Expert Disk Utilization report. If the problem is limited to one disk array, the best solution is to move data to balance workload across the subsystem. The other option is to move data to another subsystem, or to change the way that the application uses this data.
- Low CACHE hit ratio. The ESS Expert Cache report, RMF Cache reports, and LISTDATA output can be used to confirm this. If you are suffering from poor cache hit ratios there is little that you can do. You should look at the ESS as a whole, using ESS Expert, and check that activity is balanced across both clusters of the ESS.
- NVS full condition. If the NVS is overcommitted you will see an increase in the values reported in the DFW BYPASS field of the RMF Cache Activity report. If this is a persistent performance indicator you may want to spread activity across more ESS subsystems.

CONN time

Connect time is time as measured by the channel subsystem during which the device is actually connected to the CPU through the path (channel, control unit, DASD) and transferring data. This time is considered good in most cases, because it is transferring data.

If you see devices with poor performance and high connect times then the cause is probably application related. High connect times are associated with large data transfers, either due to the use of large block sizes or to activities like DB2 pre-fetch that schedules large I/O transfers. If the application is not reporting poor response and other users of the volume are not impacted, then no further work is required. High connect time is an indication of large amounts of data being transferred. An application, such as DB2, transfers large I/Os for maximum efficiency.

DB DLY

The Device Busy Delay field is part of the Direct Access Device Activity report. Currently, RMF adds the control unit queuing delay for an I/O into this field. Control unit queuing is

calculated by the control unit and is passed back to the host through a channel frame. Control unit queue occurs when there are extent conflicts for two or more I/Os that would ordinarily operate concurrently. These I/Os may be from the same system and arrive at the controller through different paths of the same PAV device or they can come from different z/OS systems. True device busy delay will only occur in PAV active with Device Reserve. RMF provides a measure of reserve activity. Without reserves, the device busy can be attributed to control unit queuing.

You have to find the bottleneck if the major portion is PEND. PEND time accounts for ‘Wait for channel’, ‘Wait for director (director port delay time)’, ‘Wait for control unit (control unit delay time)’, and ‘Wait for device (device busy time and control unit queuing time)’. But normally, the major reasons of bad PEND time are ‘Wait for channel’ and ‘Wait for device’. The latter is caused by shared DASD contention frequently. Multiple Allegiance should reduce PEND time where the cause is device busy; however, if PEND time is still being reported then you should examine the mix of data sets on the volume and determine which data set(s) is causing the problem. The solution to the problem may be to move or isolate some data sets, possibly on custom volumes. You should also check to ensure that the problem is not due to channel use.

9.11.5 Analyze cache performance

The RMF Cache Subsystem Activity report provides useful information for analyzing the reason of high DISC time. Figure 9-21 shows a sample report.

```
                                       C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                              PAGE    1
            OS/390                   SYSTEM ID FCO1             DATE 02/18/2003            INTERVAL 14.59.526
            REL. 02.10.00            RPT VERSION 02.10.00       TIME 00.30.00

SUBSYSTEM  2105-01     CU-ID  B410     SSID 0404     CDATE  02/18/2003     CTIME 00.30.01      CINT  14.59
TYPE-MODEL 2105-F20

------------------------------------------------------------------------------------------------------------------------------------
                                                       CACHE SUBSYSTEM STATUS
------------------------------------------------------------------------------------------------------------------------------------

SUBSYSTEM STORAGE           NON-VOLATILE STORAGE         STATUS
CONFIGURED      8192.0M     CONFIGURED       192.0M      CACHING                    - ACTIVE
AVAILABLE       7333.2M     PINNED              0.0      NON-VOLATILE STORAGE       - ACTIVE
PINNED             0.0                                   CACHE FAST WRITE           - ACTIVE
OFFLINE            0.0                                   IML DEVICE AVAILABLE       - YES

------------------------------------------------------------------------------------------------------------------------------------
                                                      CACHE SUBSYSTEM OVERVIEW
------------------------------------------------------------------------------------------------------------------------------------
                                                                                                  (2)
TOTAL I/O    162755    CACHE I/O    162755    CACHE OFFLINE        0
TOTAL H/R     1.000    CACHE H/R     1.000

CACHE I/O     -------------READ I/O REQUESTS-------------    -------------------WRITE I/O REQUESTS----------------------          %
REQUESTS         COUNT      RATE       HITS      RATE     H/R      COUNT      RATE      FAST      RATE      HITS      RATE     H/R      READ
NORMAL          162705     181.0     162692     181.0   1.000         18       0.0        18       0.0        18       0.0   1.000     100.0
SEQUENTIAL          32       0.0         25       0.0   0.781          0       0.0         0       0.0         0       0.0     N/A     100.0
CFW DATA             0       0.0          0       0.0     N/A          0       0.0         0       0.0         0       0.0     N/A       N/A
TOTAL           162737     181.0     162717     181.0   1.000         18       0.0        18       0.0        18       0.0   1.000     100.0

-----------------------CACHE MISSES-----------------------        ------------MISC------------         ------NON-CACHE I/O-----
REQUESTS        READ     RATE    WRITE      RATE   TRACKS     RATE                       COUNT    RATE                  COUNT    RATE
                                                                   DFW BYPASS              0     0.0       ICL              0     0.0
NORMAL            13      0.0        0       0.0       21      0.0  CFW BYPASS              0     0.0       BYPASS           0     0.0
SEQUENTIAL         7      0.0        0       0.0      175      0.2  DFW INHIBIT             0     0.0       TOTAL            0     0.0
CFW DATA           0      0.0        0       0.0                    ASYNC (TRKS)           75     0.1
TOTAL             20     RATE      0.0

---CKD STATISTICS---      ---RECORD CACHING---                  (1)                          (3)
WRITE              24     READ MISSES          0
WRITE HITS         24     WRITE PROM          17
```

Figure 9-21 RMF Cache Subsystem Activity report

The report in Figure 9-21 shows statistics on a subsystem basis. There is also a report on a logical volume basis. You have to use the daily subsystem report as an indication of cache utilization, and use the device report for in-depth problem determination.

You can expect the following reasons for high DISC times:

- High DISK<>CACHE transfer rate

- High DISK utilization
- Low CACHE hit ratio
- NVS full condition (DFW retry)

High disconnect time may be caused by a high disk-to-cache transfer rate. This may be indicated by a high staging/destaging rate (number 1 in Figure 9-21 on page 348). If disconnect time is being caused by high disk utilization, you can see this in the ESS Expert Disk Utilization report, as RMF does not report physical disk utilization (RMF RAID Rank report does provides I/O rate and MB/s for ranks). High disconnect time may also be a result of a low cache hit ratio (number 2 in Figure 9-21 on page 348, or in the ESS Expert Cache report). If the cause of a high disconnect time is a NVS full condition, this is indicated by high DFW BYPASS counts (number 3 in Figure 9-21 on page 348).

Basically, you have to do ‘data isolation’ to solve high DISC time. For example, if high DFW retry counts cause performance degradation, the scope of the degradation will be a cluster. You can only move the data to solve it.

9.11.6 Analyze channel path activity

Using the ESS and its sophisticated new functions (PAV and Multiple Allegiance) allows for increased data transfer rates compared with older DASD. Therefore, channel utilization may become a bottleneck. If you recognize high PEND time, you also have to check out your channel path activity in your RMF report. You should configure at least 16 ESCON or four FICON channel paths per z/OS image to the ESS. We also recommend that you keep the ESCON channel utilization lower than 60 percent. 60 percent is at the end of normal operating ranges for DASD subsystems. Performance degradation would be noticeable during host adapter failure unless you keep the channel utilization lower than 60 percent. For FICON, the recommendation is 50 percent.

9.11.7 FICON RMF information

In this section we analyze the I/O response time components when the attachment is done using FICON, and how RMF presents the respective information in its reports.

Channel utilization

The primary RMF report of interest for FICON is the Channel Path Activity report (see Example 9-4).

Example 9-4 RMF Channel Path Activity report

```
.                                        C H A N N E L   P A T H   A C T I V I T Y
                                                                                                                           PAGE    1
           OS/390                   SYSTEM ID FC01           DATE 02/18/2003           INTERVAL 14.59.526
           REL. 02.10.00            RPT VERSION 02.10.00     TIME 00.30.00             CYCLE 1.000 SECONDS

  IODF = 76   CR-DATE: 01/14/2003   CR-TIME: 12.19.11   ACT: POR          MODE: LPAR       CPMF: EXTENDED MODE
-------------------------------------------------------------------------------------------------------------------------------------
                                                   DETAILS FOR ALL CHANNELS
-------------------------------------------------------------------------------------------------------------------------------------
 CHANNEL PATH    UTILIZATION(%)      READ(MB/SEC) WRITE(MB/SEC)      CHANNEL PATH    UTILIZATION(%)      READ(MB/SEC) WRITE(MB/SEC)
 ID TYPE SHR    PART  TOTAL    BUS    PART  TOTAL   PART  TOTAL      ID TYPE SHR    PART  TOTAL    BUS    PART  TOTAL   PART  TOTAL

 06 OSD   Y     1.17   3.89  14.97    0.00   0.00   0.00   0.00      18 CNC_P Y     0.00   0.16
 07 OSD   Y     0.50   3.88  14.97    0.00   0.00   0.00   0.00      19 CNC_P Y     0.00   0.00
 0D CNC_P Y     0.00   0.16                                          1A CNC_P Y     0.00   0.50
 0E CNC_P Y     0.00   0.15                                          1B CNC_P     OFFLINE
 0F CNC_P Y     0.00   0.43                                          1C CNC_P Y     0.00   0.88
 10 CNC_P Y     0.00   0.00                                          1D CNC_P Y     0.00   0.00
 12 CTC_P Y     0.34   0.34                                          20 FC    Y     4.64  17.51  25.63    1.70  14.49   0.21   0.40
 17 CNC_P Y     0.00   0.15                                          22 FC    Y     4.64  17.52  25.63    1.70  14.50   0.22   0.39

 24 FC    Y     4.63  17.53  25.65    1.70  14.51   0.22   0.40      32 CNC_P     OFFLINE
 26 FC    Y     3.29   6.86  13.81    0.00   0.00   2.48   5.16      38 CNC_P     OFFLINE
```

```
28 FC    Y     0.00  26.11  38.59    0.00   0.00   0.00  29.23      39 CNC_P     OFFLINE
2D OSD       OFFLINE                                                 3A CNC_P     OFFLINE
2E OSD       OFFLINE                                                 ED '24'      OFFLINE
2F OSD       OFFLINE
```

FICON channels can be identified in the TYPE column for their type beginning with FC:

- Type FC indicates a native FICON channel.
- Type FC_S indicates a switched native FICON channel.
- Type FCV indicates a FICON bridge channel that connects to an ESCON control unit via a bridge card in a 9032 Model 5 ESCON director.

For a given FICON channel there are three entries under UTILIZATION (%):

- PART denotes the FICON processor utilization due to this logical partition (this field will be blank in BASIC mode or if EMIF is not installed).
- TOTAL denotes the FICON processor utilization for the sum of all the LPARs.
- BUS denotes the FICON internal bus utilization for the sum of all the LPARs.

The FICON processor is busy for channel program processing, which includes the processing of each individual channel command word (CCW) in the channel program, and some setup activity at the beginning of the channel program and clean-up at the end.

The FICON bus is busy for the actual transfer of command and data frames from the FICON channel to the adapter card, which is connected via the FICON link to the director or control unit. The FICON bus is also busy when the FICON processor is polling for work to do. This is why one can see anywhere from 5 to 12 percent FICON bus utilization on the RMF Channel Activity report during time intervals when there are no I/Os active on that channel. There is no accurate way of determining actual FICON processor utilization, that is, there is no way of knowing when the FICON processor is busy vs. when it is idle. RMF *estimates* FICON processor utilization by comparing the actual number of command and data sequences processed with a maximum number determined by benchmarks.

The actual FC channel processor and bus utilizations as reported by RMF will vary by workload. In general, small data transfer sizes will drive FC processor utilization higher than bus utilizations, while large data transfer sizes will drive the FC bus utilization higher than the processor utilization.

Figure 9-22 on page 351 shows the processor and bus utilizations that are reported on the RMF Channel Activity report for 4 KB read hit I/O operations on a zSeries900 (1 Gbit FICON adapters used). At the knee of the response time curve, for this very simple 4 KB read hit benchmark program, FICON processor utilizations are about 80 percent and FICON bus utilizations are about 40 percent.

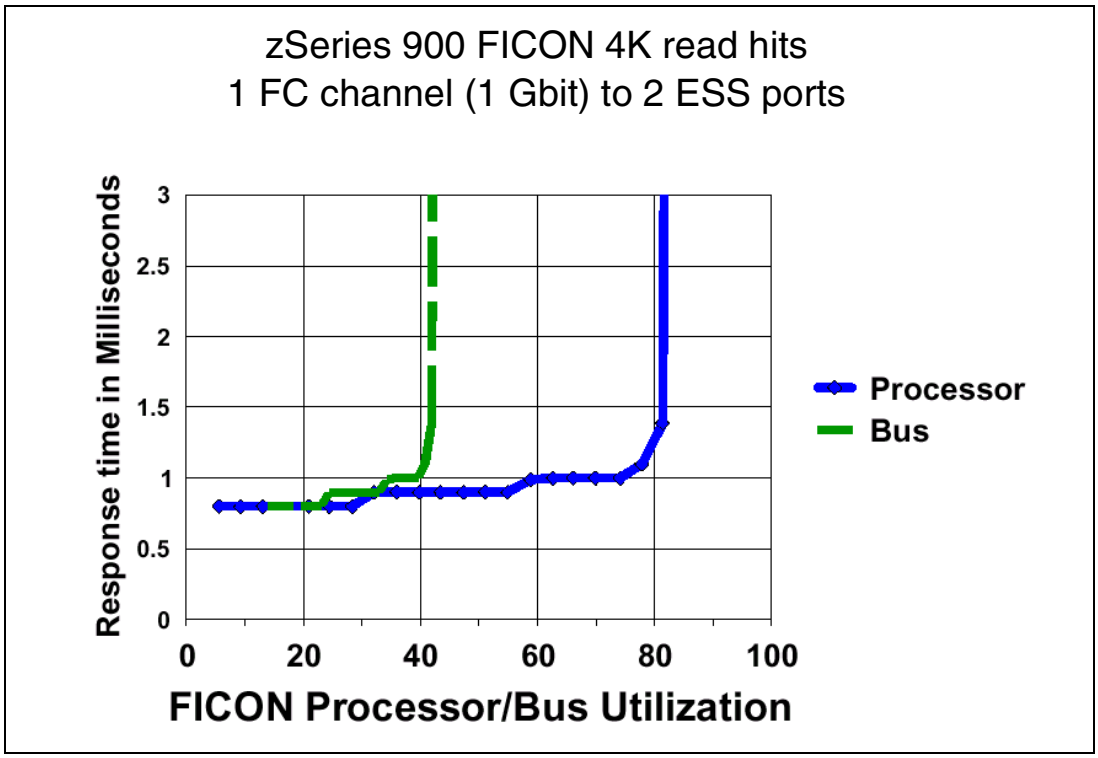


Figure 9-22 FICON processor/bus utilization - 4 K read hits

Figure 9-23 on page 352 shows the processor and bus utilizations that are reported on the RMF Channel Activity report for 27 K or half-track read hit I/O operations on the zSeries 900 (1 Gbit FICON adapters used). In this example the knee of the response time curve occurs at 60 percent FICON bus utilization and 40 percent FICON processor utilization.

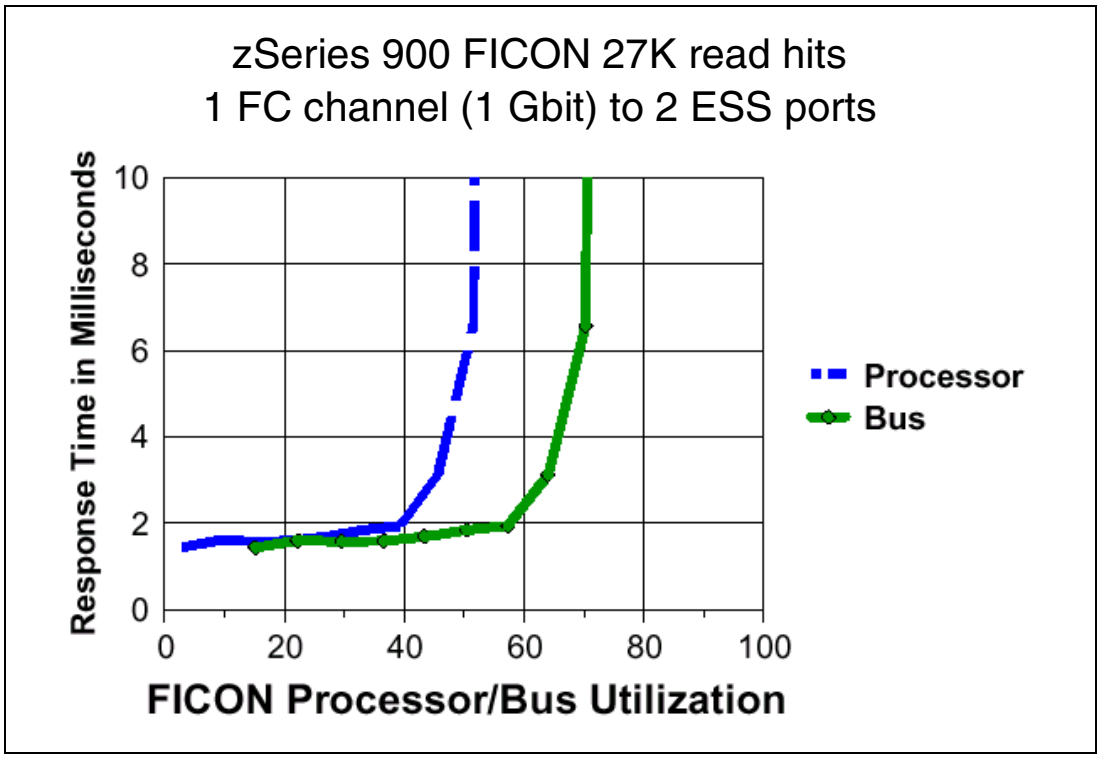


Figure 9-23 FICON 27 K read hit utilization curves

In general, to achieve good response times for the production workloads, the maximum of both the FICON processor and FICON bus utilization should be kept less than 50 percent.

Channel throughput

FICON channels also provide bandwidth information (MB/sec) not available for ESCON channels. This is provided separately for reads and writes since the Fibre Channel link is full duplex, at both the logical partition level (PART) and the entire system level (TOTAL).

Figure 9-24 on page 353 shows the MB/sec that can be achieved with 4 K, 27 K, and 6x27 K I/O operations using 1 Gbit FICON adapters. 2 Gbit FICON adapters provide roughly double the bandwidth.

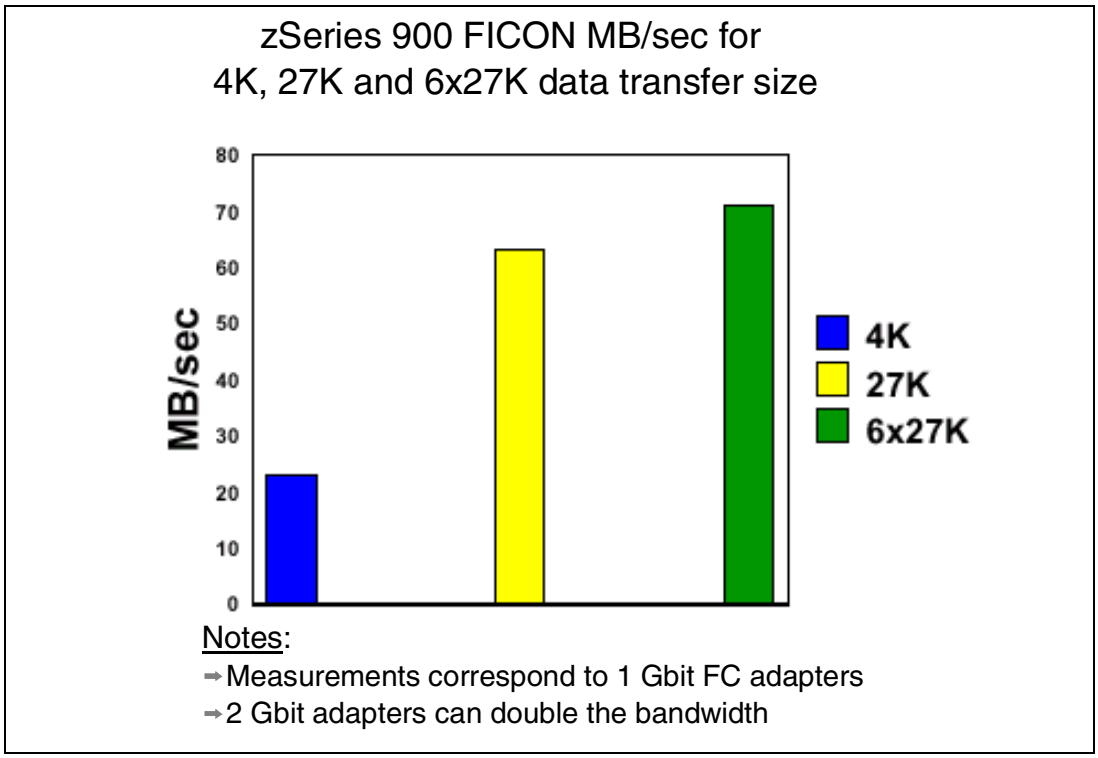


Figure 9-24 FICON data transfer rates

As you can see, with data transfer sizes of only 4 K bytes, the maximum attainable bandwidth is just over 20 MB/sec, while data transfer sizes of 27 K or 6x27 K bytes are capable of 60 to 70 MB/sec.

For FICON channels it is possible to estimate the average number of bytes transferred per SSCH by dividing the MB/sec of a FICON channel from the Channel Path Activity report (Example 9-4 on page 349) by the total SSCH/sec processed by a FICON channel from the I/O Queuing Activity report (Example 9-5). The total SSCH/sec processed by a FICON channel can be determined by adding up all of the 'CHPID taken' fields on the I/O Queuing Activity report for each LCU that a single FICON channel is connected to.

Example 9-5 RMF I/O Queuing Activity report

```
.                                        I/O   Q U E U I N G   A C T I V I T Y
                                                                                                                PAGE    1
          OS/390                   SYSTEM ID FC01            DATE 02/18/2003           INTERVAL 14.59.526
          REL. 02.10.00            RPT VERSION 02.10.00      TIME 00.30.00             CYCLE 1.000 SECONDS

TOTAL SAMPLES =   900     IOP   ACTIVITY RATE  AVG Q LNGTH     IODF = 76   CR-DATE: 01/14/2003   CR-TIME: 12.19.11   ACT: POR

                           00     1563.846        0.00
                           01      834.665        0.00
                           02      553.693        0.00
                                                                                    DELAY    % ALL
LCU    CONTROL UNITS     DCM GROUP   CHAN     CHPID    % DP   % CU  CONTENTION      Q     CH PATH
                       MIN MAX DEF  PATHS     TAKEN    BUSY   BUSY     RATE       LNGTH    BUSY

0038   B000                          20     17.774           0.00
                                     22     17.861           0.00
                                     24     17.917           0.00
                                      *     53.552    0.00   0.00      0.000      0.00    53.69

0039   B100                          20     23.999           0.00
                                     22     23.998           0.00
                                     24     24.029           0.00
                                      *     72.027    0.00   0.00      0.000      0.00    53.69

003C   B400                          20     85.462           0.00
```

```
                          22     85.465          0.00
                          24     85.451          0.00
                           *     256.38    0.00  0.00      0.000     0.00    53.69

003F  B700                20     62.204          0.00
                          22     62.476          0.00
                          24     62.234          0.00
                           *     186.91    0.00  0.00      0.000     0.00    53.69
```

If the average data transfer sizes of your channel programs are greater than 27 K bytes, then you should compare the FICON bus utilizations and the MB/sec fields on your RMF Channel Activity Report to graphs in Figure 9-23 on page 352 and Figure 9-24 on page 353 to assess the maximum capability of FICON channels for your workload.

It is important to note that the maximum FICON bus utilization achieved in these sample measurements we are presenting is about 80 percent, and not 100 percent as one might expect. This is due to the fact that the bus utilization represents the number of cycles that the bus was busy compared to the theoretical maximum possible bus busy cycles based on the cycle time of the bus. In the process of transferring command and data frames across the bus, there are gaps that occur between transfers depending on the arrival pattern and the mix of different channel programs running at any one point in time, so it is therefore not possible to achieve 100 percent FICON bus utilization with normal channel program processing.

Response time

For a native FICON, it will also be of interest to consult the RMF Direct Access Device Activity report. Here one can examine the AVG RESP TIME column and various response time components for activity to the LCUs attached to the FICON channels. In particular, AVG CONN TIME for large block size transfers should be significantly less for native FICON channels than for the same transfer size on ESCON or FICON Bridge channels due to the 100 MB/sec (200 MB/sec for 2 Gbit FICON) link transfer speeds of native FICON.

With native FICON, you should continue to analyze your I/O activity by looking at the DASD activity reports, just as you did with ESCON channels. If response time is a problem, then the *response time* components need to be looked at. If *disconnect time* is a problem, then an increase in CU cache size might help. If IOSQ time is a problem, then ESS with Parallel Access Volumes might help. If *pending* or *connect times* are too high, then one can look at the FICON processor and bus utilizations. If either one of these utilizations is above 50 percent then overuse of the FICON channel could be contributing to additional pending and connect time delays.

If, on the other hand, pending and connect times are high and FICON channel utilizations are less than 50 percent, then overuse of a FICON director port or control unit port could be contributing factors. If FICON channels from multiple CECs are connected to the same director destination port, then one must add up the activity from all the CECs to determine the total destination port activity. This total activity level should be less than the knee of the curve points depicted in the measurement results illustrated in Figure 9-22 on page 351 and Figure 9-23 on page 352.

Connect time

One of the basic differences between FICON and ESCON channel performance is the *connect time* component of response time. Since an ESCON channel is only capable of executing one I/O at a time, the amount of time that it takes to execute the protocol plus data transfer components of connect time is relatively constant from one I/O operation to the next with the same exact channel program.

With FICON, however, connect time can vary from one execution of a channel program to another. This is a side effect of the multiplexing capability of FICON. Since both the channel

and the control unit can be concurrently executing multiple I/O operations, the individual data transfer frames of one I/O operation might get queued up behind the data transfer frames of another I/O operation. So the connect time of an I/O with FICON is dependent upon the number of I/Os that are concurrently active on the same FICON channel, link, and control unit connection. Multiplexing also means that the start and end of the connect time for one FICON I/O operation can overlap the start and end of the connect time for several other FICON I/O operations.

With ESCON, the additional queueing delays caused by having multiple I/Os concurrently active appear in the *pending* or *disconnect time* components of response time. If the same workload with the same activity rate and the same level of I/O concurrency is run on FICON channels instead of ESCON channels, then one could see the *pending* and *disconnect* time components of response time decrease and the *connect* time component increase for small data transfer sizes. For large data transfers, the improved connect time due to the high link transfer speed will most likely offset any increased time due to multiplexing queueing delays.

Figure 9-25 illustrates the improvement in connect time for 1 Gbit FICON compared to ESCON for large data transfers when I/Os were active to 512 different device volumes across four FICON channels or 128 device volumes per FICON during a one-minute RMF interval.

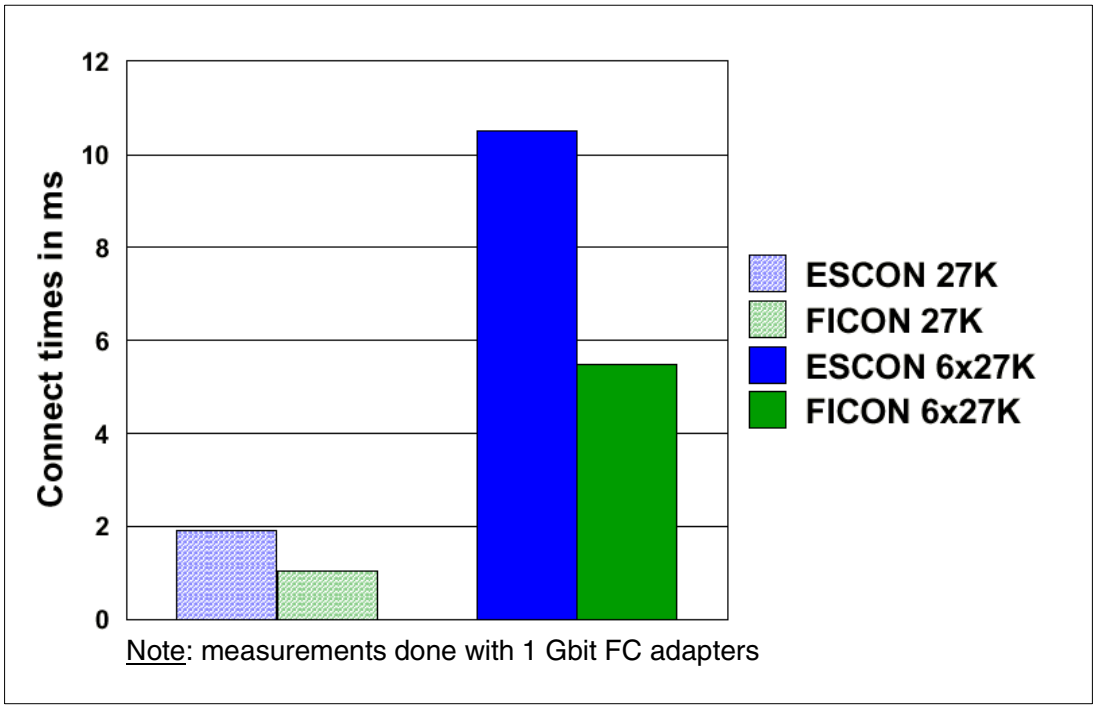


Figure 9-25 FICON vs. ESCON connect times

A few other fields on the RMF I/O Queueing Activity report will be different for FICON vs. ESCON. For example, the %DP BUSY and %CU BUSY should be 0 for FICON due to the elimination of destination port busy and control unit busies with FICON, and the %ALL CH PATH BUSY field is meaningless for FICON.

Disconnect time

FICON channel utilizations are not impacted by cache miss activity. With ESCON, when a cache miss occurs, the ESS must disconnect from the channel, service the cache miss, and then attempt to reconnect to the channel again. The additional overhead involved in disconnecting and reconnecting reduces the maximum throughput capabilities of an ESCON

channel. One must also be careful not to drive ESCON channel utilizations too high since the probability of a successful re-connection decreases as ESCON channel utilizations increase.

With FICON channels, disconnect and reconnect protocols are eliminated. When the frames of data for one cache miss are ready to be transferred to the channel, they take their position on the queue along with any other frames that may be ready to transfer from any of the other I/O operations that are concurrently active.

When the ESS services a FICON cache miss, it simply keeps track of the amount of time it takes to perform this operation for RMF reporting purposes only. This time appears as *disconnect* time on the Direct Access Device Activity report even though an actual disconnect operation never occurred (that is, the FICON channel and the ESS never had any formal communications about this event).

In fact, from a FICON channel perspective, a cache miss is treated the same as a cache hit, except that it takes longer for the data transfer to start. The utilization of the FICON channel processor and bus will be the same for the same activity level regardless of whether the control unit cache is achieving a 30 percent, 60 percent, 90 percent, or any other cache hit ratio. For workloads with relatively low control unit cache hit ratios, it is therefore possible to achieve greater than 4 to 1 FICON channel consolidation by connecting a single FICON channel to many different physical control units.

Summary

In summary, the basic architecture and design differences between FICON and ESCON result in many changes to the performance data that appears on RMF reports and its interpretation. Additional information in the form of FICON processor and bus utilizations and read and write MB/sec is provided to help analyze the multiplexing capability of FICON.

Since ESCON is only capable of executing one I/O operation at a time, RMF reports the time that the entire CHPID path is busy for ESCON channel utilization. With FICON, we must consider the individual components of the total CHPID path such as the FICON channel processor and bus, the fibre link, the switch destination port, and the control unit port adapter microprocessor and bus. The charts and examples that were presented in this section should help you assess the maximum capability that FICON channels can deliver with your particular your workload.

The guides, examples, and recommendations presented in this chapter are generically valid, and as so can be used. For a less generic and more detailed approach that considers your particular environment, the Disk Magic modeling tool is available to IBM personnel and business partners that can help you in the capacity and configuration planning activities.

10

iSeries servers

This chapter gives an overview of the iSeries storage architecture and basic guidelines to help you plan how to configure your ESS for optimal capacity and performance when connected to iSeries servers. The general guidelines presented in this chapter are intended to work with most user situations, but may not account for the peculiarities of all installations. Your IBM representative or business partner can assist when planning a more detailed approach that considers all the particular factors of any user installation.

The information in this chapter can be complemented with additional information that can be found at http://www.ibm.com/support/techdocs (where the search can be refined by asking for iSeries and ESS).

10.1 iSeries servers

When attaching an iSeries to an ESS, the iSeries will be like any other Fixed Block Architecture (FBA) server, whether SCSI or Fibre Channel connected. Basically the ESS presents a set of LUNs to the iSeries, like it does for any other open systems server.

One thing that distinguishes the iSeries is the LUN sizes it will be using. With a SCSI adapter the LUNs will report into the iSeries as the different models of device type 9337. And with a Fibre Channel adapter connection, the LUNs will report into the iSeries as the different models of the 2105 device type. The models will depend on the size of LUNs that have been configured.

The other distinguishing characteristics of the iSeries come on an upper layer on top of the preceding architectural characteristics described so far for the FB servers. This is the Single Level Storage concept that the iSeries uses. This is a powerful characteristic that makes the iSeries a unique server.

10.2 Single level storage

Both the main memory of the iSeries and the storage disks are treated as a very large virtual address space, known as *single level storage*. This is probably the most significant differentiation of the iSeries when compared to other open systems. As far as applications on the iSeries are concerned, there is really no such thing as a disk unit.

Storage management and caching of data into main storage is completely automated based on expert cache algorithms. Storage management automatically spreads the data across the disk arms or disk drives (or across LUNs for ESS disks) and continues to add records to files until specified threshold levels are reached.

Single level storage is efficient. Regardless of how many application programs need to use an object, only one copy of it is required to exist. This makes the entire main storage of an iSeries server a fast cache for disk storage.

10.3 Expert Cache

Expert Cache is a set of algorithms that executes in the main central processing unit (CPU). Expert cache uses designated pools in main storage to cache database operations. The size of the pools is adjusted dynamically or controlled by the system operator as a function of time.

By caching in main storage, the system eliminates access to the storage devices and reduces associated I/O traffic. Expert Cache works by minimizing the effect of synchronous DASD I/O on a job. The best candidates for performance improvement by using expert cache are jobs that are most affected by synchronous DASD I/Os.

Benchmark tests using a simulated commercial workload environment called CPW (or Commercial Processing Workloads) let us conclude that Expert Cache is beneficial at all times for external disk configurations. The read and write cache algorithms of ESS work favorably with algorithms of Expert Cache to improve system performance.

Figure 10-1 on page 359 compares the performance of dedicated ESS DASD when running with OS/400 Expert Cache function, both off (*FIXED) and on (USRDFN). As you can see from the example, Expert Cache works well with the ESS.

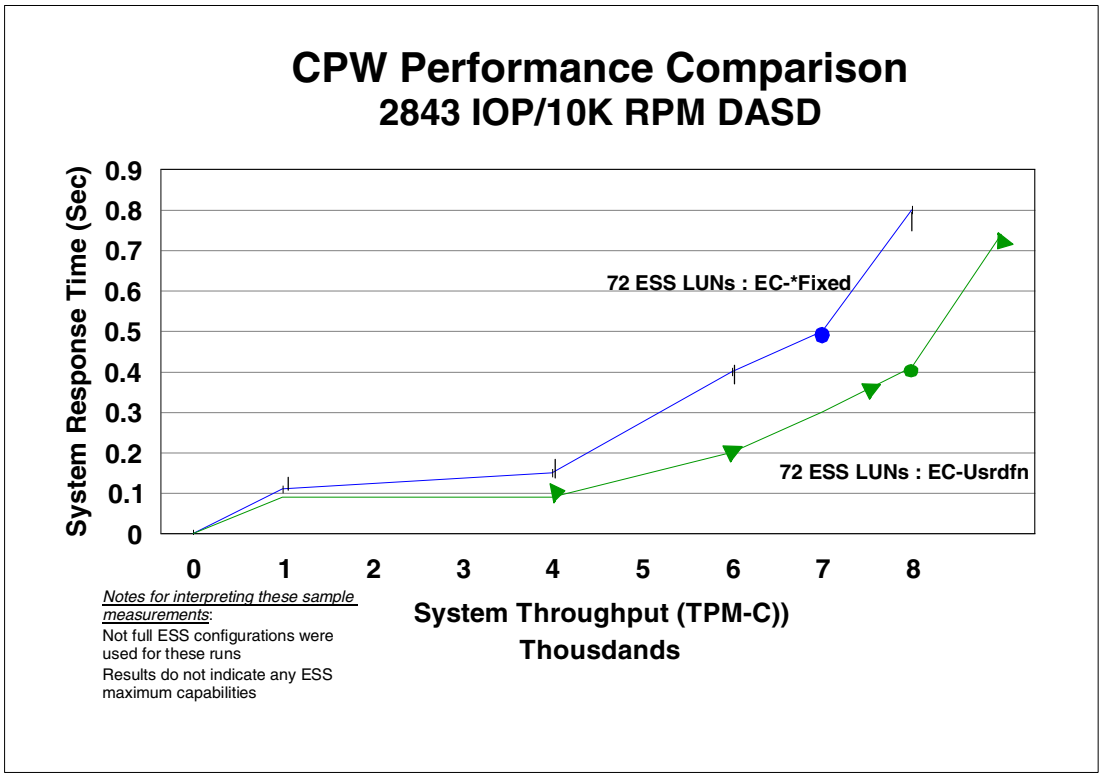


Figure 10-1 ESS and iSeries Expert Cache

10.4 Direct access storage devices

Direct Access Storage Devices (DASD) or disk volumes can either be internal to the iSeries or attached externally. Disk volumes are grouped in auxiliary pools (ASPs). Disk volumes can be protected or unprotected. Protection can be mirroring or RAID-5.

