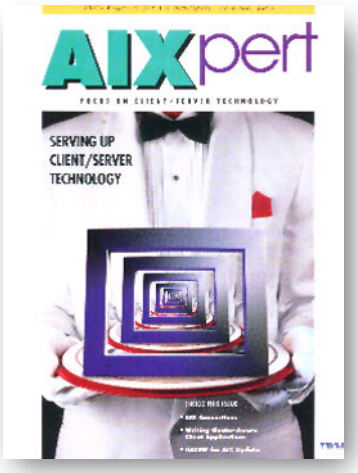
AIXpert DIGITAL VERSION IBM

TABLE OF CONTENTS



Commentary

Serving Up Client/Server
By George Noren

Client/Server

AIX Connections
By Bret R. Olszewski and Kay Chang

DCE Cell Performance: High Water Marks
By Bob Russell

Writing Cluster-Aware Client Applications
By Steven Kohler and Thomas Casey

PDM Implementation Framework
By Eddie Ho, Peter Stoll, and Eric Dunn

DCE DFS Interoperability in Data Sharing Environments
By Jean Pehkonen

AIX

HACMP for AIX: Version 4.1.1 Update
By Daniel P. Cox

Using PCMS to Control AIX Software Development
By Sohail Haque

Support

IBM's Porting Center of the East
By Valerie Paul

Flowmark—A Workflow Manager
By Carolyn Cummiskey

Communications

Global Networking Using X.25
By Eddie Ho and John Ellis

Q&A

AIX Questions
Compiled by Daryl Green

NOVEMBER
1995

AIXpert BACK TO MAIN MENU

Serving Up Client/Server

You have heard a lot about Client/Server as a way to structure functionality on a network. Perhaps your head is a-buzz with ideas that have no real grounding in how you put this Client/Server stuff to work. We have taken the challenge in this issue to "put a face" on Client/Server—to show some new products in that arena, and to discuss new aspects of existing products. If you are wondering where to go from now until Network Centric Computing becomes a reality, this issue will help give you some ideas.

With Version 4.1.4 of AIX comes a new product that helps to integrate more than 10,000 AIX solutions with both IBM-compatible and Macintosh personal computers. Read about AIX Connections to find out how this product puts the power of an AIX server on every client screen. This issue brings two treatments of an essential technology for client/server computing: Distributed Computing Environment (DCE). For pointers about laying out your client/server installation using DCE, be sure to see the DCE Cell Performance article. To explore putting DCE into a heterogeneous environment that includes other technologies (such as NFS and AFS), read the DCE DFS Interoperability article. For a successful installation from the real-world, check out "PDM Implementation Framework," which explains how IBM is using networks to solve complex data management problems.

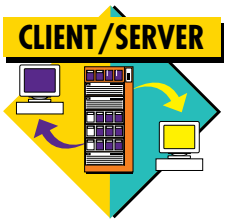
Client/server also highlights the problem of keeping the network and servers running dependably. This issue gives you two articles that address a solution to that problem: High Availability. Read about the new version of HACMP to learn of the latest improvements in that product. A second HACMP article tells you how to write cluster-aware client applications. To extend your networking into a global arena, read about X.25, a globally available networking technology.

Solution developers should see the article about FlowMark to learn of an opportunity to join the Flowmark Partners in Development program. Also read about the porting facilities that are now available on the East Coast at the Solution Partnership Center—East. An article about a Configuration Management Control System (PCMS) and the popular AIX Questions section complete the issue.

George Noren

George Noren, IBM Corporation, Internal Zip 4103, 11400 Burnet Road, Austin, TX 78758. Internet: geo@austin.ibm.com. Since joining IBM in September 1979, Mr. Noren has written manuals for System/34, System/36™, and AIX on both the RT® and RISC System/6000® platforms, and was a member of the InfoExplorer™ design team. He has also worked as system administrator for several AIX server machines and their clients, and is currently responsible for the Prototype Evaluation Labs in Austin. Mr. Noren studied engineering at Illinois Institute of Technology, holds a BA in English from the University of Minnesota and an MBA from St. Edwards University in Austin.

George Noren

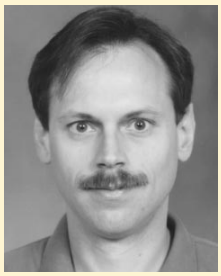


AIX Connections

By Bret R. Olszewski and Kay Chang

AIX Connections has established a new standard in UNIX-to-PC connectivity. The combination of services, with product-level robustness and performance, should prove attractive to companies seeking to integrate AIX® applications with PC clients. This article discusses the architecture and performance characteristics of AIX Connections.

Today's complex business environments have many computers interlinked to enterprise data—and often the Internet. Frequently, many different types of computers, both clients and servers, are connected by a variety of protocols and network hardware. This potpourri increasingly challenges those employed to maintain order.



Bret R. Olszewski

Kay Chang

AIX Connections Overview

AIX Connections is a feature of AIX 4.1.4 that was implemented with a simple goal: to provide the premier solution for integrating IBM-compatible and Macintosh® personal computers with UNIX® solutions. Its benefits include access to AIX scalability, consolidated system management, and ease of cooperation and integration with more than 10,000 AIX solutions.

AIX provides unmatched scalability, encompassing hardware from notebook computers to massively parallel systems. Since this is done with binary compatibility, growth is virtually unlimited. With AIX Connections as a PC server, networked PCs can use all of the robust AIX features. For example, printing to a high-speed laser printer can be as simple as redirecting your LPTx.

One thing is certain in the world of PCs—they are all different. Network operating systems implement incompatible protocols, administration, and filesystem structure. Using network operating systems to configure and administer multiple servers is complex and expensive.

But consolidating onto a single AIX Connections solution allows files and printers to be shared by different types of networked PCs. When one user generates a file on a Macintosh, it can be used by another user on a PC running Microsoft® Windows™ 3.1—transparent to both users. This works because AIX's Journaled File System is used directly and without abstraction, allowing files to be controlled through AIX's security apparatus for consistent use by all services. User administration, which is also consistent with AIX, does not require redundant updates as some other solutions.

Since all AIX Connections are implemented consistently, there are no incompatibilities with other AIX solutions. Files and printers are shared. Protocol stacks allow AIX Connections to coexist on the network with other solutions (such as databases) as well as customer-written applications, shown in Figure 1.

AIX Connections Components

AIX Connections includes the following elements:

LSserver: LSserver consists of network protocols, such as NetBEUI, TCP/IP, Server Message Block (SMB), and file, print, and terminal emulation services. LSserver provides services to PC clients running OS/2®, DOS, Windows 3.1, Windows 95, Windows for Workgroups, and Windows NT™ workstations. SMBs are the basic units of the interface, originally developed by IBM, Intel®, and Microsoft, then standardized by X/Open™. AIX Connections provides NetBIOS Control Block (NCB), X/Open Transport Interface (XTI), and Transport Library Interface (TLI)

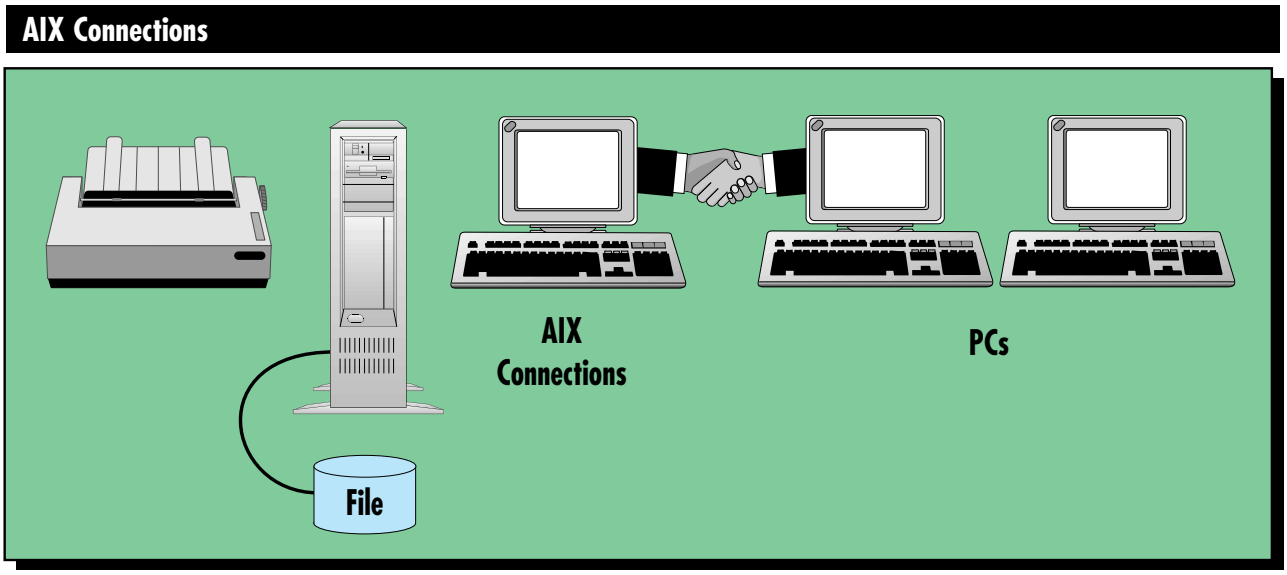


Figure 1. AIX Connections

libraries for program interfaces that map into SMB protocols.

NWserver: NWserver consists of network protocols, such as IPX/SPX, and provides file, print, and terminal emulation services. NWserver provides services to PC clients running OS/2, DOS, Windows 3.1, Windows 95, Windows for Workgroups, and Windows NT workstations. AIX Connections provides XTI and TLI libraries for programming interfaces that are mapped into NetWare Core Protocol (NCP) protocols for file and print.

MACserver: MACserver consists of Apple® Macintosh network protocols such as AppleTalk® Data Stream Protocol (ADSP) and AppleTalk Transaction Protocol (ATP). Apple File Protocol (AFP) and Apple Print Protocol are both based on ATP. MACserver provides file and print services.

TNclient: TNclient gives an AIX host the opportunity to access either SMB or NetWare® network operating systems as a client. For example, an AIX system can mount a filesystem that is actually served from another network operating system. Available services include the following:

- Protocols and requester to SMB servers, such as OS/2 LAN Server, Windows NT Advanced Server, and Microsoft LAN Manager for logon, filesystem mounts, and printer services
- Protocols and requester to NetWare servers, such as the Novell® NetWare Server and NetWare Server for AIX for logon, filesystem mounts, and printer services

The combination of these elements, along with the extensive TCP/IP-based services of AIX, present a wide-ranging solution to the "tower of babel" computing environment we face today.

Architecture

Figure 2 shows the internal architecture of AIX Connections. The diagram shows three sets of entities: protocol services, directory services, and daemons. Interlinked with each of those are AIX services—some explicitly stated such as sockets, and some implicitly invoked, like the filesystem. The architecture effectively consists of four parts:

- **AIX services:** The architecture focuses on using as much of the capability provided by AIX as possible. Instead of creating unique functions, AIX Connections depends on AIX services, such as Data Link Provider Interface (DLPI), Streams, TLI, and security.
- **Communication protocols:** AIX Connections requires that various communication protocol stacks, such as TCP/IP, IPX, and NetBIOS, be implemented.
- **Control processes for file, print, and TTY:** Daemons provide support for service requests initiated by clients via protocols.
- **Programming interfaces:** Control processes use these interfaces for service. Interfaces are also provided for customers to write applications.