When the iSeries DASDs are external, like when using an ESS, the disk devices are mapped into the Logical Unit Numbers (LUNs) that are carved in the ESS ranks. In the ESS LUNs are striped across a rank, with the ranks being either RAID-5 or RAID-10 protected.

10.5 LUNs allocation and capacity

In the ESS, for FBA-attached servers, disk device allocation is based on the LUNs that are carved into the ESS ranks. The LUNs map to the disk devices that the FBA servers operating systems attach in the ESS. LUNs are striped across the hard disk drives that make up the ESS rank, and the size of the LUN is set when it is defined to the ESS.

The ESS can accommodate all iSeries disks with the exception of the load source unit. Nevertheless, a mirror image of the OS/400 load source unit can be located in ESS for recovery purposes. Load source mirroring requires OS/400 V4R3.

When attaching FBA servers to the ESS, the thing that distinguishes the iSeries is the LUN sizes it will be using: 4.19, 8.59, 17.54, 35.16, 36.00, and 70.56 GB. With a SCSI adapter (#6501) attachment, the LUNs will report into the iSeries as the different models of device type 9337. And with a Fibre Channel disk adapter (#2766) connection, the LUNs will report

into the iSeries as the different models of the 2105 device type. The models will relate to the size of LUNs that have been configured.

Table 10-1 shows the number of iSeries LUNs that can be configured in an ESS rank and the device types that are emulated as seen from the iSeries.

Table 10-1 Configurable LUNs per array

Array size for ISeries volume size	**9337-58C 9337-58A 4.19 GB**	**9337-59C 9337-59A 2105-A01 2105-A81 8.59 GB**	**9337-5AC 9337-5AA 2105-A02 2105-A82 17.54 GB**	**9337-5CC 9337-5CA 2105-A05 2105-A85 35.16 GB**	**9337-5BC 9337-5BA 2105-A03 2105-A83 36.00 GB**	** 2105-A04 2105-A84 70.56 GB**
9.1 GB 6+P+S	12 LUNs + 2.25 GB	6 LUNs + 1.04 GB	2 LUNs + 17.48 GB	1 LUN + 17.41 GB	1 LUN + 16.57 GB	0 LUNs
9.1 GB 7+P	14 LUNs + 2.63 GB	7 LUNs + 1.22 GB	3 LUNs + 8.70 GB	1 LUN + 26.18 GB	1 LUN + 25.34 GB	0 LUNs
18.2 GB 6+P+S	25 LUNs + 0.35 GB	12 LUNs + 2.13 GB	5 LUNs + 17.45 GB	2 LUNs 34.87 GB	2 LUNs + 33.19 GB	1 LUN + 34.63 GB
18.2 GB 7+P	29 LUNs + 1.11 GB	14 LUNs + 2.49 GB	6 LUNs + 17.44 GB	3 LUNs + 17.24 GB	3 LUNs + 14.73 GB	1 LUN + 52.17 GB
36.4 GB 6+P+S	50 LUNs + 0.75 GB	24 LUNs + 4.31 GB	11 LUNs + 17.41 GB	5 LUNs + 34.62 GB	5 LUNs + 30.43 GB	2 LUNs + 69.32 GB
36.4 GB 7+P	58 LUNs + 2.28 GB	28 LUNs + 5.03 GB	13 LUNs + 17.39 GB	6 LUNs + 34.54 GB	6 LUNs + 29.51 GB	3 LUNs + 33.83 GB
72.8 GB 6+P+S	N/A	49 LUNs + 0.03 GB	23 LUNs + 17.23 GB	11 LUNs + 34.08 GB	11 LUNs + 24.86 GB	5 LUNs + 68.08 GB
72.8 GB 7+P	N/A	57 LUNs + 1.48 GB	27 LUNs + 17.23 GB	13 LUNs + 39.91 GB	13 LUNs + 23.02 GB	6 LUNs + 67.67 GB

Note: The residual GBs may be configured as AS/400 volumes with a smaller size. For example, a residual 17.41 GB can be defined as 2 x 8.59 GB LUNs.

From the ESS perspective, the size of a logical device has no effect on its performance. ESS does not serialize I/O operations on the basis of logical devices. However, you must select the size of the LUNs based on operating system considerations as well as storage administrator considerations. iSeries only allows one outstanding I/O operation per disk device at a time. Hence, to ensure that iSeries sustains enough I/O operations, you may find that more and smaller disks devices (LUNs in the ESS) can perform better than fewer larger disks devices.

You should also consider for the iSeries, that each Fibre Channel host bus adapter (HBA) supports 32 LUNs and each SCSI adapter supports 16 LUNs. LUN size must be large enough to ensure that each host bus adapter can allocate the available capacity. For example, you want to have one Fibre Channel HBA addressing 420 GB in the ESS. One alternative is define 22 17.54 GB LUNs and four 8.59 GB LUNs. But, if you define only 8.59 GB LUNs, you can address maximum 32 * 8.59 GB, that is, 274 GB. You need LUNs greater than 8.59 GB in this case to address the 420 GB.

Protected and non-protected LUNs

With its RAID architecture, the ESS emulated 9337 and 2105 volumes are treated as protected logical volumes (9337 models that end in C), which prohibits software mirroring. To solve this, the ESS permits disk units to be defined as non-protected models (9337 models that end in A). Software mirroring is only allowed on non-protected 9337s and 2105s.

From an ESS perspective, all iSeries volumes are defined on RAID ranks and are protected within the ESS. The ESS Specialist Add Volumes panel allows you to define the volume as Unprotected.

10.6 Hard disk drives capacity and speed

The size and speed of the physical disks (HDDs, hard disk drives, or DDMs, disk drive modules) of the ESS disk eight-packs can have significant effect on the performance.

The following general guidelines can help you when deciding the hard disk drives capacity and speed of the disk eight-packs, when the ESS will be connected to iSeries servers:

- Applications with high write activity rates will benefit if more DDMs are used, as compared to applications with the same access density but lower write activity rate.
- 18 GB 15 K rpm DDMs are most likely to be a good choice for applications with access densities greater than 2 IO/sec/GB.
- 18 GB 10 K rpm and 36 GB 15 K rpm DDMs are most likely to be a good option for applications with access densities between 1 and 2 IO/sec/GB.
- 36 GB 10 K rpm and 73 GB 15 K rpm may be used by applications that do not have high performance requirements; or applications that make good use of cache with very high cache hit ratios (greater than 95 percent); or applications with low access rates relative to the amount of storage, less than 1 IO/sec/GB.
- 73 GB 10 K RPM or 145 GB DDMs tend to offer a very attractive choice due to their ability to deliver large capacity in a small footprint. However, this capacity should be balanced with the performance needs.

The previous considerations for hard disk drive capacity and speed are generic and as such can be used. For a more detailed and accurate approach that takes into consideration the particularities of your processing environment you should contact your IBM representative who can assist you with the modelling tools for ESS capacity and configuration planning.

10.7 Host adapters

The iSeries can attach to the ESS using SCSI and Fibre Channel host adapters. The general recommendations for host attachment also apply for the iSeries. The information discussed in the following sections can further be complemented with the recommendations in 5.5, “SCSI” on page 134, and 5.6, “Fibre Channel” on page 136.

10.7.1 SCSI host adapter

The ESS can be attached to the AS/400 or iSeries using the SCSI disk controller feature #6501 and emulating 9337 disks drives. OS/400 V3R1 supports 4.19 and 8.59 GB disks drives, and V4R2 and later support 4.19, 8.59, 17.54, 36, and 35.17 GB disk drives. #6501 is supported only by earlier than V4R5 OS/400 versions.

Up to 16 LUNs are supported per #6501 adapter, 8 LUNs per #6501 port. The maximum number of #6501 adapters supported with an ESS is sixteen, which will use the sixteen ESS SCSI Host Adapters (32 SCSI ports).

The #6501 can be installed in the AS/400 system unit and/or in the I/O expansion tower. The #6501 is not supported in the #5065 Storage/PCI expansion tower or in the 170, 600, and S10 system units.

Note: The #6501 SCSI adapter was withdrawn from marketing July 31st, 2001.

For optimum performance we recommend that you to configure a maximum of two #6501 adapters per AS/400 tower, and no more than three. All towers should have two #6501 adapters before the third is added.

For SCSI host attachments the number of SCSI ports determines the ultimate capability of the ESS to transfer data to the host. Use the following general guidelines to ensure that your plans include an adequate number of ports:

- 80 GB of storage for each attached SCSI Fast-Wide port. The #6501 adapter provides two SCSI Fast-Wide ports.
- If you have collected workload information, estimate a data transport rate of 10–15 MB per seconds for each #6501 SCSI port.

We recommend ordering and configuring the maximum number of SCSI host adapters to achieve maximum bandwidth.

10.7.2 Fibre Channel

With OS/400 V5R1, the iSeries supports external SAN connections via native Fibre Channel adapters. The PCI Disk Fibre Channel Adapter (feature #2766) is used to connect the iSeries to the ESS.

Fibre Channel adapters in iSeries are supported in Models 270 or 8xx. They are not supported on the older PCI-based system 6xx and 7xx models.

The #2766 Fibre Channel adapter is supported in most locations in the iSeries system units and expansion towers. However, this adapter is not supported under the embedded IOP in the system CEC.

The #2766 requires a dedicated IOP in all positions and is capable of driving the IOP at a full rated speed. The iSeries configurator accurately positions and ties this adapter to an IOP.

Information on total storage capacity supported by a particular iSeries model can be found in the publication *IBM @server iSeries and AS/400e™ System builder*, SG24-2155, and it should not be exceeded.

With OS/400 Version 5 Release 1 the #2766 Fibre Channel adapter supports:

- 1 Gbps adapter speed
- Point-to-point and quick loop topologies
- Distance of 500 metres for 50 µ cables and of 175 metres for 62.5 µ cables
- Maximum of two cascaded switches
- Maximum of one target per iSeries Fibre Channel adapter
- Maximum of 32 LUNS per FC adapter

With the introduction of OS/400 Version 5 Release 2, the #2766 Fibre Channel adapter supports:

- 2 Gbps adapter speed
- Full switch in addition to point-to-point and quick loop topologies
- Distance of 300 metres for 50 µ cables and 150 metres for 62.5 µ cables if working at 2 Gbps
- More than 2 cascaded switches

- Multi-target support
- Maximum of 32 ESS LUNs

The 2 Gbps adapter speed is enabled with feature #2766 simply loading V5R2 of OS/400. The adapters are auto-sensing and run either 1 Gbps or 2 Gbps. To gain the full benefit of this change, all elements in the SAN infrastructure need to be enabled for 2 Gbps.

With multi-target support with OS/400 V5R2, many ESSs can be supported from a single Fibre Channel adapter initiator, but the total number of addressable LUNs remains 32. This means, for example, that a single iSeries Fibre Channel disk adapter can have 16 LUNs on each of two ESSs.

The #2766 is capable of running up to 100 MB/s or 200 MB/s, so fewer Fibre Channel host adapters are required on the ESS to achieve the same throughput as multiple SCSI cards. As a rule of thumb, between three to four SCSI #6501 adapter cards can be replaced by a single Fibre Channel adapter card working at 100 MB/s (1 Gbps).

You should not plan the capacity based on the maximum bandwidth that the adapter can deliver. A good approach is to make planning decisions assuming a throughput capability of around 60 MB/sec per Fibre Channel port. You can use this general guideline when you know your application's throughput demand.

Sometimes there is little or no information about throughput demands of the application and you still must estimate the number of ports required. A good guideline for this situation is to estimate one 1 Gbit Fibre Channel port for every 400 GB of data.

10.8 iSeries performance and monitoring tools

This section gives an overview of the tools you can use to monitor the ESS performance when connected to iSeries servers.

10.8.1 IBM TotalStorage Expert

The ESS Expert component of the IBM TotalStorage Expert can be used to obtain performance information of the ESS relative to any one of its attached servers. This tool can run on an AIX or Windows 2000 server that does not need to be dedicated.

The ESS Expert presents information summarized in reports that include:

- Number of I/O requests for the entire storage server in total and separated among various physical disk groupings
- Read and write cache hit ratio
- Cache to/from disk operations (stage/destage)
- Disk read and write response time
- Disk utilization

The information provided by the ESS Expert can be complemented with the information provided by the PM @server iSeries monitoring tool, which we discuss in the following section. The ESS Expert is further described in detail in 4.5, "TotalStorage Expert performance reports and other tools" on page 122.

10.8.2 PM eServe™r iSeries

PM @server Series is an integrated function within OS/400 that automates many of the steps required for the capacity planning and performance analysis functions that are key to face the challenges of an e-business on-demand era.

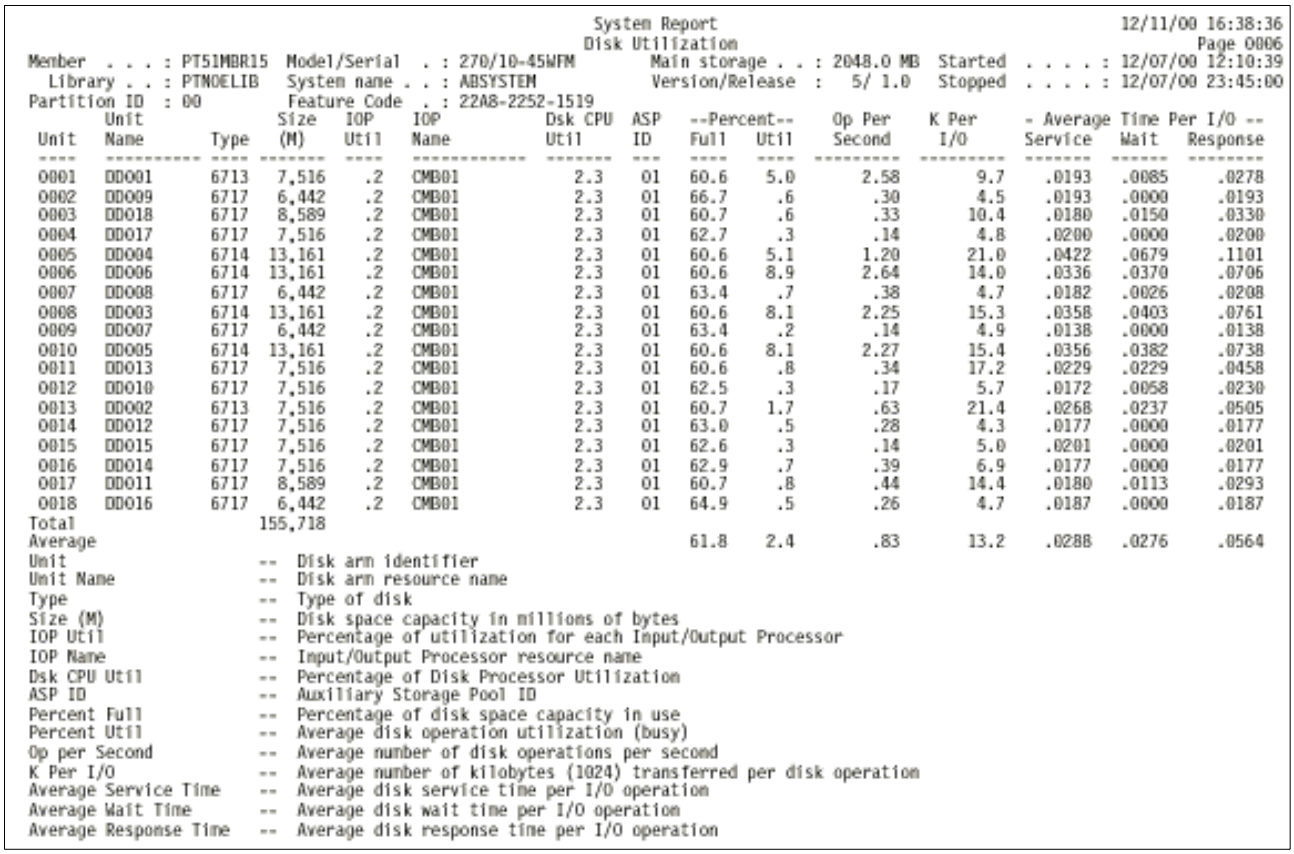
Simply activate PM @server iSeries and the OS/400 Collection Services will automatically collect system utilization information. This information can include CPU utilization, disk capacity and utilization, response time, throughput, and application and user utilization.

PM eServer iSeries Disk utilization reports

Performance reports provided by PM @server iSeries for analyzing disk performance can complement the ESS Expert reports.

The following considerations apply when reviewing PM @server iSeries disk reports:

- Total storage utilization, in terms of capacity, for each ASP is a very important factor to consider in an iSeries environment. Because of this, in some PM @server iSeries reports, disk utilization is not related to workload but to capacity. So you must observe the performance reports, if disk utilization refers to the percentage of time the drive is found busy or if it refers to the percentage of capacity being used.
- In the PM @server iSeries Disk Utilization reports (see Figure 10-2 on page 365) the units are not physical disk drives when they are in the ESS. Units correspond to LUNs carved in the ESS. So for high utilization reported figures, the ESS ranks should be reviewed using the ESS Expert reports.
- The ESS ranks utilization should be reviewed also for the units that show high response times in the Performance Monitor Disk Utilization report.
- In general, complementing the ESS Expert information with the PM @server iSeries reports information is the recommended approach.

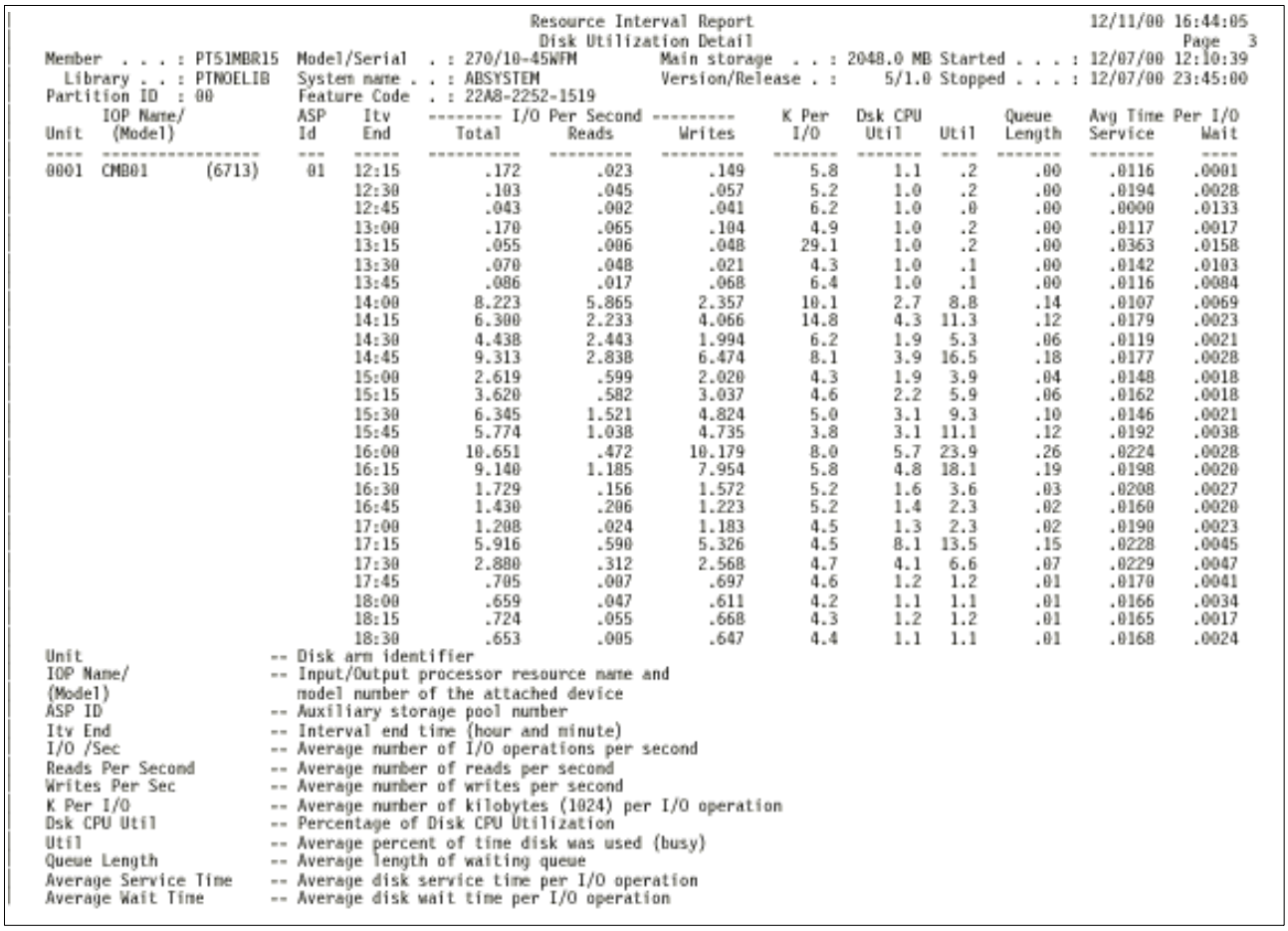


```
                                                System Report                                                   12/11/00 16:38:36
                                                Disk Utilization                                                        Page 0006
Member  . . . : PT51MBR15  Model/Serial  . : 270/10-45WFM       Main storage . . : 2048.0 MB  Started  . . . . : 12/07/00 12:10:39
  Library . . : PTMOELIB   System name . . : ABSYSTEM           Version/Release  :   5/ 1.0   Stopped  . . . . : 12/07/00 23:45:00
Partition ID  : 00         Feature Code  . : 22A8-2252-1519
        Unit              Size   IOP    IOP            Dsk CPU  ASP   --Percent--    Op Per    K Per     - Average Time Per I/O --
 Unit   Name       Type    (M)   Util   Name           Util     ID    Full   Util    Second     I/O      Service   Wait   Response
 ----   ---------- ---- -------  ----   ------------  -------  ---   ----   ----  ---------  ---------  -------  ------  --------
 0001   DD001      6713   7,516    .2   CMB01            2.3    01    60.6    5.0       2.58        9.7    .0193   .0085     .0278
 0002   DD009      6717   6,442    .2   CMB01            2.3    01    66.7     .6        .30        4.5    .0193   .0000     .0193
 0003   DD018      6717   8,589    .2   CMB01            2.3    01    60.7     .6        .33       10.4    .0180   .0150     .0330
 0004   DD017      6717   7,516    .2   CMB01            2.3    01    62.7     .3        .14        4.8    .0200   .0000     .0200
 0005   DD004      6714  13,161    .2   CMB01            2.3    01    60.6    5.1       1.20       21.0    .0422   .0679     .1101
 0006   DD006      6714  13,161    .2   CMB01            2.3    01    60.6    8.9       2.64       14.0    .0336   .0370     .0706
 0007   DD008      6717   6,442    .2   CMB01            2.3    01    63.4     .7        .38        4.7    .0182   .0026     .0208
 0008   DD003      6714  13,161    .2   CMB01            2.3    01    60.6    8.1       2.25       15.3    .0358   .0403     .0761
 0009   DD007      6717   6,442    .2   CMB01            2.3    01    63.4     .2        .14        4.9    .0138   .0000     .0138
 0010   DD005      6714  13,161    .2   CMB01            2.3    01    60.6    8.1       2.27       15.4    .0356   .0382     .0738
 0011   DD013      6717   7,516    .2   CMB01            2.3    01    60.6     .8        .34       17.2    .0229   .0229     .0458
 0012   DD010      6717   7,516    .2   CMB01            2.3    01    62.5     .3        .17        5.7    .0172   .0058     .0230
 0013   DD002      6713   7,516    .2   CMB01            2.3    01    60.7    1.7        .63       21.4    .0268   .0237     .0505
 0014   DD012      6717   7,516    .2   CMB01            2.3    01    63.0     .5        .28        4.3    .0177   .0000     .0177
 0015   DD015      6717   7,516    .2   CMB01            2.3    01    62.6     .3        .14        5.0    .0201   .0000     .0201
 0016   DD014      6717   7,516    .2   CMB01            2.3    01    62.9     .7        .39        6.9    .0177   .0000     .0177
 0017   DD011      6717   8,589    .2   CMB01            2.3    01    60.7     .8        .44       14.4    .0180   .0113     .0293
 0018   DD016      6717   6,442    .2   CMB01            2.3    01    64.9     .5        .26        4.7    .0187   .0000     .0187
Total                  155,718
Average                                                            61.8    2.4        .83       13.2    .0288   .0276     .0564
Unit                --  Disk arm identifier
Unit Name           --  Disk arm resource name
Type                --  Type of disk
Size (M)            --  Disk space capacity in millions of bytes
IOP Util            --  Percentage of utilization for each Input/Output Processor
IOP Name            --  Input/Output Processor resource name
Dsk CPU Util        --  Percentage of Disk Processor Utilization
ASP ID              --  Auxiliary Storage Pool ID
Percent Full        --  Percentage of disk space capacity in use
Percent Util        --  Average disk operation utilization (busy)
Op per Second       --  Average number of disk operations per second
K Per I/O           --  Average number of kilobytes (1024) transferred per disk operation
Average Service Time    --  Average disk service time per I/O operation
Average Wait Time       --  Average disk wait time per I/O operation
Average Response Time   --  Average disk response time per I/O operation
```

Figure 10-2 System Report - Disk Utilization

10.8.3 PM eServer iSeries interval reports

PM @server iSeries also presents interval reports. These interval reports provide information of the system operation during the data collection period. Disk Utilization Detail, illustrated in Figure 10-3 on page 366, is very useful for sizing your ESS. You must consolidate the information for the peak 15 minutes intervals where throughput and response time are relevant for your applications and get the peak I/Os per second, KB per I/O, and also read-to-write ratio. With this information, your IBM representative will be able to help you estimate more accurately the ESS configuration that will adequately support your iSeries workload demands.



```
                                        Resource Interval Report                                    12/11/00 16:44:05
                                         Disk Utilization Detail                                          Page   3
Member  . . . : PT51MBR15  Model/Serial  . : 270/10-45WFM        Main storage  . . : 2048.0 MB Started . . . : 12/07/00 12:10:39
  Library . . : PTNOELIB   System name . . : ABSYSTEM           Version/Release . :     5/1.0 Stopped . . . : 12/07/00 23:45:00
Partition ID  : 00         Feature Code  . : 22A8-2252-1519
      IOP Name/            ASP    Itv    -------- I/O Per Second ---------     K Per   Dsk CPU          Queue    Avg Time Per I/O
Unit   (Model)             Id     End      Total      Reads      Writes          I/O     Util    Util   Length   Service     Wait
----  ----------------     ---    -----    ---------  ---------  ---------     -------  -------  ----  -------  -------     ----
0001  CMB01      (6713)     01    12:15        .172       .023       .149         5.8      1.1    .2      .00     .0116    .0001
                           01    12:30        .103       .045       .057         5.2      1.0    .2      .00     .0194    .0028
                           01    12:45        .043       .002       .041         6.2      1.0    .0      .00     .0000    .0133
                           01    13:00        .170       .065       .104         4.9      1.0    .2      .00     .0117    .0017
                           01    13:15        .055       .006       .048        29.1      1.0    .2      .00     .0363    .0158
                           01    13:30        .070       .048       .021         4.3      1.0    .1      .00     .0142    .0103
                           01    13:45        .086       .017       .068         6.4      1.0    .1      .00     .0116    .0084
                           01    14:00       8.223      5.865      2.357        10.1      2.7   8.8      .14     .0107    .0069
                           01    14:15       6.300      2.233      4.066        14.8      4.3  11.3      .12     .0179    .0023
                           01    14:30       4.438      2.443      1.994         6.2      1.9   5.3      .06     .0119    .0021
                           01    14:45       9.313      2.838      6.474         8.1      3.9  16.5      .18     .0177    .0028
                           01    15:00       2.619       .599      2.020         4.3      1.9   3.9      .04     .0148    .0018
                           01    15:15       3.620       .582      3.037         4.6      2.2   5.9      .06     .0162    .0018
                           01    15:30       6.345      1.521      4.824         5.0      3.1   9.3      .10     .0146    .0021
                           01    15:45       5.774      1.038      4.735         3.8      3.1  11.1      .12     .0192    .0038
                           01    16:00      10.651       .472     10.179         8.0      5.7  23.9      .26     .0224    .0028
                           01    16:15       9.140      1.185      7.954         5.8      4.8  18.1      .19     .0198    .0020
                           01    16:30       1.729       .156      1.572         5.2      1.6   3.6      .03     .0208    .0027
                           01    16:45       1.430       .206      1.223         5.2      1.4   2.3      .02     .0160    .0020
                           01    17:00       1.208       .024      1.183         4.5      1.3   2.3      .02     .0190    .0023
                           01    17:15       5.916       .590      5.326         4.5      8.1  13.5      .15     .0228    .0045
                           01    17:30       2.880       .312      2.568         4.7      4.1   6.6      .07     .0229    .0047
                           01    17:45        .705       .007       .697         4.6      1.2   1.2      .01     .0170    .0041
                           01    18:00        .659       .047       .611         4.2      1.1   1.1      .01     .0166    .0034
                           01    18:15        .724       .055       .668         4.3      1.2   1.2      .01     .0165    .0017
                           01    18:30        .653       .005       .647         4.4      1.1   1.1      .01     .0168    .0024
Unit                    -- Disk arm identifier
IOP Name/               -- Input/Output processor resource name and
(Model)                    model number of the attached device
ASP ID                  -- Auxiliary storage pool number
Itv End                 -- Interval end time (hour and minute)
I/O /Sec                -- Average number of I/O operations per second
Reads Per Second        -- Average number of reads per second
Writes Per Sec          -- Average number of writes per second
K Per I/O               -- Average number of kilobytes (1024) per I/O operation
Dsk CPU Util            -- Percentage of Disk CPU Utilization
Util                    -- Average percent of time disk was used (busy)
Queue Length            -- Average length of waiting queue
Average Service Time    -- Average disk service time per I/O operation
Average Wait Time       -- Average disk wait time per I/O operation
```

Figure 10-3 Interval Report - Disk Utilization detail

11

Understanding your workload

This chapter typifies and discusses the different kinds of workloads that applications can generate. This characterization can be useful for understanding performance documents, as well as to categorize the different workloads in your installation.

11.1 General workload types

Following are the workload type definitions as used in several of the IBM performance documents.

11.1.1 Standard workload

This workload is characterized by random access of 4 KB records, with a mix of 70 percent reads and 30 percent writes. This is also characterized by moderate read hit ratios in the disk subsystem cache (approximately 50 percent). This workload might be representative of a variety of online applications (for example, SAP R/3 applications, many database applications, and file servers).

11.1.2 Read intensive cache unfriendly workload

This workload is characterized by very random 4 KB reads. The accesses are extremely random, such that virtually no cache hits occur in the external cache. This might be representative of some decision support or business intelligence applications, where virtually all of the cache hits are absorbed in host memory buffers.

11.1.3 Sequential workload

In many user environments, sequential performance is critical due to the heavy use of sequential processing during the batch window. The types of sequential I/O requests that play an important role in batch processing cover a wide range.

11.1.4 Batch jobs

Batch workloads have some common characteristics, such as:

- Frequently, a mixture of random data base accesses, skip-sequential, pure sequential, and sorting.
- Large data transfers and high path utilizations.
- Often constrained to operate within a particular window of time while online operation is restricted or shut down. Poorer or better performance is often not recognized unless it impacts this window.

11.1.5 Sort jobs

Most sorting applications (like the z/OS DFSORT™) are characterized by large transfers for input, output, and work data sets.

Adding to the preceding list, Table 11-1 further provides a summary of the different workload types' characteristics.

Table 11-1 Workload types

Workload type	Characteristics	Representative of
Sequential read	Large record reads - QSAM half track - Open 64 KB blocks Large files from disk	Database backups Large queries Batch Reports

Workload type	Characteristics	Representative of
Sequential write	Large record writes Large files to disk	Database restores and loads Batch
z/OS Cache uniform	Random 4 KB record R/W ratio 3.4 Read hit ratio 84%	Average database CICS®/VSAM IMS
z/OS Cache standard	Random 4 KB record R/W ratio 3.0 Read hit ratio 78%	Representative of 'typical' database conditions
z/OS Cache friendly	Random 4 KB record R/W ratio 5.0 Read hit ratio 92%	Interactive Legacy software
z/OS Cache hostile	Random 4 KB record R/W ratio 2.0 Read hit ratio 40%	DB2 logging
Open read-intensive	Random 4 KB record Read % = 67% Hit ratio 28%	Very large DB DB2
Open standard	Random 4 KB record Read % = 70% Hit ratio 50%	OLTP File system
Open read-intensive	Random 4 KB record Read % = 100% Hit ratio 0%	Decision support Warehousing Large DB inquiry

11.2 Database workloads

Analyzing and discussing database workload characteristics can be a very broad subject. In this section we are going to limit our discussion to a DB2 I/O situation as an example of the database workload demands. The information discussed in this chapter can be further complemented with the information in Chapter 12, "Databases" on page 375.

DB2 environments can often be difficult to typify since there can be wide differences in I/O characteristics. DB2 Query has high read content and is sequential in nature. Transaction environments have more random content, and are sometimes very cache unfriendly, but some other times are very good hit ratios. DB2 has also implemented several changes that affect I/O characteristics, such as sequential pre-fetch and exploitation of I/O priority queuing. Users need to understand the unique characteristics of their installation's processing before generalizing about DB2 performance. However, users running DB2 on zSeries will benefit from the performance improvements provided by the ESS, which will translate into fewer performance tuning issues, shorter duration for large queries, higher logging rates, and faster elapsed times for DB2 utilities such as loads and reorganizations of DB2 tables.

11.2.1 DB2 query

DB2 query workloads can typically be characterized by:

- High read content
- Large transfer size

A DB2 query workload should mostly have the same characteristics as a sequential read workload. ESS sequential pre-fetch algorithms and the exploitation of z/OS-enhanced CCWs provide very good performance for most DB2 queries.

DB2 queries can also benefit from the ESS multiple allegiance and PAV features. Multiple allegiance provides improved device response times for those users exploiting the data sharing features of DB2 V4 or later releases. Without multiple allegiance each system must wait for the other system's I/O to complete before its work can start. When running a query workload alongside a transaction processing workload, the transaction workload usually suffers more than the queries. Multiple allegiance removes the requirement for the transaction workload to wait until a long query finishes before accessing a disk—thus improving the transaction response time without any impact to the query workload.

PAV also allows greater sharing of logical volumes in DB2. Use of DB2 parallel queries in a non-PAV environment requires that each partition is placed on a separate disk volume to ensure the parallelism required. PAV allows multiple partitions to reside on the same logical volume, without any performance penalty. This simplifies data placement for DB2.

11.2.2 DB2 logging

DB2 logging is mostly a very cache unfriendly workload with a high sequential write component. ESS performance for DB2 logging is excellent due to its high sequential write capability.

11.2.3 DB2 transaction environment

DB2 transaction workloads can be characterized by:

- Low to moderate read hits, depending upon the size of DB2 buffers.
- May qualify as cache unfriendly for some applications.
- Deferred writes can cause low write hit ratios.
- Deferred write chains with multiple locate-record commands in chain.
- Can be low read/write ratio, due to reads being satisfied in large DB2 buffer pool.

The ESS-enhanced cache algorithms, together with the high subsystem bandwidth, and minimal RAID-5 write penalty (none for RAID-10), mean that the ESS can support both high subsystem throughput and high transaction rates for DB2 transaction-based workloads.

One of DB2's main advantages is the exploitation of a large buffer pool in processor storage. When managed properly the buffer pool can avoid a large percentage of the accesses to disk. Depending on the application and the size of the buffer pool this can translate to poor cache hit ratios for what in DB2 is called synchronous reads. PAV can be used to increase the throughput to a device even if all accesses are read misses, because each logical volume is spread across seven physical DDMs.

DB2 administrators often require that table spaces and their indexes are placed on separate volumes. This is for both availability and performance reasons. In the ESS data availability is ensured by the RAID implementations (RAID-5 and RAID-10), and performance is improved by PAV. PAV allows real concurrent access to both a table space and an index that reside on the same logical disk device.

11.2.4 DB2 utilities

DB2 utilities such as loads, reorganizations, copies, and recovers have the advantage of the superior sequential bandwidth of the ESS backend (SSA device adapters and ranks), coupled with the concurrency of PAVs. PAVs in particular will keep DB2 utilities that are run

concurrently, such as image copies and reorganizations, from having as large an adverse impact on transaction response times as they do on non-PAV subsystems.