The description below illustrates how these components are used by AIX Connections.

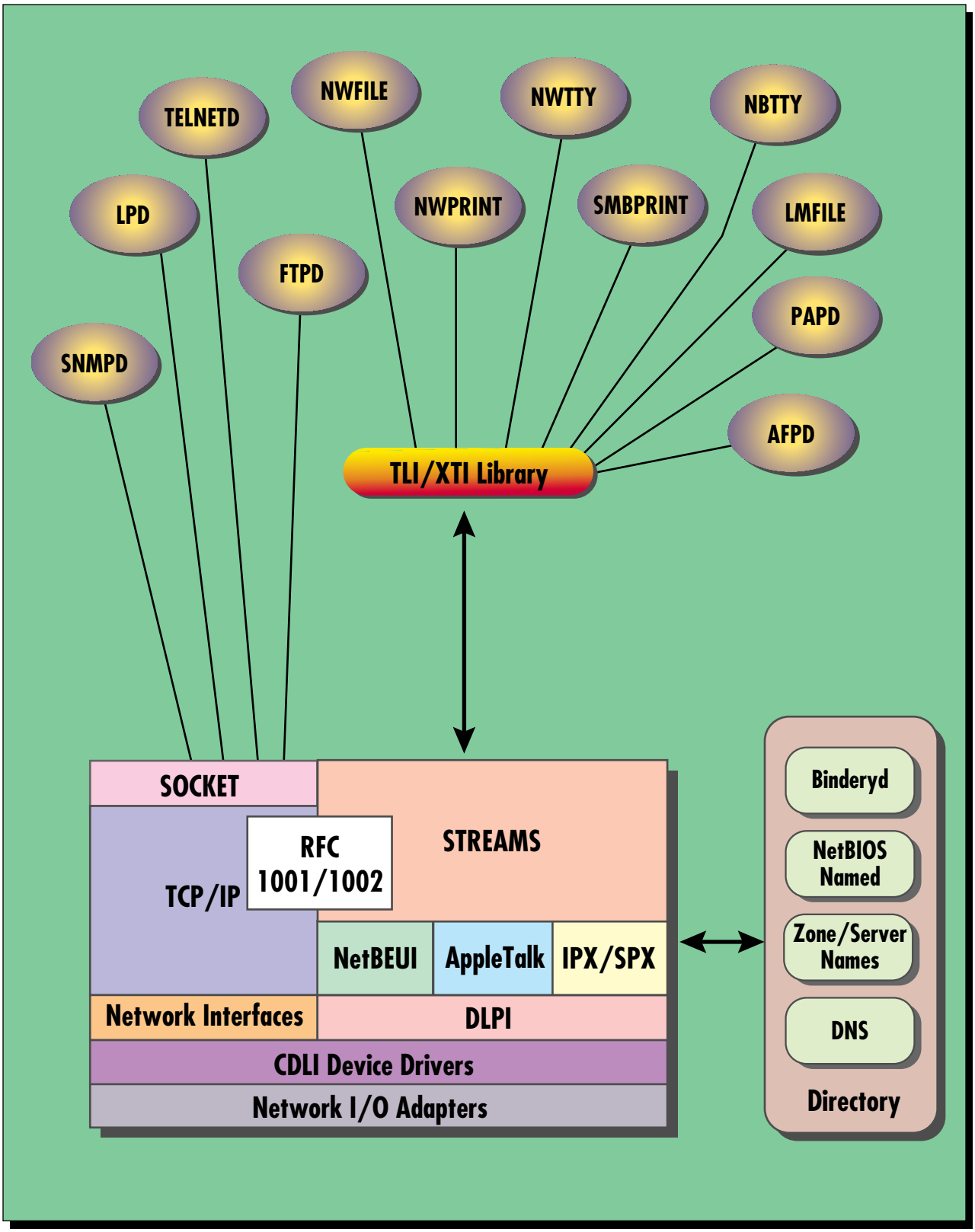


Figure 2. Internal Architecture of AIX Connections

Base AIX Facilities

AIX Connections uses various base I/O services and lower-level protocols, including LAN network device drivers, network interface for TCP/IP, and data link protocol drivers such as DLPI.

The DLPI driver provides two main functions:

- To bridge between network device drivers below and STREAMS protocol drivers above
- To process primitives for implementing IEEE® 802.2 services

It uses both connectionless and connection-oriented services. STREAMS drivers, which require connection-oriented services such as NetBEUI, must depend on the connection-oriented features. Others such as Datagram Delivery Protocol (DDP) and IPX use only the connectionless capability.

Network and Transport Protocols

AIX Connections protocols such as NetBEUI, AppleTalk, and IPX/SPX are implemented primarily via Streams. NetBEUI, which depends primarily on connection-oriented services, ties its session map to the connection-oriented DLPI state, where NetBEUI sessions are brought down when the link-level connections are disconnected.

AppleTalk Datagram Delivery Protocol (ADP), the connectionless service, assumes the logical network connection is always present. AppleTalk Session Protocol (ASP) provides the transport connection for end-to-end file service.

IPX, which originated from the XNS protocol invented by Novell, is connectionless, whereas SPX is connection-oriented. The SPX protocol is implemented by a STREAMS module pushed on top of the IPX driver.

RFC1001/1002, somewhat like TP0 of Open System Interconnection (OSI), implements a null layer using TCP/IP as a transport. This exists primarily for interpreting a mapping that is defined by a NetBIOS service to the TCP/IP service and protocol.

File Services

LMfile, a component of the LSserver, provides a file access service communication over SMB protocols. It requires NetBEUI or TCP/IP transport that is accessible via Transport Library Interface.

NWfile, a component of the NWserver, provides file access service communication over NCP protocols. It requires IPX access via TLI.

AFPD, a component of the MACserver, provides file access service using Apple File Protocol. It requires the Apple Session Protocol via TLI. The File Services components (LMfile, NWfile, and AFP) not only provide file access to the PC clients, but also perform necessary user and group security. They all support the file security scheme used by AIX called Access Control List (ACL).

Print Services

SMBprint provides an interface to the AIX spooler by catching the client's Print SMB requests. It then converts and routes them to the AIX print queue.

NWprint provides an interface to the AIX spooler for catching the client's print NCP requests, then converting and routing them to the AIX print queue.

Papserver provides services that handle print requests from the Macintosh clients and send them to the AIX printer queue.

Terminal Emulation Services

NBtty uses the SMB protocol to provide pseudo-TTY emulation and login service. NWtty is a terminal emulation similar to Novell's Network Virtual Terminal (NVT) function. No terminal emulation for the Macintosh is provided.

Directory Services

Because of the evolutionary nature of various PC protocols, directory services such as naming conventions differ. PC clients expect file names and directory structures on the server to be exactly like their local filesystems. AIX Connections directory services must include the flexibility to handle the PC client assumptions, which results in various ways of handling different namespaces and conventions.

AIX is the common denominator underneath these various resource names. A server name is restricted to a convention, which enables any PC client to read names and understand them. File length can be restricted, or users and group names can be limited. The fundamental underlying element of namespace and security is the AIX user; each PC client session is mapped to an AIX user. This mapping of disparate features of various network operating systems into AIX conventions helps to achieve interoperability.

Performance Aspects

Having explored the extensive capabilities of the architecture, let's explore how it works and performs in a practical setting.

AIX Connections provides the premier solution for integrating IBM-compatible and Macintosh personal computers with UNIX solutions.

About BAPCo

Business Applications Performance Corporation (BAPCo) is a non-profit corporation that develops and distributes a set of objective performance benchmarks based on popular software applications and industry-standard operating systems. It was founded to foster the creation of meaningful benchmarks in an applications environment.

BAPCo benchmarks are unique because they are based on software packages commonly found in retail computer software stores. Using benchmarks based on business applications enables users to conduct evaluations of system handling realistic workloads in environments users might often encounter. This approach provides meaningful data for evaluating systems because the results are relevant to demands they would typically place on systems through day-to-day workloads.

Membership

Membership is open to any organization interested in contributing to the corporation's goals and purposes, while adhering to the corporation's code of conduct. Current members include companies such as Compaq®, Dell®, Hewlett-Packard®, IBM, Intel, InfoWorld®, Microsoft, AT&T® Global Information Solutions®, DEC®, and Unisys®, plus many more. Additional members are invited to contribute to the development of the benchmarks.

Conducting Benchmarks

Since they do not have a lab facility, BAPCo has designed benchmarks so that testing can be done on a company's own systems, following the standards set by BAPCo. BAPCo's method of designing benchmarks focuses on an actual PC user's level of usage. All participating members in the benchmark development groups must follow this guideline. Participation in these groups is open and encouraged. Each participating company receives one vote on the developed benchmark to ensure objectivity.

BAPCo will provide large, long-running, realistic scripts and data that test a wide range of applications and system resources. BAPCo's Workload Characterization methodology is based on modeling user activities at the functional level. The mode is subsequently used to develop scripts similar to the end-users' workloads. To do this, they use the correct functional mix and appropriate data sets (as derived from the models) along with the most popular applications in various categories. This type of characterization produces workloads that closely resemble actual usage patterns of the intended audience.

Benchmark Results

BAPCo will publish its guidelines for executing and reporting benchmark results. It will also clearly define system configurations and formatting and disclose rules for test results.

All BAPCo benchmark licensees (companies that have purchased the benchmark suite) may publish benchmark results obtained on their machines. BAPCo will also publish its own quarterly newsletter, *The BAPCo Report*, which will include informative system-level performance articles and performance benchmark system numbers.

BAPCo Products

SYSmark is a family of benchmark suites from BAPCo continuing from the SYSmark92 and SYSmark93 suite. SYSmark95 for Windows will be used by those interested in evaluating system-level performance of PCs running Windows applications. SYSmark for Windows NT is the industry's first cross-architecture benchmark on Intel, Alpha, and MIPS platforms. SYSmark/FS for Servers is intended as a tool to evaluate the performance of a file server in a network environment servicing requests from network clients.

For more information about BAPCo, call 408-988-7654 or write to:

BAPCo
2200 Mission College Boulevard
M/S RN2-02
Santa Clara, CA 95052-8122

Environments

Knowing the environment or market is key to selecting a benchmark to measure performance. The benchmark must be representative of the work that customers perform, so that the system is tuned to customer requirements.

The traditional environment includes file and print servers that are segmented into three classes:

1. **Servers that provide a wide range of services to PC clients**, including file and print services. Most, if not all, executables reside on the server. This is generally the most performance-sensitive environment.

2. **Servers that provide primarily file access.** These servers function primarily to allow clients to share files. Executables are not usually served and print services are not provided. These servers often support many clients with mission-critical data and have very high reliability requirements.

3. **Servers that provide primarily print services.** Often a site will attempt to centralize print services onto a large server. RISC System/6000® servers are particularly good candidates for such a system.

File Server Benchmarks

Several viable PC-based file server benchmarks exist today. Benchmarking file servers can take place in two dimensions: responsiveness and capacity. A responsiveness benchmark may measure latency of a send/receive request from a client to a server. Another responsiveness benchmark might be to have a client read a file as fast as possible from a server. Both tests are interesting because responsiveness is important for PC productivity applications. Unfortunately, responsiveness tests are frequently not indicative of the capacity of a server since a server may need to service requests from many clients simultaneously. A throughput benchmark is required for capacity measurements.

Three good capacity benchmarks were evaluated for use with AIX Connections: IBM's Claire,

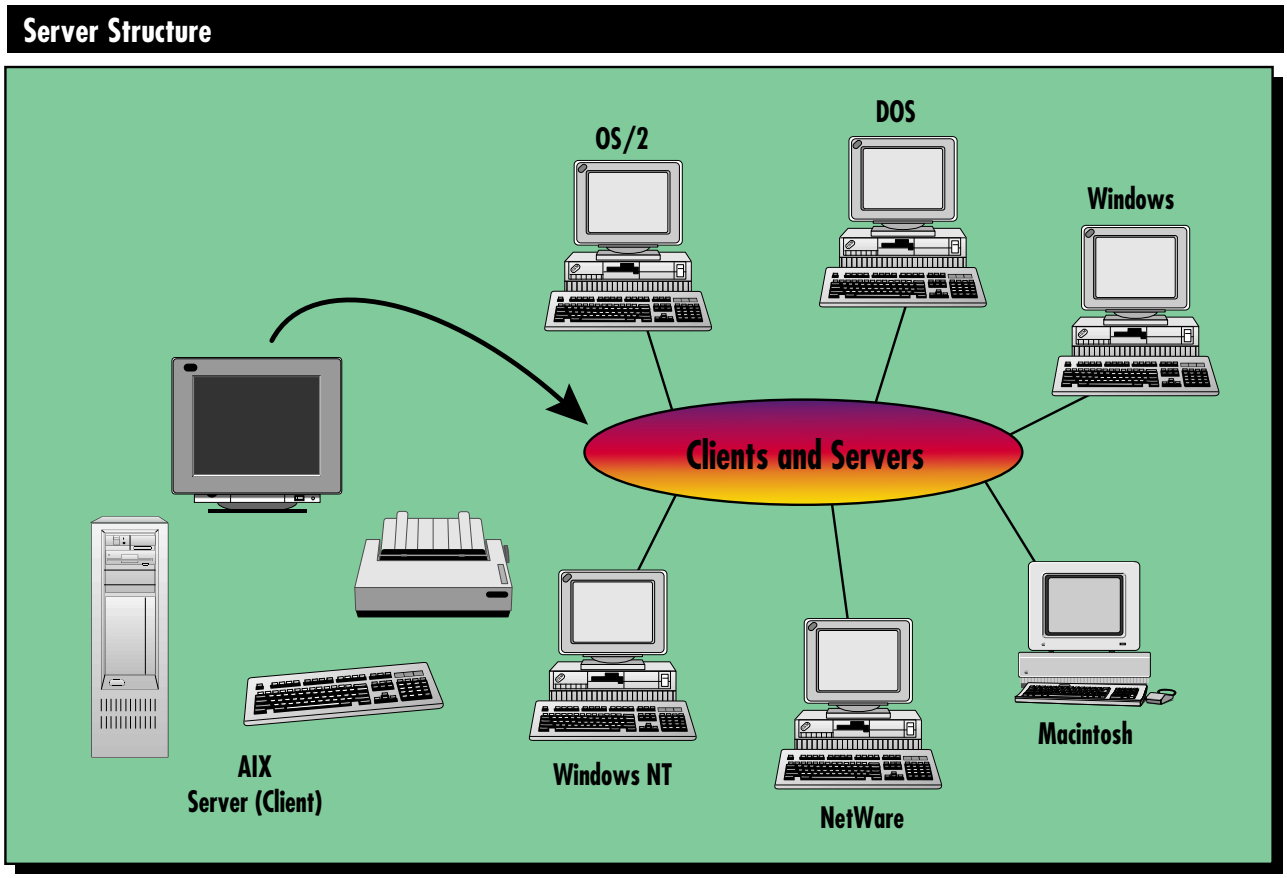


Figure 3. Server Structure

Ziff-Davis' PC Bench 3.0, and the BAPCo file server benchmark. All are based on existing PC applications or their measurements. In this article, when we refer to the BAPCo benchmark, we are referring to the file server benchmark. This is important because BAPCo supports several benchmarks. Currently, the BAPCo file server benchmark is the most recent and sophisticated of those evaluated. Since it is used heavily by the IBM LAN Server team, it was selected for measuring AIX Connections.

As of this writing, the BAPCo benchmark standardization is not complete. Therefore, the results discussed should not be compared with future, authorized results.

BAPCo File Server Benchmark

The BAPCo benchmark is composed of a set of scripted sessions on Lotus 1-2-3® for Windows, Word for Windows, dBASE IV®, Harvard Graphics®, cc:Mail®, Freelance Graphics® for Windows, Paradox® 3.5, WordPerfect® for Windows, and Microsoft Excel. The benchmark, which runs on IBM-compatible personal computers, is currently not applicable to Apple products.

The benchmark is constructed so that each PC client has a directory structure with unique data for the benchmark; the executables are shared on a directory structure. Each client executes the same scripts, but the order is staggered to reduce overlap. So user01 will begin in application Lotus 1-2-3 and end in Microsoft Excel and user02 will begin in Word for Windows and end in Lotus 1-2-3. In spite of the staggering, there are often periods in the benchmark when multiple clients are executing the same application. The DOS executables are local to the client, but the Windows 3.1 executables, drivers, libraries, and configuration files are accessed from the server.

The benchmark includes other common server functions such as printing and locking. Each BAPCo session includes 19 file prints. The dBASE portion of the benchmark includes record locking.

The output metric of BAPCo is scripts per minute—the number of scripts executed (nine per client times the number of clients) divided by the execution time. This is one of BAPCo's major problems: it is not accurate to compare results on dissimilar hardware configurations because client performance matters. Client performance depends on the processor, cache, graphics subsystem, and the efficiency of the network adapters used. Additionally, the amount of memory available on the client can have a significant impact on benchmark results. Response time is not measured.

The benchmark is structured so the PC clients do not emulate user think time or typing-rate delays between key strokes and mouse movements; therefore, each PC runs the applications as fast as it can. This is beneficial to the cost of running the benchmark since fewer PCs are

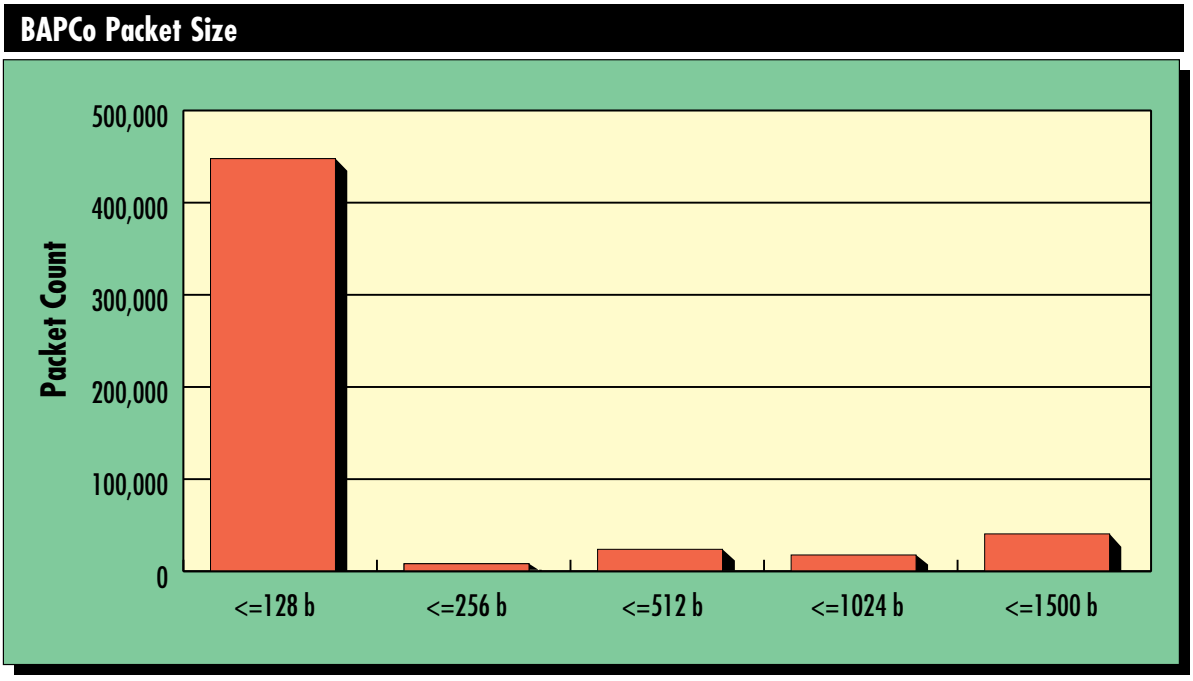


Figure 4. BAPCo packet size measures on NWserver

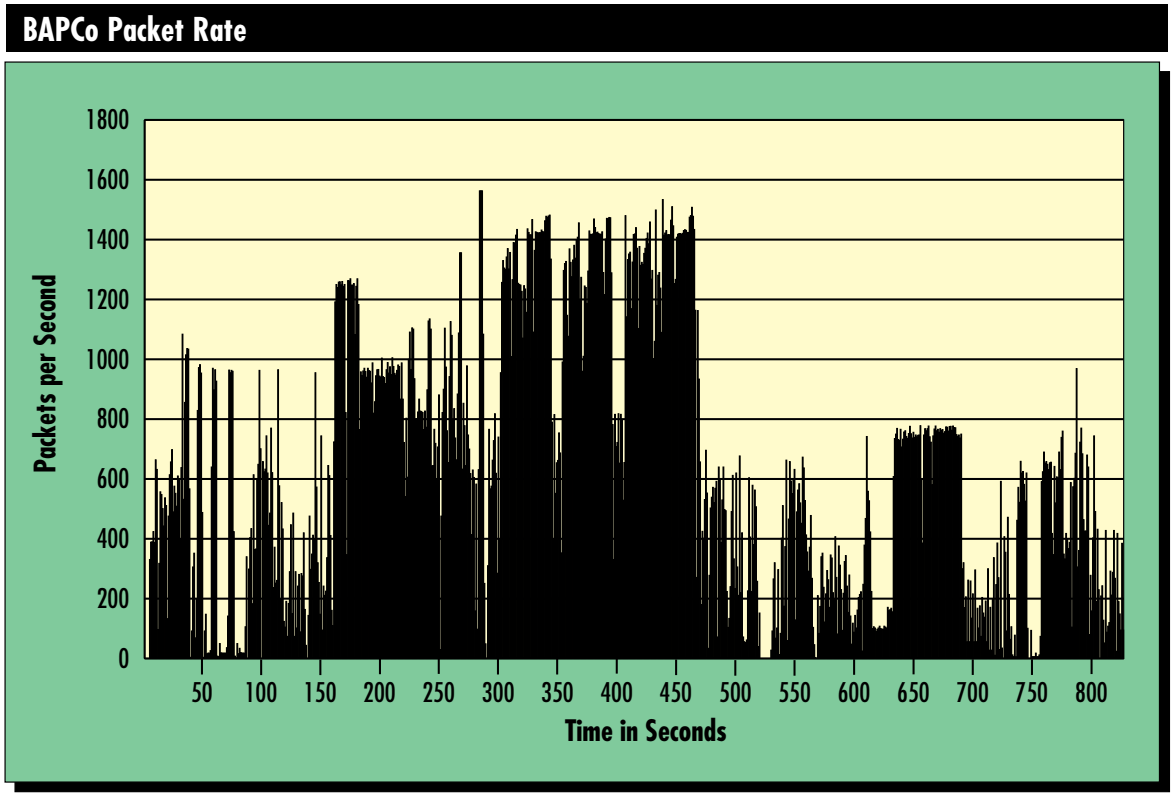


Figure 5. Packet Rate over time for BAPCo using NWserver

needed to saturate a server. It is also a potential problem, since the benchmark does not measure scalability effects that the server might encounter while supporting hundreds or perhaps thousands of real PCs.