11.3 Application workloads

This section categorizes different types of common applications according to their I/O behavior. The application behavior is typified in four categories:

1. Need for high throughput. These applications need MB/second, and the more the better. Transfers are large read only I/Os and typically sequential access Data Base Management Systems (DBMSs) are used; however, some random DBMS access may exist.
2. Need for high throughput and R/W mix, similar to category 1 (large transfer sizes). In addition to 100 percent read operations, this situation will have a mixture of read and writes in the 70/30 and 50/50 ratios. Here the DBMS is typically sequential, but random and 100 percent write operations also exist.
3. Need for high I/O rate and throughput. This category requires both performance characteristics of IO/s and MB/second. Depending upon the application, the profile is typically sequential access, medium to large transfer sizes (16 KB, 32 KB, and 64 KB), and 100/0, 0/100, and 50/50 R/W ratios.
4. Need for high I/O rate. With many users and many different applications running simultaneously, the category could consist of any of the following: Small to medium sized transfers (4 KB, 8 KB, 16 KB, and 32 KB), 50/50 and 70/30 R/W ratios, and a random DBMS.

These workload categories are summarized in Table 11-2, and the common applications that can be found at any installation are classified following this categorization.

Table 11-2 Application workload types

Category	Application	Read/write ratio		
4	General file serving	All simultaneously	4 KB - 32 KB	Random and sequential
4	Online transaction processing	50/50, 70/30	4 KB, 8 KB	Random
4	Batch update	50/50	16 KB, 32 KB	Random and sequential
1	Data mining	100/0	32 KB, 64 KB, or larger	Mainly sequential, some random
1	Video on demand	100/0	64 KB or larger	Sequential
2	Data warehousing	100/0, 70/30, 50/50	64 KB or larger	Mainly sequential, random easier
2	Engineering and scientific	100/0, 0/100, 70/30, 50/50	64 KB or larger	Sequential
3	Digital video editing	100/0, 0/100, 50/50	32 KB, 64 KB	Sequential

Category	Application	Read/write ratio		
3	Image processing	100/0, 0/100, 50/50	16 KB, 32 KB, 64 KB	Sequential
4	Batch update	50/50	16 KB, 32 KB	Random and sequential

11.3.1 General file serving

This application type consists of many users, running many different applications, all with varying file access sizes and mixtures of read/write ratios, all occurring simultaneously. Applications could include an NT file server, LAN storage, disk arrays, and even Internet/intranet servers. There is no standard profile here other than the ‘chaos’ principle of file access. This application is considered here because this profile covers almost all transfer sizes and R/W ratios.

11.3.2 Online transaction processing

This application category typically has many users, all accessing the same disk storage subsystem and a common set of files. The file access typically is under control of a DBMS and each user may be working on the same or unrelated activities. The I/O requests are typically spread across many files, therefore the file sizes are typically small and randomly accessed. Typical applications consist of a network file server or a disk subsystem being accessed by a sales department entering order information.

11.3.3 Data mining

Databases are the repository of most data, and every time information is needed some type of database is accessed. Data mining is the process of extracting valid, previously unknown, and ultimately comprehensible information from large databases and using it to make crucial business decisions. This application category consists of a number of different operations, each of which is supported by a variety of techniques such as rule induction, neural networks, conceptual clustering, association discovery, and so on. In these applications the DBMS will only extract large sequential or possibly random files depending on the DBMS access algorithms.

Video on demand

Video on demand consists of video playback that can be used to broadcast quality video for either satellite transmission or commercial applications like in room movies. Fortunately for the storage industry, the current data rates needed for this type of transfer have been reduced dramatically due to data compression developments. A broadcast quality video stream now only needs about 3.2 MB/second bandwidth to serve a single user. These advancements have reduced the need for higher speed interfaces and can be serviced with the current interfaces. However, these applications are now demanding numerous concurrent users interactively accessing multiple files within the same storage subsystem. This requirement has changed the environment of video applications in that the storage subsystems will be specified by the number of video streams that they can service simultaneously. In this application the DBMS will only extract large sequential files.

Data warehousing

A data warehouse supports information processing by providing a solid platform of integrated, historical data from which to do analysis. A data warehouse organizes and stores the data needed for informational and analytical processing over a long historical time. A data

warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data used to support the management's decision making process. A data warehouse is always a physically separate store of data that spans a spectrum of time, and the relationships found in the data warehouse are many.

An example of data warehousing is a design around financial applications and functions, such as loans, savings, bank card, and trust for a financial institution. In this application there are basically three kinds of operations: The initial loading, the access, and the updating of the data. However, due to the fundamental characteristics of a warehouse these operations can occur simultaneously. At times this application could perform 100 percent reads when accessing the warehouse; 70 percent reads and 30 percent writes when accessing data while record updating occurs simultaneously; or even 50 percent reads and 50 percent writes when the user load is heavy. Keep in mind that the data within the warehouse is a series of snapshots and once the snapshot of data is made, the data in the warehouse does not change. Therefore, there is typically a higher read ratio when using the data warehouse.

11.3.4 Engineering and scientific applications

The engineering and scientific arena includes hundreds of different applications. Some typical applications are CAD, Finite Element Analysis, simulations and modeling, large scale physics applications, and so on. Some transfers could consist of 1 GB of data for 16 users, while others may require 20 GB of data and hundreds of users. The engineering and scientific areas of business are more concerned with the manipulation of spatial data as well as of series data. This application typically goes beyond standard relational DBMS systems, which manipulate only flat (two-dimensional) data. Spatial or multi-dimensional issues and the ability to handle complex data types are commonplace in engineering and scientific applications.

Object-Relational DBMS (ORDBMS) are now being developed, and they not only offer traditional relational DBMS features, but will additionally support complex data types. Object storage and manipulation can be done, and complex queries at the database level can be made. Object data is data about real world objects, including information about their location, geometry, and topology. Where location describes their position, geometry relates to their shape, and topology includes their relationship to other objects. These applications essentially have an identical profile to that of the data warehouse.

11.3.5 Digital video editing

Digital video editing (DVE) is very popular in the movie industry. The idea that a film editor can load entire feature films onto disk storage and interactively edit and immediately replay the edited clips has become a reality. This application combines the ability to store huge volumes of digital audio and video data onto relatively affordable storage devices to process a feature film. In the near future, when films are being shot on location, there will be no need for standard 38 mm film because all cameras will be directly fed into storage devices and 'takes' will be immediately reviewed. If the 'take' does not turn out as expected, it will be redone immediately. DVE has also been used to generate the latest high tech films that require sophisticated computer-generated special effects.

Depending on the host and operating system used to perform this application, transfers are typically medium to large in size and access is always sequential. Image processing consists of moving huge image files for the purpose of editing. In these applications the user is regularly moving huge high-resolution images between the storage device and the host system. These applications service many desktop publishing and workstation applications. Editing sessions can include loading large files of up to 16 MB into host memory, where users edit, render, modify, and eventually store back onto the storage system. High interface

transfer rates are needed for these applications or the users will waste huge amounts of time waiting to see results. If the interface can move data to and from the storage device at over 32 MB/second then an entire 16 MB image can be stored and retrieved in less than one second. The need for throughput is all important to these applications and, along with the additional load of many users, I/O operations per second are also a major requirement.

12

Databases

This chapter reviews the major IBM database systems and the performance considerations when they are used with the IBM TotalStorage Enterprise Storage Server. In this chapter we limit our discussion to the following situations:

- DB2 in a z/OS environment
- DB2 in an open environment
- IMS in a z/OS environment

The information and discussion contained in this chapter can further be complemented with the following IBM publications:

- *Administration Guide of DB2 for OS/390*, SC26-8957
- *DB2 for OS/390 Administration Guide*, SC26-8952

12.1 DB2

In this section we provide a description of the characteristics of the different database workloads, as well as of the types of data-related objects used by DB2 (in a z/OS environment). Also discussed in this section are the performance considerations and general guidelines when using DB2 with the ESS, as well a description of the tools and reports that can be used for monitoring DB2.

12.1.1 Understanding your database workload

To better understand and position the performance of your particular database system, it is helpful to first learn about the following very common database profiles and their unique workload characteristics.

DB2 online transaction processing (OLTP)

OLTP databases are among the most mission-critical and widely deployed of all. The primary defining characteristic of OLTP systems is that the transactions are processed in real-time or online and often require immediate response back to the user. Examples would be:

- A point of sale terminal in a retail business
- An automated teller machine (ATM) used for bank transactions
- A telemarketing site processing sales orders and checking the inventories

From a workload perspective, OLTP databases typically:

- Process a large number of concurrent user sessions.
- Process a large number of transactions using simple SQL statements.
- Process a single database row at a time.
- Are expected to complete transactions in seconds, not minutes or hours.

OLTP systems process the day-to-day operation of businesses and, therefore, have strict user response and availability requirements. They also have very high throughput requirements and are characterized by large amounts of database inserts and updates. They typically serve hundreds, or even thousands, of concurrent users.

Decision support systems (DSSs)

DSSs differ from the typical transaction-oriented systems in that they most often use data extracted from multiple sources for the purpose of supporting end-user decision making. The types of processing consist of:

- Data analysis applications using pre-defined queries
- Application-generated queries
- Ad-hoc user queries
- Reporting requirements

DSS systems typically deal with substantially larger volumes of data than OLTP systems due to their role in supplying users with large amounts of historical data. Whereas 100 GB of data would be considered large for an OLTP environment, a large DSS system could be 1 terabyte of data or more. The increased storage requirements of DSS systems can also be attributed to the fact that they often contain multiple, aggregated views of the same data.

While OLTP queries are mostly related to one specific business function, DSS queries are often substantially more complex. The need to process large amounts of data results in many CPU intensive database sort and join operations. The complexity and variability of these types of queries must be given special consideration when estimating the performance of a DSS system.

12.1.2 DB2 overview

DB2 is a database management system based on the relational data model. Most users choose DB2 for applications that require good performance and high availability for large amounts of data. This data is stored in data sets mapped to DB2 table spaces and distributed across DB2 databases. Data in table spaces is often accessed using indexes that are stored in index spaces.

Data table spaces can be divided in two groups: *System table spaces* and *user table spaces*. Both of these have identical data attributes. The difference is that system table spaces are used to control and manage the DB2 subsystem and user data. System table spaces require the highest availability and some special considerations. User data cannot be accessed if the system data is not available.

In addition to data table spaces, DB2 requires a group of traditional data sets not associated to table spaces that are used by DB2 to provide data availability: The backup and recovery data sets.

In summary, the three main data set types in a DB2 subsystem are:

- DB2 system table spaces
- DB2 user table spaces
- DB2 backup and recovery data sets

The following sections describe the different objects and data sets that DB2 uses.

12.1.3 DB2 storage objects

DB2 manages data by associating it to a set of DB2 objects. These objects are logical entities, and some of them are kept in storage. The following are DB2 data objects:

- TABLE
- TABLESPACE
- INDEX
- INDEXSPACE
- DATABASE
- STOGROUP

A complete description of all DB2 objects can be found in the IBM publication *DB2 for OS/390 Administration Guide*, SC26-8952. Following we briefly describe each of them.

TABLE

All data managed by DB2 is associated to a table. The table is the main object used by DB2 applications.

TABLESPACE

A table space is used to store one or more tables. A table space is physically implemented with one or more data sets. Table spaces are VSAM linear data sets (LDS). Because table spaces can be larger than the largest possible VSAM data set, a DB2 table space may require more than one VSAM data set.

INDEX

A table can have one or more indexes (or can have no index). An index contains keys. Each key may point to one or more data rows. The purpose of an index is to get direct and faster access to the data in a table.

INDEXSPACE

An index space is used to store an index. An index space is physically represented by one or more VSAM LDS data sets.

DATABASE

Database is a DB2 representation of a group of related objects. Each of the previously named objects has to belong to a database. DB2 databases are used to organize and manage these objects.

STOGROUP

A DB2 storage group is a list of storage volumes. STOGROUPs are assigned to databases, table spaces, or index spaces when using DB2 managed objects. DB2 uses STOGROUPs for disk allocation of the table and index spaces.

Installations that are SMS managed can define STOGROUP with VOLUME(*). This specification implies that SMS assigns a volume to the table and index spaces in that STOGROUP. In order to do this, SMS uses ACS routines to assign a storage class, a management class and a storage group to the table or index space.,

12.1.4 DB2 data set types

As already mention, DB2 uses system and user table spaces for the data, as well as a group of data sets not associated with table spaces that are used by DB2 to provide data availability; these are backup and recovery data sets.

DB2 system table spaces

DB2 uses databases to control and manage its own operation and the application data.

- The catalog and directory databases

  Both databases contain DB2 system tables. DB2 system tables hold data definitions, security information, data statistics, and recovery information for the DB2 system. The DB2 system tables reside in DB2 system table spaces.

  The DB2 system table spaces are allocated when a DB2 system is first created. DB2 provides the IDCAMS statements required to allocate these data sets as VSAM LDSs.

- The work database

  The work database is used by DB2 to resolve SQL queries that require temporary work space. Multiple table spaces can be created for the work database.

DB2 application table spaces

All application data in DB2 is organized in the objects as described in 12.1.3, “DB2 storage objects” on page 377.

Application table spaces and index spaces are VSAM LDS data sets with the same attributes as DB2 system table spaces and index spaces. The difference between system and application data is made only because they have different performance and availability requirements.

DB2 recovery data sets

In order to provide data integrity, DB2 uses data sets for recovery purposes. Following is a brief description of these DB2 recovery data sets. These data sets are described in further detail in the IBM publication *Administration Guide of DB2 for OS/390*, SC26-8957.

- Bootstrap data set

  DB2 uses the bootstrap data set (BSDS) to manage recovery and other DB2 subsystem information. The BSDS contains information needed to restart and to recover DB2 from any abnormal circumstance. For example, all log data sets are automatically recorded with the BSDS. While DB2 is active, the BSDS is open and updated.

  DB2 always requires two copies of the BSDS because they are critical for data integrity.

- Active logs

  The active log data sets are used for data recovery and to ensure data integrity in case of software or hardware errors.

  DB2 uses active log data sets to record all updates to user and system data. The active log data sets are open as long as DB2 is active. Active log data sets are reused when the total active log space is used up, but only after the active log (to be overlaid) has been copied to an archive log.

  DB2 supports dual active logs. It is strongly recommend that you use dual active logs for all DB2 production environments.

- Archive logs

  Archive log data sets are DB2 managed backups of the active log data sets. Archive logs data sets are automatically created by DB2 whenever an active log is filled. DB2 supports dual archive logs, and it is recommended that you use dual archive log data sets for all production environments.

  Archive log data sets are sequential data sets that can be defined on disk or on tape, migrated and deleted with standard procedures.

12.2 DB2 with the ESS - Performance recommendations

When using an ESS, the following generic recommendations will be useful when planning for a good DB2 performance. These are generic recommendations and as such can be used.

For a more detailed and accurate approach that takes into consideration the particularities of your DB2 environment, you should contact your IBM representative who can assist you with the ESS capacity and configuration planning.

12.2.1 Know where your data resides

DB2 storage administration can be done using SMS to simplify disk use and control, or also without using SMS. In both cases it is very important that you know where your data resides.

If you want optimal performance from ESS, do not treat it totally like a ‘black box’. Understand how DB2 tables map to underlying volumes, and how the volumes map to RAID arrays.

Establish a storage allocation policy that uses as many ESS ranks as practically wise.

12.2.2 Balance workload across ESS resources

You can balance workload activity across ESS resources doing the following:

- Spread DB2 data across ESS boxes if practical.
- Spread DB2 data across clusters in each ESS.
- Spread DB2 data across ESS device adapters.
- Spread DB2 data across as many ranks as practical.

Figure 12-1 illustrates this technique. Assuming that there are eight ESS ranks available for DB2, spread your data through all the arrays. For each storage group, select a group of volumes from each array. In this way, table spaces assigned to one storage group will be spread across arrays, adapters, and clusters in an ESS and across ESSs.

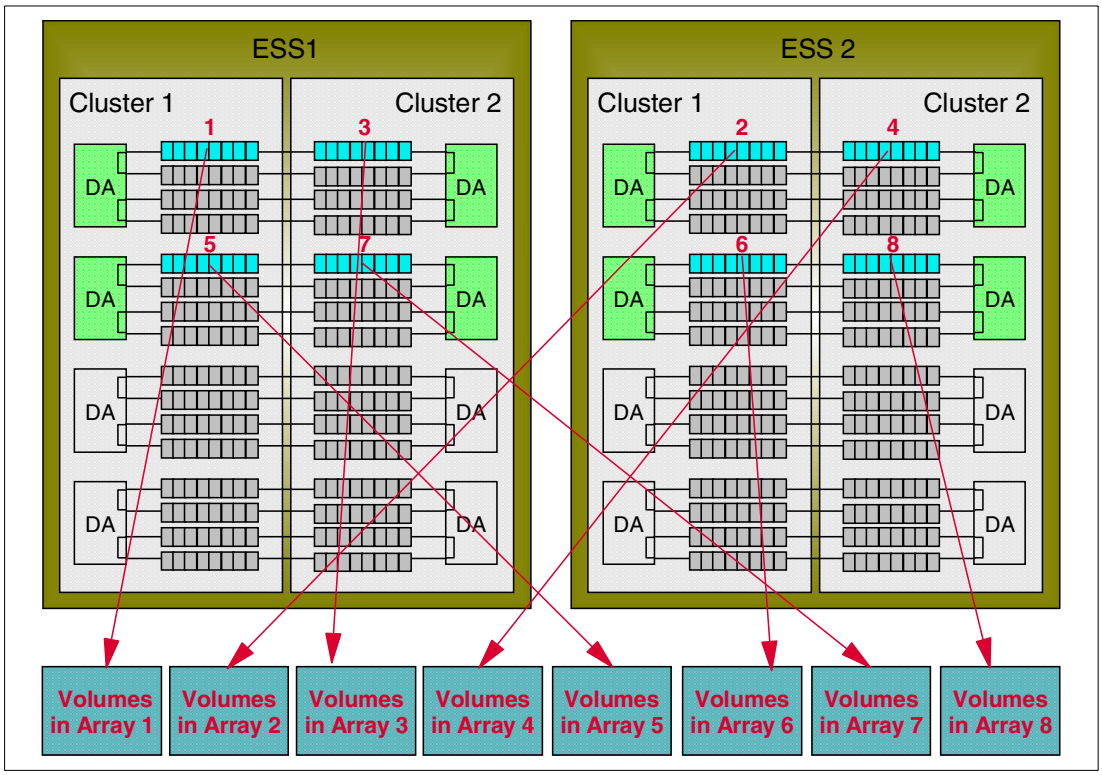


Figure 12-1 Allocating DB2 volumes, spread the data

In addition, consider the following:

- You may intermix tables and indexes, and also system, application, and recovery data sets, on ESS ranks. The I/O activity will be more evenly spread, and skew effects avoided that otherwise you could see if the components were isolated.
- You can use a vertical mapping where volumes are spread evenly across all arrays assigned to DB2 in the ESS.

12.2.3 Take advantage of VSAM data striping

Before VSAM data striping was available, in a multi-extent, multi-volume VSAM data set sequential processing did not present any type of parallelism. This meant that when an I/O operation was executed for an extent in a volume, no other activity from the same task was scheduled to the other volumes.

VSAM data striping addresses this problem with two modifications to the traditional data organization:

- The records are not placed in *key ranges* along the volumes; instead they are organized in stripes.
- Parallel I/O operations are scheduled to sequential stripes in different volumes.

By striping data the VSAM control intervals (CIs) are spread across multiple devices. This format allows a single application request for records in multiple tracks and CIs to be satisfied by concurrent I/O requests to multiple volumes.

The result is improved data transfer to the application. The scheduling of I/O to multiple volumes in order to satisfy a single application request is referred as an I/O *path packet*.

If you refer again to Figure 12-1 on page 380 you can see that we can stripe across arrays, across device adapters, across clusters, and across ESSs.

If you plan to enable VSAM I/O striping refer to the following publication for additional discussion: *DB2 for z/OS and OS/390 Version 7 Performance Topics,* SG24-6129.

12.2.4 Large volumes

With ESS large volume support, zSeries users can allocate the larger capacities available in the ESS, with the set of 256 device addresses that each LSS provides. From the ESS perspective, the capacity of a volume does not determine its performance. From the z/OS perspective, PAVs reduce or eliminate any additional enqueues that may originate from the increased I/O on the larger volumes. From the storage administration perspective, configurations with larger volumes are simpler to manage.

Before large volume support, OS/390 and z/OS limited the maximum volume size to 10017 cylinders (3390-9). With large volumes support, the maximum supported is up to 32760 cylinders.

Measurements oriented to determine how large volumes can impact DB2 performance have shown that similar response times can be obtained when using larger volumes as when using the smaller 3390-3 standard size volumes (refer to 9.5.2, “Larger vs. smaller volumes performance examples” on page 328, for a discussion). Additionally, in Figure 12-2 on page 382 you can see the results for a DB2 transaction+query processing comparing standard size 3390-3 vs. an equivalent total capacity using six larger volumes.

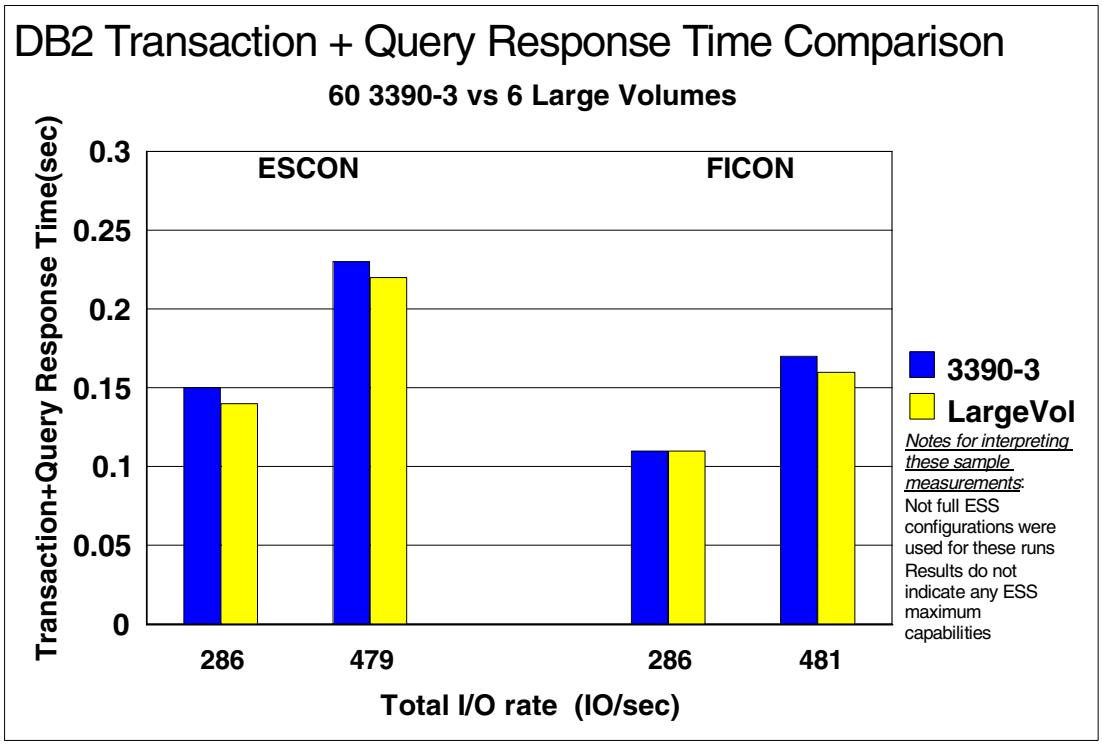


Figure 12-2 DB2 transaction + query response time comparison

With fewer large volumes similar performance can be achieved. However, system administration is simplified. Large volume support and volume size performance considerations are discussed in detail in 9.5, “Logical volume sizes” on page 326.

12.2.5 Additional capacity planning considerations

Some additional considerations when planning the ESS requirements for your database server performance demands are the following.

Throughput

How do you know what will be the throughput requirement? The best way, and not always an option, is to measure the performance of a current production system. If you do not have a running system to measure, consider reviewing benchmark results that have been completed for similar applications.

Workload type

It will be important to know what workload type your application fits in, so the general recommendations can be applied. In 12.1.1, “Understanding your database workload” on page 376, two main workload types were discussed: Online transaction processing and decision support systems.

Disk Magic

Despite that the generic recommendations can be used as such, for a more detailed and accurate estimation that takes into consideration the particularities of your DB2 environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

12.2.6 Monitoring the ESS performance

For a discussion of the ESS performance monitoring tools and recommendations refer to 12.8, “Monitoring tools in a database environment” on page 400.

12.3 IMS

This section discusses IMS, its logging, and the performance considerations when IMS uses the IBM TotalStorage Enterprise Storage Server for its data.

12.3.1 IMS overview

IMS consists of three components, the Transaction Manager (TM) component, the Database Manager (DB) component, and a set of system services that provides common services to the other two components.

IMS Transaction Manager

The IMS Transaction Manager provides a network with access to the applications running under IMS. The users can be people at terminals or workstations, or other application programs.

IMS Database Manager

The IMS Database Manager provides a central point of control and access to the data that is processed by IMS applications. The Database Manager component of IMS supports databases using IMS’s own hierarchic database model. It provides access to the databases from the applications running under the IMS Transaction Manager, the CICS transaction monitor, and z/OS batch jobs.

It provides functions for preserving the integrity and maintaining the databases. It allows multiple tasks to access and update the data, while ensuring the integrity of the data. It also provides functions for reorganizing and restructuring the databases.

The IMS databases are organized internally using a number of IMS’s own internal database organization access methods. The database data is stored on disk storage using the normal operating system access methods.

IMS system services

There are a number of functions that are common to the Database Manager and Transaction Manager:

- Restart and recovery of the IMS subsystems failures
- Security: Controlling access to IMS resources
- Managing the application programs: Dispatching work, loading application programs, providing locking services
- Providing diagnostic and performance information
- Providing facilities for the operation of the IMS subsystem
- Providing an interface to other z/OS subsystems that interface with the IMS applications.

12.3.2 IMS logging

The IMS logging is one of the most write-intensive operations in a database environment.

During IMS execution, all information necessary to restart the system in the event of a failure is recorded on a system log data set. The IMS logs are made up of the following.

IMS log buffers

The log buffers are used to write the information that needs to be logged.

Online log data sets (OLDS)

The OLDS are data sets that contain all the log records required for restart and recovery. These data sets must be pre-allocated on DASD and will hold the log records until they are archived.

The OLDS are made of multiple data sets that are used in a wrap-around manner. At least three data sets must be allocated for the OLDS to allow IMS to start, while an upper limit of 100 is supported.

Only complete log buffers are written to OLDS, to enhance performance. Should any incomplete buffers need to be written out, they are written to the WADS.

Write ahead data sets (WADS)

The WADS is a small direct access data set that contains a copy of committed log records that are in OLDS buffers, but have not yet been written to OLDS.

When IMS processing requires writing of a partially filled OLDS buffer, a portion of the buffer is written to the WADS. If IMS or the system fails, the log data in the WADS is used to terminate the OLDS, which can be done as part of an emergency restart, or as an option on the IMS Log Recovery Utility.

The WADS space is continually reused after the appropriate log data has been written to the OLDS. This data set is required for all IMS systems, and must be pre-allocated and formatted at IMS start-up when first used.

System log data sets (SLDS)

The SLDS is created by the IMS log archive utility, preferably after every OLDS switch. It is usually placed on tape, but can reside on disk. The SLDS can contain the data from one or more OLDS data sets.

Recovery log data sets (RLDS)

When the IMS log archive utility is run, the user can request creation of an output data set that contains all of the log records needed for database recovery. This is the RLDS and also known to DBRC. The RLDS is optional.

12.4 ESS considerations for IMS

Some of the benefits of using the IBM TotalStorage Enterprise Storage Server in an IMS environment are the following:

- IMS takes advantage of the ESS PAV function (described in detail in 9.2, “Parallel Access Volumes” on page 308) that allows multiple concurrent I/Os to the same volume at the same time from applications running on a z/OS system image.
- Less disk contention when accessing the same volumes from different systems in an IMS data sharing group, using ESS Multiple Allegiance function (described in 9.3, “Multiple Allegiance” on page 324).

- Faster paging devices that can help avoid performance problems for IMS transactions.
- ESS supports track, partial track, and record level caching, which avoids the potential inefficiencies of only one type of cache stage operation. ESS monitors the operations and automatically switches between the different types of caching, according to which provides the best performance advantage to IMS and other systems.
- IMS exploitation of DASD fast write. DASD fast write (DFW) or Fast Write performs write operations at cache speeds and has the potential for improving the performance of critical IMS data sets.

  Writing to disk is not required to complete a DFW operation. A copy of the data to be written is placed in the ESS host adapter and in NVS, and the storage controller returns channel-end and device-end. Where databases are frequently updated, standard caching and DFW can be very beneficial.
- By using FICON channels with ESS to attach logging data sets, database data sets and IMS system data set volumes, performance improvements can be obtained by IMS:
  - The IMS OLDS logging bandwidth increases significantly when using FICON paths vs. using ESCON paths. Also the channel path busy percentage dramatically drops.
  - Running IMS-intensive I/O workloads, database response times are similar or better when using FICON vs. ESCON plus providing a 4 to 1 consolidation in the number of connections.

The most significant impacts of using ESS can be observed in IMS logging and massive I/O processes such as BMPs and database utilities.

12.5 IMS with the ESS - Performance recommendations

When using an ESS, the following generic recommendations will be useful when planning for a good IMS performance. These are generic recommendations and as such can be used.

For a more detailed and accurate approach that takes into consideration the particularities of your IMS environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

12.5.1 Balance workload across ESS resources

You can balance workload activity across ESS resources by doing the following:

- Spread IMS data across ESS boxes if practical.
- Spread IMS data across clusters in each ESS.
- Spread IMS data across ESS device adapters.
- Spread IMS data across ESS ranks.

Figure 12-1 on page 380 illustrates this technique. The point is to spread the IMS data over arrays, adapters, and clusters in an ESS and across ESSs.

12.5.2 Large volumes

With ESS large volume support, zSeries users can allocate the larger capacities available in the ESS, with the set of 256 device addresses that each LSS provides. From the ESS perspective, the capacity of a volume does not determine its performance. From the z/OS perspective, PAVs reduce or eliminate any additional enqueues that may originate from the increased I/O on the larger volumes. From the storage administration perspective, configurations with larger volumes are simpler to manage.

Before large volume support, OS/390 and z/OS limited the maximum volume size to 10017 cylinders (3390-9). With large volumes support, the maximum supported is up to 32760 cylinders.

Measurements oriented to determine how large volumes can impact IMS performance have shown that similar response times can be obtained when using larger volumes as when using the smaller 3390-3 standard size volumes (refer to 9.5.2, “Larger vs. smaller volumes performance examples” on page 328, for a discussion on large volumes).

Figure 12-3 illustrates the device response times when using thirty-two standard 3390-3 volumes vs. four larger volumes (that add a similar total capacity). The results show that with the larger volumes the response times are similar to the standard size 3390-3 volumes.

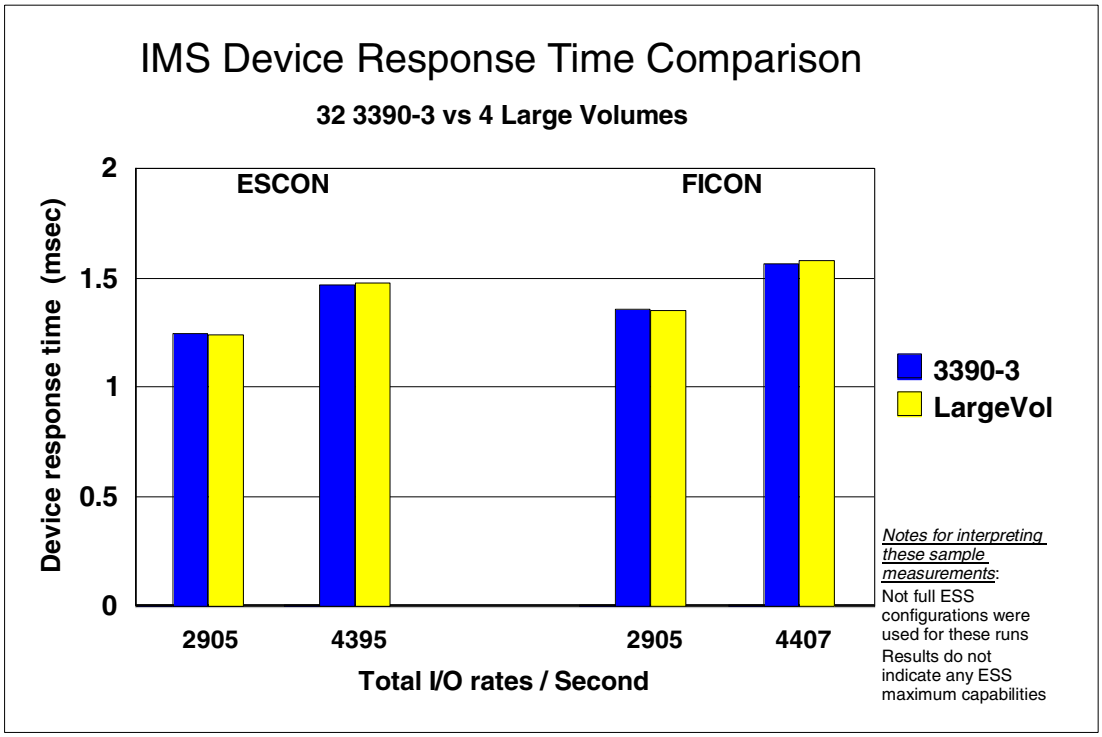


Figure 12-3 IMS device response times - Large volumes vs. standard 3390-3

Figure 12-4 on page 387 shows the IMS transaction response times, which for all practical purposes were the same when using larger volumes as compared to the standard size 3390-3 volumes.

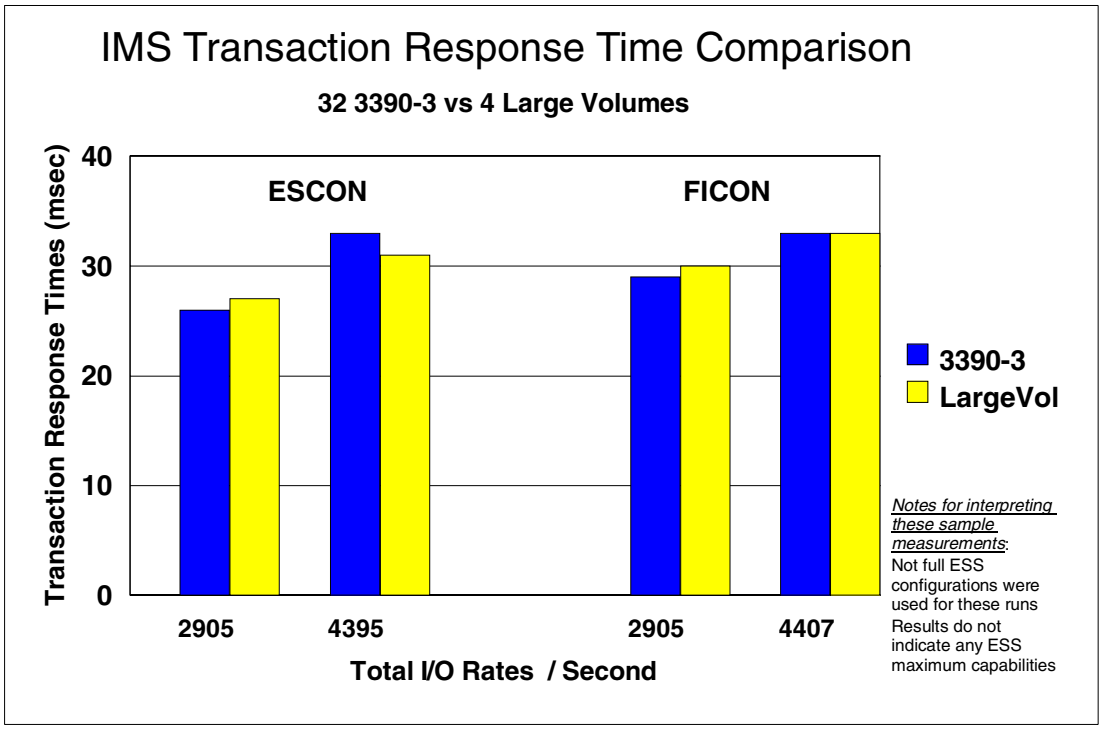


Figure 12-4 IMS transaction response time - Large volumes vs. standard 3390-3

The dynamic alias tuning function of the Workload Manager assigns the additional PAV aliases as needed, thus ensuring the reduction or elimination of the IOSQ times.

12.5.3 Additional capacity planning considerations

Some additional considerations when planning the ESS requirements for your database server performance demands are the following.

Throughput

How do you know what will be the throughput requirement? The best way, and not always an option, is to measure the performance of a current production system. If you do not have a running system to measure, consider reviewing benchmark results that have been completed for similar applications.

Workload type

It will be important to know what workload type your application fits in, so the general recommendations can be applied. In 12.1.1, “Understanding your database workload” on page 376, two main workload types were discussed: Online transaction processing and decision support systems.

Disk Magic

Despite that the generic recommendations can be used as such, for a more detailed and accurate estimation that takes into consideration the particularities of your IMS environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

12.5.4 Monitoring the ESS performance

For a discussion of the ESS performance monitoring tools and recommendations refer to 12.8, “Monitoring tools in a database environment” on page 400.

12.6 DB2 UDB - Open environment

This section discusses the performance considerations when using the DB2 Universal Database™ (DB2 UDB) with the IBM TotalStorage Enterprise Storage Server in an open environment.

The information presented in this section is further discussed in detail in *IBM ESS and IBM DB2 UDB Working Together*, SG24-6262.

12.6.1 DB2 UDB storage concepts

DB2 Universal Database (DB2 UDB) is IBM’s object-relational database for UNIX, Linux, OS/2, and Windows operating environments.

The database object that maps the physical storage is the *tablespace*. Figure 12-5 illustrates how DB2 UDB is logically structured and how the tablespace maps the physical object.

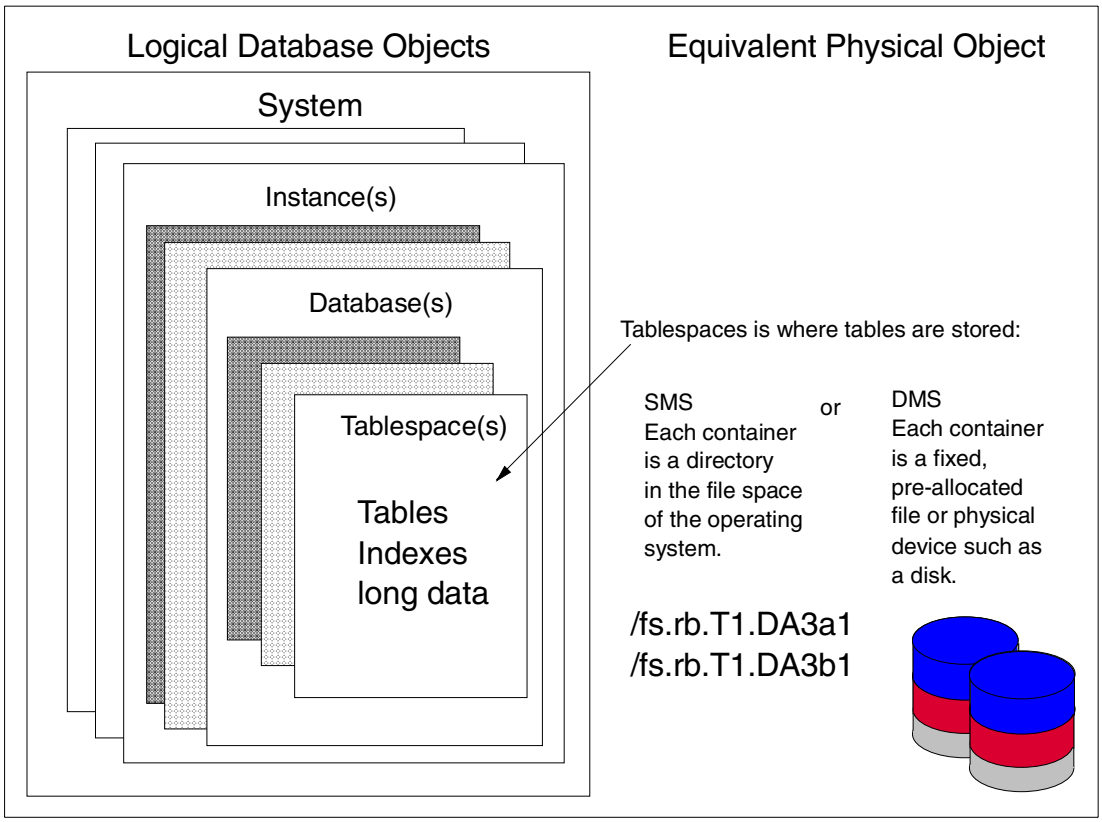


Figure 12-5 DB2 UDB logical structure

Instances

An *instance* is a logical database manager environment where databases are cataloged and configuration parameters are set. An instance is similar to an image of the actual database manager environment. You can have several instances of the database manager product on the same database server. You can use these instances to separate the development environment from the production environment, tune the database manager to a particular environment, and protect sensitive information from a particular group of users.

For partitioned database systems, all data partitions will reside within a single instance.

Databases

A relational database structures data as a collection of database objects. The primary database object is the table (a defined number of columns and any number of rows). Each database includes a set of system catalog tables that describe the logical and physical structure of the data, configuration files containing the parameter values allocated for the database, and recovery log(s).

DB2 UDB allows multiple databases to be defined within a single database instance. Configuration parameters can also be set at the database level, thus allowing you to tune, for example, memory usage and logging.

Nodes and database partitions

A *node* number in DB2 UDB terminology is equivalent to a *data partition*. Databases that are comprised of multiple data partitions and that reside on an SMP system are also called multiple logical node (MLN) databases.

Nodes are identified by the physical system (CPU node) where they reside by a unique node number. The node number, which can be from 0 to 999, uniquely defines a node. Node numbers must be in ascending sequence (gaps in the sequence are allowed).

The configuration information of the database is stored in the catalog node. The catalog node is the node from which you create the database.

Nodegroups

A *nodegroup* is a set of one or more database partitions. For non-partitioned implementations (all editions except for EEE), the nodegroup is always made up of a single partition.

Partitioning map

When a nodegroup is created, a *partitioning map* is associated with it. The partitioning map in conjunction with the partitioning key and hashing algorithm is used by the database manager to determine which database partition in the nodegroup will store a given row of data. Partitioning maps do not apply to non-partitioned databases.

Containers

A *container* is the way of defining where on the storage device will the database objects be stored. Containers may be assigned from file systems by specifying a directory. These are identified as PATH containers. Containers may also reference files that reside within a directory. These are identified as FILE containers and a specific size must be identified. Containers may also reference raw devices. These are identified as DEVICE containers, and the device must already exist on the system before the container can be used.

All containers must be unique across all databases; a container can belong to only one tablespace.

Table spaces

A database is logically organized in *table spaces*. A table space is a place to store tables. To spread a table space over one or more disk devices you simply specify multiple containers.

For partitioned databases, the table spaces reside in nodegroups. In the create tablespace command execution, the containers themselves are assigned to a specific node in the nodegroup, thus maintaining the 'shared nothing' character of DB2 UDB EEE.

Tablespaces can be either system managed space (SMS) or data managed space (DMS). For an SMS table space, each container is a directory in the file system, and the operating system file manager controls the storage space (LVM for AIX). For a DMS table space, each container is either a fixed-size pre-allocated file, or a physical device such as a disk (or in the case of the ESS, a vpath), and the database manager controls the storage space.

There are three main types of user table spaces: *Regular* (index and/or data), *temporary*, and *long*. In addition to these user-defined table spaces, DB2 requires a system table space, the catalog table space, to be defined. For partitioned database systems this catalog table space resides on the catalog node.

Tables, indexes, and LOBs

A table is a named data object consisting of a specific number of columns and unordered rows. Tables are uniquely identified units of storage maintained within a DB2 tablespace. They consist of a series of logically linked blocks of storage that have been given the same name. They also have a unique structure for storing information that allows the information to be related to information on other tables

When creating a table you can choose to have certain objects, such as indexes and large object (LOB) data, stored separately from the rest of the table data. In order to do this, the table must be defined to a DMS table space.

Indexes are defined for a specific table and assist in the efficient retrieval of data to satisfy queries. They can also be used to assist in the clustering of data.

Large objects (LOBs) can be stored in columns of the table. These objects, although logically referenced as part of the table, may be stored in their own table space when the base table is defined to a DMS table space. This allows for more efficient access of both the LOB data and the related table data.

Pages

Data is transferred to and from devices in discrete blocks that are buffered in memory. These discrete blocks are called *pages*, and the memory reserved to buffer a page transfer is called an I/O buffer. DB2 UDB supports various page sizes including 4 k, 8 k, 16 k, and 32k.

When an application accesses data randomly, the page size determines the amount of data transferred. This corresponds to the size of data transfer request done to the ESS, which is sometimes referred to as the *physical record*.

Sequential read patterns can also influence the page size selected. Larger page sizes for workloads with sequential read patterns can enhance performance by reducing the number of I/Os.

Extents

An *extent* is a unit of space allocation within a container of a table space for a single table space object. This allocation consists of multiple pages. The extent size (number of pages) for an object is set when the table space is created.

- An extent is a group of consecutive pages defined to the database.
- The data in the tables spaces is striped by extent across all the containers in the system.

Bufferpools

A *bufferpool* is main memory allocated in the host processor to cache table and index data pages as they are being read from disk or being modified. The purpose of the bufferpool is to improve system performance. Data can be accessed much faster from memory than from disk; therefore, the fewer times the database manager needs to read from or write to disk (I/O) the better the performance. Multiple buffer pools can be created.

DB2 pre-fetch (reads)

Pre-fetching is a technique for anticipating data needs and *reading ahead* from storage in large blocks. By transferring data in larger blocks, fewer system resources are used and less time is required.

Sequential pre-fetch reads consecutive pages into the buffer pool before they are needed by DB2. List pre-fetches are more complex. In this case the DB2 optimizer optimizes the retrieval of randomly located data.

The amount of data being pre-fetched determines the amount of parallel I/O activity. Ordinarily the database administrator should define a pre-fetch value large enough to allow parallel use of all of the available containers.

Consider the following example:

- A tablespace is defined with a page size of 16 KB using raw DMS.
- The tablespace is defined across four containers, each container residing on a separate logical device, and the logical devices are on different ESS ranks.
- The extent size is defined as 16 pages (or 256 KB).
- The prefetch value is specified as 64 pages (number of containers * extent size).
- A user makes a query that results in a tablespace scan, which then results in DB2 performing a prefetch operation.

The following will happen:

- DB2, recognizing that this prefetch request for 64 pages (1 MB) evenly spans four containers, will do four parallel I/O requests, one against each of those containers. The request size to each container will be 16 pages (or 256 KB).
- The AIX Logical Volume Manager will break the 256 KB request to each AIX logical volume into smaller units (128 KB is the largest), and pass them on to the ESS as ‘back-to-back’ requests against each ESS logical disk.
- As the ESS receives a request for 128 KB (if the data is not in cache), four ranks will operate in parallel to retrieve the data.
- After receiving several of these requests, the ESS will recognize that these DB2 prefetch requests are arriving as sequential accesses, causing the ESS sequential prefetch to take effect. This will result in all of the disks in all four ESS ranks to operate concurrently, staging data to the ESS cache, to satisfy the DB2 prefetch operations.

Page cleaners

Page cleaners are present to make room in the buffer pool before pre-fetchers read pages on disk storage and move them into the buffer pool. For example, if a large amount of data has been updated in a table, many data pages in the buffer pool may be updated but not written into disk storage (these pages are called dirty pages). Since pre-fetchers cannot place fetched data pages onto the dirty pages in the buffer pool, these dirty pages must be flushed to disk storage and become *clean* pages so that pre-fetchers can place fetched data pages from disk storage.

Logs

Changes to data pages in the buffer pool are logged. Agent processes updating a data record in the database update the associated page in the buffer pool and write a log record into a log buffer. The written log records in the log buffer will be flushed into the log files asynchronously by the logger. With UNIX, you can see a logger process (db2loggr) for each active database using the **ps** command.

To optimize performance neither the updated data pages in the buffer pool nor the log records in the log buffer are written to disk immediately. They are written to disk by page cleaners and the logger, respectively.

The logger and the buffer pool manager cooperate and ensure that the updated data page is not written to disk storage before its associated log record is written to the log. This behavior ensures that the database manager can obtain enough information from the log to recover and protect a database from being left in an inconsistent state when the database has crashed as a result of an event, such as a power failure.

Parallel operations

DB2 UDB extensively uses *parallelism* to optimize performance when accessing a database. DB2 supports several types of parallelism:

- Query
- I/O
- Utility

Query parallelism

There are two dimensions of query parallelism: *Inter-query parallelism* and *intra-query parallelism*. Inter-query parallelism refers to the ability of multiple applications to query a database at the same time. Each query executes independently of the others, but they are all executed at the same time. Intra-query parallelism refers to the simultaneous processing of parts of a single query, using *intra-partition parallelism*, *inter-partition parallelism*, or both.

- Intra-partition parallelism subdivides what is usually considered a single database operation, such as index creation, database loading, or SQL queries, into multiple parts, many or all of which can be run in parallel within a single *database partition.*
- Inter-partition parallelism subdivides what is usually considered a single database operation, such as index creation, database loading, or SQL queries, into multiple parts, many or all of which can be run in parallel across multiple partitions of a partitioned database on one machine or on multiple machines. Inter-partition parallelism only applies to EEE.

I/O parallelism

When there are multiple *containers* for a tablespace, the database manager can exploit parallel I/O. Parallel I/O refers to the process of writing to, or reading from, two or more I/O devices simultaneously. This can result in significant improvements in throughput.

DB2 implements a form of data striping by spreading the data in a tablespace across multiple *containers*. In storage terminology, the part of a stripe that is on a single device is a *strip*. The DB2 term for strip is *extent*. If your tablespace has three containers, DB2 will write one extent to container 0, the next extent to container 1, the next extent to container 2, then back to container 0. The stripe *width*—a generic term not often used in DB2 literature—is equal to the number of containers, or three in this case.

Extent sizes are normally measured in numbers of DB2 pages.

Containers for a tablespace would ordinarily be placed on separate physical disks, allowing work to be spread across those disks, and allowing disks to operate in parallel. Since the ESS logical disks are striped across the rank, the database administrator can allocate DB2 containers on separate logical disks residing on separate ESS arrays. This will take advantage of the parallelism both in DB2 and in the ESS. For example, four DB2 containers residing on four ESS logical disks on four different 7+P ranks will have data spread across 32 physical disks.

Utility parallelism

DB2 utilities can take advantage of intra-partition parallelism. They can also take advantage of inter-partition parallelism: Where multiple database partitions exist, the utilities execute in each of the partitions in parallel.

The load utility can take advantage of intra-partition parallelism and I/O parallelism. Loading data is a CPU-intensive task. The load utility takes advantage of multiple processors for tasks, such as parsing and formatting data. It can also use parallel I/O servers to write the data to containers in parallel.

In a partitioned database environment, the Auto Loader utility takes advantage of intra-partition, inter-partition, and I/O parallelism by parallel invocations of the LOAD command at each database partition where the table resides.

During index creation, the scanning and subsequent sorting of the data occurs in parallel. DB2 exploits both I/O parallelism and intra-partition parallelism when creating an index. This helps to speed up index creation when a CREATE INDEX statement is issued, during restart (if an index is marked invalid), and during the reorganization of data.

Backing up and restoring data are heavily I/O-bound tasks. DB2 exploits both I/O parallelism and intra-partition parallelism when performing backup and restore operations. Backup exploits I/O parallelism by reading from multiple table space containers in parallel, and asynchronously writing to multiple backup media in parallel. Refer to the BACKUP DATABASE command and the RESTORE DATABASE command discussion in *DB2 Command Reference for Common Server V2,* S20H-4645, for information on how to enable parallelism for these utilities.

12.7 DB2 UDB with the ESS - Performance recommendations

When using an ESS, the following generic recommendations will be useful when planning for good DB2 UDB performance. These are generic recommendations and as such can be used.

For a more detailed and accurate approach that takes into consideration the particularities of your DB2 UDB environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

12.7.1 Know where your data resides

Know where your data resides. Understand how DB2 containers map to ESS logical disks, and how those logical disks are distributed across the ESS ranks. Spread DB2 data across as many ESS ranks as possible.

If you want optimal performance from ESS, do not treat it completely like a black box. Establish a storage allocation policy that allocates data using several ESS ranks. Understand how DB2 tables map to underlying logical disks, and how the logical disks are allocated across the ESS arrays.

One way of making this process easier to manage is to maintain a modest number of ESS logical disks. Balance workload across ESS resources. Establish a storage allocation policy that allows a balanced workload activity across RAID arrays. You can take advantage of the inherent balanced activity and parallelism within DB2, spreading the work for DB2 partitions and containers across the ESS arrays. This applies to both OLTP and DSS workload types. If you do that, and have planned sufficient resource, then many of the other decisions become secondary.

The following general recommendations can be considered:

- DB2 query parallelism allows workload to be balanced across CPUs and, if DB2 Universal Database Enterprise-Extended Edition (EEE) is installed, across data partitions.
- DB2 I/O parallelism allows workload to be balanced across containers.

As a result, you can balance activity across ESS resources by following these rules:

- Span ESS boxes.
- Span clusters within a box.
- Span disk adapters.
- Engage as many arrays as possible.

Figure 12-6 on page 395 illustrates this technique for a single table space consisting of eight containers.

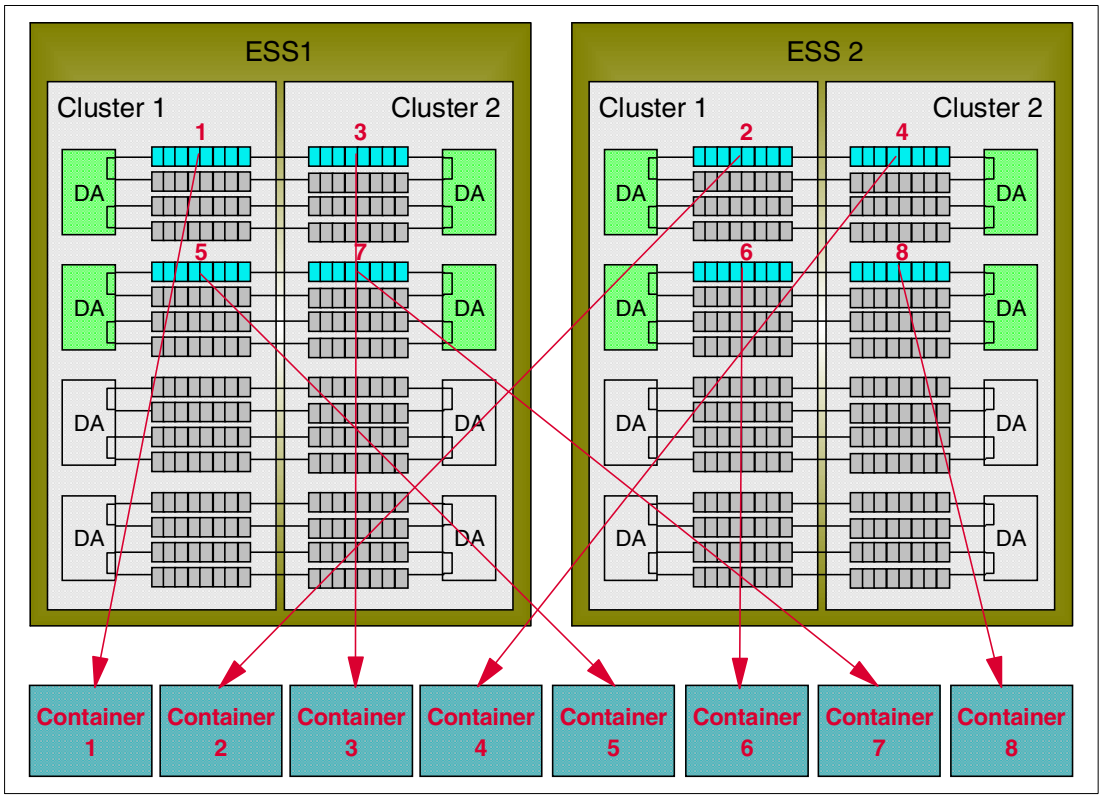


Figure 12-6 Allocating DB2 containers using a "spread your data" approach

In addition, consider the following:

- You may intermix data, indexes, and temp spaces on the ESS ranks. Your I/O activity will be more evenly spread and thus will avoid the skew effect, which you would otherwise see if the components were isolated.
- For EEE systems, establish a policy that allows partitions and containers within partitions to be spread evenly across ESS resources. You can choose either a horizontal mapping, in which every partition has containers on every available ESS array; or a vertical mapping, in which DB2 partitions are isolated to specific arrays, with containers spread evenly across those arrays.
- For EEE systems, selection of horizontal or vertical mapping approaches may be influenced as you consider the number of DB2 partitions, the number of arrays, and how future growth will influence those factors. The vertical mapping approach works well as long as the number of ESS ranks is an even multiple of the number partitions. Otherwise, the horizontal approach is probably best.

12.7.2 Use DB2 to stripe across containers

Use the inherent striping of DB2, placing containers for a tablespace on separate ESS logical disks on different ESS ranks. This will eliminate the need for using underlying operating system or logical volume manager striping.

Look again at Figure 12-6. In this case, we are striping across arrays, across disk adapters, across clusters, and across ESS boxes. This can all be done using the striping capabilities of DB2's container and *shared nothing* concept. This eliminates the need to employ AIX logical volume striping.

12.7.3 Selecting DB2 logical sizes

The three settings in a DB2 system that primarily affect the movement of data to and from the disk subsystem work together. These are:

- Page size
- Extent size
- Prefetch size

Page size

Page sizes are defined for each tablespace. There are four supported page sizes: 4 K, 8 K, 16 K, and 32 K. Some factors that affect the choice of page size include:

- The maximum number of records per page is 255. To avoid wasting space on a page, do not make page size greater than 255 times the row size plus the page overhead.
- The maximum size of a table space is proportional to the page size of its table space. In SMS, the data and index objects of a table have limits, as shown in Table 12-1. In DMS, these limits apply at the tablespace level.

Table 12-1 Page size relative to tablespace size

Page size	Maximum data/index object size
4 KB	64 GB
8 KB	128 GB
16 KB	256 GB
32 KB	512 GB

Select a page size that can accommodate the total expected growth requirements of the objects in the tablespace.

For OLTP applications that perform random row read and write operations, a smaller page size is preferable, because it wastes less buffer pool space with unwanted rows. For DSS applications that access large numbers of consecutive rows at a time, a larger page size is better, because it reduces the number of I/O requests that are required to read a specific number of rows.

Tip: Experience indicates that page size can be dictated to some degree by the type of workload. For pure OLTP workloads a 4 KB page size is recommended. For a pure DSS workload a 32 KB page size is recommended. For a mixture of OLTP and DSS workload characteristics we recommend either an 8 KB page size or a 16 KB page size.

Extent size

If you want to stripe across multiple arrays in your ESS, then assign a logical disk from each array to be used as a DB2 container. During writes, DB2 will write one extent to the first container, the next extent to the second container, and so on until all eight containers have been addressed before cycling back to the first container. DB2 stripes across containers at the tablespace level.

Since ESS stripes at a fairly fine granularity (32 KB), selecting multiples of 32 KB for extent size makes sure that multiple ESS disks are used within an array when a DB2 prefetch occurs.

I/O performance is fairly insensitive to selection of extent sizes, mostly due to the fact that ESS employs sequential detection and prefetch. For example, even if you picked an extent

size such as 128 KB, which is smaller than the full array width (it would involve accessing only four disks in the array), the ESS sequential prefetch would keep the other disks in the array busy.

Tip: A good starting point would be to set extent size equal to one complete stripe of disks in the LUN. For example, for a LUN striped across a 6+P array, set it to 6*32 K = 196 K.

Prefetch size

The tablespace prefetch size determines the degree to which separate containers can operate in parallel.

Although larger prefetch values might enhance throughput of individual queries, mixed applications would generally operate best with moderate-sized prefetch and extent parameters. You will want to engage as many arrays as possible in your prefetch, to maximize throughput.

It is worthwhile to note that prefetch size is tunable. By this we mean that prefetch size can be altered after the tablespace has been defined and data loaded. This is not true for extent and page size that are set at table space creation time and cannot be altered without re-defining the table space and re-loading the data.

Tip: The prefetch size should be set so that as many arrays as desired can be working on behalf of the prefetch request. For other than the ESS, the general recommendation is to calculate prefetch size to be equal to a multiple of the extent size times the number of containers in your tablespace. For the ESS you may work with a multiple of the extent size times the number of arrays underlying your tablespace.

12.7.4 Selecting the ESS logical disk sizes

The ESS gives you great flexibility when it comes to disk allocation. This is particularly helpful, for example, when you need to attach multiple hosts. However, this flexibility can present a challenge as you plan for future requirements.

The ESS supports a high degree of parallelism and concurrency on a single logical disk. As a result, a single logical disk the size of an entire array achieves the same performance as many smaller logical disks. However, you must consider how logical disk size affects both the host I/O operations and the complexity of your organization's systems administration.

Smaller logical disks provide more granularity, with its associated benefits. But it also increases the number of logical disks seen by the operating system. Select an ESS logical disk size that allows for granularity and growth without proliferating the number of logical disks.

You should also take into account your container size and how the containers will map to AIX logical volumes and ESS logical disks. In the simplest situation, the container, the AIX logical volume, and the ESS logical disk will be the same size.

Tip: Try to strike a reasonable balance between flexibility and manageability for your needs. Our general recommendation is that you create no fewer than two logical disks in an array, and the minimum logical disk size should be around 16 GB. Unless you have an extremely compelling reason, standardize a unique logical disk size throughout the ESS.

Among the advantages and the disadvantages between larger and smaller logical disks sizes, we have the following:

- Advantages of smaller size logical disks:
  - Easier to allocate storage for different applications and hosts.
  - Greater flexibility in performance reporting; for example, ESS Expert reports statistics for logical disks.
- Disadvantages of smaller size logical disks:

  Small logical disk sizes can contribute to proliferation of logical disks, particularly in SAN environments and large configurations. Administration gets complex and confusing.
- Advantages of larger size logical disks:
  - Simplifies understanding of how data maps to arrays.
  - Reduces the number of resources used by the operating system.
  - Storage administration is simpler, thus more efficient and less chances for mistakes.
- Disadvantages of larger size logical disks:

  Less granular storage administration and the resulting less flexibility in storage allocation.

Examples

Let us assume a 6+P array with 36 GB disk drives. Suppose you wanted to allocate disk space on your 16-array ESS as flexibly as possible. You could carve each of the 16 arrays up into 8G logical disks, resulting in 26 logical disks per array (with a little left over). This would yield a total of 16 * 26 = 416 logical disks. Then you could implement 8-way multi-pathing, which in turn would make 8 * 416 = 3328 hdisks visible to the operating system.

Not only would this create an administratively complex situation, but at every reboot the operating system would query each of those 3328 disks. Reboots could take a long time.

Alternatively, you could have created just 16 large logical disks. With multi-pathing and attachment of eight Fibre Channel ports, you would have 8 * 16 = 128 hdisks visible to the operating system. Although this number is large, it is certainly more manageable; and reboots would be much faster. Having overcome that problem, you could then use the operating system logical volume manager to carve this space up into smaller pieces for use.

There are problems with this large logical disk approach as well, however. If the ESS is connected to multiple hosts or it is on a SAN, then disk allocation options are limited when you have so few logical disks. You would have to allocate entire arrays to a specific host; and if you wanted to add additional space, you would have to add it in array-size increments. Furthermore, you could not utilize the full bandwidth of the ESS by striping across all arrays if only some arrays were accessible by a host.

This problem is less severe if you know your needs well enough to say that your ESS will never be connected to more than one host. Nevertheless, in some versions of UNIX an hdisk can be assigned to only one logical volume group. This means that if you want an operating system volume group that spans all arrays of the ESS, you are limited to a single volume group for the entire ESS.

DB2 can use containers from multiple volume groups, so this is not technically a problem for DB2. So, if you want the ability to do disk administration at the volume group level (exports, imports, backups, and so on) then you will not be very pleased with a volume group that is three to eleven terabytes in size.

12.7.5 Multi-pathing

Use ESS multi-pathing along with DB2 striping to ensure balanced use of Fibre Channel or SCSI paths.

Multi-pathing is the hardware and software support that provides multiple avenues of access to your data from the host computer. When using ESS, this means you need to provide at least two Fibre Channel or SCSI connections to each host computer from any component being multi-pathed. It also involves some additional considerations when configuring the ESS host adapters and volumes.

ESS multi-pathing requires the installation of the IBM Subsystem Device Driver (SDD) software on your host computer. SDD is discussed in 5.8, “Subsystem Device Drivers (SDD) - Multipathing” on page 149, and in 6.4, “SDD commands for AIX, HP-UX, and Sun Solaris” on page 176.

There are several benefits from using multi-pathing: Higher availability, higher bandwidth, and easier management. A high availability implementation is one in which your application can still access data using an alternate resource if a component fails. Easier performance management means that the multi-pathing software automatically balances the workload across the paths.

12.7.6 General capacity planning considerations

Some additional considerations when planning the ESS requirements for your database server performance demands are the following.

Throughput

How do you know what will be the throughput requirement? The best way, and not always an option, is to measure the performance of a current production system. If you do not have a running system to measure, consider reviewing benchmark results that have been completed for similar applications.

Workload type

It will be important to know what workload type your application fits in, so the general recommendations can be applied. In 12.1.1, “Understanding your database workload” on page 376, two main workload types were discussed: Online transaction processing and decision support systems.

Disk Magic

Despite that the generic recommendations can be used as such, for a more detailed and accurate estimation that takes into consideration the particularities of your DB2 UDB environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

12.7.7 Monitoring the ESS performance

For a discussion of the ESS performance monitoring tools and recommendations refer to 12.8, “Monitoring tools in a database environment” on page 400.

Also refer to 6.3, “Common UNIX performance monitoring tools” on page 168, to review specific operating system’s monitoring tools.

12.8 Monitoring tools in a database environment

In this section we present the tools that can be used to monitor the ESS performance in a database processing environment. The information presented in this section can be complemented with the information in 9.10, “ESS performance monitoring tools” on page 341, and in 9.11, “ESS performance monitoring for z/OS” on page 342.

12.8.1 RMF monitoring

Four RMF reports can be used to obtain data for analyzing database disk activity. A complete description of the reports discussed in this section can be found in *z/OS V1R4.0 RMF Report Analysis,* SC33-7991.

Cache Subsystem Activity report

There are three Cache Subsystem Activity reports:

- Cache Subsystem Status

  This report gives the amount of cache storage and nonvolatile storage (NVS) installed, as well as the current status of the cache. Caching status must be active.

- Cache Subsystem Overview

  This report, presented in Example 12-1, gives information on the number of I/O requests that were solved in cache (cache hits).

  – Under MISC, the DFW BYPASS field reports NVS overuse, and ASYNC (TRKS) displays the data flow between cache and disks. A high value of ASYNC I/Os with a BYPASS=0 is an indicator of a heavy workload, but the NVS is adequate.

  – The CKD STATISTICS column reports the existence of old channel programs, which can cause performance degradation if they are still used. Some system-related tools can still use those channel programs.

  – Under CACHE MISSES there are four sets of data:

    • NORMAL and SEQUENTIAL lines show, respectively, synchronous and asynchronous I/O misses.

    • The TRACKS and RATE columns display staging activity from disks to cache. In particular, the *sequential prefetch* activity is accounted by number of tracks read and by rate of the read at the end of the SEQUENTIAL line.

  – CFW DATA is positive when DFSORT uses cache sortwork files.

- Cache Subsystem Activity

  This report gives, for all online volumes attached to the subsystem, the specific utilization of the cache. It also consolidates this information at the LCU level. This information is often correlated with the LCU view of the DEVICE report.

To produce these reports specify: REPORTS (CACHE (SUBSYS)).

Example 12-1 Cache Subsystem Activity report

```
0-------------------------------------------------------------------------------------------------------------------------------
                                                CACHE SUBSYSTEM OVERVIEW
 -------------------------------------------------------------------------------------------------------------------------------
OTOTAL I/O   133364   CACHE I/O   133364   CACHE OFFLINE        0
 TOTAL H/R    0.917   CACHE H/R    0.917
-CACHE I/O    ------------READ I/O REQUESTS-------------    ----------------------WRITE I/O REQUESTS----------------------         %
 REQUESTS       COUNT     RATE      HITS     RATE     H/R      COUNT     RATE      FAST     RATE      HITS     RATE     H/R       READ
ONORMAL         28683     31.9     17688     19.7   0.617      66234     73.6     66234     73.6     66234     73.6   1.000       30.2
 SEQUENTIAL     11005     12.2     10995     12.2   0.999      27442     30.5     27441     30.5     27441     30.5   1.000       28.6
 CFW DATA           0      0.0         0      0.0     N/A          0      0.0         0      0.0         0      0.0     N/A        N/A
OTOTAL          39688     44.1     28683     31.9   0.723      93676    104.1     93675    104.1     93675    104.1   1.000       29.8
```

```
-----------------------CACHE MISSES-----------------------
 REQUESTS      READ    RATE   WRITE    RATE  TRACKS    RATE

 NORMAL       10995    12.2       0     0.0   11224    12.5
 SEQUENTIAL      10     0.0       1     0.0     308     0.3
 CFW DATA         0     0.0       0     0.0
0TOTAL        11006    RATE    12.2
----CKD STATISTICS---     ---RECORD CACHING---
0WRITE           1470     READ MISSES        0
 WRITE HITS      1453     WRITE PROM     60886

-----------MISC-----------          ------NON-CACHE I/O-----
                COUNT   RATE                        COUNT   RATE
DFW BYPASS          0    0.0          ICL               0    0.0
CFW BYPASS          0    0.0          BYPASS            0    0.0
DFW INHIBIT         1    0.0          TOTAL             0    0.0
ASYNC (TRKS)    14550   16.2
```

Direct Access Device Activity report

This reports informs response times by volume and LCU. Response time of a specific volume consists of pending time, disconnect time, connect time, and queuing time (see 9.2.1, “Response time components” on page 308, for a detailed discussion). Example 12-2 shows part of this report. The important fields are:

- LCU, number of the Logical Control Unit.
- DEVICE ACTIVITY RATE, rate per second at which start subchannel (SSCH) instructions to the device completed successfully.
- AVG RESP TIME, response time in milliseconds.
- AVG IOSQ TIME, queuing time in IOSQ on the device.
- AVG PEND TIME, pending time.
- AVG DISC TIME, disconnect time.
- AVG CONN TIME, connect time mainly for data transfer. To estimate the path utilization:

  (AVG CONN TIME * DEVICE ACTIVITY RATE/1000)*100

  As an example, an average connect of 4.5 ms with 1200 I/O/sec gives 540 percent, which means a minimum of six paths are required for this workload level. Checking the channel path activity reports for the different LPARS sharing this LCU enables to balancing the activity (540 percent) over the defined paths.
- % DEV UTIL, percentage of device utilization shows the percentage of times RMF has found this device busy. This is a good indicator of demand contention on this volume.

To get standard reporting by LCU, specify: REPORTS (DEVICE (DASD)).

Example 12-2 Direct Access Device Activity report

```
1                              D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y
                                                                                                PAGE    5
            OS/390                    SYSTEM ID AVIA            START 10/24/2002-09.30.00  INTERVAL 000.15.00
            REL. 02.10.00             RPT VERSION 02.10.00      END   10/24/2002-09.45.00  CYCLE 1.000 SECONDS
-
 TOTAL SAMPLES =    900   IODF = 01   CR-DATE: 02/01/2002   CR-TIME: 15.21.55     ACT: POR
-                                        DEVICE   AVG  AVG   AVG  AVG  AVG   AVG  AVG  AVG    %      %      %     AVG    %      %
 STORAGE  DEV  DEVICE   VOLUME PAV  LCU  ACTIVITY RESP IOSQ  DPB  CUB  DB    PEND DISC CONN   DEV    DEV    DEV   NUMBER ANY    MT
  GROUP   NUM  TYPE     SERIAL          RATE     TIME TIME  DLY  DLY  DLY   TIME TIME TIME   CONN   UTIL   RESV  ALLOC  ALLOC  PEND
-         2246 33903    OSASYS   3  0026   0.322   1.0  0.0        0.0  0.0   0.2  0.0  0.8   0.01   0.01   0.0    6.0   100.0   0.0
          2247 33903    OSACF2   3  0026   0.320   3.5  0.0        0.0  0.0   0.2  0.6  2.7   0.03   0.04   0.0    6.0   100.0   0.0
0                        LCU      0026 293.277   2.7  0.4        0.0  0.0   0.2  1.2  0.8   0.11   0.27   0.0   77.4   100.0   0.0
OSGPRDCRI 3200 33903    PRDC02   3  0027  30.997   3.5  0.0        0.0  0.0   0.2  0.3  3.1   3.17   3.43   0.0    5.9   100.0   0.0
 SGTMPPRD 3201 33903    TMPP01   3  0027   0.212   7.1  0.0        0.0  0.0   0.2  0.0  6.9   0.05   0.05   0.0    3.0   100.0   0.0
          3202 33903    DBC241   4  0027  10.961   2.6  0.0        0.0  0.0   0.2  1.5  0.9   0.24   0.66   0.0    1.0   100.0   0.0
          3203 33903    DBC242   4  0027  12.873   2.5  0.0        0.0  0.0   0.2  1.4  0.9   0.28   0.73   0.0    1.0   100.0   0.0
          3204 33903    DBC243   4  0027  12.326   2.6  0.0        0.0  0.0   0.2  1.5  0.9   0.27   0.73   0.0    1.0   100.0   0.0
```

I/O Queuing Activity report

The I/O Queuing Activity report is used to analyze the pathing behavior. To get this report, specify REPORTS (IOQ).

Channel Path Activity report

The Channel Path Activity report identifies performance contentions associated with the channel paths. To produce this report, specify REPORTS (CHAN).

Review the following fields:

- CHANNEL ID is the hexadecimal number of the channel path identifier (CHPID).
- PATH SHR: A value of Y indicates that the channel link (physical channel) is shared between one or more LPARs.
- PARTITION UTILIZATION (%) is the percentage of physical channel path utilization by the LPAR.
- TOTAL UTILIZATION (%) is the percentage of physical channel path utilization that all LPARS of this CPC use. This is the aggregate view of channel utilization.