BAPCo Benchmark on AIX

The server hardware configuration used for this article included a RISC System/6000 Model E20 (100 MHz 604) with 64 MB of memory, a single 1 GB SCSI disk, and two PCI Ethernet™ adapters. The client configuration included a set of IBM P330 (Pentium™ 90 MHz) systems, each with 8 MB of memory. Clients were uniformly distributed on two Ethernet segments, each segment directly attached to an adapter on the server.

This section will discuss the profile of BAPCo on AIX, using NWserver and NetWare for AIX.

Like most networking benchmarks, the BAPCo benchmark can be viewed as a mix of distinct transactions. Since the transaction types were defined by the communications protocols used, the number of transactions or requests to complete a BAPCo script varies when comparing NetWare for AIX, NWserver, LSserver over NetBEUI, and LSserver over TCP/IP (RFC). At the most atomic level, we can view the number of network packets (it is possible for a protocol transaction to span multiple network packets) as a measure of work. As with NFS, most packets are quite small. Figure 4 shows the distribution of network packet sizes as measured from a single-client execution of the BAPCo benchmark on Ethernet. Remember that Ethernet only allows packets of up to 1500 bytes.

The distribution of network packets over time shows a "bursty" pattern, as expected from end user applications. Figure 5 shows the number of packets sent and received per second during a single-client execution of BAPCo. The area of high packet rates, from approximately 150 seconds to 450 seconds, is where dBASE IV and Harvard Graphics execute. Obviously, these are the most I/O intensive applications in the mix. A single client drives this packet distribution over Ethernet. Since the client in BAPCo does not emulate user think time, it executes as fast as possible. The faster the client—in CPU, graphics, and network adapter efficiency—the more load it can generate on the server. Although fast clients have the horsepower to drive tremendous network load, they rarely do so. Clearly, network bandwidth of faster LANs, such as 100 Mbit Ethernet, are sorely needed as client performance

NetWare Request	Percentage of Total Requests
File Read Requests	77%
File Write Requests	15%
File Open Requests	5%
Directory Requests	3%
Other Requests	<1%

Figure 6. BAPCo script of NetWare request profile

and network-centric applications continue to increase.

Moving further up the food chain of a file server operation, we can measure the basic transaction request types for protocols. Figure 6 shows the NetWare request counts for a single client execution of BAPCo as measured from `sconsole` on NetWare for AIX. Note that the request mix is dominated by file reads and writes.

Figure 7 shows BAPCo's file I/O usage on the server. It indicates the total megabytes of data transferred to and from the filesystem as well as the number of reads and writes for the most frequently used files in the benchmark.

The table data also implies the small sizes of reads and writes used in the benchmark. In particular, consider the 00000008.$$$ file used in Harvard Graphics. The file is opened three times in the benchmark for importing and exporting graphics data. Each file read and write seems to average only eight types. Thus, while the Harvard Graphics script does not move a large amount of data, the CPU consumed is quite high because of the many system calls required to write the data. Since most filesystem reads and writes are cached to memory, the system disk utilization is low.

This BAPCo analysis leads to two conclusions. First, the networked environment is sensitive to the number of clients (and their network bandwidth requirements) on each LAN segment. Normally, it requires at least two Ethernet segments to feed an E20 server. Second, under most circumstances, the CPU will be the limiting factor in server performance. The next section examines how one server actually performs under load testing.

BAPCo Testing on NWserver

This section details measurements performed on an E20 server running NWserver with varying client load. One useful feature of NWserver is that there is no need for complex tuning parameters. In fact, the only two parameters available are the client shell and the enabling or disabling of packet burst on the server (disabled in our case). This makes it less complex to run at high utilization than NetWare for AIX. Under all measures, the I/O wait time was small, even with only a single disk on the server system.

File	MB Transferred	#Filesystem reads	#Filesystem writes	Application
dbase.ovl	9.6	12013	0	dBASE IV
wp.fil	5.7	6343	0	WordPerfect
tmpdnlba.Sdb	4.3	2839	2658	dBASE IV
customer.mdx	1.8	6176	0	dBASE IV
win386.exe	1.6	1683	0	Windows 3.1
paradox.aux	1.4	3601	0	Paradox 3.5
user.exe	1.2	1528	0	Windows 3.1
flwmain.exe	1.1	1522	0	Freelance
customer.dbf	0.9	4740	734	dBASE IV
00000008.$$$	0.7	43522	43522	Harvard Graphics

Figure 7. Filesystem measure of BAPCo with NWserver

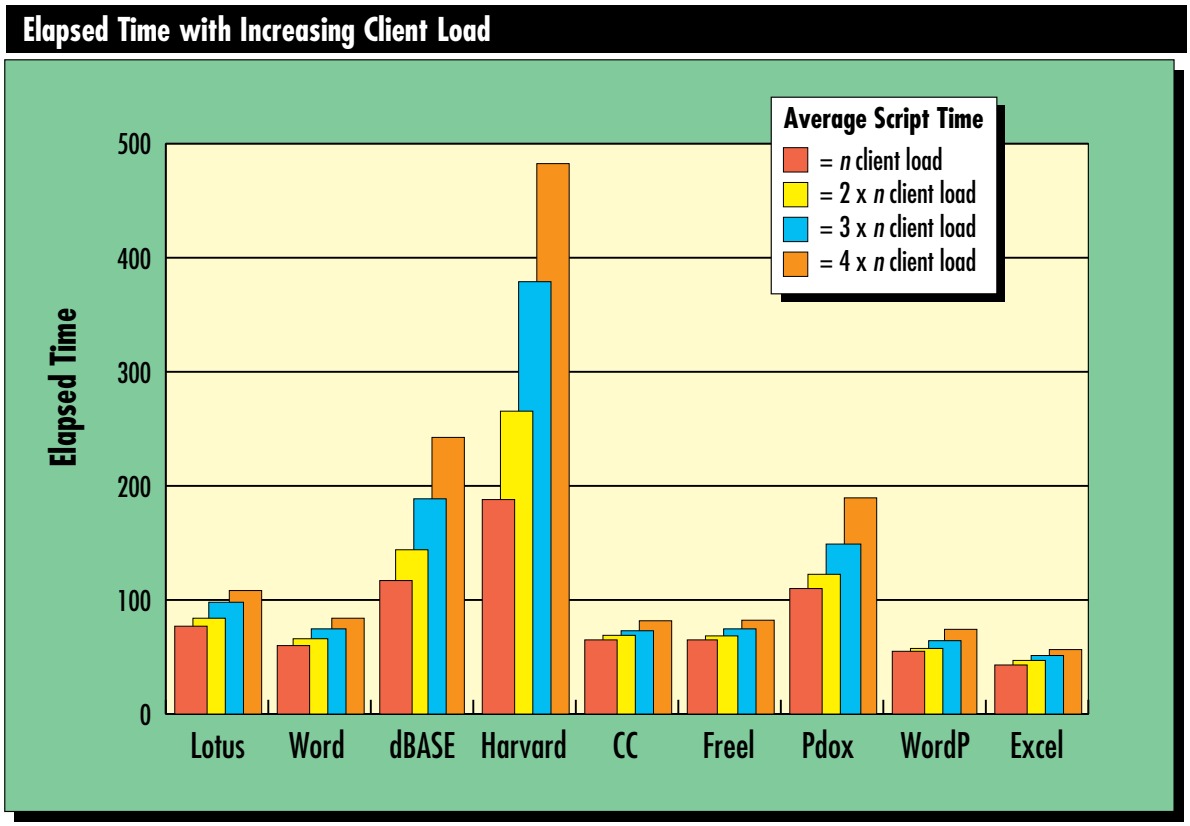


Figure 8. NWserver per script elapsed time with increasing client load

The CPU consumption of the workload is primarily system time, reflecting protocol processing and filesystem activity.

As load on the server is increased by running more clients, the time to complete each script on the client also increases because of queuing for resources, both network and server CPU. Figure 8 shows the increasing elapsed time for the scripts with increasing load.

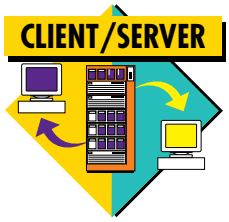
The CPU utilization increases from 41.5% in the lowest case to 96.8% in the heaviest case. The applications that are most I/O intensive, namely dBASE IV, Harvard Graphics, and Paradox 3.5, suffer the most increase in runtime when the server load is increased. Other applications are less sensitive to server load, obviously employing more of their time executing in the client, drawing screens, and doing calculations.

Clearly, NWserver scales well, running user applications up to very high CPU utilization. However, keystroke-intensive and mouse movement-intensive operations do begin to experience occasional noticeable response time delays at high CPU utilization. Typically, you would not want to run a server at such high utilization, but it is good to know that high-load transient behavior can be handled gracefully.

Although it is not evident from the measurements, we have learned a lot by tuning AIX Connections using the BAPCo benchmark. We have improved the NetBIOS/NetBEUI protocol stack, improved STREAMS services, and removed redundant filesystem operations. The combination of these improvements has almost tripled performance improvement over the initial implementation.

Bret R. Olszewski, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Olszewski is a senior programmer working on MP performance. He joined IBM in 1989 and has worked on various aspects of AIX performance. He has a BS in Computer Science from the University of Minnesota.

Kay Chang, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Ms. Chang is an advisory programmer in AIX communication architecture, currently working on AIX Connections. She has an MS in Computer Science from Wright State University in Dayton, Ohio.



DCE Cell Performance: High Water Marks

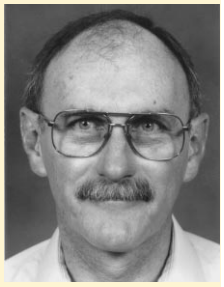
By Bob Russell

Will DCE Security and Cell Directory Services handle the needs of a 10,000-user enterprise? This article discusses the tests and environments designed to address this performance and capacity question. This study has yielded some hardware and configuration high water marks that can be helpful in planning large-scale cell topologies.

Many large customers are preparing to consolidate their corporate networks into a single Distributed Computing Environment (DCE) cell. Others plan to do this under the administration of the IBM Warp Server (WS) using DCE Directory and Security Services (DSS). DSS will be based on the Open Software Foundation® (OSF®) DCE Version 1.1. Both DSS and the DCE-enabled client are in beta test and scheduled to ship in late 1995.

Recently, a large IBM customer, which we will call BigCo, decided to consolidate their enterprise under the IBM WS product. BigCo has a 10,000-user LAN environment. The capacity and performance limitations of the current LAN Server 4.0 (LS4) domain control and administration subsystems preclude expansion to a 10,000-user domain.