12.8.2 ESS Expert

The ESS Expert component of the IBM TotalStorage Expert gathers performance information from the ESS and stores it in a relational database. The information is summarized in reports that can include:

- Number of I/O requests for the entire storage server in total and separated among various physical disk groupings
- Read and write cache hit ratio
- Cache to/from disk operations (stage/destage)
- Disk read and write response time
- Disk utilization

The ESS Expert is described in further detail in 4.4, “IBM TotalStorage Expert” on page 104.

13

ESS Copy Services

The IBM TotalStorage Enterprise Storage Server provides a powerful set of copy functions suitable for the uninterrupted operations and the business continuance requirements that organizations currently face. These copy functions are also the way for installations to do their remote data migrations and off-site backups.

In this chapter we discuss the performance and capacity planning considerations for:

- FlashCopy
- Peer-to-Peer Remote Copy (PPRC)
- Extended Remote Copy (XRC)

All these ESS Copy Services functions are optional features of the ESS Model 800.

The information discussed in this chapter can be complemented with the following IBM publications and publications:

- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *Implementing ESS Copy Services on S/390,* SG24-5680
- *Implementing ESS Copy Services on UNIX and Windows NT/2000,* SG24-5757
- *IBM TotalStorage Enterprise Sorage Server PPRC Extended Distance,* SG24-6568

13.1 FlashCopy

Taking backups of user data traditionally has taken a considerable amount of time. Usually backups have been taken outside prime shift because of their duration and the consequent impact to normal operations. Databases must be closed to create consistency, and online systems are normally shut down for the duration of the backups.

With the IBM TotalStorage Enterprise Storage Server, the backup time can be reduced to a minimal amount of time when the FlashCopy function is used. FlashCopy creates an instant point-in-time copy of the logical volumes, with minimal impact to the applications.

With FlashCopy, you get an instant time zero (T0) copy (target) of an ESS volume (source) as you invoke the command. In a minimal amount of time, the source and target volumes are both immediately available for application use. The actual data copy operation is then (optionally) performed in the background while the source and target volumes are receiving reads and writes from the applications. Host applications are not aware of the background copy processing taking place.

For all server platforms, FlashCopy can be controlled using a Web browser by means of the ESS Copy Services Web user interface (WUI) of the ESS. Under z/OS, FlashCopy can also be invoked using DFSMSdss, or TSO commands. For selected open systems servers the ESS also provides the ESS Copy Services command-line interface (CLI) for invocation and management of FlashCopy tasks. FlashCopy support is also available for the zSeries servers running the Linux operating system.

FlashCopy Version 1

The original implementation of FlashCopy in the ESS, namely FlashCopy Version 1, had some considerations that needed to be observed:

- FlashCopy Version 1 worked at the volume level only.
- The source and target volumes needed to be in the same ESS logical subsystem (LSS).
- A source and a target volume could only be involved in one FlashCopy relationship at a time.

FlashCopy Version 1 is available as an optional feature of the ESS. For a detailed discussion of FlashCopy Version 1 functions you can refer to *Implementing ESS Copy Services on UNIX and Windows NT/2000,* SG24-5757.

FlashCopy Version 2

With FlashCopy Version 2 the previous considerations have changed, and also new functionality has been added. FlashCopy Version 2 provides all the functions of FlashCopy Version 1, plus the following enhancements:

- In addition to volume copy, data set FlashCopy is also possible for z/OS users.
- Multiple FlashCopy relationships can be present on a volume and extent level.
- When doing extent FlashCopy, target tracks need not to be in the same location as source tracks. Target tracks can even be on the same volume as the source tracks.
- Source and target volumes can be on the same or different LSSs.
- Incremental copies of established FlashCopy relationships can be done.
- Consistency groups can be enabled when establishing FlashCopy relationships.

FlashCopy Version 2 is available as an optional feature of the ESS. Current FlashCopy Version 1 users can upgrade to Version 2, provided they install ESS Licensed Internal Code (LIC) 2.2.0 or later.

For a detailed discussion of FlashCopy Version 2 functions you can refer to *Implementing ESS Copy Services on S/390,* SG24-5680, and *Implementing ESS Copy Services on UNIX and Windows NT/2000,* SG24-5757.

13.1.1 Operation

FlashCopy provides a point-in-time copy of an ESS logical volume. The point-in-time copy function gives you an immediate copy of the original data as it looked at the specific point in time when the FlashCopy was requested. This is known as the T0 (time-zero) copy.

Note: For users that have FlashCopy Version 2 installed, this discussion also applies to the general understanding of how data set FlashCopy operates.

As Figure 13-1 on page 406 illustrates, when FlashCopy is invoked, a relationship (or session) is established between the source and target volumes of the FlashCopy pair. This includes creation of necessary bitmaps and metadata information needed to control the copy operation. This process takes only a very short moment to complete. After this FlashCopy command initialization, the FlashCopy relationship is established, then control returns to the operating system and the T0 copy of the source volume is available for use.

As soon as the pair has been established, both the source and the target volumes are immediately available for reads and writes from the applications. At the same time a background task starts copying the tracks from the source to the target volume. Optionally, you can suppress this background copy task. This is efficient, for example, if you are doing a temporary copy just to take a backup to tape.

The FlashCopy relationship ends when the background copy task completes. If the FlashCopy was requested with the no-background copy option, then the relationship must be explicitly ended by command.

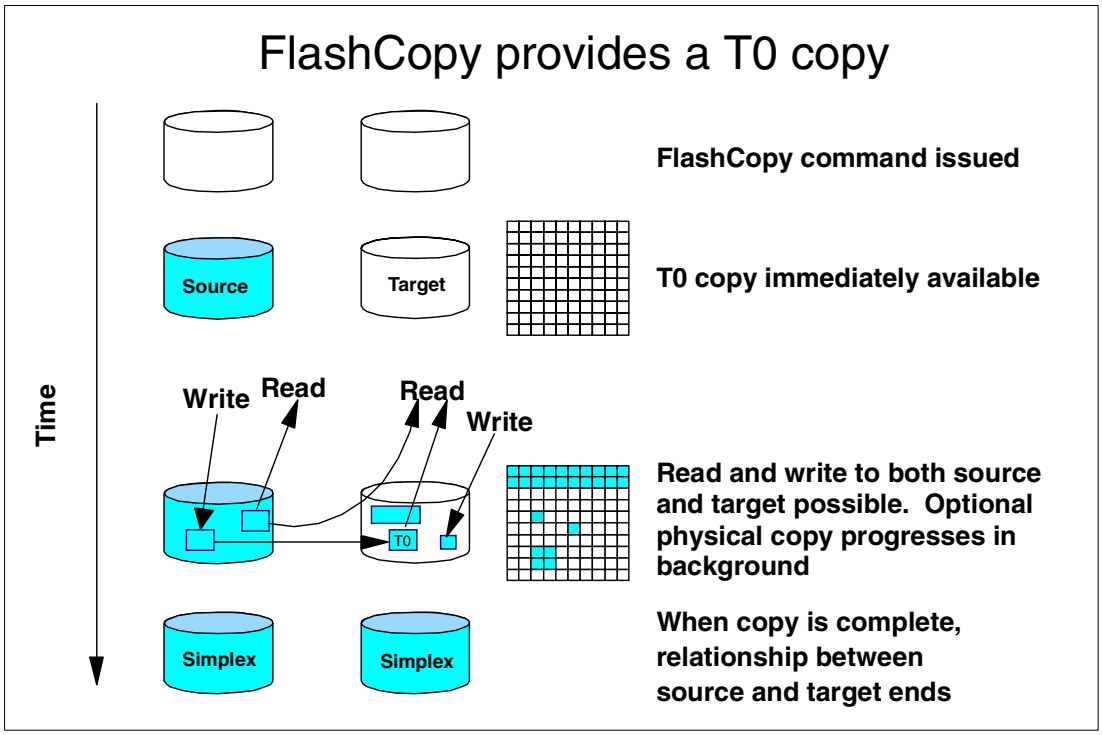


Figure 13-1 FlashCopy - Starting and ending the relationship

At the time when FlashCopy is started the (optional) background copy task starts copying data from the source to the target. While the background copy is in process, the target volume data can be read or written as though all the data had been physically copied.

The ESS keeps track of which data has been copied from source to target. As Figure 13-2 on page 407 shows, if an application wants to read some data from the target that has not yet been copied to the target, the data is read from the source; otherwise, the read can be satisfied from the target volume.

As Figure 13-2 on page 407 shows, when an application wants to modify data on either the source or target volume and that data has yet not been copied, a copy-on-demand operation is invoked to first copy the track to the target volume. The following reads to this track on the target volume will be satisfied from the target volume. When all tracks have been copied to the target volume, and the FlashCopy relationship will automatically end.

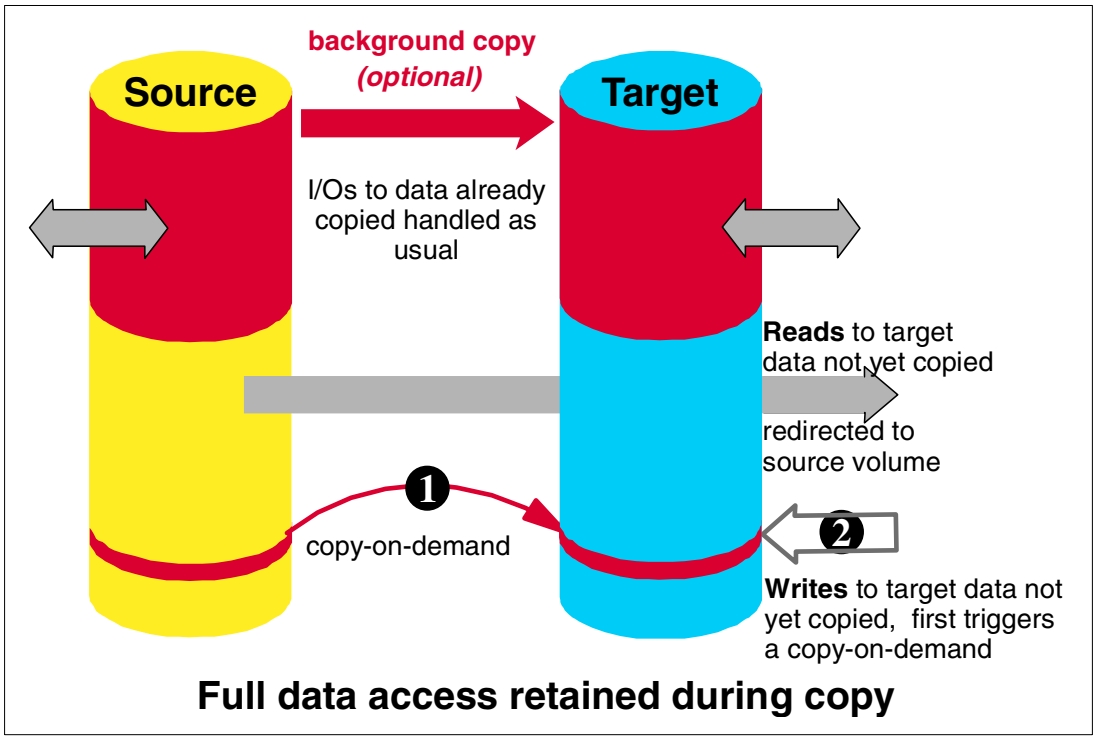


Figure 13-2 I/O processing with FlashCopy

Applications do not have to be stopped for FlashCopy. However, you have to manage the data consistency across different volumes. This will normally require that you quiesce applications for the short duration to establish the FlashCopy session. One example of things to consider is that only data on the physical disk is copied, while data in the buffers in the application server will not be copied. So either applications have to be frozen consistently, or you have to invoke built-in functions that maintain the application consistency.

13.1.2 Performance considerations

There are a number of considerations on FlashCopy performance, which we discuss in this section:

- Time necessary for FlashCopy to complete
- Application performance
- Background copy
- Additional load on the ESS

The FlashCopy guidelines and recommendations that we discuss in this chapter are generic and as such can be used. For a more detailed and accurate approach that takes into consideration the particularities of your environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

Time taken for FlashCopy to complete

FlashCopy Version 1 operates on an entire logical volume. The process of establishing the copy relationship and making the copy available takes typically a few seconds. The size of the logical volume makes very little difference to this time. With FlashCopy Version 2 the establish time for the FlashCopy relationship has been reduced even further.

Figure 13-3 illustrates an example of FlashCopy Version 1 establish times and withdraw times for open systems LUNs with the no-background copy option selected on an ESS Model 800. When requesting the background copy, the establish times can be slightly higher. The measurements compare the establish time for 1 and for 224 open systems LUNs for both the Web-user interface and the command-line interface invocation.

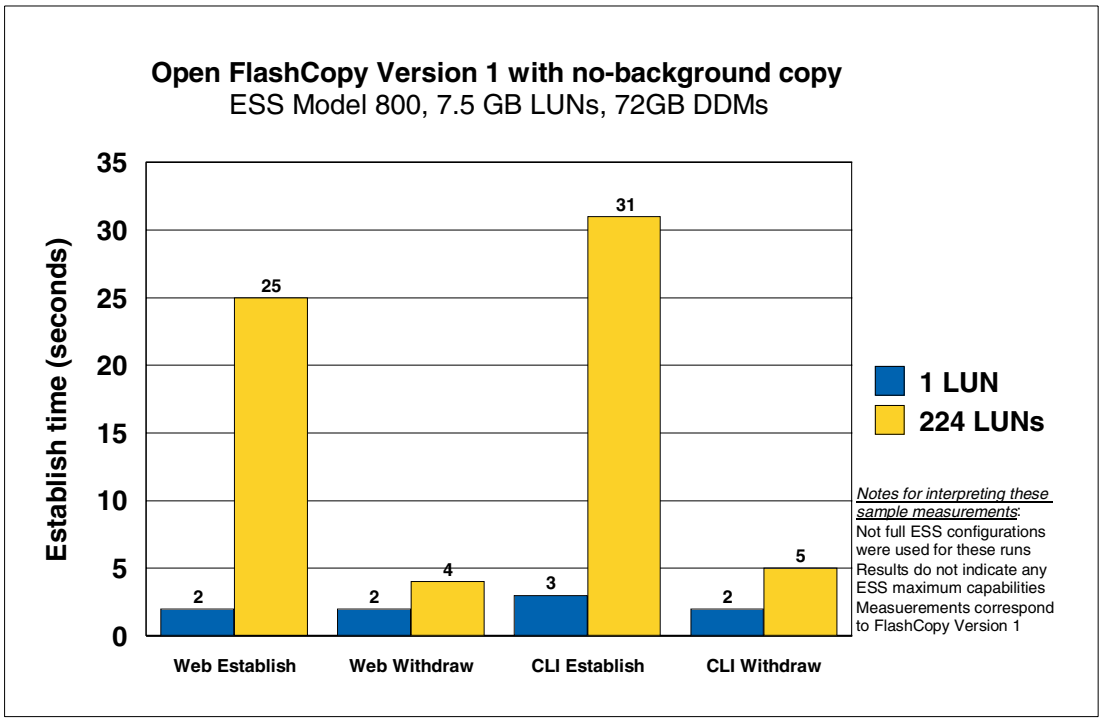


Figure 13-3 Open FlashCopy with NOCOPY establish and withdraw times

Figure 13-4 on page 409 illustrates FlashCopy establish times when requested upon different number of volumes with the background copy option, and invoked using TSO commands. The establish times with no-background copy would be about 10 percent faster.

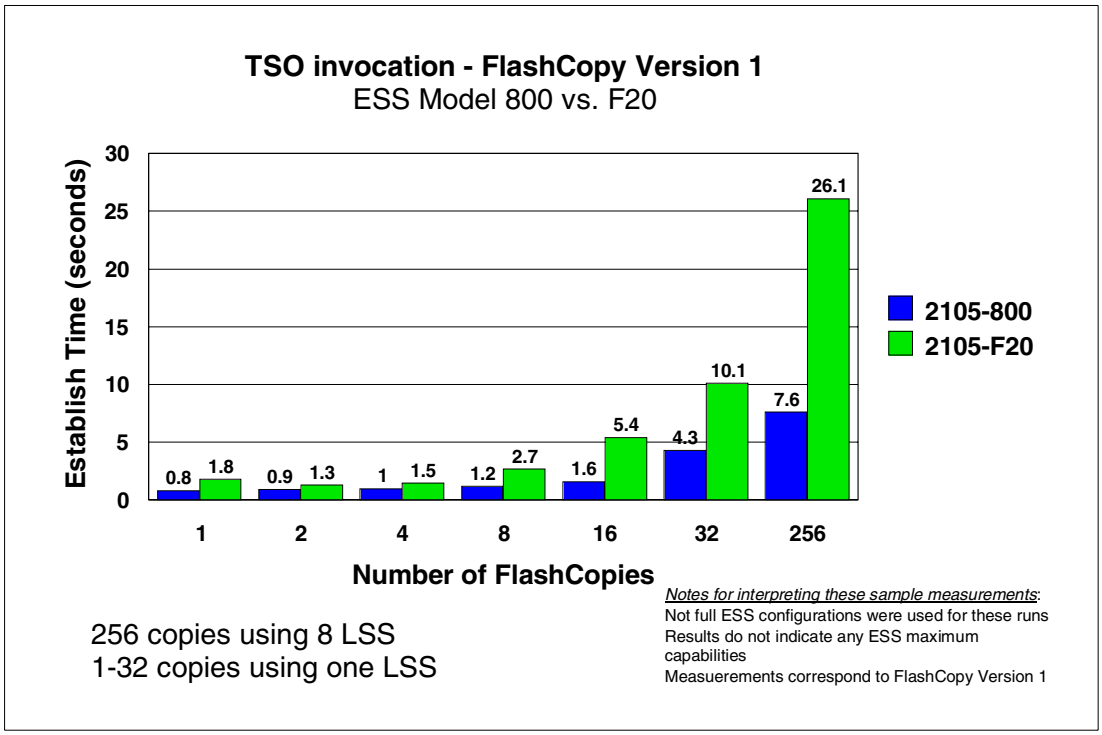


Figure 13-4 FlashCopy establish time - TSO invocation and different number of volumes

You can see that with the ESS Model 800 the establish times significantly improve as compared to the model F20, and this is especially notorious when a large quantity of volumes is established (as the 256 column shows).

Note: The examples shown in Figure 13-3 and Figure 13-4 are for FlashCopy Version 1. With FlashCopy Version 2 there have been significant improvements in the FlashCopy establish times. With FlashCopy Version 2 you can see reductions of up to 10 times in the establish times as compared to FlashCopy Version 1.

Application performance

With traditional backup methods, applications had to be stopped for the duration of the backup, which could take a long time. FlashCopy allows backups to be created in a minimal amount of time, just the very short duration of the FlashCopy initialization (establish) time. Then the optional background physical copy proceeds in parallel with the application doing its normal read and write activity, unaware of FlashCopy activity.

FlashCopy is a hardware implementation of the ESS, and uses a very small amount of cache to hold bitmaps and metadata. FlashCopy processing is indiscernible to the applications, as the (optional) copy is managed by the ESS with a low priority and without compromising resources needed for application's I/O processing.

Background copy

Once FlashCopy has done the initialization, then the optional background copy is started. In the case of FlashCopy Version 1, because the source and target volumes must be in the same LSS, the data is only moved across the same ESS device adapter DA, which provides plenty of bandwidth.

Figure 13-5 illustrates the data rates when a FlashCopy Version 1 background data copy for open systems LUNs is requested. The data rates are for one, two, and four volume copies within the same rank and also between two ranks. You can see that there is hardly any difference in the data rates with just one pair, but as the number of concurrent pairs increases then using two ranks starts to provide significantly more throughput.

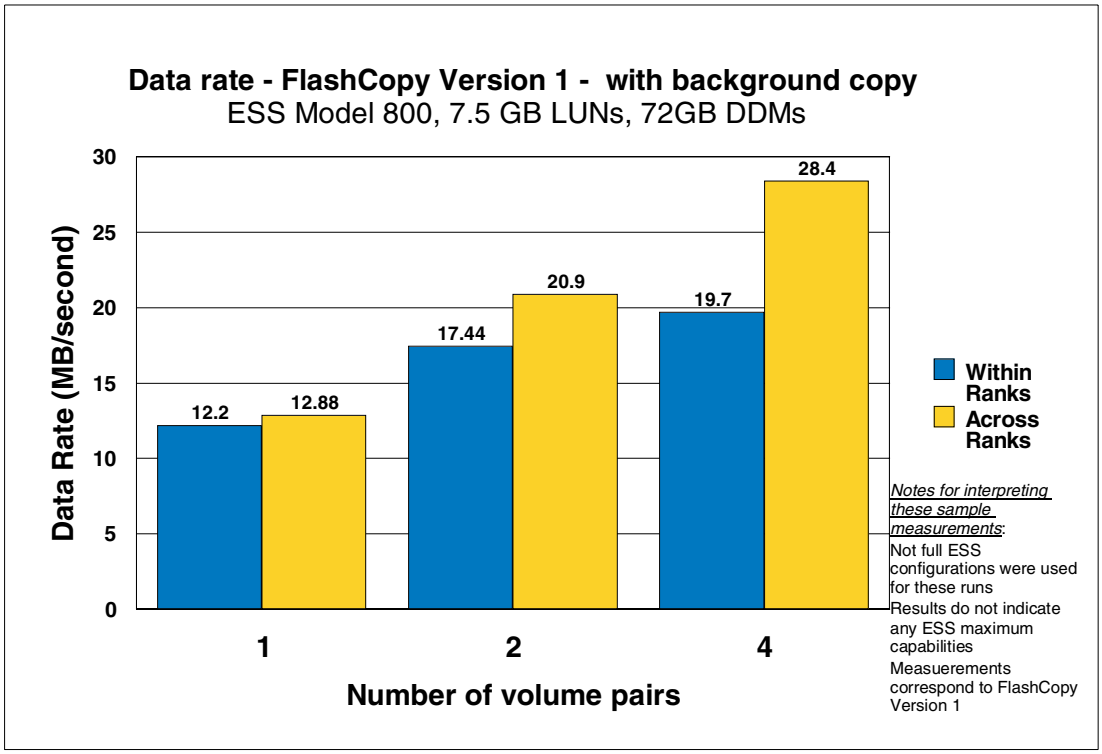


Figure 13-5 Open FlashCopy background copy rate

Additional load on the storage control

The process of taking the FlashCopy has minimal impact on the throughput of the ESS; however, you may be putting some additional load on the ESS because of operational changes. For example, if you use FlashCopy to take an instant backup of your data, and then dump the target volumes to tape, you need to plan for sufficient capacity to handle the additional dump processing.

Without FlashCopy you may have previously dumped data to tape, and dedicated as many storage resources as possible to the dump jobs in order for them to complete as fast as possible. With FlashCopy, you are able to run dump jobs in parallel with the applications; therefore, you want to ensure that the storage resources are adequately balanced to provide enough bandwidth for the dump jobs running concurrently with the applications I/Os.

You can also offload the FlashCopy job to a secondary server using PPRC. See Figure 13-6 on page 411 for and example of a split mirror backup using PPRC to accomplish an offload of the data from a primary ESS to a secondary ESS.

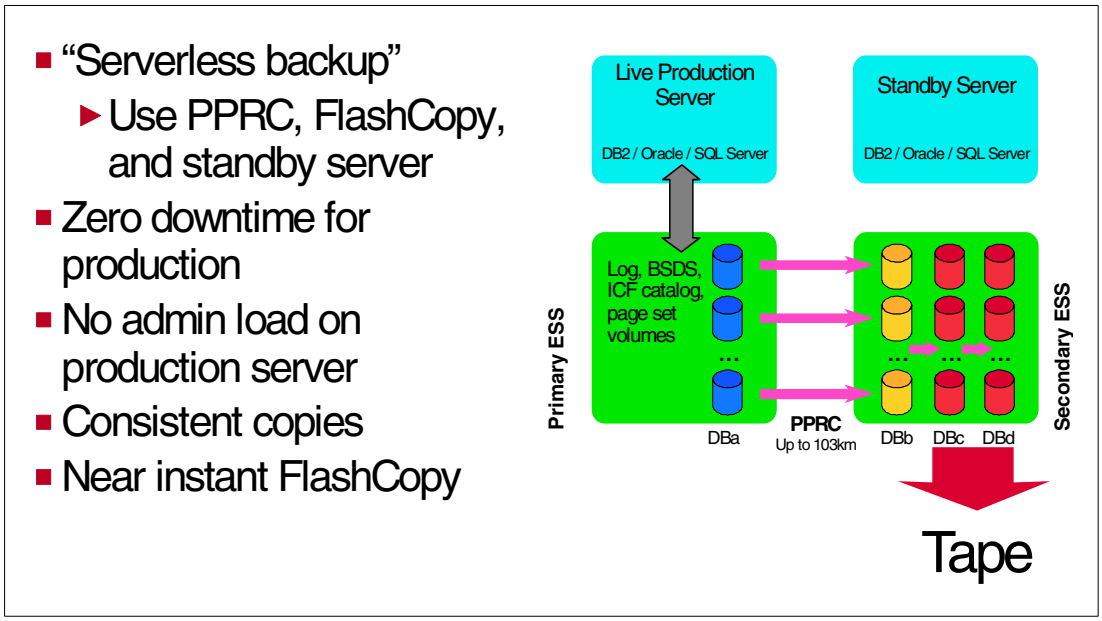


Figure 13-6 SAP R/3 split mirror backup

13.1.3 Planning for FlashCopy

When planning to use FlashCopy, certain considerations apply. Some of these considerations are valid if you are using FlashCopy Version 1 and are no more valid for FlashCopy Version 2. Additionally some new considerations apply for FlashCopy Version 2, as some new functionality has been added.

- When planning to use FlashCopy Version 1, you must plan for sufficient storage capacity within each LSS for the target volumes also. This consideration does not apply to FlashCopy Version 2, as the source and target can be on different LSSs.
- Target volumes need to not be dedicated to any particular source, but a target can only be the target for only one source at a time.
- With FlashCopy Version 1 a FlashCopy source volume can have only one target volume; with FlashCopy Version 2 a source can have up to 12 targets. Therefore, the number of target volumes you must plan for depends on the number of concurrent FlashCopy sessions you will be running.
- FlashCopy Version 1 works on a volume basis only, so in this case source and target refer to volumes. z/OS users with FlashCopy Version 2 can request data set copies also, so in this case source and target can refer either to extents of tracks or full volumes (the full extent that makes up a volume).

RAID ranks

The IBM TotalStorage Enterprise Storage Server Model 800 can have its ranks configured as either RAID-5 or RAID-10. You can FlashCopy between logical volumes with different RAID array types, as Figure 13-7 on page 412 illustrates.

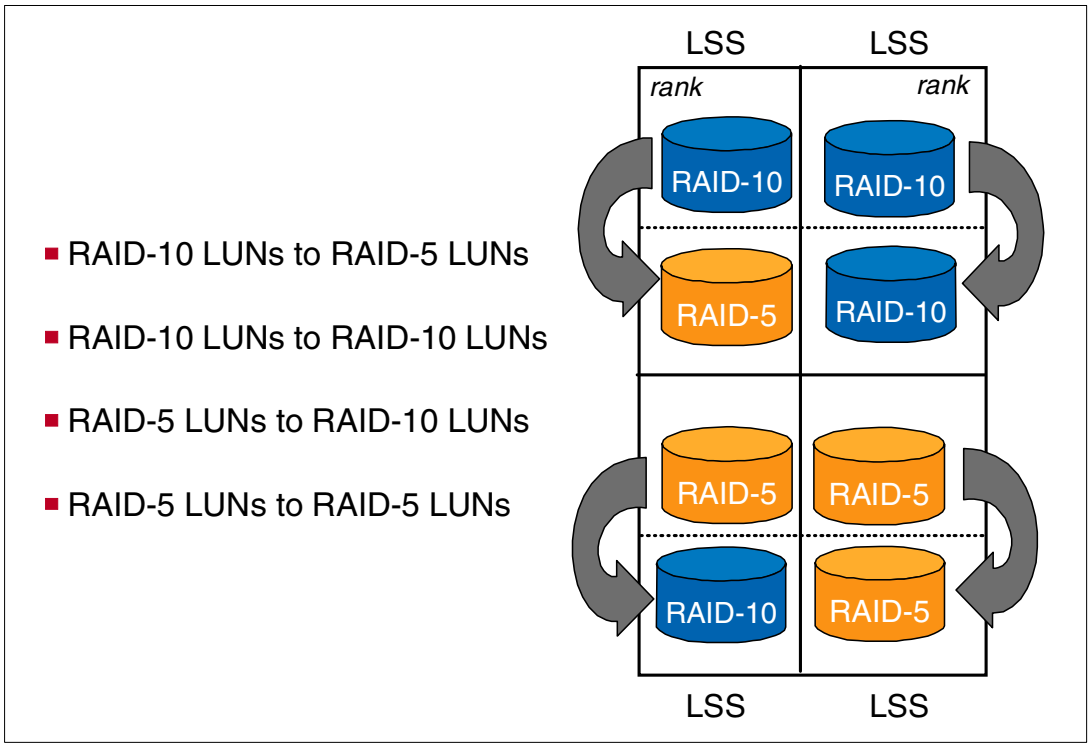


Figure 13-7 RAID FlashCopy options

Different sources to the same set of targets

With FlashCopy Version 1, source and target volumes need to be in the same LSS. In some cases this needs some planning considerations, for example, if several versions of the backup need to be taken or more than one database needs to be backed up during the shift, that would exhaust the available volumes in the LSS. One possible solution is to reuse the same set of target volumes for different source volumes backups. How do we accomplish this?

Say, for example, that you are flashing four RAID-10 volumes to another four RAID-10 volumes in the same or different arrays in the same LSS. For this example we will discuss flashing between volumes in the same array and assigning them to server 2 (see Figure 13-8 on page 413) to offload the dump I/O activity.

You can see in the example in Figure 13-8 on page 413 that we are copying the source volumes and then backing up (dumping to tape) the target volumes with a TSM server at a scheduled time. This allows an offline backup of the production database without any downtime to it. Once the backup of the target volumes is complete then the FlashCopy volume pairs can be withdrawn in preparation for the next FlashCopy tasks.

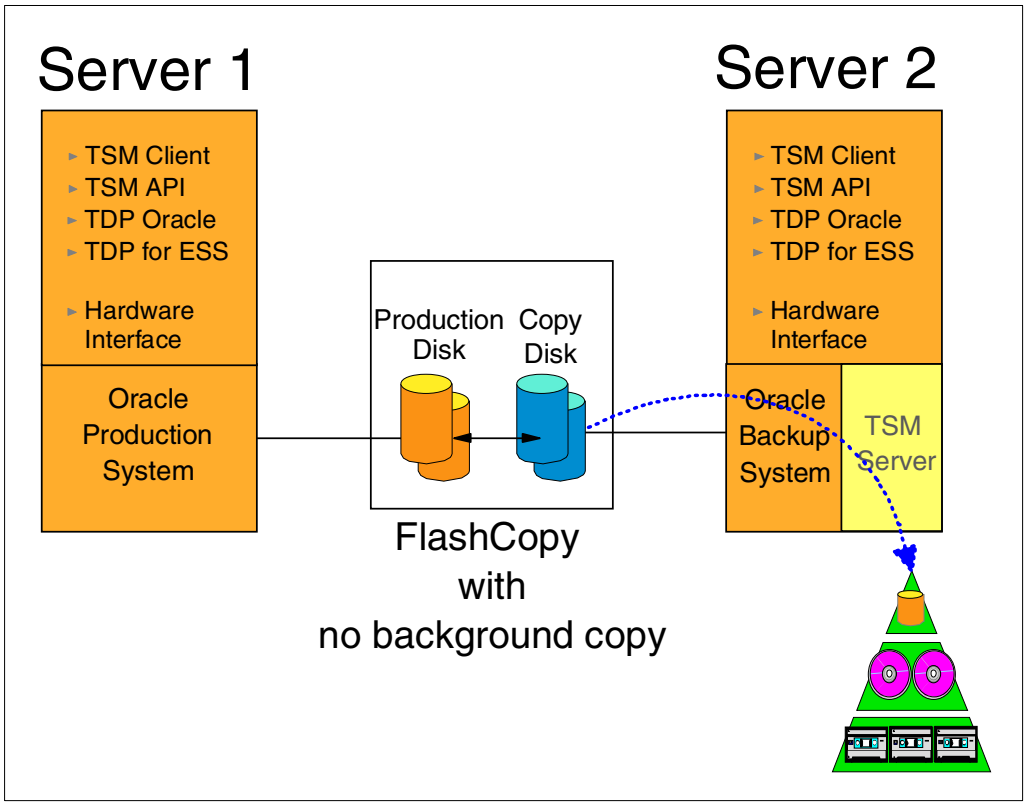


Figure 13-8 Offloading the copy disks to tape

In Figure 13-9 on page 414 you can see that twenty LUNs are allocated in the same array. Eight are 16 GB in size, four are 8 GB in size, four are 4 GB in size, and the last four are 2 GB is size. In our example, database A resides on the first four 16 GB volumes, database B resides on the 8 GB volumes, database C resides on the 4 GB volumes, and database D resides on the 2 GB volumes. If we are going to designate our target volumes as the second set of 16 GB volumes and make them our shared FlashCopy pool, we could FlashCopy database A at 8:00 p.m., then back it up. At 12:00 a.m. we could FlashCopy database B to the same set of target volumes, and so forth, as illustrated in Figure 13-9 on page 414. Remember that as we are requesting no-background copy, then a withdraw will be needed after each database backup is complete.

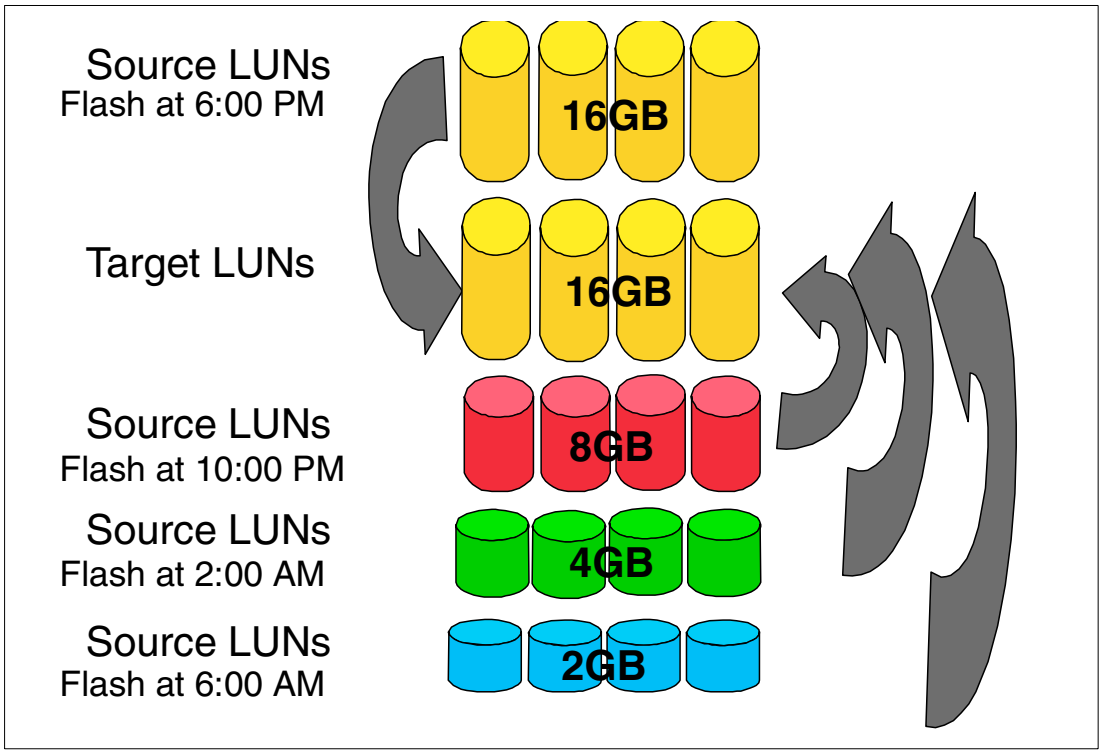


Figure 13-9 Flashing different sources to same targets at different times

As you can see, we can FlashCopy all sixteen source volumes in the same array to the same four 16 GB target volumes if we do it one at a time, at different times of the day. As you can see from this example, you can save space doing some careful planning beforehand.

Dedicated source to dedicated targets

If you have enough volumes, you can simplify your tasks and copy the databases following the conventional method of having a dedicated target volume for every source volume. Also, if you are planning on using FlashCopy for database refreshes on the same or different target hosts, then you need to dedicate a target volume for each source volume to be copied. In this case you would use the background copy option when requesting the FlashCopy.

13.2 Peer-to-Peer Remote Copy

Peer-to-Peer Remote Copy (PPRC) is a hardware solution for mirroring logical volumes from an ESS at a local site (primary volumes) where the application runs, onto the volumes of an ESS at a remote site (secondary volumes). PPRC is a remote copy solution for the open systems servers and for the zSeries servers.

Two modes of PPRC are available with the IBM TotalStorage Enterprise Storage Server:

- *PPRC synchronous mode*, for synchronous real-time mirroring between ESSs located up to 103 km apart. In this mode of operation the ESS synchronously mirrors the updates done to the primary volumes.
- *PPRC Extended Distance mode* (PPRC-XD), for non-synchronous data copy over continental distances (using channel extenders) with excellent application performance.

PPRC can be managed using a Web browser to interface with the ESS Copy Services Web user interface on all supported platforms. PPRC can also be operated using commands for

selected open systems servers that are supported by the ESS Copy Services command-line interface (CLI). In addition, for the z/OS users, the TSO commands can be used to control PPRC; and for the z/OS, z/VM, and VSE/ESA™ users the ICKDSF utility can be used.

PPRC is a hardware solution, thus it is application independent. The PPRC feature must be installed on both the local and the remote ESS. Because the copy function occurs at the ESS level, the application does not need to know of its existence. PPRC requires no application changes.

Linux

Support for the ESS PPRC functions is now available for zSeries servers running the Linux operating system.

13.2.1 PPRC operation

In synchronous PPRC mode an I/O does not complete until it is acknowledged from the remote ESS. Updates to primary PPRC volumes go first into the cache and NVS of the primary ESS. The updates are then sent to the secondary ESS over the ESCON links. When the data is in cache and NVS is at the secondary site, the receipt is acknowledged by the secondary ESS, and then the primary ESS signals the application that the I/O is complete. Destage from cache to the back-end disk drives on both the primary and the secondary ESS is performed asynchronously. This process is illustrated in Figure 13-10.

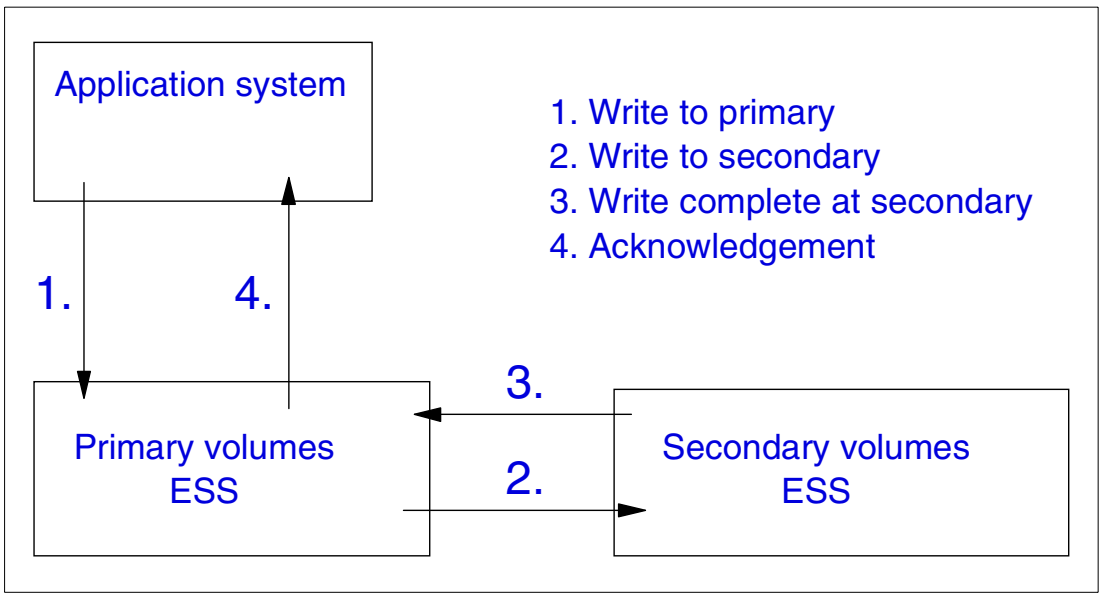


Figure 13-10 PPRC - Synchronous operation

The local site ESS is also called primary ESS if it contains at least one PPRC primary volume, while the remote ESS is called secondary ESS if it contains at least one PPRC secondary volume (primary and secondary volumes are also referred to as source and target volumes, respectively). An ESS can act as primary and secondary at the same time if it has PPRC source and target volumes. This mode of operation is called bi-directional PPRC.

PPRC guarantees that the secondary copy is up-to-date by ensuring that the primary volume update will be successfully completed only when the primary ESS receives acknowledgment that the secondary copy has been successfully updated.

This synchronous technique of PPRC also ensures that application-dependent writes will be applied on the secondary volumes in the same sequence as in the primary volumes, thus providing application consistency at every moment.

13.2.2 PPRC implementation on the ESS

As with other PPRC implementations, you can establish PPRC pairs only between storage control units of the same type, which means that primary and secondary must both be ESSs with the optional PPRC function enabled. PPRC between different models of the ESS is supported.

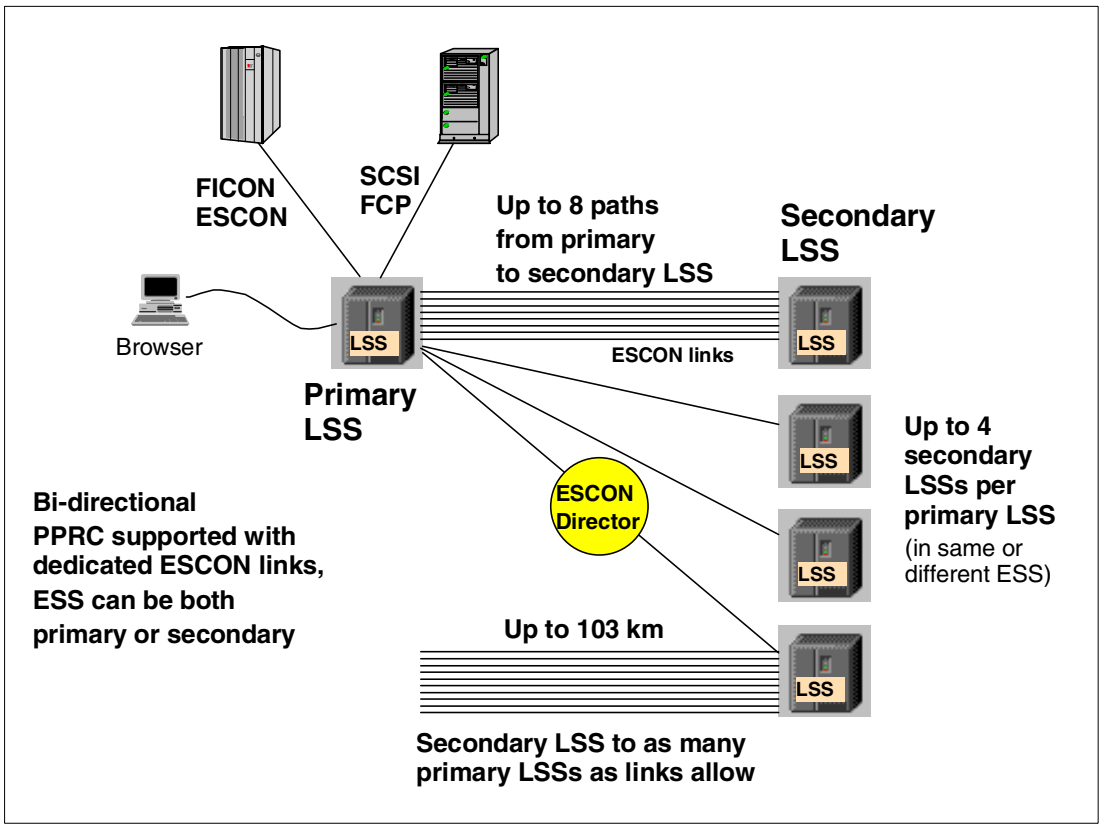


Figure 13-11 PPRC configuration options with ESS

ESCON links

The primary and secondary ESS subsystems must be connected by ESCON links, which PPRC uses to transfer the mirrored data. The number of links is limited by the number of available ESCON ports in each ESS.

As Figure 13-12 on page 417 shows, PPRC links are unidirectional. This means that a physical ESCON link can be used to transmit data from the primary ESS to the secondary ESS. If you want to set up a bi-directional PPRC configuration with source and target volumes on each ESS, you need ESCON PPRC links in each direction.

The number of links depends on the write activity to the primary volumes. The direction of a link is fixed when the first logical path is defined on it. The direction can be changed only by first un-defining all the existing paths from it.

Primary PPRC ESCON ports are dedicated for PPRC use and cannot be used for S/390 host connections. The ESCON ports on the secondary ESS do not need to be dedicated for PPRC

use. They can also be used for host communications when the ports are connected to an ESCON director.

There are two ESCON ports on an ESCON adapter card. Each port on the adapter can be used independently for PPRC and/or host connections.

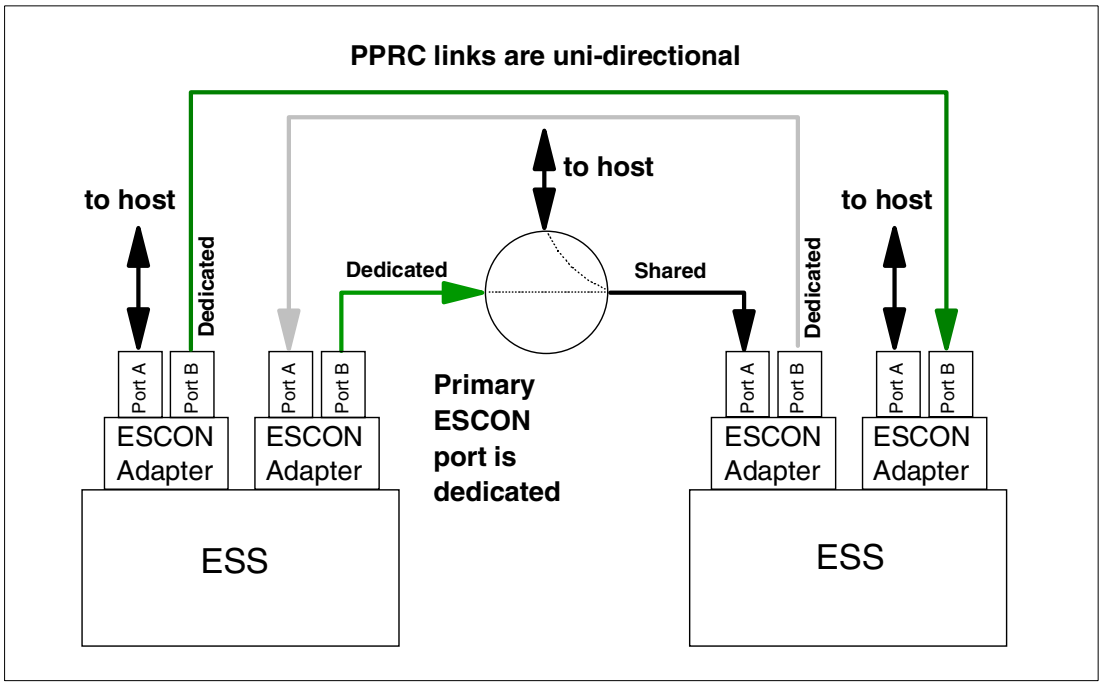


Figure 13-12 PPRC links

ESCON links between the ESSs are required also when you use PPRC to mirror open systems logical volumes. To connect beyond the native ESCON link range of 3 km, ESCON Directors, channel extenders over Wide Area Network (WAN) lines, or Dense Wave Division Multiplexors (DWDM) over dark fibers may be used.

PPRC logical paths

Before PPRC volume pairs can be defined, logical paths must be established between the ESS logical subsystems (LSS). You establish logical paths between control unit images of the same type over the physical ESCON links.

The PPRC logical paths are established between the LSSs. Up to eight logical paths can be established for an LSS pair. A primary LSS can be connected to up to four secondary LSSs, which may be on different ESSs. A secondary LSS can be connected to as many primary LSSs as ESCON links are available (see Figure 13-11 on page 416).

Logical paths between different LSS pairs can share the same physical ESCON links. While the number of logical paths is limited to eight, you can have more physical ESCON links between two ESSs. This allows you to spread the load of different LSS pairs over more physical links.

PPRC volume pairs

PPRC mirroring can be established between logical volumes in LSSs for which logical paths have been defined. The secondary volume has to be of at least the same size as the primary volume. In S/390 environments, the primary and secondary volumes have to be of the same device geometry. PPRC between 3380 and 3390 format devices is not supported.

A volume can be involved in only one PPRC relationship at a time. Both the primary and secondary PPRC volumes can become a FlashCopy source volume, but not a FlashCopy target. On the other hand, FlashCopy source and target volumes can become both PPRC primary and secondary volumes.

13.2.3 PPRC Extended Distance (PPRC-XD)

Peer-to-Peer Remote Copy Extended Distance (PPRC-XD) is a non-synchronous copy option for both open systems and zSeries servers. This means that host updates to the source volumes do not have to wait for the update to be confirmed by the secondary ESS. As a result, applications' I/O response times do not suffer as in synchronous mirroring.

PPRC-XD brings new flexibility to the IBM TotalStorage Enterprise Storage Server and PPRC. PPRC-XD can operate at very long distances, even continental distances, well beyond the 103 km supported for PPRC synchronous transmissions—with minimal impact on the applications. Distance is limited only by the network and channel extenders' technology capabilities.

The non-synchronous operation of PPRC-XD, together with its powerful throughput (copying sets of track updates only) and the supported channel extenders improvements, make PPRC-XD an excellent copy solution at very long distances and with minimal application performance impact. PPRC Extended Distance is an excellent solution for remote copy solutions involving:

- Data copy
- Data migration
- Off-site backup
- Transmission of database logs
- Application recovery implementations based on periodic point-in-time copies

PPRC-XD comes as part of the optional PPRC feature.

PPRC-XD operation

As soon as the updates to the PPRC-XD primary volumes are in the cache and NVS of the primary ESS, a Channel End and Device End is returned to the host, as would be the case if the volumes were not in a PPRC relationship (see Figure 13-13 on page 419).

The primary ESS keeps track of updated tracks and periodically sends them in batches to the secondary ESS. This is a throughput-oriented, very efficient method of non-synchronous mirroring.

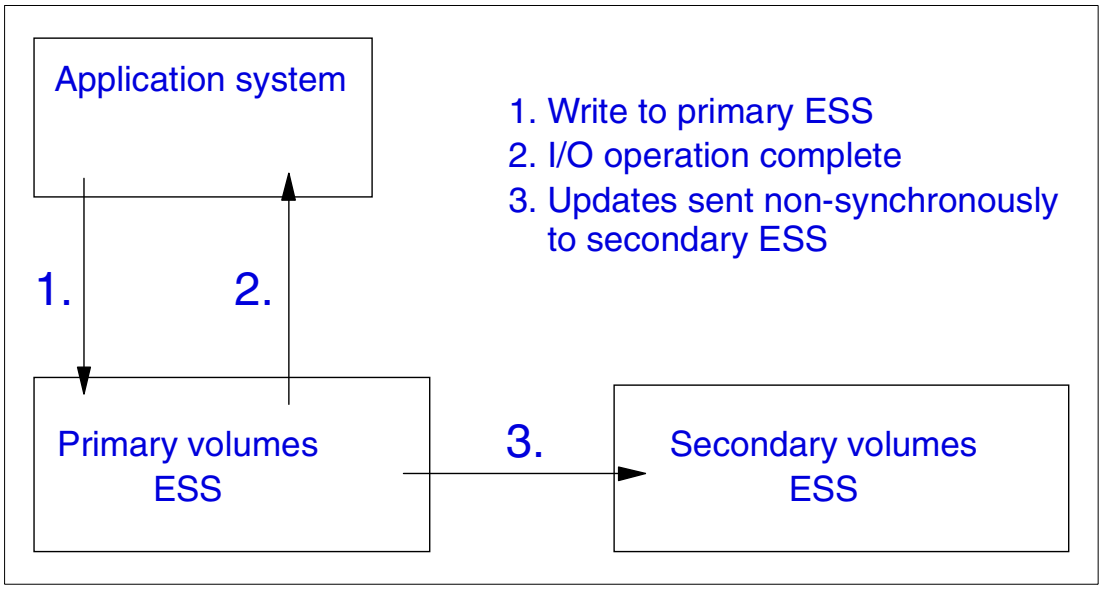


Figure 13-13 PPRC-XD - Non-synchronous operation

The same configuration rules regarding ESCON links, logical paths, and volume pairs apply to PPRC-XD as synchronous PPRC.

Data consistency

While in PPRC-XD state, the volume pairs remain in duplex-pending XD state. When the application is doing writes on the primary volumes, the updates to secondary volumes are done non-synchronously. There is no certainty that the sequence of the updates on the secondary volumes will be in the same order as on the primary volumes. This means that secondary volumes are keeping a *fuzzy* copy of the data. Application-dependent writes are not assured to be applied in the same sequence as written on the primary.

Because of the non-synchronous characteristics of PPRC-XD, at any time there will be a certain amount of application-updated data that will not be reflected at the secondary volumes. This data corresponds to the tracks that were updated since the last batch of tracks was sent to the secondary ESS.

13.2.4 Asynchronous cascading PPRC

PPRC Version 2 incorporates a new powerful function: Asynchronous cascading PPRC. Asynchronous cascading PPRC provides a long-distance remote copy solution for zSeries and open systems environments by allowing PPRC secondary volumes (involved in a PPRC synchronous relationship) to also simultaneously serve as a PPRC primary volume in a PPRC Extended Distance (PPRC-XD) relationship to a remote site. This new capability enables the creation of three-site or two-site asynchronous cascading PPRC configurations.

Figure 13-14 on page 420 illustrates a typical implementation of a PPRC Version 2 cascading implementation. This is a replication solution that will typically copy data synchronously (PPRC-SYNC) from the local site volumes to the intermediate site volumes, and from there the data is copied non-synchronously (PPRC-XD) to the remote site volumes at any distance. Asynchronous cascading PPRC is the basic enabler of long-distance disaster recovery solutions.

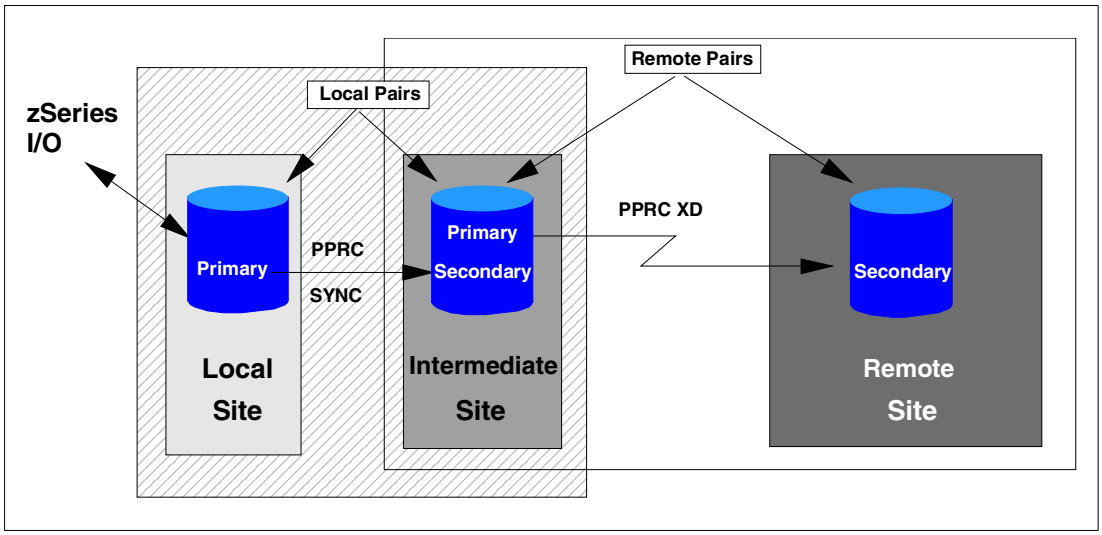


Figure 13-14 PPRC Version 2 - Asynchronous cascading

Operational control for PPRC Version 2 is provided by the ESS Specialist and Copy Services CLI. For zSeries and S/390 environments, operational control for zSeries and S/390 volumes involved in PPRC configurations can be managed by IBM's Geographically Dispersed Parallel Sysplex (GDPS®) offering. GDPS, an industry-leading e-business availability solution available through IBM Global Services, is a multi-site solution designed to provide the capability to manage remote copy configurations and storage, automate Parallel Sysplex operational tasks, and perform failure recovery—from a single point of control.

PPRC Version 2 is available as an optional feature of the ESS. Current PPRC Version 1 users can upgrade to Version 2, provided they install ESS Licensed Internal Code (LIC) 2.2.0 or later.

Performance considerations

When evaluating the performance that can be expected from a cascaded PPRC implementation, both the synchronous (local to intermediate) and the non-synchronous (intermediate to remote) considerations should be regarded. Basically, the applications will be affected by the synchronous local pair conditions of the configuration. Your IBM representative can help you when planning your particular cascaded implementation.

13.2.5 Performance considerations

In this section we discuss general considerations on PPRC performance. This discussion is useful before addressing the next section, 13.2.6, “Planning for PPRC” on page 426.

There have been a lot of enhancements to the way two ESSs communicate over ESCON links compared to the PPRC implementation on previous ESS disks. The ESCON protocol has been streamlined, less handshaking is done, and larger ESCON frames are transmitted between two ESSs.

Several factors affect the performance you can achieve in a PPRC environment. Some of these factors are related to the configuration, such as ESCON distance and number of PPRC links that are used. Others have to do with the workload characteristics such as the data block sizes, read/write ratio, or type or processing (random vs. sequential). The amount of write activity is a key factor in the performance of the PPRC implementation.

Normally, the main objective is to protect a critical workload with the minimum impact on application response times. In the synchronous copy technique, the mirroring adds an overhead to the response time of each write I/O. Additionally, processing the PPRC writes puts more work on the ESS, thus affecting its throughput.

When sizing an ESS for PPRC you must consider both the impact on write response time, and the impact on subsystem throughput. The high performance characteristics of the ESS enable it to handle extremely high throughput rates.

Note: The PPRC guidelines and recommendations discussed in this chapter are generic and as such can be used. For a more detailed and accurate approach that takes into consideration the particularities of your environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

Establish performance

The process of getting the primary and secondary PPRC volumes in sync is called the *initial establish*. Figure 13-15 shows PPRC establish data rates for one, two, three, four, and one hundred and ninety two volumes, at a zero kilometers distance. The measurements were done on ESS Model F20s with eight 32-bit ESCON links. The aggregated 126.6 MB/sec data rate over eight links was the result of each link operating at 15.8 MB/sec.

Note that the increase in throughput beyond four establish tasks was not significant. Be careful when establishing PPRC pairs while applications are running. Even though the ESS prioritizes host I/Os, heavy load on the PPRC links due to the establish activity will affect applications response times as volumes become duplexed.

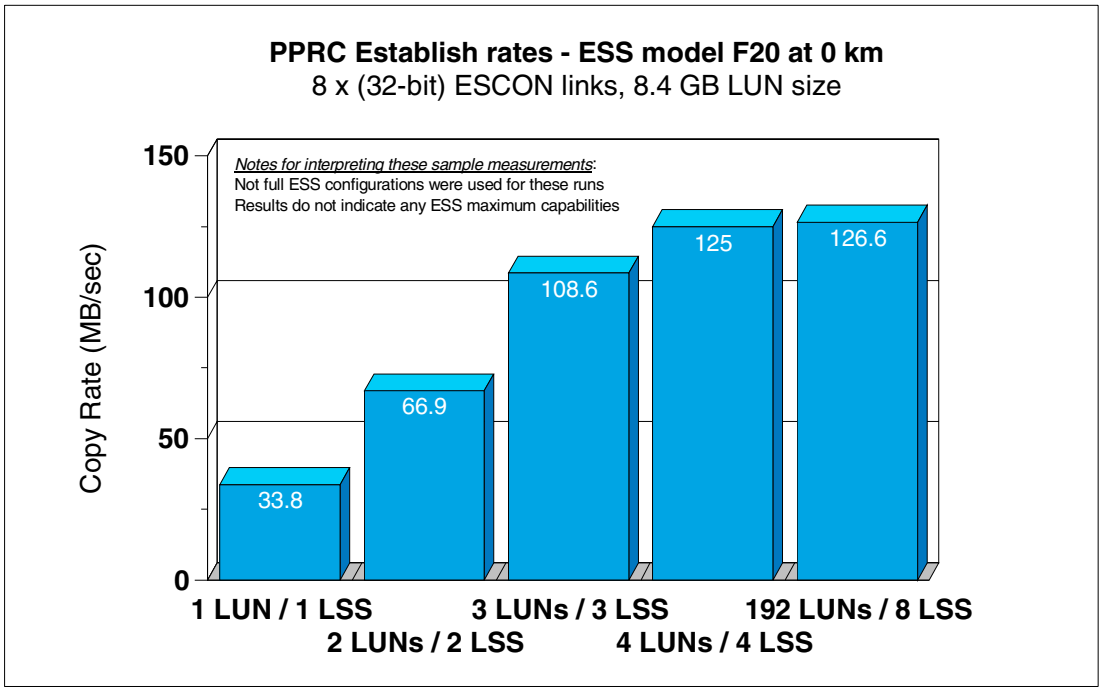


Figure 13-15 PPRC establish rates - ESS Model F20 with 32-bit ESCON adapters

The ESS Model 800 64-bit ESCON adapters provide more throughput as compared to the 32-bit ESCON adapters. And with the 64-bit adapters in the ESS Model 800 the gains are still significant when using more than four links, as compared to the 32-bit adapters in an F20 that gets almost into a plateau at four links. In fact, the ESS Model 800 establish rate scales well

up to 16 links, that is, doubling the number of links doubles the throughput (see Table 13-1 and see also Figure 13-20 on page 427).

Figure 13-16 shows how distance affects the PPRC establish rates. The measurements were done with an ESS model F20 using one 32-bit ESCON link with an Inrange 9801 as the channel extender. The top bar is the data rate with direct ESCON connection (no channel extender) at 0 km distance; the second bar is the data rate at 0 km using the extender. From Figure 13-16 you can see that as the distance increases, the data rate on the link decreases proportionally—but there is no data rate *droop* effect.

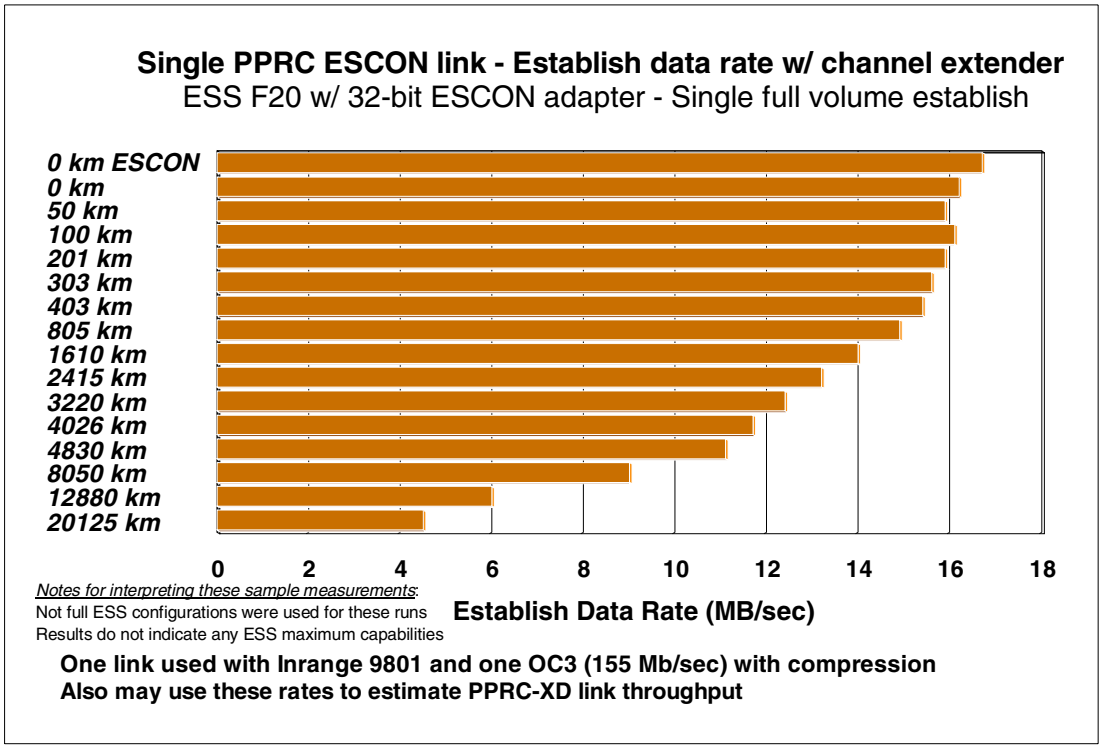


Figure 13-16 PPRC pair establish at several distances

The ESS Model 800 single link establish times are nearly 18 MB/sec at 0 KM distance when using the 64-bit ESCON adapters. The establish rates are identical whether the ESS 800 has the Standard or the Turbo processors, as the rate is determined by the link capacity (see Table 13-1). Table 13-1 illustrates the data rates that can be expected for the PPRC establish process at different distances using eight and sixteen links, for the ESS Model 800 with 64-bit ESCON adapters.

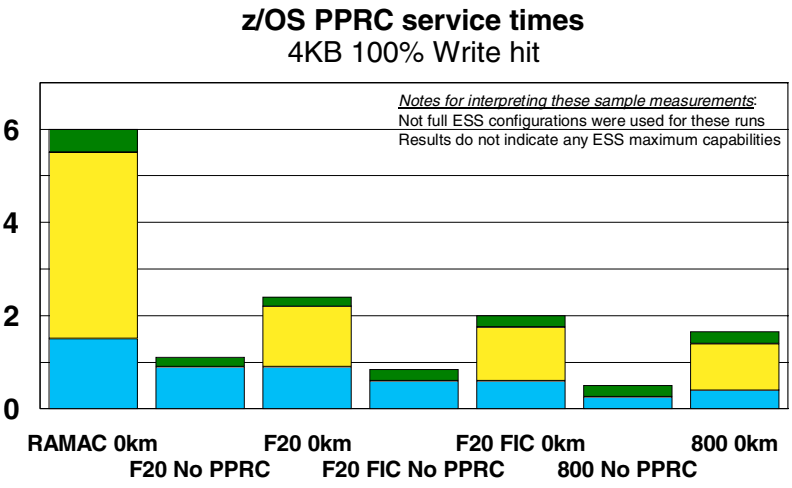
Table 13-1 ESS Model 800 PPRC establish data rates with 64-bit ESCON adapters

Model	PPRC links/distance	Establish data rate (open systems)
ESS 800 - Turbo	8/0 km	137 MB/sec
ESS 800 - Turbo	16/0 km	265 MB/sec
ESS 800 - Turbo	8/75km	77 MB/sec
ESS 800 - Standard	8/0 km	137 MB/sec

Write performance

In a synchronous operation, when the volumes are already in the full duplex condition, every update to a primary volume is first copied to the secondary volume for the I/O operation to complete. This synchronous overhead increases the response time that the applications see on its I/O operations. This increase due to PPRC appears as part of the disconnect time component of the service time (see 9.2.1, “Response time components” on page 308, for a discussion on the service time components).

Figure 13-17 illustrates the disconnect times that appear even at 0 Km distances as compared to non-PPRC operations, for different ESS and non-ESS disk models. Distance is a factor that proportionally increases the disconnect time. When planning for PPRC in your installation, in order to estimate the synchronous overhead that your applications will see, you should take into consideration your particular workload characteristics also. Your IBM representative can help you with the required modelling to obtain these results.



Conn	1.5	0.9	0.9	0.6	0.6	0.25	0.4
Disc	4	0	1.3	0	1.15	0	1
Pend	0.5	0.2	0.2	0.25	0.25	0.25	0.25
	6	1.1	2.4	0.85	2	0.5	1.65

Figure 13-17 z/OS PPRC service time ESS Model 800 vs. ESS Model F20

ESCON distance

PPRC is supported on the ESS at distances up to 103 Km. PPRC eliminates the ESCON data rate droop effect of ESCON. This means that the data rate does not decline with distance. Delays over distance are due to propagation delays only.

Figure 13-18 on page 424 illustrates an example of the effect that a distance of 75 Km has on the I/O response time of PPRC volumes for a standard workload. Distance proportionally increases the disconnect time component of the I/O response time, as the signal needs more time to travel. When planning for PPRC in your installation, in order to estimate the effect that the distance will have on your application’s I/O. In addition to the distance you should also take in consideration the particularities of your workload. Your IBM representative can help you with the required modelling to obtain these results.

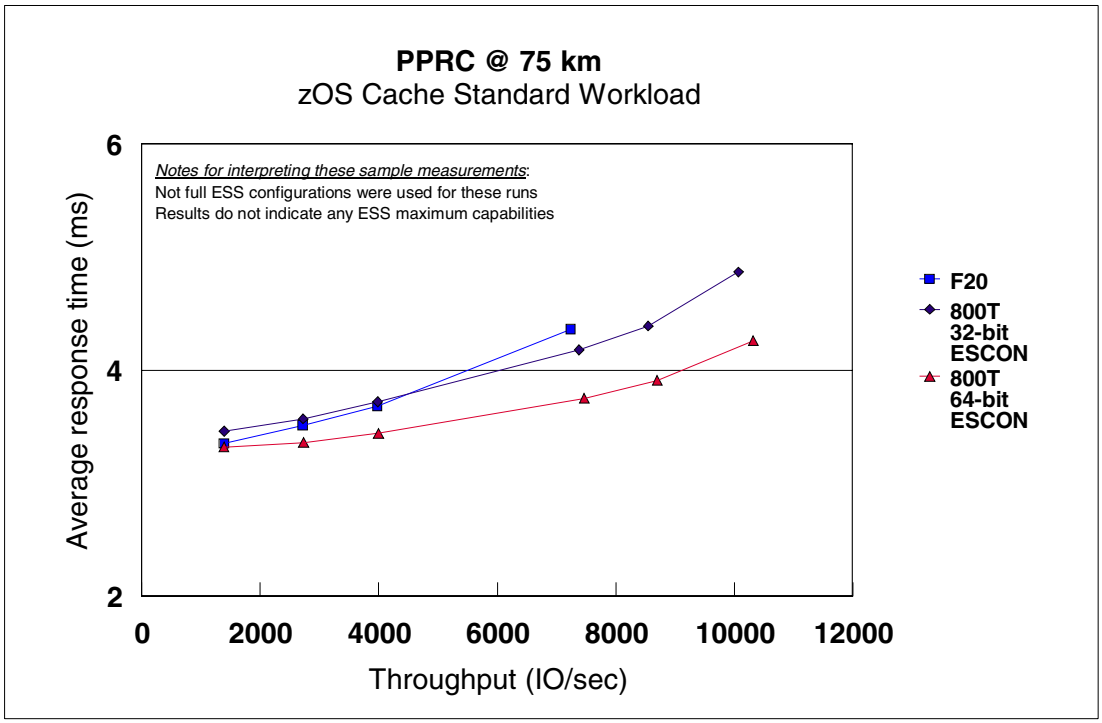


Figure 13-18 PPRC at 75 km distance

Sequential write performance

PPRC may reduce the write sequential throughput depending on distance, transfer size, and block size. This may impact S/390 batch processing and some heavy sequential UNIX or Windows workloads.

Figure 13-19 on page 425 illustrates PPRC throughputs on open systems volumes with sequential write workload, at different distances and using a different number of links on ESS Model 800 with 64-bit and 32-bit ESCON adapters. The workload is 100 percent sequential writes. Every I/O that the application does is copied to the secondary. With no PPRC, the host can do about 313 I/Os per second. When PPRC is activated on all volumes, the host I/O data rate drops down as the PPRC links throughput becomes the limiting factor.

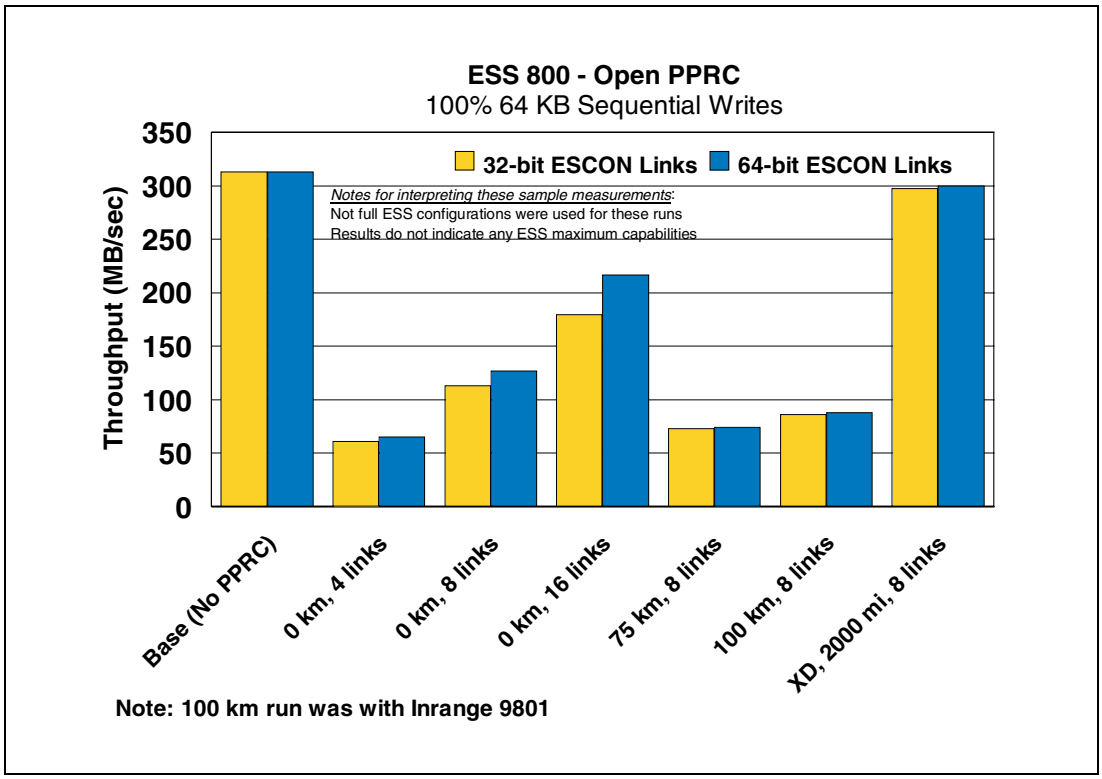


Figure 13-19 PPRC link throughput

Other conclusions from the analysis of Figure 13-19 are:

- Distance proportionally affects the synchronous PPRC operation. But for PPRC-XD at two thousand miles, distance has no significant impact on throughput as compared to non-PPRC throughput.
- On the ESS Model 800, increasing the number of links increases the throughput almost proportionally. This is in contrast to previous ESS models, where more than eight links provided no significant benefits.
- The 64-bit ESCON adapters provide significantly more throughput on local distances than the 32-bit ESCON adapters.
- The channel extender provides better throughput at long distances, with the break even point at around 75 Km. You can see in Figure 13-19 that at 100 Km the channel extender was providing slightly more throughput that at 75 Km. Under 75 Km, DWDMs may be more efficient. If using networking devices for your PPRC implementation, you should contact your channel extender or DWDM vendor to properly assess the performance capabilities of the solution.