In WS, DCE Security and Directory services will replace the LAN Server domain security and directory functions. BigCo's LAN Server 4.0 Domain Control Database (DCDB) and `NET.ACC` will be migrated into the DCE CDS namespace and the DCE Security registry. The existing clients do not need to be upgraded because the replacement of BigCo's current LS4 subsystems will be completely transparent to existing LAN Requesters.



Bob Russell

This article evaluates the capacity and performance of the underlying DCE services used by WS relative to BigCo's planned 10,000-user environment.

Customer Requirements

BigCo, like many enterprises, is preparing to consolidate a large corporate network that consists of several LAN Server domains into a single DCE cell.

The core of BigCo's network is a 16 Mbits-per-second Token-Ring backbone with several local departmental LAN segments. A larger fiber-optic network ties the metropolitan area's LAN segments and several remote offices through 56 KB and T1 Wide Area Networks (WANs).

The heart of BigCo's network is a LAN segment containing their LAN Server domain controllers, database, print, file, and Lotus Notes® servers. BigCo was able to provide transactional and network utilization data about their current server LAN segment, which helped to structure a meaningful workload and lab environment.

BigCo Statistics

The following BigCo statistics are relevant to this study:

- BigCo provided graphs of the logon activity throughout the business day. The aggregation of the peak-hour activity for each of BigCo's five major LAN Server 4.0 domains is 1,534 logons per hour.
- The Sniffer analysis of BigCo's 16 Mbits-per-second server LAN segment shows their network is 38% to 43% utilized, with 1,332 to 1,453 frames per second and 567 to 587

bytes per frame. The network has 918 to 922 active network addresses.

Statistics from other LS4 customers were used to populate the DCE Security registry and CDS namespace. These statistics show an average of 50 users per domain in the LS4 security database (`NET.ACC`). The average population of the Domain Control Database (DCDB) is eight shared resources per domain and one subdirectory per user.

Considering these populations and BigCo's requirements, the initial population of the DCE test cell for each 1,000 users will be 1,000 accounts. DCE Security registry will have 20 groups of principals. The DCE Cell Directory namespace will contain 900 directories and 100 objects.

In reality, a pure WS and DSS environment will have a much different mixture of DCE directories and objects. Each WS domain controller will create only nine CDS directories. Since all aliases, names, and resources will be stored in these directories as DCE objects, these tests were conducted under the worst-case assumptions for memory, disk, and performance.

The DCE configuration program for DCE clients and DSS-enabled clients creates one directory and four objects on behalf of each configured client. Some hints for managing the resources and performance of the `/.:/hosts` directory tree will be discussed in the "DCE Cell Directory Services Performance" section.

Testing Approach

The system was measured in a fully loaded condition, running a modified version of our Point of Sale (POS) benchmark application. The average POS Remote Procedure Call (RPC) data size of 3,250 bytes was reduced so it did not reach the 16 Mbits-per-second Token-Ring capacity at throughput levels consistent with 10,000 users. The size was reduced by eliminating the POS Catalog RPC, which transfers 16,384 bytes of data. Eliminating the catalog also reduced the average number of RPCs per customer sale from 5.5 to 4.5. The data size was reduced to 329 bytes, which is more consistent with BigCo's 567 bytes per frame on the network. The network utilization in our tests ranged between 25% and 40%, which is consistent with the utilization reported by BigCo.

An optional function of the POS benchmark is to periodically perform a DCE Login. For these tests, the frequency was set at one DCE Login for every 15 customer sales distributed randomly. Each time a DCE Login was performed, a different principal name was used from the full population of the DCE Security registry. In a one-hour test of the 90 MHz Pentium DCE Security server, 25,920 DCE Logins were performed with a population of 10,000 principals, thus providing full coverage of the DCE Security registry.

The POS benchmark performs one CDS namespace lookup of a CDS object for every customer sale. Since POS does not perform CDS directory lookups, a script that randomly lists directories and subdirectories from the full CDS population was created to avoid artificially efficient caching by the CDS primary and secondary servers.

Cell Population

In a DCE cell of 10,000 users, the memory and disk sizes of DCE Security and CDS entities become significant. DCE Security accounts and principals require 1 KB each, or 10 MB for 10,000. For these tests, a 32 MB DCE Security server is sufficient—16 MB for OS/2 and DCE, and 10 MB for the registry.

DCE Cell Directory Services (CDS) has two types of entities: objects and directories. Objects—exported programs, aliases, names, and resource definitions—require 1 KB each, or 10 MB for 10,000. Directories require 14.2 KB each, or 142 MB for 10,000.

When CDS is distributed across several CDS replicas[1], the memory requirement for each replica would only need to be enough to contain the directories assigned to that CDS replica. For example, with one CDS primary server and nine CDS secondary servers with 10,000 subdirectories distributed evenly on the 10 replicas, only 14.2 MB would be required for the CDS namespace on each replica.

Since the cell root directory is replicated on all the servers, memory adequate for the root contents would also need to be considered. In these tests, 1,000 CDS objects were added to the root directory, which placed a 1 MB additional requirement on all 10 CDS replicas. For example, 16 MB for OS/2 and DCE, 1 MB for the root directory, and 142 MB for 10,000 directories

[1]A DCE cell has one primary CDS server and can also have one or many secondary servers. CDS primary and secondary servers are called *replicas*. Each replica contains the cell root directory, and may contain all or some of the directories that comprise the total CDS namespace.

equals 159 MB. If the namespace is distributed across 10 replicas, then the 142 MB can be divided by 10; therefore, 14.2 MB plus 17 MB equals a 31.2 MB memory requirement for each replica.

The `/.:/hosts` directory tree created by DCE configuration was moved to a dedicated CDS replica. Because of the available memory on our test hardware, some tests were run with a full population of 10,000, while others were run with a reduced population of 1,000. Sufficient testing was performed to quantify the performance difference between the two population levels and to provide a simple algorithm to bridge between them.

In the tests with a CDS primary and some number of CDS secondary servers, half the number of directories and subdirectories were created on each server, and each server's top directory was replicated on one other CDS server. By having the top directory replicated on another server, DCE automatically maintains its contents on the second replica.

Customer Sales per Minute	RPC Calls per Second	CDS Lookups per Second	DCE Logins per Second	Percent 16 Mbits per Second Network Utilized
8,000	933	139	8.9	80%
7,000	817	117	7.8	70%
6,000	700	100	6.7	60%
5,000	583	83	5.6	50%
4,000	467	67	4.4	40%
3,000	350	50	3.3	30%
2,000	233	33	2.2	20%
1,000	117	17	1.1	10%

Figure 1. Workload conversion table

For example, in the test with one CDS primary and eight secondaries (described later), each CDS replica contains one top directory, `/.:/TestDir0` to `/.:/TestDir8`. Each of these directories contains 500 subdirectories, `/.:/TestDir0/D1000` to `D1499`. Also, each CDS replica contained a read-only copy (replica) of the tree from one other server. Finally, each replica has a copy of the 1,000 objects in the root directory `/.:/TestObject000` to `TestObject999`.

The total CDS population was 1,000 objects and 9,000 directories. Since IBM DCE 1.2 for OS/2 and AIX was used for these tests, replication of the DCE Security server is not yet available. A single DCE Security server was used in all configurations tested. The first release of WS and DSS will support security replication; it is currently available on DCE/6000 for AIX Version 1.3. The first release of WS and DSS will require at least one OS/2 DCE Security replica to support the function required by LS4 legacy servers and clients.[2]

A variety of OS/2, AIX, and Windows DCE clients were used to drive the workload for these tests. A total of 56 physical clients were used, with each client running one or more client processes. There were 200 logical sessions. The combined horsepower of these clients was sufficient to drive about 8,500 POS customer sales per minute. This was enough to fully utilize the lower-horsepower DCE server configurations, but not some of the high-end RISC System/6000 configurations.

Measurement Methodology

The methodology for measuring the performance in the following tests was to measure the system in a busy state. In addition to the metrics shown in Figures 2, 5, 6, and 7, there are other performance factors to consider, which occur while the tests are running. Figure 1 is a conversion table to determine the other workload factors from the metric of interest in Figures 2, 5, 6, and 7.

DCE Security Performance

Three different OS/2-based DCE Security servers were measured: 486-33 MHz, 486-66 MHz, and a Pentium 90. No RISC System/6000 models were used because of insufficient client hardware to drive an interesting workload.

Figure 2 shows the DCE Security server performance, expressed as DCE Logins per second. The measurements were taken in a steady state, while the DCE Security server was nearly 100% utilized. The DCE Logins per second in Figure 2 are based on a registry population of 1,000 accounts and principals. When the population

[2]Initial tests have been conducted of DCE Security replication on the OSF 1.1 base under development. These tests indicate that the scale-up characteristics of DCE Security replication will be similar to the CDS replication characteristics presented in this article.

was increased to 10,000, the throughput decreased by 18%. This suggested a 2% reduction in maximum performance for each additional 1,000 accounts and principals above the base 1,000.

DCE Login and other functions contacting the DCE Security server have two options for obtaining the network address of the DCE Security server. (Figure 3 shows the performance of both options for a 386-25 MHz client.)

The first option (the default) is to look up the address of the DCE Security server from CDS. The DCE Security server location is stored in the CDS namespace as an object in the CDS root directory. Using CDS to locate the DCE Security server every time it is needed has two negative performance impacts:

- Figure 3 shows that the response time for the client is more than two times longer than when CDS is not used (see option 2).
- The load on the DCE CDS server is high. For example, using a RISC System/6000 Model 570 CDS server with a DCE Login rate of 8,000 DCE Logins per hour, the CDS server is 40% utilized with no other CDS activity.