In order to maximize sequential throughput you should spread activity across the ESS device adapters (DAs), loops, and RAID ranks to provide the maximum internal bandwidth for your workload.

PPRC-XD performance

PPRC-XD, due to its non-synchronous technique, has practically no impact on host write performance. Figure 13-19 illustrates how PPRC-XD is not affected by distance.

13.2.6 Planning for PPRC

The following are some guidelines for planning your ESS PPRC configuration. The PPRC guidelines and recommendations that we discuss in this section are generic and as such can be used.

For a more detailed and accurate approach that takes into consideration the particularities of your environment you should contact your IBM representative, who can assist you with the ESS capacity and configuration planning.

Disk capacity guidelines

The optimum disk capacity is dependant on your workload characteristics. The disk size that you select for the secondary ESS has no impact on performance. If you are planning to move production work to the secondary site, you must consider the performance of the secondary ESS.

Cache/NVS

To keep the overall response time performance when adding PPRC to your configuration, consider increasing the cache-to-backend ratio. The cache-to-backend ratio of the secondary storage control should be suitable to handle the primary I/O activity if the secondary is invoked for disaster purposes.

PPRC links

PPRC uses ESCON channels between the primary and secondary ESS. The ESS Model 800 supports 64-bit ESCON host adapters. The 64-bit ESCON host adapter provides improved performance over the predecessor 32-bit ESCON host adapter.

The number of links you need depends on the write intensity to the primary volumes, that is, the I/O rate and read-to-write ratio. A minimum of four links is recommended for availability reasons, one for each host adapter bay.

Figure 13-20 on page 427 illustrates the increase in throughput when the number of PPRC links is expanded. Results are shown for both ESS Model 800 Turbo processor with 32-bit and 64-bit ESCON adapters and for ESS Model F20 (with 32-bit ESCON host adapters). The example corresponds to a workload with 100 percent write hits at 0 km distance.

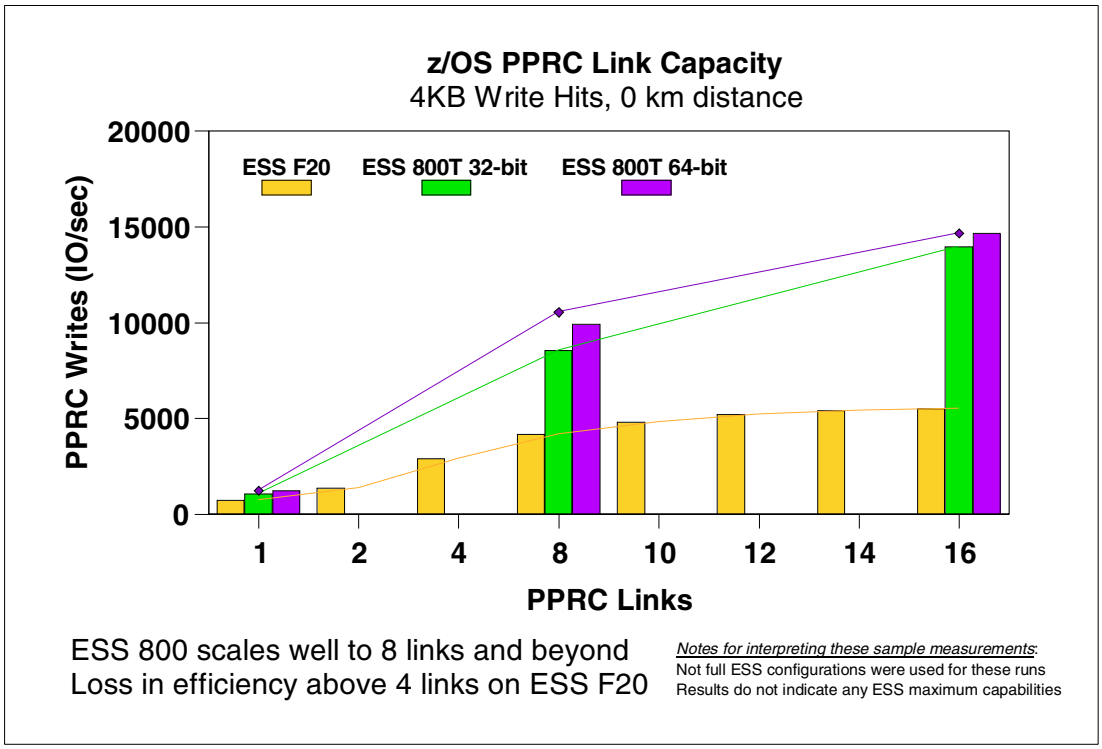


Figure 13-20 z/OS PPRC link capacity

Figure 13-20 shows that the ESS Model 800 clearly outperforms the F20. It also shows that better throughput is achieved when using the 64-bit ESCON adapters. You can see that the ESS Model 800 scales well to 16 links; that is, by adding links you get almost proportional increase in performance. On the other hand, with the ESS Model F20 more than eight PPRC links provided no significant improvement. When more than eight PPRC links are used on an ESS Model 800, the Turbo option is recommended for the best possible performance.

13.3 Extended Remote Copy (XRC)

Extended Remote Copy (XRC) is a copy function available for the z/OS users. It involves the System Data Mover (SDM) component of DFSMS/MVS and, hence, works only in z/OS and OS/390 environments. XRC maintains a copy of the data asynchronously, at a remote location. The asynchronous characteristics of XRC make it suitable for continental distance implementations. It is a combined hardware and software solution that offers data integrity and data availability that can be used as part of business continuance solutions, for workload movement, and for data migration. XRC is an optional feature on the ESS.

XRC can be controlled using TSO/E commands, or through the DFSMSdfp™ Advanced Services ANTRQST Macro, which calls the System Data Mover API.

Managing a large set of mirrored volumes over long distances requires automation for monitoring and decision making. The GDPS/XRC automation package offering provides a solution to this requirement.

The following IBM publications can be used to complement the XRC discussion presented in this section:

- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *Implementing ESS Copy Services on S/390,* SG24-5680

13.3.1 XRC operation

XRC operation is illustrated in Figure 13-21. The flow of operations is as follows:

1. The application on the primary system writes to the primary volumes.
2. The application I/O is signalled completed when the data is written to primary ESS cache and NVS, that is, channel end and device end are returned to the primary system.
3. The ESS groups the updates into record sets, which are asynchronously offloaded from the cache to the SDM system. As XRC uses this asynchronous copy technique, the performance impact on primary applications is minimal.
4. The record sets, perhaps from multiple primary ESSs, are processed into consistency groups (CGs) by the SDM. The CGs contain records that have their order of update preserved across multiple LCUs within an ESS, across multiple ESSs, and across other storage subsystems participating in the same XRC session. This preservation of order is absolutely vital for dependent write I/Os such as databases and logs. The creation of CGs guarantees that XRC will copy data with update sequence integrity.
5. When a CG is formed, it is written from the SDM real storage buffers to the journal data sets.
6. Immediately after the CG has been hardened on the journal data sets, the records are written to their corresponding secondary volumes. Those records are also written from SDM's real storage buffers. Because of the data in transit between the primary and secondary sites, the currency of the data on secondary volumes lags slightly behind the currency of the data at the primary site.
7. The control data set is updated to reflect that the records in the CG have been written to the secondary volumes.

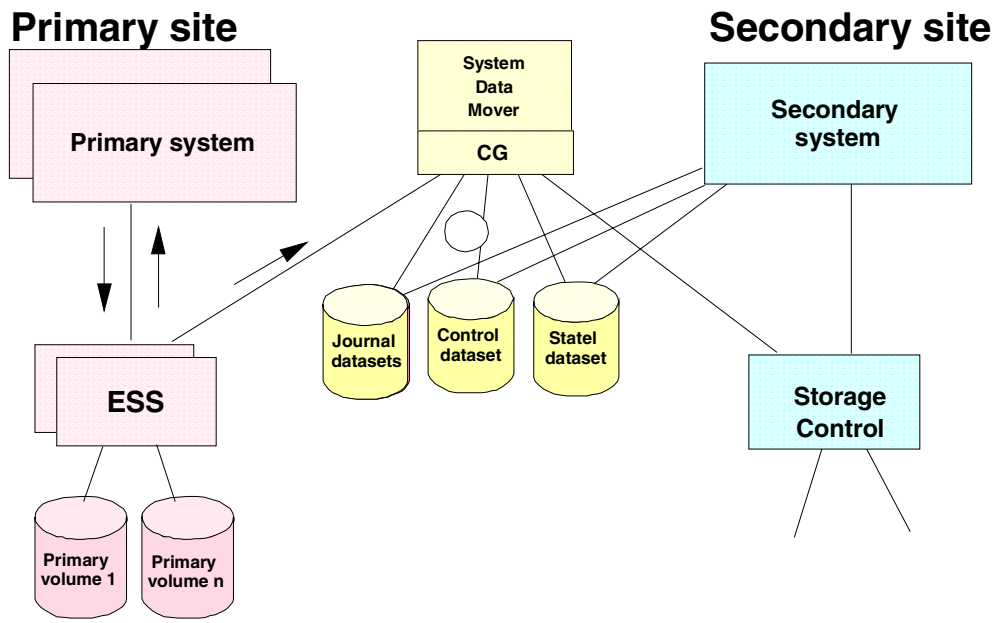


Figure 13-21 XRC operation

Applications doing write I/O to the primary volumes get the device end (DE) status (write I/O complete) as soon as the data is secured in the cache and NVS of the primary ESS. This

means that the applications' write I/O operations have no overheads due to the write to the secondary volumes.

The SDM needs to have access to all primary ESSs with XRC volumes, as well as to the target, or secondary, disk storage controls. In this way the SDM acts as the coordinator for all the disk storage subsystems, and can assure data integrity across multiple boxes. Applying updates in the correct sequence on the secondary disk subsystem is ensured by the SDM.

An XRC session can include volumes of any storage control that supports XRC, including the IBM TotalStorage Enterprise Storage Server, the IBM 9390 Models 1 and 2, the IBM 3990 Model 6, and some non-IBM storage controls that support XRC. The secondary volumes can reside on any storage control; the secondary storage controls do not need to support XRC.

Coupled Extended Remote Copy

Coupled Extended Remote Copy (CXRC) expands the capability of XRC so that very large installations that have configurations consisting of thousands of primary volumes can be assured that all their volumes can be recovered to a consistent point in time. In a disaster recovery situation, for the recovered data of a particular application to be usable the data must be recovered to a consistent point in time. CXRC provides this capability by allowing multiple XRC sessions to be coupled with a coordination of consistency between them. This way, all the volumes in all the coupled sessions can be recovered to a consistent point in time.

Dynamic workload balancing

The objective XRC's dynamic workload balancing algorithm is to balance the write activity from primary systems vs. SDM's capability to offload cache during write peaks or a temporary lack of resources to SDM, and with minimal impact to primary systems.

In situations where the SDM offload rate falls behind the primary systems write activity, data starts to accumulate in cache. This is dynamically detected by the primary ESS microcode, and it responds by slowly but progressively reducing available write bandwidth for the primary systems, thus giving the SDM a chance to catch up.

The algorithm implemented with the 3990-6 used either path blocking or device level blocking. With the ESS, only device level blocking is used. The update rate for a volume continues unrestricted unless a volume reaches a residual count threshold waiting to be collected by the SDM. Whenever that threshold is exceeded, application updates to the single volume are paused to allow the SDM to read them from the cache of the subsystem.

By using the DONOTBLOCK parameter of the XADDPAIR command, you can request XRC to not block specific devices. This option can be used for IMS WADs, DB2 logs, CICS logs, or spool data sets, which use small block sizes, do numerous updates, and are critical to application response time.

Unplanned outage support

On the IBM 3990 Model 6 storage controller, XRC pairs could be suspended only for a short time or when the System Data Mover was still active. This was because the bitmap of changed cylinders was maintained by the System Data Mover in the software. This software implementation allowed a re-synchronization of pairs only during a planned outage of the System Data Mover or the secondary disk subsystem. Instead of this, the ESS starts and maintains a bitmap of changed tracks in the hardware, in non-volatile storage (NVS), when the connection to the System Data Mover is lost or an XRC pair is suspended by command.

The bitmap is used in the re-synchronization process when you issue the XADD SUSPENDED command to re-synchronize all suspended XRC pairs. Copying only the

changed tracks is much faster compared to a full copy of all data. With ESS's XRC support, a re-synchronization is now possible for a planned, as well as an unplanned, outage of one of the components needed for XRC to operate.

13.3.2 XRC performance

The points to consider when discussing the performance of XRC are:

- Write performance; the impact on application I/O response time
- Subsystem performance; the performance of the ESS overall
- SDM performance

Write performance

XRC is an asynchronous remote mirroring solution. This means that the copy of the updated data to the secondary site occurs asynchronously to the application write I/O processing.

Application updates do not have to wait until the update is committed to the secondary volume, so there is a negligible impact on the application's I/O resulting from XRC.

Subsystem performance

Because of XRC processing there is some additional work performed by the primary ESS, for example, maintaining the update group. After each write I/O is complete (DE status to the application) the ESS retains updated data in the cache until the SDM can read the update group. There is also the additional workload resulting from the SDM reading of the updates from the ESS.

Our recommendation is that you evaluate the cache size required to satisfy your primary application needs, and then plan for XRC buffers. This may mean that you should install more cache (if possible), or increase your cache-to-backend ratio by spreading the volumes across multiple ESSs. Your IBM representative can help you determine the cache needed for XRC running the Disk Magic modelling tool.

The use of hardware bitmaps in the ESS during suspend processing, as well as during the loss of the SDM/ESS links, allows re-synchronization of the XRC pairs without complete volume copies. This significantly reduces data movement and also minimizes the use of cache by XRC, which provides much more consistent application performance, even if XRC is forced into SUSPEND mode.

New performance-enhanced CCWs

The ESS supports performance-enhanced channel command words (CCWs) that allow reading or writing more data with fewer CCWs and thus reducing the overhead of previous CCW chains. The System Data Mover will take advantage of these performance-enhanced CCWs for XRC operations on an ESS.

Establish performance

XRC does not require dedicated ESCON channels for mirroring purposes. The SDM needs to have access to the primary ESS. XRC takes advantage of the new read track CCW to maximize the performance of the establish process. XRC processes multiple ADDPAIR statements concurrently, with the performance depending on the available bandwidth. Processing multiple concurrent ADDPAIR statements reduces the average time per volume. For example, if you establish 32 concurrent volume pairs, the average time per volume is 83 minutes. You can control the rate of XADDPAIR establishes using XRC parameters.

Utility devices

For any z/OS host system to read data from a disk storage control unit that storage control unit must provide a device address. This is a requirement of the S/390 architecture that also applies when reading from the ESS. So, when the SDM issues channel commands to read updates from the primary ESS, it has to specify a device address even though the data in the cache in reality belongs to several different devices. This device address used by the SDM to offload updates from the primary subsystem cache is called the *utility device*.

If SDM has multiple XRC storage control sessions in one LCU, SDM needs one utility device per storage control session. It is important that the utility device is not heavily used by the primary system because this could prevent SDM from getting access to the device, and the offload process could slow down.

XRC uses two types of implementations of utility device. With the UTILITY parameter of the XSET command you can set the type of utility device that XRC will use.

- Fixed utility device

  In this implementation XRC always reads data from the ESS using the specified device, thus eliminating application contention for devices.

- Floating utility device

  In this implementation XRC uses a floating device in the ESS. This means that the microcode in the ESS will continuously monitor the usage of utility device candidates, select the least used device, and pass this address to the SDM.

  When using floating utility devices, all primary devices in the XRC storage control session are eligible to be used as a utility device.

SDM performance

The performance of the SDM is dependent on the processing resources, the processor memory, the number of write I/Os being processed, and the block size of the write I/Os.

You can reduce SDM's MIP requirements by increasing the amount of memory available for the SDM. The recommended real memory requirement is 35 MB per primary LSS for each XRC session (remember, the ESS may be defined as up to 16 LSSs).

If you have more than 1500–1800 volumes in a single XRC session, you may need to split these into two or more XRC sessions. Up to five XRC sessions are allowed in each z/OS image.

13.3.3 Planning for XRC

The performance that you will achieve with XRC on the ESS is dependent on your workload as well as the configuration that you will be using. This section discusses some guidelines for planning your XRC configuration. The XRC guidelines and recommendations that we discuss in this section are generic and as such can be used.

For a more detailed and accurate approach that takes into consideration the particularities of your environment you should contact your IBM representative, who can assist you with the capacity and configuration planning.

ESS configuration

When sizing an ESS for XRC you must accommodate the additional workload from the SDM reads. This can be estimated by adding the number of write I/Os of the current workload to the total number of I/Os.

The size of the secondary storage control does not impact performance, although you may need to consider sizing it to accommodate production workload on the secondary storage controls in the event of a disaster.

The secondary site storage controls must be configured to handle all of the primary site writes plus the I/Os to the journal, control, and state data sets as a minimum. But they must also be capable of supporting the I/O activity related to the primary application's requirements in a disaster recovery situation.

Although there is no requirement to use ESSs as secondary storage subsystems, doing so will give you all the benefits of the ESS. In addition, it also makes it possible to use XRC in a copy-back implementation. With ESSs at the secondary you can also combine XRC and FlashCopy functions.

SDM to primary site bandwidth

One of the key elements in the sizing of an XRC configuration is to determine the required bandwidth between the SDM and the primary ESSs. This has to be done by analyzing the workload to determine the peak requirements.

The peak period analyzed should be short enough so that lower workload levels do not average the peak workload down. Typically you should analyze online requirements during the 5-minute peak period, and batch requirements during the 15-minutes peaks. Taking this approach, you can still benefit from the dynamic workload balancing function of XRC, without really compromising application throughput or response time.

The challenge can be to find the peak intervals, because the period with the highest write activity is not necessarily the period with the highest bandwidth requirement. Typically the write activity during daytime processing can be high, but this processing uses short write block sizes. The required bandwidth can therefore be higher during batch processing, even if the write rate is lower, because of larger block sizes.

You could use a report generator tool to identify peaks of write I/Os and data transfer based on information in SMF record type 74. Subtype 1 (device activity) contains information about I/O rates and connect times, and subtype 5 (cache activity) contains information about the number of reads and number of writes per device. Make sure that known peak days such as end of month processing, are included in the analyses.

SDM data sets

There are three data sets required for XRC: The journal, the control, and the state data sets, generally referred to as the control data sets.

The control data set has pointers to the journal data sets indicating the last set of consistency groups written to the secondary volumes and the amount of data written to the journal.

You can allocate the control data set in one of two ways, as a sequential data set or as a PDSE. SDM detects which type of control data set you have allocated and supports the specified data set type. We recommend the sequential allocation; this provides a much higher performance than the PDSE allocation.

The state data set records the status of the XRC session, including the timestamps of the latest consistency groups journaled and committed to disk.

The state and control data sets do not need to reside on dedicated volumes, and can share a volume.

The third data set is the XRC journal data set; this is used by the XRC recovery process. The journal data set contains checkpoint records of changes applied to the secondary volumes. You can define a minimum of two journal data sets, up to 16 journal data sets for an XRC session. We recommend that these are defined in pairs.

The performance of the journal data set is critical to SDM performance. Journaling is an area that can easily become a bottleneck and contribute to SDM slow down. If not properly configured, the result could be that the consistency groups would accumulate in the SDM's address space because the journal data sets did not have the necessary performance.

Journal data sets have a high sequential write requirement. We recommend that these are defined as single extent striped sequential data sets.

Define, preferably, four to eight journal data sets for smaller XRC configurations (for up to about 750 volumes). For large XRC configurations, define from eight to 16 journal data sets.

FICON

The SDM benefits greatly from the higher bandwidth FICON channels provide to read the updates from the primary ESS. The transfer bandwidth for a single thread read for XRC is at least five times better than with ESCON due to the larger block sizes of data transfers.

The improved bandwidth of FICON together with the longer un-repeated distance support (up to 100 km) results in a considerable improvement over ESCON channels, when using direct channel attachment. These capabilities position XRC as a disaster recovery solution in the range of metropolitan areas. With ESCON, the recommendation is channel extenders through telecom lines beyond 25 km. With FICON, there is no need to use channel extenders until the distance exceeds 100 km.

A

UNIX shell scripts

This appendix includes scripts helpful to manage disk devices and monitor I/O for servers attached to the ESS. Implementation of these scripts is described in 6.5, “AIX-specific I/O monitoring commands” on page 184.

Introduction

The scripts presented in this appendix were written and tested on AIX servers, but could be modified to work with SUN Solaris and HP-UX.

By downloading the Acrobat PDF version of this publication, you should be able to copy and paste these scripts for easy installation on your host systems. To function properly, the scripts presented here rely on:

- An AIX host running AIX 4.3.3ML10+ or AIX 5L™
- Subsystem Device Driver (SDD) for AIX Version 1.3.1.0 or later
- ESS Utility package Version 1.0.6 or later

The scripts presented in this appendix are:

- vgmap
- lvmap
- vpath_iostat
- ess_iostat
- test_disk_speeds

Attention: These scripts are provided on an ‘as is’ basis. They are not supported or maintained by IBM in any formal way. No warranty is given or implied, and you cannot obtain help with these scripts from IBM.

VGMAP

The **vgmap** script displays which vpaths a volume group uses and also which rank each vpath belongs to. Use this script to determine if a volume group is made up of vpaths on several different ranks and which vpaths to use for creating striped logical volumes.

An example output of the **vgmap** command is shown in Example A-1.

Example: A-1 VGMAP output

```
# vgmap testvg

PV_NAME         RANK    PV STATE  TOTAL PPs   FREE PPs
testvg:
vpath0          1100    active      502         502
vpath2          1000    active      502         502
```

```
#!/bin/ksh
###########################################
# VGMAP
# usage: vgmap <vgname>
#
# Displays ESS logical disks and RANK ids for each
# disk in the volume group
#
# Note: the script depends on correct lssdd info in
# /tmp/lssdd.out
#
# Before running the first time, run:
# lssdd > /tmp/lssdd.out
#
# Author: Pablo Clifton pablo.clifton@usa.net
```

```
# Date: Feb 28, 2003
#########################################

lssddfile=/tmp/lssdd.out
workfile=/tmp/work.$0
sortfile=/tmp/sort.$0

# AIX
lsvg -p $1 | grep -v "PV_NAME" > $workfile
echo "\nPV_NAME   RANK          PV STATE        TOTAL PPs    FREE PPs    Free D"

for i in `cat $workfile | grep vpath | awk '{print $1}'`
do
        #echo "$i ... rank"
        rank=`grep -w $i $lssddfile | awk '{print $12}' | head -n 1`
        sed "s/$i /$i    $rank/g" $workfile > $sortfile
        cp $sortfile $workfile
done

cat $workfile
rm $workfile
rm $sortfile
######################### THE END ####################
```

LVMAP

The **lvmap** script displays which vpaths and ranks a logical volume uses. Use this script to determine if a logical volume spans vpaths on several different ranks. The script does not tell you if a logical volume is striped or not. Use **lslv <lv_name>** for that information or modify this script.

An example output of the **lvmap** command is shown in Example A-2.

Example: A-2 LVMAP output

```
# lvmap stripedlv

LV_NAME   RANK           COPIES        IN BAND        DISTRIBUTION
stripedlv:
vpath2    1000           004:000:000   100%          000:004:000:000:000
vpath4    1300           004:000:000   100%          000:004:000:000:000
vpath10   1400           004:000:000   100%          000:004:000:000:000
```

```
#!/bin/ksh
#########################################
# LVMAP
# usage: lvmap <lvname>
#
# displays logical disk and rank ids for each
# disk a logical volume resides on

# Note: the script depends on correct lssdd info in
# /tmp/lssdd.out
#
# Before running the first time, run:
# lssdd > /tmp/lssdd.out
#
# Author: Pablo Clifton pablo.clifton@usa.net
```

```
# Date: Feb 28, 2003
#########################################

lssddfile=/tmp/lssdd.out
workfile=/tmp/work.$0
sortfile=/tmp/sort.$0

lslv -l $1 | grep -v " COPIES " > $workfile

for i in `cat $workfile | grep vpath | awk '{print $1}'`
do
        #echo "$i ... rank"
        rank=`grep -w $i $lssddfile | awk '{print $12}' | head -n 1`
        sed "s/$i /$i    $rank/g" $workfile > $sortfile
        cp $sortfile $workfile
done

echo "\nLV_NAME   RANK           COPIES        IN BAND        DISTRIBUTION"
cat $workfile

rm $workfile
rm $sortfile
######################### THE END ####################
```

VPATH_IOSTAT

The **vpath_iostat** script is a a wrapper program for AIX that converts iostat information based on hdisk devices to vpaths instead.

The **vpath_iostat** script depends on the **lssdd** command (included in the ESSUTIL package for AIX) and **iostat**. The script first builds a map file to list hdisk devices and their associated vpaths and then converts **iostat** information from hdisks to vpaths.

To run the script, make sure the ESS UTILITY **lssdd** and SDD are working properly—that is, all volume groups are using vpaths instead of hdisk devices and **lssdd** reports the correct information.

The command syntax is:

```
vpath_iostat (control c to break out)
```

or

```
vpath_iostat <interval> <iteration>
```

An example of the output **vpath_iostat** produces is shown in Example A-3.

Example: A-3 VPATH_IOSTAT output

```
garmo-aix: Total VPATHS used:        8    16:16 Wed 26 Feb 2003   5 sec interval
garmo-aix Vpath:          MBps           tps             KB/trans        MB_read         MB_wrtn
garmo-aix vpath0          12.698         63.0            201.5           0.0             63.5
garmo-aix vpath6          12.672         60.6            209.1           0.0             63.4
garmo-aix vpath14         11.238         59.8            187.9           0.0             56.2
garmo-aix vpath8          11.314         44.6            253.7           0.0             56.6
garmo-aix vpath2          6.963          44.2            157.5           0.0             34.8
garmo-aix vpath12         7.731          30.2            256.0           0.0             38.7
garmo-aix vpath4          3.840          29.4            130.6           0.0             19.2
garmo-aix vpath10         2.842          13.2            215.3           0.0             14.2
--------------------------------------------------------------------------------------
```

```
garmo-aix   TOTAL READ:   0.00    MB      TOTAL WRITTEN:  346.49  MB
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   70.00   MB/sec
```

```
#!/bin/ksh
#####################################################################
# Usage:
#       vpath_iostat  (default: 5 second intervals, 1000 iterations)
#       vpath_iostat <interval> <count>
#
# Function:
#       Gather IOSTATS and report on ESS VPATHS instead of disk devices
#       AIX hdisks
#       HP-UX [under development ]
#       SUN   [under development ]
#       Linux [under development ]
#
# Note:
#       vpath_iostat depends on valid VPATH ids from the LSSDD command which
#       is part of the ESS Utilities
#
#       A small amount of free space < 1MB is required in /tmp
#
# Author:       Pablo Clifton   pablo.clifton@usa.net
# Date: Feb 28, 2003
#####################################################################

#########################################################
# set the default period for number of seconds to collect
# iostat data before calculating average
period=5
iterations=1000

essfile=/tmp/disk-vpath.out             # File to store output from lssdd command
ifile=/tmp/lssdd.out                    # Input file containing LSSDD info

ds=`date +%d%H%M%S`             # time stamp
hname=`hostname`                # get Hostname
ofile=/tmp/vstats               # raw iostats
wfile=/tmp/wvfile               # work file
wfile2=/tmp/wvfile2             # work file
pvcount=`iostat | grep hdisk | wc -l | awk '{print $1}'`

#########################################
# Create a list of the vpaths this system uses
# Format:       hdisk           ESS-vpath
# LSSDD output MUST BE correct or the IO stats reported
# will not be correct
#########################################
if [ ! -f $ifile ]
then
  echo "Collecting LSSDD info for disk to vpath map..."
  lssdd > $ifile
fi

cat $ifile  | awk '{print $4 "\t" $3}' > $essfile

#####################################
# ADD INTERNAL SCSI DISKS to RANKS list
#####################################
for internal in `lsdev -Cc disk | grep SCSI | awk '{print $1}'`
```

```
do
        echo "$internal $internal" >> $essfile
done

##########################################
# Set interval value or leave as default
if [[ $# -ge 1 ]]
then
        period=$1
fi

#######################################
# Set <iteration> value
if [[ $# -eq 2 ]]
then
        iterations=$2
fi

############################################################
#  ess_iostat <interval> <count>

i=0
while [[ $i -lt $iterations ]]
do
        iostat $period 2  > $ofile              # run 2 iterations of iostat
                                                # first run is IO history since boot

        grep hdisk $ofile > $ofile.temp # only gather hdisk info- not cd
                                                        # other devices

        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | sort +4 -n -r | head -n 100 > $wfil
e

        ########################################
        #Converting hdisks to vpaths....           #
        ########################################
        for j in `cat $wfile  | awk '{print $1}'`
        do
           vpath=`grep -w $j $essfile | awk '{print $2}'`
           sed "s/$j /$vpath/g" $wfile > $wfile2
           cp $wfile2 $wfile
        done


        ########################################
        # Determine Number of different VPATHS used
        ########################################
        numvpaths=`cat $wfile | awk '{print $1} ' | grep -v hdisk | sort -u | wc -l`

        dt=`date +"%H:%M %a %d %h %Y"`

        print "\n$hname: Total VPATHS used: $numvpaths  $dt     $period sec interval"
        printf "%s\t%s\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n"  "$hname" "Vpath:" "MBps" "tps"
"KB/trans" "MB_read" "MB_wrtn"


        ########################################
        # Sum Usage for EACH VPATH and Internal Hdisk
        ########################################
```

```
        for x in `cat $wfile |  awk '{ print $1}' | sort -u`
        do
        cat $wfile | grep -w $x | awk '{ printf ("%4d\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n" ,
$1, $2, $3, $4, $5, $6) }' | awk 'BEGIN {
 }
                { tmsum=tmsum+$2 }
                { kbpsum=kbpsum+$3 }
                { tpsum=tpsum+$4 }
                { kbreadsum=kbreadsum+$5 }
                { kwrtnsum=kwrtnsum+$6 }

        END {
                if ( tpsum > 0 )
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
vpath, kbpsum/1000, tpsum, kbpsum/tpsum , k
breadsum/1000, kwrtnsum/1000)
                else
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
vpath, kbpsum/1000, tpsum, "0", kbreadsum/1
000, kwrtnsum/1000)
             }' hname="$hname" vpath="$x" >> $wfile2.tmp

        done


        ##########################################
        # Sort VPATHS/hdisks by NUMBER of TRANSACTIONS
        ##########################################
        if [[ -f $wfile2.tmp ]]
        then
                cat $wfile2.tmp | sort +3 -n -r
                rm $wfile2.tmp
        fi


        ############################################################
        # SUM TOTAL IO USAGE for ALL DISKS/LUNS over INTERVAL
        ############################################################
        #Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
        # field 5 read field 6 written
        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | awk  'BEGIN { }
                { rsum=rsum+$5 }
                { wsum=wsum+$6 }

                END {
                rsum=rsum/1000
                wsum=wsum/1000

                printf
("-------------------------------------------------------------------------------------
-\n")

                if ( divider > 1 )
                {
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n", hname, "TOTAL READ: ",
rsum, "MB", "TOTAL WRITTEN: ", wsum, "MB"
 ) }
```

```
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n\n\n", hname, "READ SPEED:
", rsum/divider, "MB/sec", "WRITE SPEED:
 ", wsum/divider, "MB/sec" )
                 }'  hname="$hname"  divider="$period"

        let i=$i+1

done

rm $ofile
rm $wfile
rm $wfile2
rm $essfile

################## THE END #################
```

ESS_IOSTAT

The **ess_iostat** script is a a wrapper program for AIX that converts **iostat** information based on hdisk devices to ranks instead.

The **ess_iostat** script depends on the **lsess** command (included in the ESSUTIL package for AIX) and **iostat**. The script first builds a map file to list hdisk devices and their associated ranks and then converts **iostat** information from hdisks to ranks.

To run the script, make sure the ESS UTILITY **lsess** is working properly and enter:

```
ess_iostat (control c to break out)
```

or

```
ess_iostat <interval> <iteration>
```

An example of the **ess_iostat** output is shown in Example A-4.

Example: A-4 ESS_IOSTAT output

```
# ess_iostat 5 1

garmo-aix: Total RANKS used:       12      20:01 Sun 16 Feb 2003   5 sec interval
garmo-aix  Ranks:       MBps             tps           KB/trans       MB_read        MB_wrtn
garmo-aix  1403         9.552           71.2            134.2           47.8            0.0
garmo-aix  1603         6.779           53.8            126.0           34.0            0.0
garmo-aix  1703         5.743           43.0            133.6           28.8            0.0
garmo-aix  1503         5.809           42.8            135.7           29.1            0.0
garmo-aix  1301         3.665           32.4            113.1           18.4            0.0
garmo-aix  1601         3.206           27.2            117.9           16.1            0.0
garmo-aix  1201         2.734           22.8            119.9           13.7            0.0
garmo-aix  1101         2.479           22.0            112.7           12.4            0.0
garmo-aix  1401         2.299           20.4            112.7           11.5            0.0
garmo-aix  1501         2.180           19.8            110.1           10.9            0.0
garmo-aix  1001         2.246           19.4            115.8           11.3            0.0
garmo-aix  1701         2.088           18.8            111.1           10.5            0.0
-------------------------------------------------------------------------------------------
garmo-aix    TOTAL READ:   430.88  MB         TOTAL WRITTEN:  0.06    MB
garmo-aix    READ SPEED:    86.18 MB/sec      WRITE SPEED:    0.01   MB/sec
```

```
#!/bin/ksh
```

```
#set -x
#####################################################################
# Usage:
#       ess_iostat  (default: 5 second intervals, 1000 iterations)
#       ess_iostat <interval> <count>
#
# Function:
#       Gather IOSTATS and report on ESS RANKS instead of disk devices
#       AIX hdisks
#       HP-UX
#       SUN
#       Linux
#
# Note:
#       ess_iostat depends on valid rank ids from the LSESS command which
#       is part of the ESS Utilities
#
#       A small amount of free space < 1MB is required in /tmp
#
# Author:       Pablo Clifton   pablo.clifton@usa.net
# Date: Feb 28, 2003
#####################################################################

#########################################################
# set the default period for number of seconds to collect
# iostat data before calculating average
period=5
iterations=1000

essfile=/tmp/lsess.out          # File to store output from lsess command

ds=`date +%d%H%M%S`             # time stamp
hname=`hostname`                        # get Hostname
ofile=/tmp/rstats                       # raw iostats
wfile=/tmp/wfile                        # work file
wfile2=/tmp/wfile2              # work file
pvcount=`iostat | grep hdisk | wc -l | awk '{print $1}'`

#########################################
# Create a list of the ranks this system uses
# Format:       hdisk           ESS-rank
# LSESS output MUST BE correct or the IO stats reported
# will not be correct
#########################################
lsess | awk '{print $1 "\t" $9}' > $essfile


#######################################
# ADD INTERNAL SCSI DISKS to RANKS list
#######################################
for internal in `lsdev -Cc disk | grep SCSI | awk '{print $1}'`
do
        echo "$internal $internal" >> $essfile
done


###########################################
# Set interval value or leave as default
if [[ $# -ge 1 ]]
then
```

```
        period=$1
fi


#######################################
# Set <iteration> value
if [[ $# -eq 2 ]]
then
        iterations=$2
fi


#############################################################
#  ess_iostat <interval> <count>

i=0
while [[ $i -lt $iterations ]]
do
        iostat $period 2  > $ofile              # run 2 iterations of iostat
                                                        # first run is IO history since
boot

        grep hdisk $ofile > $ofile.temp # only gather hdisk info- not cd
                                                        # other devices

        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | sort +4 -n -r | head -n 100 > $wfil
e

        ########################################
        #Converting hdisks to ranks....           #
        ########################################
        for j in `cat $wfile  | awk '{print $1}'`
        do
           rank=`grep -w $j $essfile | awk '{print $2}'`
           sed "s/$j /$rank/g" $wfile > $wfile2
           cp $wfile2 $wfile
        done


        ########################################
        # Determine Number of different ranks used
        ########################################
        numranks=`cat $wfile | awk '{print $1} ' | grep -v hdisk | cut -c 1-4| sort -u -n |
wc -l`
        dt=`date +"%H:%M %a %d %h %Y"`

        print "\n$hname: Total RANKS used: $numranks    $dt     $period sec interval"
        printf "%s\t%s\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n"  "$hname" "Ranks:" "MBps" "tps"
"KB/trans" "MB_read" "MB_wrtn"


        ########################################
        # Sum Usage for EACH RANK and Internal Hdisk
        ########################################
        for x in `cat $wfile |  awk '{ print $1}' | sort -u`
        do
        cat $wfile | grep -w $x | awk '{ printf ("%4d\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n" ,
$1, $2, $3, $4, $5, $6) }' | awk 'BEGIN {
 }
```

```
                { tmsum=tmsum+$2 }
                { kbpsum=kbpsum+$3 }
                { tpsum=tpsum+$4 }
                { kbreadsum=kbreadsum+$5 }
                { kwrtnsum=kwrtnsum+$6 }

        END {
                if ( tpsum > 0 )
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
rank, kbpsum/1000, tpsum, kbpsum/tpsum , kb
readsum/1000, kwrtnsum/1000)
                else
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
rank, kbpsum/1000, tpsum, "0", kbreadsum/10
00, kwrtnsum/1000)
             }' hname="$hname" rank="$x" >> $wfile2.tmp

        done


        ##########################################
        # Sort RANKS/hdisks by NUMBER of TRANSACTIONS
        ##########################################
        if [[ -f $wfile2.tmp ]]
        then
                cat $wfile2.tmp | sort +3 -n -r
                rm $wfile2.tmp
        fi


        ############################################################
        # SUM TOTAL IO USAGE for ALL DISKS/LUNS over INTERVAL
        ############################################################
        #Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
        # field 5 read field 6 written
        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | awk  'BEGIN { }
                { rsum=rsum+$5 }
                { wsum=wsum+$6 }

                END {
                rsum=rsum/1000
                wsum=wsum/1000

                printf
("-------------------------------------------------------------------------------------------
-\n")

                if ( divider > 1 )
                {
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n", hname, "TOTAL READ: ",
rsum, "MB", "TOTAL WRITTEN: ", wsum, "MB"
 ) }

               printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n\n\n", hname, "READ SPEED:
", rsum/divider, "MB/sec", "WRITE SPEED:
 ", wsum/divider, "MB/sec" )
                }'  hname="$hname"  divider="$period"

        let i=$i+1
```

```
done

rm $ofile
rm $wfile
rm $wfile2
rm $essfile
################################ THE END ########################
```

TEST_DISK_SPEEDS

Use the **test_disk_speeds** script to test a vpath and record the speed at different times throughout the day to get an *average* read speed a rank is capable of in your environment. Use **lsess** or **lssdd** to determine which ranks the vpaths reside on.

You can change the amount of data read, the block size, and the vpath by editing the script and changing the variables:

```
tsize=100 # MB
bs=128 # KB
vpath=vpath0 # disk to test
```

An example of the output for **test_disk_speeds** is shown in Example A-5.

Example: A-5 TEST_DISK_SPEEDS example

```
# test_disk_speeds
vpath0 43.0 MB/sec  100 MB      bs=128k
```

```
#!/bin/ksh
#########################################################
# test_disk_speeds
# Measure disk speeds using dd
#
# tsize = total test size in MB
# bs    = block size in KB
# testsize= total test size in KB; tsize*1000

# count = equal to the number of test blocks to read which is
#         testsize/bsize
# Author: Pablo Clifton pablo.clifton@usa.net
# Date: February 28, 2003
#########################################################
# SET these 2 variables to change the block size and total
# amount of data read. Set the vpath to test
tsize=100 # MB
bs=128 # KB
vpath=vpath0 # disk to test
#########################################################
let testsize=$tsize*1000
let count=$testsize/$bs

# calculate start time, dd file, calculate end time
stime=`perl -e "print time"`
dd if=/dev/$vpath of=/dev/null bs="$bs"k count=$count
etime=`perl -e "print time"`
```

```
# get total run time in seconds
let totalt=$etime-$stime
let speed=$tsize/$totalt

printf "$vpath\t%4.1f\tMB/sec\t$tsize\tMB\tbs="$bs"k\n" $speed
######################### THE END ##############################
```

B

I/O terminology

This appendix contains the definitions of terms used to describe I/O operations in UNIX, Windows NT, and z/OS environments. Also presented in this chapter is the description of an I/O operation in each of these environments.

z/OS terminology

In this section we define the components of a z/OS I/O operation. The discussion in this section can be complemented with the discussion in 9.2.1, “Response time components” on page 308).

Components of a DASD I/O operation

The response time of an I/O to a DASD device is divided into four different components: connect, disconnect, pending, and IOSQ. These components are reported in the RMF Monitor I Direct Access Device Activity report.

- Connect

  The part of the I/O during which data is actually transferred, protocol, search and data transfer time.

- Disconnect

  Time that an I/O request spends freed from the channel. This is the time that the I/O positions for the data that has been requested. For the old 3990/3390 devices this included:

  a. SEEK and SET SECTOR

     Moving the device to the requested cylinder and track.

  b. Latency

     Waiting for the record to rotate under the head.

  c. Rotational Position Sensing (RPS)

     Rotational delay, as the device waits to reconnect to the channel. The term "RPS delay" traditionally means waiting for an extra disk rotation because a transfer could not occur when the data was under the head. Delays of this type are now obsolete with the ESS.

  With the ESS these components exist, but how they work and what their effects are on the I/O response time have significantly changed.

- Pending (PEND)

  Time that the I/O is delayed in the path to the device. Pending time may be attributable to the channel, control unit, or director port contention, although it is often caused by shared DASD.

- IOS Queue (IOSQ)

  Time that an I/O waits because the device is already in use by another task on this system, signified by the device's UCBBUSY bit being on.

For most users reporting is based on one or both of the following measurements:

- Service time

  Connect time plus disconnect time plus pending time

- Response time

  Service time plus IOSQ time

Cache I/O operation

The DASD and CACHE components of an I/O operation are illustrated in Figure B-1. These components are reported by the RMF Monitor I Cache Subsystem Activity report.

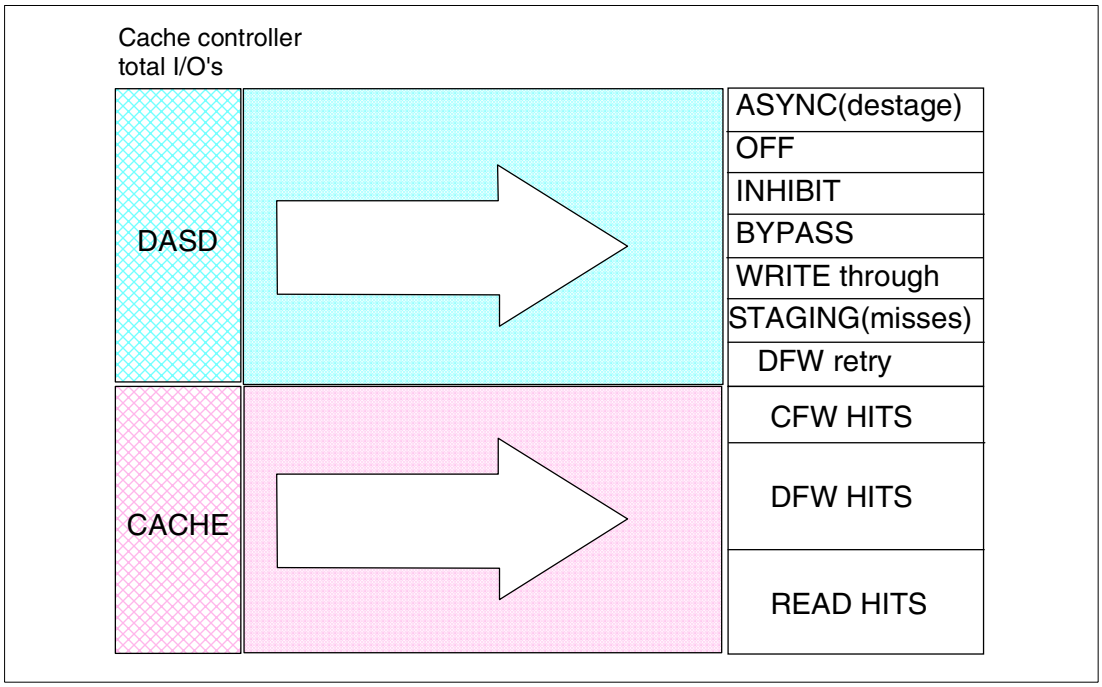


Figure B-1 Response time components of cached I/O operations

CACHE

For CACHE:

- READ HIT

  Normal and sequential read hits.

- DFW HIT

  DASD fast write (DFW) hits. In the ESS, a DFW operation always results in a hit even if the record is not in the cache. No staging is required.

- CFW HIT

  Cache fast write (CFW) write and read hits.

DASD

For DASD:

- STAGE

  Read misses and DFW misses. The record was not found in the cache. It is read or written, and that record with the rest of the track is staged into the cache. In the ESS, stages no longer occur when data for a DFW request is absent from cache. Instead, this is handled as a write ‘promote’ (and counts as a hit).

- DFWRETRY

  DASD fast write retry. The record is in cache and is about to result in a cache hit, but non-volatile storage (NVS) is full. The operation is retried by the channel subsystem, which usually results in a cache hit.

- THRU

  Write THRU to DASD. These are writes to devices behind storage controls that are not enabled for DFW. This does not apply to the ESS since DFW cannot be disabled on ESS.

- BYPASS

The bypass mode was specified in the define extent. The I/O is sent directly to DASD; the track is not promoted and the record is invalidated in cache. The ESS does not actually bypass the cache, but ensures that data that specifies bypass mode is destaged quickly.

- INHIBIT

  The inhibit mode was specified in the define extent and the record was not in the cache. Examples of inhibit include DFDSS for reads and DFSMS dynamic cache management (DCM or DCME) *maybe cache* and *never cache* storage classes. The I/O will retrieve the record in cache if it is there; if it is not in the cache, the I/O is sent directly to DASD; the track is staged into cache and marked for a fast destage. If inhibit mode was specified and the result was a cache hit, the I/O would fall into one of the cache hit categories.

- OFF

  Device is turned off for caching. Caching on the ESS is always turned on.

- ASYNC

  Asynchronous destage of records to DASD. This includes anticipatory destage to write new or updated data from cache to DASD if NVS or the cache is full. The unit is in tracks per second. All the other values given above are in I/Os per second. Async is not actually part of the I/O rate. It is the consequence of a write hit, which must eventually be written to DASD. Counting it in the I/O rate would cause the I/O to be accounted for twice, once on the write hit and again when the track was destaged. Async can offer insight into the DFW.

It should be noted that the causes of bypass and inhibit listed above may not be all inclusive. In particular, you should consult your software vendor to understand which caching mode, if any, they use. The default mode is normal or sequential. However, bypass and inhibit mode can explicitly be set in software on an I/O basis. For example, it is possible for some I/Os to a data set to use inhibit and some normal caching mode, even within the execution of a particular application program. This is true for DB2. Within DB2, one plan may access a table space in normal caching mode and another plan may optimize to access in prefetch mode, utilizing the bypass mode to accomplish this.

z/OS I/O terminology

In this section we define some of the most common z/OS terms for I/O processing. Since for z/OS the ESS attaches to the host using ESCON or FICON, we will only describe I/O operations using those architectures.

Device number

Every device (locally attached) in the configuration is represented by a device number. Device numbers define devices to the operating systems. Device numbers are:

- Assigned by system programmers
- Numbers in the range x'0000-FFFF', 16 bits, allowing for a maximum of 65,536 devices per system
- Used in commands, messages, and error recording

CHPID

Each channel is identified by a Channel Path IDentifier (CHPID).

- Eight bit (two hexadecimal digit) number that allows for up to 256 CHPIDs in the range x’00-FF’
- Status displayed using the D M=CHP command

A single ESCON or FICON channel may be shared by more than one z/OS image as long as all sharing images reside on the same physical Computer Electronic Complex (CEC).

Subchannels

Subchannels are hardware definitions. They are used by the channel subsystem to represent devices and to control I/O operations over channels to devices. A subchannel must exist for the channel subsystem to allow applications to talk to devices. Subchannels are built at power-on-reset time or as a result of CHPID re-configuration, or added or deleted dynamically using the MVS command. Every device defined in the IOCP is assigned a subchannel number:

- Used in communications between the I/O Supervisor (IOS) and the Channel Subsystem (CSS).
- Subchannel numbers are contiguous and dense.
- Are assigned consecutively (starting at x'0000') when the IOCP program processes the IODEVICE macro-instructions in the IOCP input.
- Usually not required by the operator.

When subchannels are built they are left in a disabled state during IPL processing. MVS attempts to match UCBs and subchannels, and those that have matching UCBs are enabled.

Unit address

The unit address is an ID by which a channel knows a control unit and attached devices.

- For the ESS the unit address is set up when the devices are defined using ESS Specialist.
- Two hexadecimal digits in the range x'00-FF'. The value defaults to the last two characters of the device number if no value is specified in the IODF.
- Transmitted over the channel to select device.
- Usually not required by the operator.

There is not necessarily a correspondence between the unit address, device number and subchannel number assigned to a device.

Link address

This is the ESCON Director port at which the control unit attaches to an ESCON Director. It can be displayed using ESCON Manager.

Volume serial

Each disk device (DASD) is assigned a volume serial number (VOLSER) that may be used by applications to access the device. The VOLSER is written on the volume as record 3 on track 0 cylinder 0. The value for the VOLSER is kept in the UCB for online volumes.

Data sets

Data is stored in data sets, logical collections of information. There are many different types of data set organizations that may be selected by the application.

VTOC

The location of data sets on any DASD volume is recorded in a Volume Table of Contents (VTOC). The VTOC records the physical start and end locations of the data set. There is only one VTOC on a volume and a VTOC must exist for data sets to be allocated on the volume.

UCB

z/OS uses a Unit Control Block (UCB) to represent a device. System programmers define I/O devices to the operating system into an I/O Definition File (IODF). During IPL one UCB is built for every I/O device definition found in the active IODF. If a device is not defined to z/OS then a UCB is not built and it is not possible for applications to access the device.

Figure B-2 shows how the components of z/OS I/O connect together to uniquely identify a device.

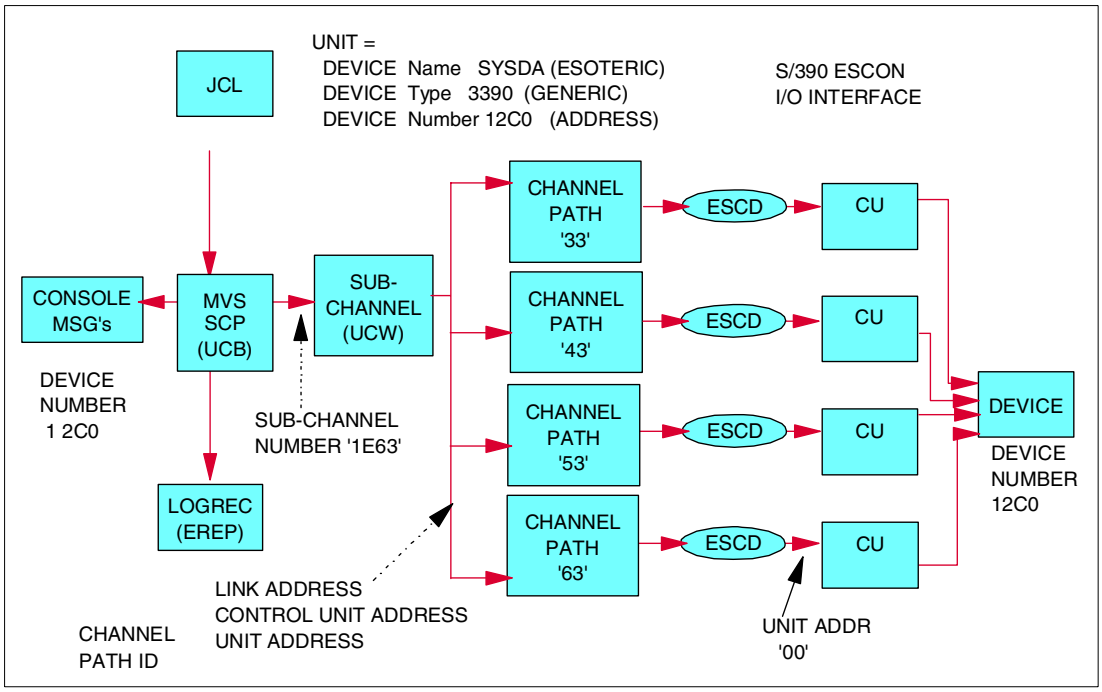


Figure B-2 Logical components of z/OS device addressing

There are at least three ways of uniquely identifying a device.

- Device number (external—operator)
- Subchannel number (internal—MVS <---> CSS)
- CHPID + destination link address (DLA) + logical address (CUA®) + unit address (UA)

Path

This is a route to a device from the UCB through the UCW (in CSS) over a channel through the control unit. Paths are:

- Controlled by operator VARY PATH and CF CHP commands.
- Status displayed in the D M=DEV(dddd) command output.

ESCON ports support up to 64 logical paths. FICON ports support up to 256 logical paths.

Control unit image

As with the link address, the logical control unit image (LCU), also known as the control unit address (CUADD), forms part of the device selection scheme in an ESCON or FICON environment. For the ESS each LCU matches an LSS (for CKD).

The reason for this is to allow the LINK to be shared among the 16 control unit images in an ESS (normally you cannot attach the same link to more than one controller because it is all

point-to-point physical connections). The CUADD statement allows us to define logical control units that now share the same ESCON or FICON links.

z/OS I/O processing flow with ESCON

The steps involved in a z/OS I/O operation using ESCON attachment are summarized in Figure B-3. It is important to understand the components of an I/O operation, because this will be very helpful when doing I/O performance analysis and problem detection.

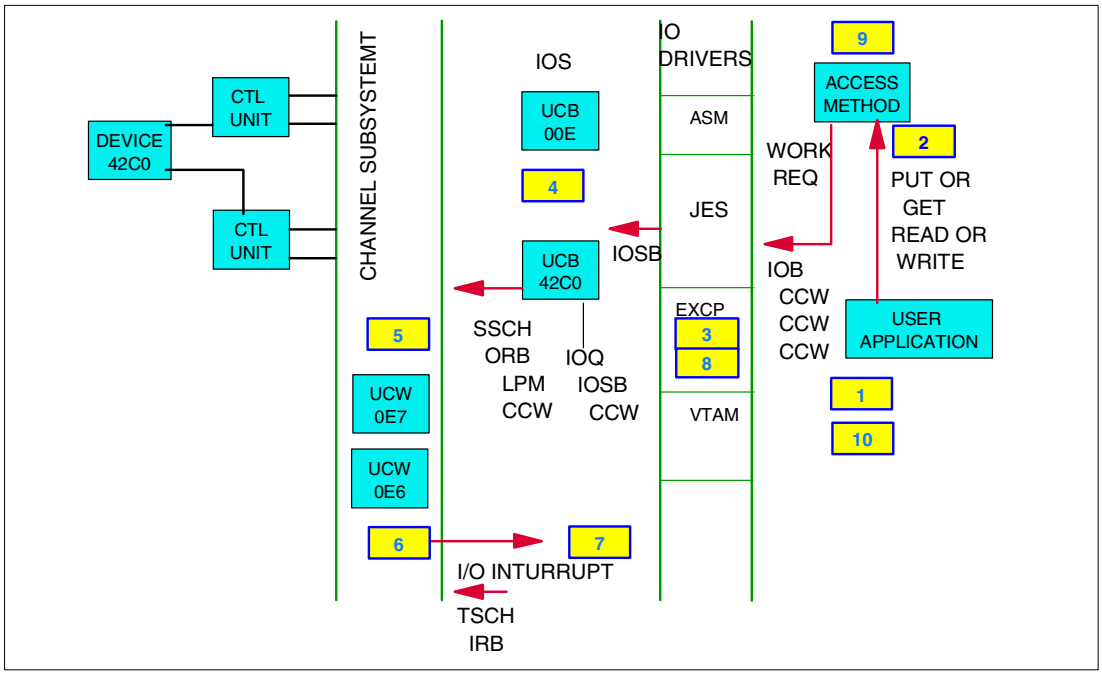


Figure B-3 Logical steps for an z/OS I/O operation

Figure B-3 illustrates the steps involved in a z/OS I/O. The process consists of the following steps:

1. The user program begins an I/O operation by issuing an OPEN macro instruction and requesting either input or output of data using an I/O macro instruction like GET, PUT, READ, or WRITE, and specifying a target I/O device.
2. An I/O macro instruction invokes an access method that interprets the I/O request and determines which system resources are required to satisfy the request. User programs can bypass the access method but then need information about the physical devices they will be accessing. By invoking access methods, user programs are device independent, as the access method deals with the physical device characteristics. z/OS provides numerous access methods. Each offers different functions. The selection of access method is based on the program that intends to access the data (sequential or random, for example).
3. To request movement of data, either the access method or the user program presents information about the operation to the EXCP processor by using the EXCP macro instruction. EXCP:
   - Translates the information, Channel Control Word (CCW) command chain addresses, and CCW data addresses, into a format acceptable by the channel subsystem
   - Fixes the pages containing the CCWs and the data buffers
   - Validates the extents requested

   – Invokes the I/O supervisor (IOS)

4. IOS places the request for I/O on the queue for the chosen I/O device in the UCB and issues the Start Subchannel (SSCH) instruction to the channel subsystem, then z/OS continues with other work until the channel subsystems indicate with an I/O interrupt that the I/O operation has completed.
5. The channel subsystem selects a channel path to initiate the I/O operation between channel and control unit or device and controls the movement of data between the channel and central storage.
6. When the I/O operation is complete, the channel subsystem signals I/O completion by generating an I/O interrupt.
7. IOS processes the interrupt by determining the status of the I/O operation, successful or otherwise, from the channel subsystem.
8. EXCP indicates that the I/O is complete by posting the access method and calling the dispatcher.
9. When appropriate, the dispatcher reactivates the access method.
10. The access method returns control to the user program, which can then continue its processing.

For a more detailed description of the processes involved in I/O operations, you can refer to *z/Architecture™ Principles of Operation*, SA22-7832.

z/OS is an interrupt driven operating system, as opposed to polling-driven operating systems. z/OS passes requests to perform I/O to the CSS and then continues processing other work until one of the following occurs:

- CSS interrupts z/OS to indicate that an I/O request is finished.
- The missing interrupt handler (MIH) alerts z/OS that the device has not responded within a user-determined period.

z/OS records the times at most of these steps by writing them into the Systems Measurement Facility (SMF) records. SMF records can be formatted and reported by programs such as Resource Measurement Facility (RMF), and also the data can be consolidated using programs like Performance Reporter for OS/390.

FICON

FICON brings some changes in the logical steps of an I/O operation. These changes result from the following basic differences between FICON and ESCON:

- While ESCON can perform only one I/O operation on a channel, with FICON multiple I/O operations can occur concurrently over a FICON link. This is made possible by the pipelining and frame multiplexing characteristics of FICON.
- With FICON the Director port busy time is eliminated, and the pending time is reduced.

UNIX and Windows NT terminology

In UNIX and Windows NT we usually are not able to monitor the actual disk I/O operation as granularly as with S/390. In general, the way in which UNIX and Windows servers access the hard disk drives is determined by the device driver for that specific device.

When it comes to disk or file access times, the only access times available from the operating systems are usually a mix of minimum, maximum, and average access times normally

displayed in milliseconds. Generally speaking, the utilities provided as standard by the operating systems are not normally able to break down an I/O operation into more distinct parts.

As the disk I/O mechanism is different between the Windows and UNIX operating systems, we discuss them separately.

UNIX disk I/O operation

In the case of UNIX, when accessing the physical disk the total response time is measured in milliseconds, and is made up of the following components:

- System call

  A request made to the operating system by an application for I/O.

- Logical file system

  Provides a consistent programming interface to applications through the system call interface. It is the logical file system layer, which then interprets the request and passes it onto the correct physical file system.

- Physical file system

  The actual implementation of the file system that is supported by the operating system. For example. Journal File System (JFS), Network File System (NFS), or CD-ROM File System (CDRFS).

- Device driver

  The actual device driver code that interfaces with the device. It is invoked when file system implementation code maps the opened file to kernel memory and reads the mapped memory.

The total time for the request in milliseconds is started when the system call is made, and ends when the system call is notified that the data is available.

The above components are a simplified breakdown of what actually happens in the kernel when a request is made. Figure B-4 on page 458 is a representation of the flow of an I/O operation through the various kernel layers in AIX.

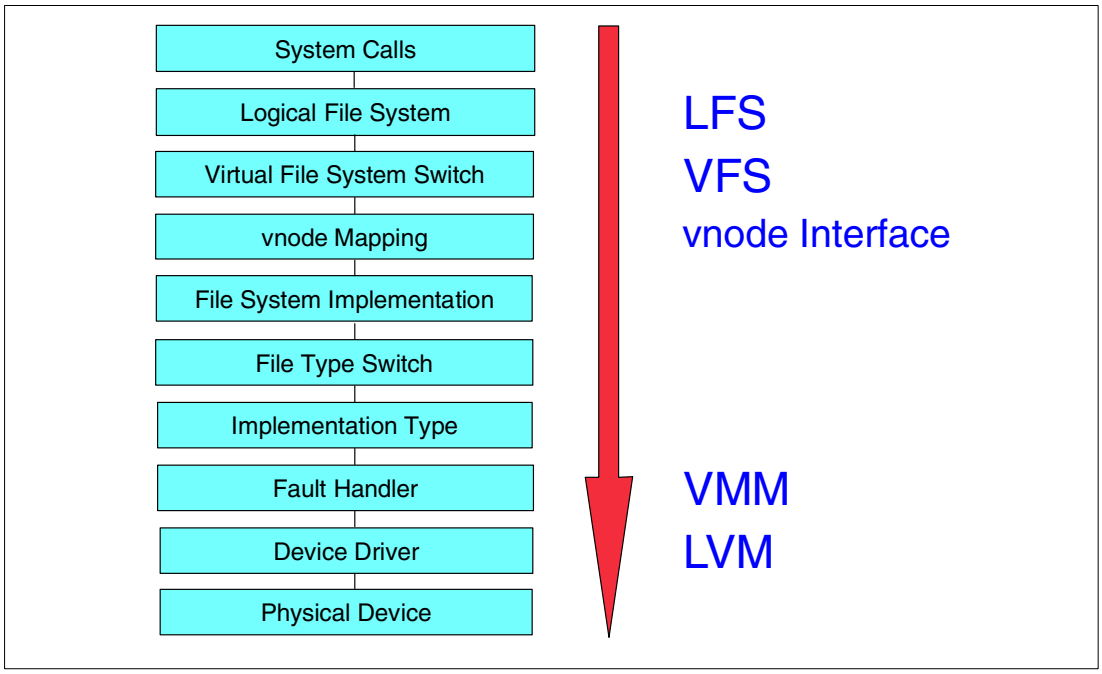


Figure B-4 AIX kernel I/O layers

The layers shown in Figure B-4 are:

Logical file system	Standard set of operations to support the system call interface
Virtual file system	Standard set of operations on the entire file system
vnode interface	Standard set of operations within the file system
File system implementation	Support of the individual file system layout
File type switch	Layer in some file system implementations that has different branches for the file type (regular file, directory, pipe, etc.)
Fault handler	Device fault handler support, provided by the AIX Virtual Memory Manager when a page of the file is not resident in memory
Device driver	Actual device driver code to interface with the device

Windows disk I/O operation

Disk I/O in Windows NT is handled by the Windows NT executive; this is the kernel-mode portion of Windows NT. The I/O manager is the portion of the Windows NT executive that handles file and network processing.

Under Windows NT, disks are partitioned to form file systems, which can be of two different types. Using the disk administrator, Windows NT can create partitions with the following types of file systems:

- The Windows NT File System (NTFS) is an advanced file system that supports journaling, extremely large storage media, and long file names. NTFS provides the highest level of security and file recovery, and is the file system of choice on Windows NT.

- The File Allocation Table (FAT) is the file system used by the MS-DOS and Windows operating systems. FAT provides support for long file names but does not provide security.

The following elements of the Windows NT Executive are used in performing an I/O operation.

- I/O manager

  The I/O manager is the part of the Windows NT Executive that manages all input and output for the operating system.

- Virtual memory manager

  The function of the virtual memory manager is to provide asynchronous I/O and mapped file I/O.

- Cache manager

  Performs file caching in Windows NT. While most caching systems allocate a fixed number of bytes for caching files in memory, the Windows NT cache dynamically changes size depending on how much memory is available.

- File system interface

  Implements a well-defined formal interface that allows it to communicate with all device drivers in the same way, without any knowledge of how the underlying devices actually work.

- Device driver

  This is the device-specific code that works in association with the Hardware Abstraction Layer (HAL) of the Windows NT Executive, to perform the disk I/O operation.

Figure B-5 is a representation of the Windows NT Executive and its components.

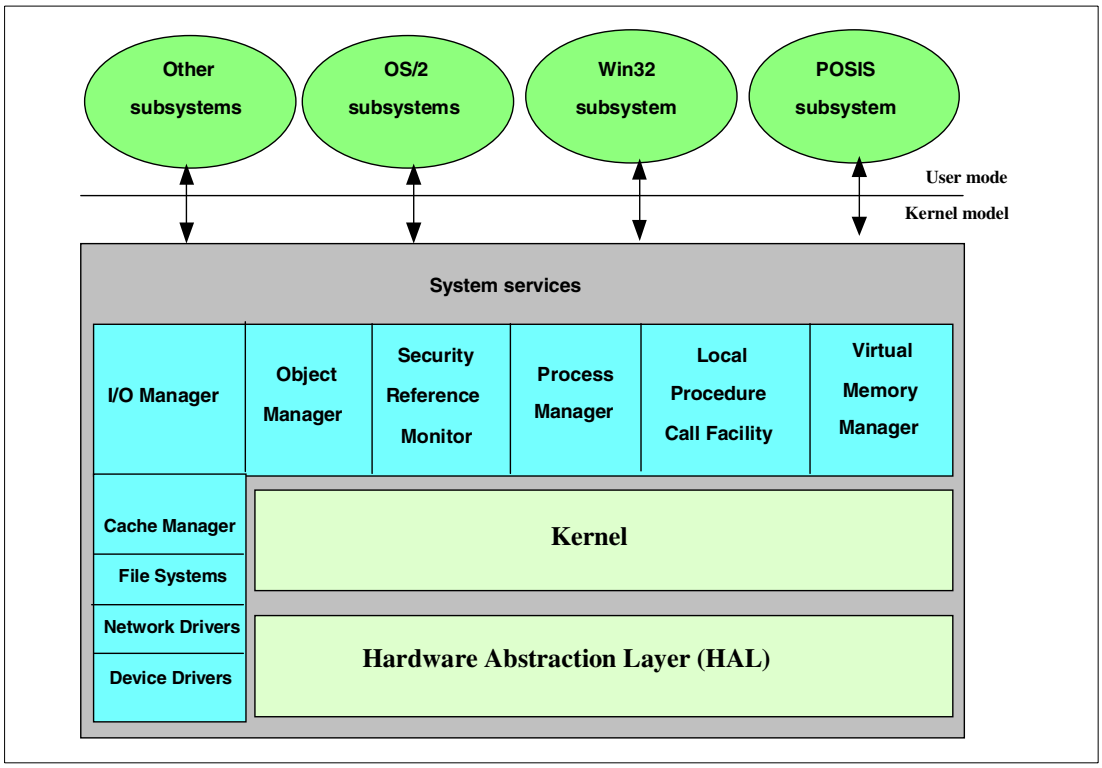


Figure B-5 The Windows NT Executive and its components

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this publication.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 463. Note that some of the documents referenced here may be available in softcopy only.

- *IBM TotalStorage Enterprise Storage Server Model 800,* SG24-6424
- *IBM TotalStorage Enterprise Storage Server: Implementing the ESS in Your Environment,* SG24-5420
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services on S/390,* SG24-5680
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services on UNIX and Windows NT/2000,* SG24-5757
- *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287
- *IBM TotalStorage Expert Hands-on Usage Guide*, SG24-6102
- *Implementing Fibre Channel Attachment on the ESS*, SG24-6113
- *Implementing Linux with IBM Disk Storage*, SG24-6261

Other publications

These publications are also relevant as further information sources:

- *IBM TotalStorage Enterprise Storage Server Host System Attachment Guide,* SC26-7446
- *IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide,* GC26-7444
- *IBM TotalStorage Subsystem Device Driver User’s Guide,* SC26-7478
- *z/OS MVS System Commands*, SA22-7627
- *Administration Guide of DB2 for OS/390*, SC26-8957
- *DB2 for OS/390 Administration Guide*, SC26-8952
- *z/OS V1R4.0 RMF Report Analysis,* SC33-7991
- *z/OS RMF Performance Management Guide*, SC33-7992
- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *z/Architecture Principles of Operation*, SA22-7832

Online resources

These Web sites and URLs are also relevant as further information sources:

- 30-day trial version of the latest VTune client software from Intel

http://developer.intel.com/software/products/eval/bonnie++/readme.html

- Complete list of the ESS fabric support

  http://www.storage.ibm.com/disk/ess/supserver.htm

- Current list of Solaris-SPARC patches and Solaris-x86 patch for Solaris 2.6, Solaris 7, and Solaris 8

  http://java.sun.com/j2se/1.3/install-solaris-patches.html

- Description and documentation on Bonnie

  http://www.coker.com.au/

- Description of the IBM TotalStorage SAN products:

  http://www.storage.ibm.com/ibmsan/products/sanfabric.html

- Discussion on how to use the LVM tool

  http://tldp.org/HOWTO/LVM-HOWTO/index.html

- Download device drivers for Sun Solaris servers

  http://wwws.sun.com/software/download/

- Enterprise Storage Server

  http://www.storage.ibm.com/hardsoft/products/ess/pdf/1012-01.pdf

- ESS/DFSMS SDM Copy Services

  http://www.storage.ibm.com/software/sms/sdm.index.html

- ESS support information

  http://www.storage.ibm.com/hardsoft/products/ess/supserver.htm

- ESS Web site

  http://www.storage.ibm.com/hardsoft/products/ess/ess.htm

- HP-UX servers device drivers download

  http://www.hp.com/country/us/eng/support.html

- Information about Iometer

  http://developer.intel.com/design/servers/devtools/iometer/

- Information on GKrellM

  http://web.wt.net/~billw/gkrellm/gkrellm.html

- Information on SUN Solaris commands and tuning options

  http://sunsolve.sun.com/handbook_pub/
  http://www.sun.com/bigadmin/collections/performance.html
  http://www.context-switch.com/reference/exscripts/perform/diskstat

- KSysguard is part of the KDE project and information and updates can be obtained at:

  http://www.kde.org

- Latest levels of the SDD and ESS utilities (ESSUTIL)

  http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105

- List of AS/400 and iSeries models to which you can attach an ESS

  http://ssddom02.storage.ibm.com/techsup/webnav.nsf/support/2105

- List of tunable kernel parameters for HP-UX Release 11i

  http://docs.hp.com/hpux/onlinedocs/TKP-90202/TKP-90202.htmlbufpages

- LVM for Linux downloaded

http://www.sistina.com/products_lvm_download.htm

- Microcode levels for RS/6000 and pSeries servers and adapters

  http://techsupport.services.ibm.com/server/mdownload

- **nmon** tool for AIX servers

  http://www-1.ibm.com/servers/esdd/articles/analyze_aix/index.html

- The most current list of Intel-based servers that can attach to the ESS

  http://www.storage.ibm.com/disk/ess/supserver_summary_open.html

- VTune NLM downloaded

  http://developer.novell.com/support/sample/tids/topt2/topt2.htm

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Index

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

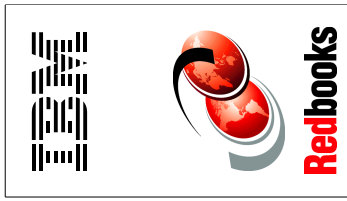
U

V

W

X

Z



IBM TotalStorage Enterprise Storage Server Model 800 Performance Monitoring and Tuning Guide

IBM TotalStorage Enterprise Storage Server Model 800 Performance Monitoring and Tuning Guide

Efficiently using ESS Model 800 capabilities

Optimizing performance in the ESS

Monitoring I/O processing in the ESS

This IBM Redbook provides guidance on the best way to configure, monitor, and manage your IBM TotalStorage Enterprise Storage Server (ESS) to achieve optimum performance. The information presented in this publication applies mainly to the ESS Model 800, but many of the discussions and recommendations can also be considered with previous F models.

This publication describes the ESS Model 800 performance features and characteristics and how they can be exploited with the different server platforms that can attach to the ESS. Then in consecutive chapters it goes over the specific performance recommendations and discussions that apply for each server environment, as well as for database and ESS Copy Services environments.

Also outlined in this publication are the various tools available for monitoring and measuring I/O performance for the different server environments, as well as monitoring performance of the entire ESS subsystem.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6422-00 ISBN 0738453242