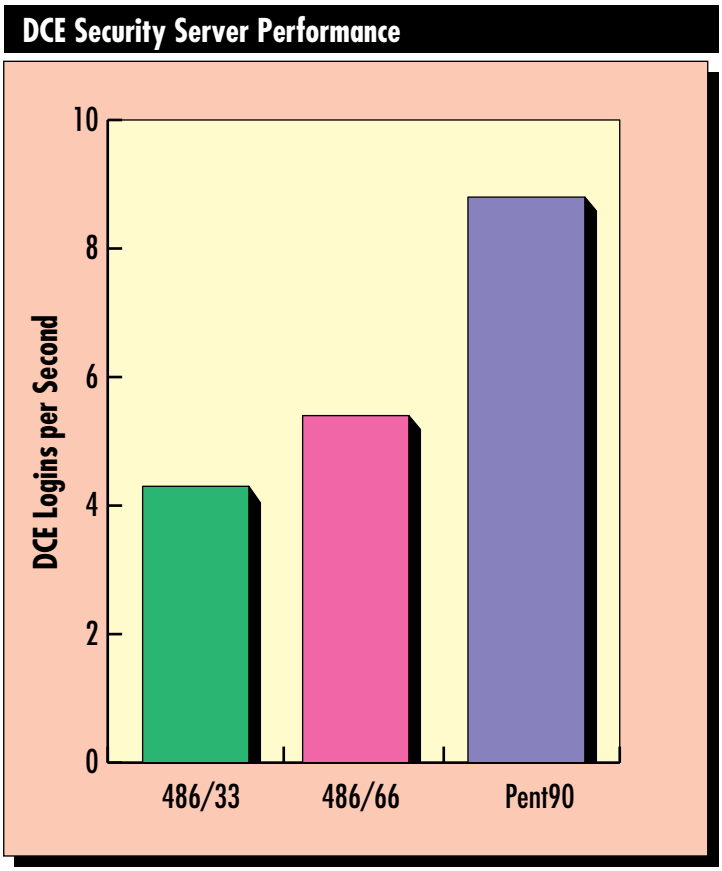


Figure 2. DCE Security Server performance

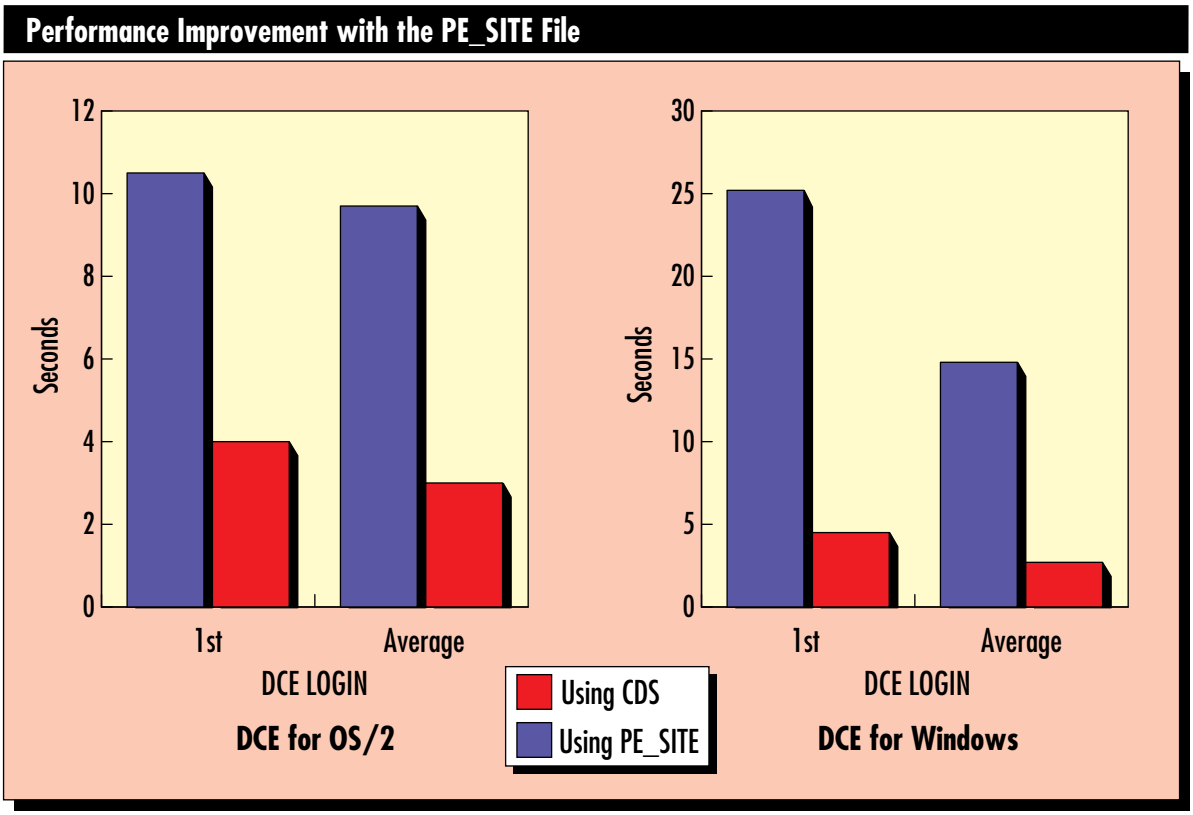


Figure 3. Using the PE_SITE file has improved performance

The second option is to allow the DCE client to store the DCE Security server's address in a file called PE_SITE on the client's hard disk. The PE_SITE option is activated in the client's CONFIG.SYS file with a SET BIND_PE_SITE=1 statement for IBM DCE for OS/2 1.2. In Warp Server using OSF DCE Version 1.1, the statement will be SET TRY_PE_SITE=1; the installation default will be 1.

The Warp Server domain controller acts as the DCE client on behalf of any legacy (LAN Server 4.0 and below) LAN requesters and additional servers.

In the next section, "DCE Cell Directory Service Performance," the results reflect option 2, using PE_SITE. If option 1 is used, 15% must be subtracted from the maximum CDS Lookups per second reported in Figures 5, 6, and 7.

The peak DCE Login activity reported by BigCo was 1,534 LAN logons per hour, or 0.426 logons per second. Since the three OS/2-based DCE Security servers tested can support 3.7 to 7.2 DCE Logins per second, DCE meets BigCo's DCE Login requirements.

The user response time of DCE Login depends on the processor speed of the DCE client. Figure 4 shows the response times for a range of DCE for OS/2 hardware platforms. These measurements were made without using the PE_SITE file. If PE_SITE is used, the improvement would be consistent with Figure 3.

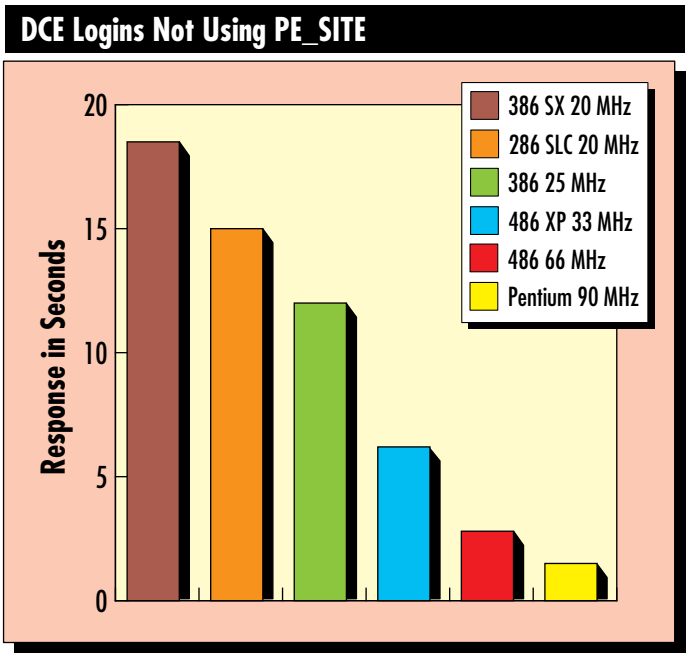


Figure 4. DCE logins that do not use PE_SITE

DCE Cell Directory Service Performance

The performance of the following DCE CDS configurations were tested:

- A stand-alone, single, primary CDS server
- A primary CDS server with three 486-66 MHz secondary servers
- A primary CDS server with one RISC System/6000 Model 990 secondary server
- A primary CDS server with six 486-33 MHz secondary servers
- A primary CDS server with eight 486-33 MHz secondary servers

The primary CDS servers studied were 486-66 MHz, Pentium 90 MHz, and RISC System/6000 Models 570, 580, and 990.

Each client has a local CDS cache, containing some part of the CDS namespace on the client's hard disk. DCE clients look up objects and directories in the CDS namespace in two ways.

The first way is to resolve the lookup in its local cache. If the client fails to find the object/directory in its local cache, the client would then call the CDS server to request a new copy of some portion of the CDS namespace.

In these tests, the client's local CDS cache satisfied all requests. This raises the question of why the CDS server is busy, since the client is satisfying the requests locally. Even though the client has the information in its cache, it must get permission from the CDS server to use the cached information.

The second way is to force the client to request a new cache from the CDS server. This occurs when either the local cache does not contain the requested information, or the cache becomes too old and a refresh is forced. The forced CDS refresh in these tests was accomplished by setting the client's cache expiration age to 0 (zero) seconds.

The main difference between the methods is that no actual data is retrieved from the server's namespace using the first method.

The CDS performance data is presented in both ways—with CDS Refresh and without CDS Refresh.

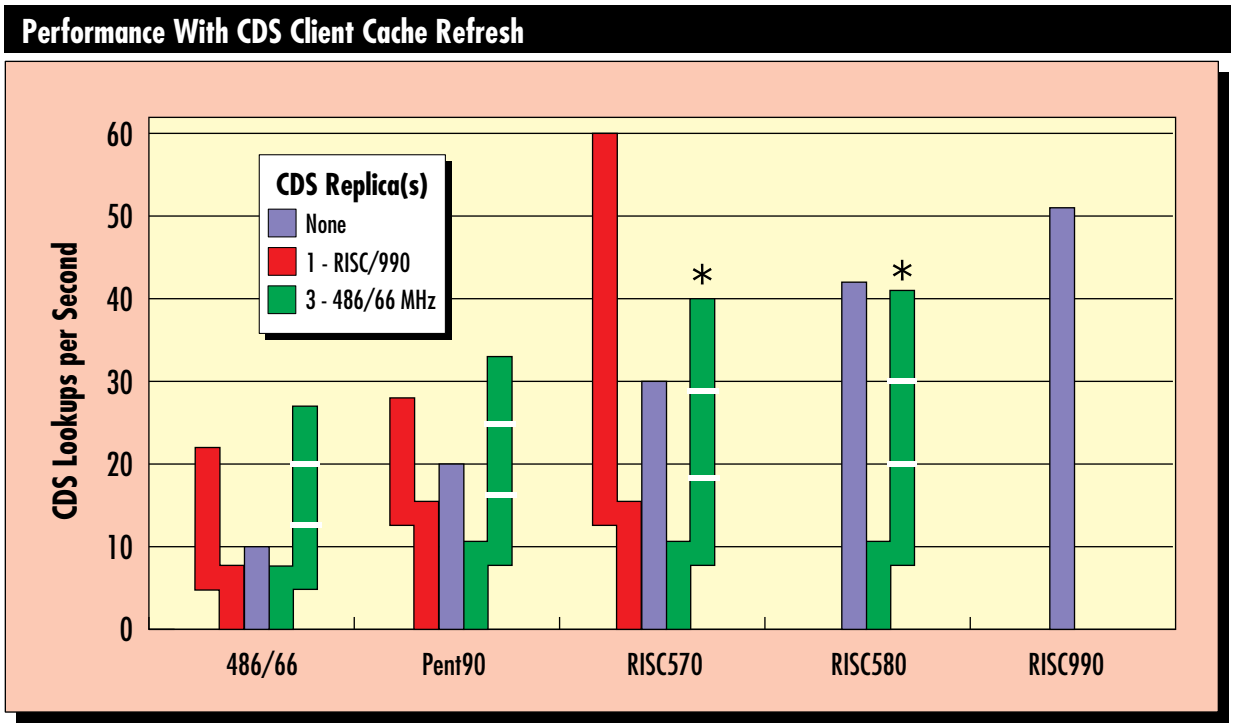


Figure 5. Performance with CDS client cache refresh

Observations on Performance Results

There are several observations of performance results:

- The throughput shown reflects nearly 100% CPU utilization of the primary CDS server.
- In each configuration, the Primary CDS server was the first to reach 100% CPU utilization, except as noted by an asterisk (*).
- If the DCE Security `PE_SITE` (described above) is not used, subtract an additional 15% from the CDS Lookups per second.

CDS Performance with CDS Client Cache Refresh

Figure 5 reflects a CDS population of 1,000 CDS objects and directories. For each additional 1,000 entries, subtract 2% from the CDS Lookups per second.

A stand-alone primary CDS server: Figure 5 shows the maximum CDS Lookups per second for the five CDS server machine types tested. (The total height of the center bar of the "cactus" is the throughput for this configuration.)

A primary CDS server with three 486-66 MHz secondary servers: The right branch of the "cactus" in Figure 5 shows the increase in performance when three 486-66 MHz secondary servers are added to the primary CDS server.

Three secondary CDS servers constituted the bottleneck in the tests for RISC System/6000 Models 570 and 580. This bottleneck can be relieved either by adding more replicas or by changing to faster replicas. The CDS Lookup requests are randomly distributed among all CDS primary and secondary servers. Therefore, if there is a great difference in horsepower, this type of bottleneck might occur. A test described later (in Figure 7) with eight CDS secondary servers demonstrates how adding more replicas can distribute the load across more low-horsepower servers.

A primary CDS server with one RISC System/6000 Model 990 secondary server: The left branch of the "cactus" in Figure 5 shows the increase in performance when one RISC System/6000 Model 990 secondary server was added to the primary CDS server. This configuration demonstrates that having a single high-horsepower CDS secondary server can be less effective than having more replicas to distribute the randomly assigned workload.

For the 486-66 MHz and the Pentium 90 primary CDS servers, the single RISC System/6000

Model 990 secondary server did not do as well as the three 486-66 MHz secondaries. This is due to the random distribution—half the requests caused the primary CDS server to reach 100%, while the Model 990 was underutilized.

In the RISC System/6000 Model 570 test, the single Model 990 looks much better, at nearly twice the stand-alone, as expected. Remember that the three 486-66 MHz secondaries were the bottleneck in these two configurations.

CDS Performance Without CDS Client Cache Refresh

For each additional 1,000 CDS entries, subtract only 1% from the CDS Lookups per second. There is less degradation for additional entries than in the "with CDS refresh" case.

Figure 6 shows the performance of the same configurations shown in Figure 5. The difference is that the clients are allowed to use their local CDS client cache.

The comments above about Figure 5 also apply to the configurations shown in Figure 6.

Since there is less interaction with the CDS servers when the CDS client cache is not refreshed, the performance for each configuration is higher in each case. In real life, there will be a mixture of lookups with and without refreshing the CDS client cache; therefore, the correct answer lies somewhere in-between.

A Primary CDS Server with Eight 486-33 MHz Secondary Servers

After completing the first two sets of tests, a question of quantity versus horsepower of the secondary CDS servers remained. The 486-33 MHz machine was chosen for this test to demonstrate that more might be better than bigger. Two tests were conducted using the Pentium 90 MHz for the primary CDS server.

The first test had six 486-33 MHz secondaries. The performance was much better than either of the earlier configurations since the work was spread across more CPUs. Figure 7 shows a greater benefit with cache refresh than without cache refresh.

The second test used eight 486-33 MHz secondary (replica) servers. The performance of this configuration is equivalent to the stand-alone RISC System/6000 Model 990, also shown in Figure 7.

Results from the six and eight CDS secondary server tests indicate that more is better. Adding

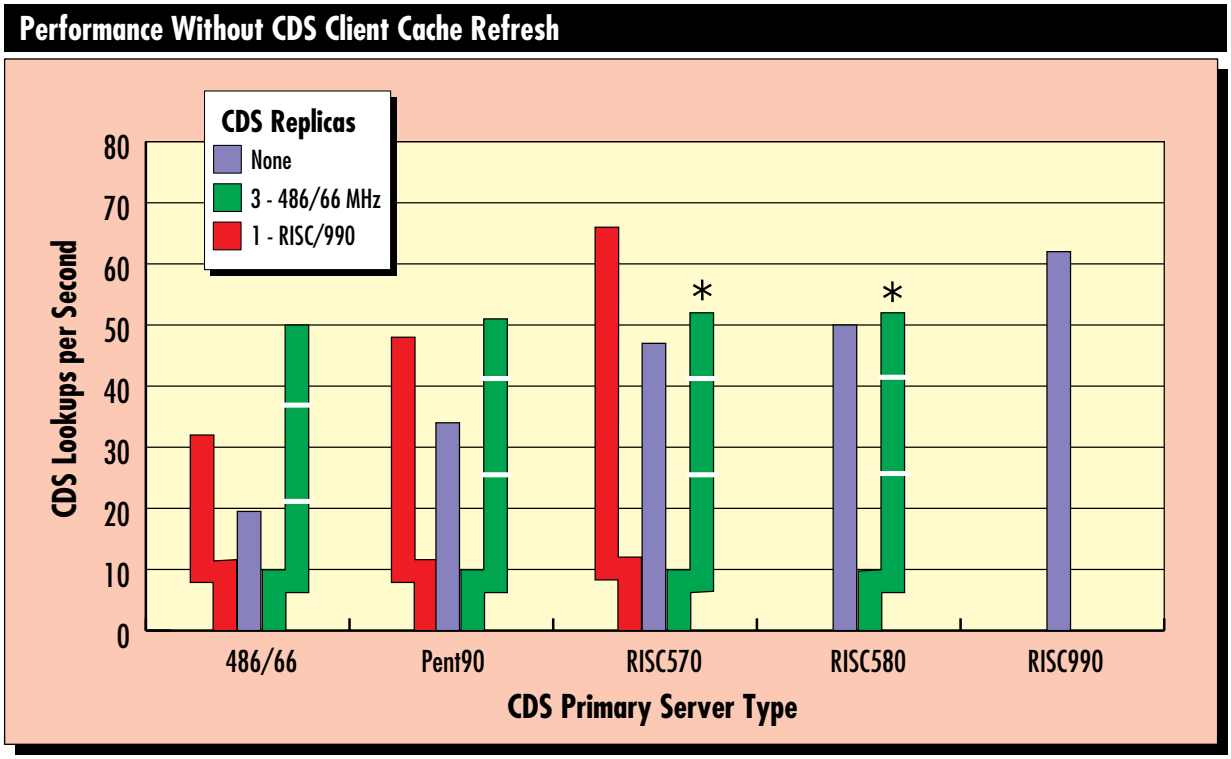


Figure 6. Performance without CDS client cache refresh

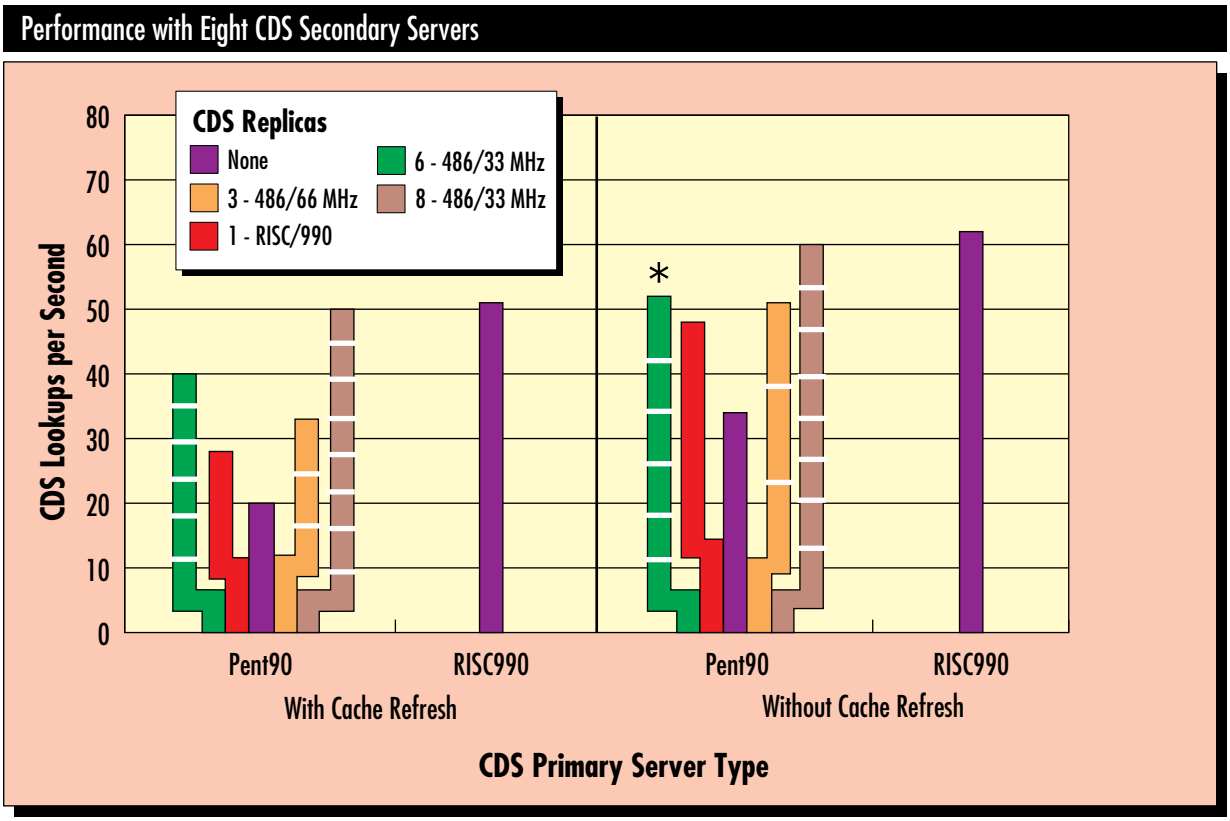


Figure 7. Performance with eight CDS secondary servers

more CDS replicas to absorb additional workload and CDS namespace population will increase the performance and capacity limits of the DCE cell.

Satisfying CDS Lookup Needs

Since BigCo did not provide specific requirements for object and directory populations and performance objectives, we can conclude from the populations of other LAN Server customers that the population is representative. Based on BigCo's network statistics, our POS benchmark application generated about the same network utilization, suggesting a similar workload. The POS benchmark performs an unnaturally high number of CDS lookups, higher than other customer applications. Therefore, the range of results achieved in these tests and the potential for improvement using additional replicas will satisfy the CDS lookup needs of BigCo.

The '/.:/hosts' Directory Tree

The DCE configuration program creates one CDS directory (`/.:/hosts/<hostname>`) and three or four CDS objects on behalf of each DCE client. In a large system such as BigCo, this would require an additional 150 MB to 180 MB of disk space for the CDS namespace. When the primary CDS server is configured, it creates the '`/.:/hosts`'directory, which is located by default on the primary CDS server. Because of the namespace disk requirement and potential degradation in CDS lookup performance, this directory should be moved to a dedicated CDS secondary server.

The `/.:/hosts` directory entries are accessed during DCE initialization by some of the management tools such as the DCE Graphical User Interface (GUI) configuration tools. Otherwise, there is little access to this directory tree. Since the performance of this replica is not critical, the memory size on this server does not need to be large enough to contain the full namespace database.

Immediately after the DCE test cell was created, we deleted the `/.:/hosts` tree from the primary CDS server and re-created it on a replica, then continued with the configuration of the other servers and clients. No other directories or objects were replicated on this secondary server.

For a 10,000-client DCE cell, we recommend 1 GB of disk space for the `/.:/hosts` directory and OS/2 `swapper.dat` file. The `swapper.dat`

file can grow to three times the size of the local CDS namespace. During periodic maintenance by the CDS server daemon, the CDS namespace files can occupy up to 2.5 times the actual size of the namespace.

The existence of `/.:/hosts/<hostname>` and its objects is a dependency of all OSF DCE platforms. Furthermore, the literal CDS pathname is hard-coded in the OSF source and cannot be distributed across multiple replicas without causing a compatibility problem with other DCE platforms.

High Performance with Ample Capacity

The DCE services supporting IBM Warp Server can support workloads consistent with BigCo's security and directory needs.

The scalability of DCE Cell Directory Services through replication can support populations and arrival rates well above the levels measured in this study.

The IBM Warp Server product will exploit DCE technology. It is a large step forward in consolidating administration and interoperability in large enterprise environments.

Bob Russell, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Russell is an advisory performance analyst for IBM PSP Systems Performance. He joined IBM in 1963 as a customer engineer in the Electric Typewriter Division. His subsequent assignments have included branch office administration and various systems analysis positions. Currently, he is evaluating the performance of distributed client/server products on the OS/2 and AIX platforms.

IBM AIX Multi-Vendor Program

AIX is being offered by several Original Equipment Manufacturers (OEMs) on their own systems. The new IBM AIX Multi-Vendor Program (AIX MVP) defines processes to ensure that trademarked AIX systems, whether from IBM or other vendors, will support a consistent programming interface. Solution providers can then develop applications based on an interface that will be portable between those systems. This helps to protect customer application investments and to broaden the market for application developers.

With AIX MVP, software developers will have a broader range of binary compatible hardware platforms running AIX than ever before. Developers can take advantage of the growing number of these compatible platforms with a single port of their applications and can use any participating AIX MVP system as a reference platform for building and maintaining their applications.

The AIX MVP Program offers the following:

- Compliance tests that can be used by AIX source licensees and compatible hardware manufacturers to ensure that their systems adhere to a consistent AIX system definition
- Support structure for AIX application developers to help resolve trademarked AIX platform compatibility issues, regardless of whether an IBM or an OEM platform is involved
- Support for the common marketing activities of solution providers, IBM, and OEM AIX platform providers through a unified AIX Application Catalog on the World Wide Web, accessible directly from OEM providers' Web pages
- Participation of AIX licensees in the evolution of AIX through their participation in an Advisory Committee, which will guide the AIX MVP program. The committee will be chaired by IBM and comprised of members from representative companies in the AIX MVP Program

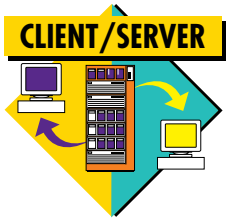
There is no charge to solution providers or customers to participate in the program.

Groupe Bull® and Motorola™ are among the current AIX licensees participating in the program. AIX MVP is planned to encompass AIX source licensees and compatible hardware manufacturers licensed to use the AIX trademark on their client and server offerings.

For More Information

Additional information is available through the World Wide Web on the Internet. To access, open the following URL: http://www.ibm.com.

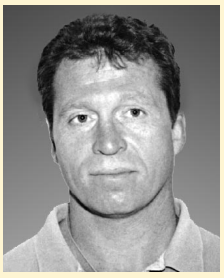
The IBM Fax Information Service allows you to receive facsimiles of prior IBM product press releases. Dial 1-800-IBM-4FAX and enter "99" at the voice menu.



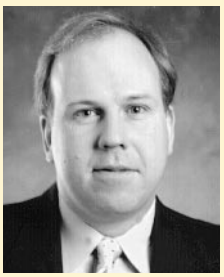
Writing Cluster-Aware Client Applications

By Steven Kohler and Thomas Casey

This article describes how application developers can use the Clinfo API routines, distributed with the HACMP for AIX software, to write "cluster-aware" client applications. Cluster-aware applications use their knowledge of a cluster's state to react intelligently to changes within that cluster. As a result, users of these client applications may not notice any change in performance or have to take any action when a change occurs within the cluster.



Steven Kohler



Thomas Casey

High availability is an essential requirement for many client/server implementations. To ensure that mission-critical data and applications are continuously available for processing, many sites have integrated high availability software into their client/server systems.

In the AIX environment, the HACMP software provides high availability services for clusters of two to eight RISC System/6000 uniprocessor and Symmetric Multiprocessor (SMP) systems and Scalable POWERparallel™ (SP) processors. HACMP allows a cluster to continue to provide data and application services critical to an installation even though a key system component—a network adapter, for example—is no longer available. When a component fails, HACMP detects the loss and shifts that component's workload to another component in the cluster. While not instantaneous, services are restored rapidly, usually within one to five minutes. The HACMP software provides recovery options for the following cluster components:

- Processors
- Networks and network adapters
- Disk and disk adapters
- Applications

When building highly available client/server systems, system engineers typically focus much planning and effort on making the server complex as fault resilient as possible. Constrained by time and money, they may overlook the client side of the "client/server" equation. Client applications, therefore, tend to be "naive." A naive client application has no knowledge of the cluster state and views the cluster as a black box that is either up or down. If a server fails, the client application typically hangs, and later must be restarted or at least reconnected to the server. Meanwhile, unable to work, stymied users ask "What's wrong? Is it my machine, the network, or what? Did I lose a lot of work?"

To build a client/server system that delivers the full benefits of high availability to its users, system engineers should address availability issues relating to client applications during the planning and implementation of that system. Clients should be cluster-aware—that is, they should be able to determine the state of the cluster and use that knowledge to react intelligently to changes within the cluster. A cluster-aware client, for example, could notify users about a problem with the cluster and instruct them to stand by while the problem is fixed. Or it could connect to an alternate server, perhaps masking the failure from the user altogether.

Writing intelligent client applications does not have to be a complicated, time-consuming task. This article demonstrates the amount of coding necessary to make client applications sufficiently robust to handle node failures transparently. We begin by briefly describing the cluster information facilities provided with HACMP. Next, we describe how the HACMP software recovers from node failure and its effect on client applications. We then describe a pair of C language routines that use functions from the Clinfo API to enable a front-end application to monitor its connection

to a Sybase® SQL Server database and, if the connection is lost as a result of a node failure, reestablish the connection to the database without having to restart the application.

Making Cluster State Information Available to Client Applications

The cluster information facilities provided with HACMP allow users to monitor the cluster state and to write intelligent client applications. These facilities include the Cluster SMUX Peer daemon, which maintains a cluster Management Information Base (MIB) that clients can access using the standard Simple Network Management Protocol (SNMP) interface, and the Cluster Information (Clinfo) daemon and libraries, which provide a simpler interface for writing cluster-aware clients.

Cluster SMUX Peer

The Cluster SMUX Peer provides SNMP support to client applications. SNMP is an industry-standard specification for monitoring and managing TCP/IP-based networks. The Cluster SMUX Peer daemon maintains a custom MIB that describes an HACMP cluster. When the Cluster SMUX Peer daemon starts running on a cluster node, it registers with the SNMP daemon, then continually gathers cluster information from the Cluster Manager daemon. The Cluster SMUX Peer daemon maintains an updated topology map of the cluster in the HACMP MIB as it tracks events and resulting states of the cluster.

Cluster Information Program

The Clinfo daemon is an SNMP-based monitor. The Clinfo daemon, running on a client machine or on a cluster node, queries the Cluster SMUX Peer daemon for updated cluster information that it stores in shared memory. Through Clinfo, information about the state of an HACMP cluster, nodes, networks, and network adapters can be made available locally to clients and applications.

Figure 1 shows the relationship between SNMP, the Cluster SMUX Peer, and Clinfo within an HACMP cluster.

Clinfo API

The Clinfo API gives the programmer a way to directly access the dynamic information about an HACMP cluster, as well as specific components within that cluster. The Clinfo API supplies both C and C++ language libraries tailored especially for this purpose. HACMP for AIX, Version 4.1 and above includes both single-threaded and multi-threaded versions of the Clinfo C and C++ API

Cluster Information Facilities

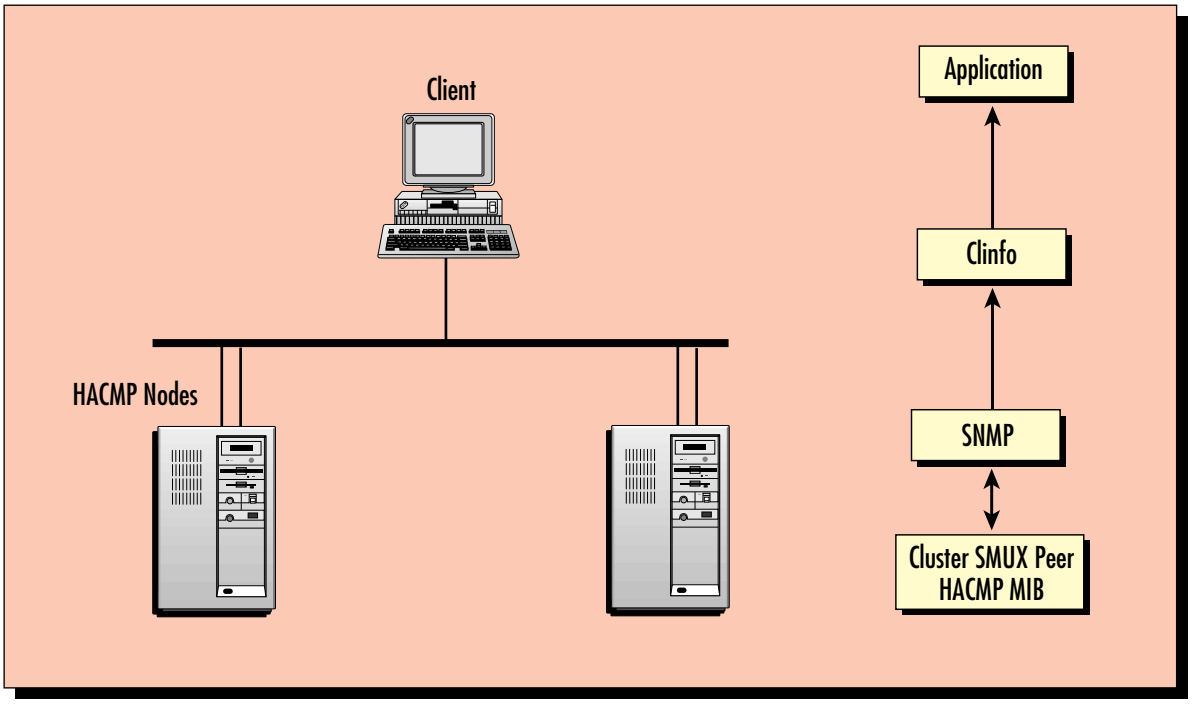


Figure 1. Cluster information facilities

libraries. The routines described in this article use the single-threaded C library.

Supported Platforms

Clinfo is supplied with the HACMP software. The source code for the Clinfo daemon and the API is distributed, so Clinfo can be ported to other machines. CLAM Associates has ported and tested Clinfo on PCs running DOS and Microsoft Windows.

Recovering from Node Failure

The HACMP software uses an agent, called a Cluster Manager, on each cluster node. The Cluster Manager monitors local hardware and software subsystems, tracks the availability of other nodes in the cluster, and monitors the availability of the other nodes by exchanging heartbeats with its neighboring nodes. If a node stops sending heartbeats, the surviving Cluster Managers on the remote nodes in the cluster take the necessary actions to get the critical applications up and running and to ensure that data has not been corrupted or lost. The available Cluster Managers take over the network interface, the volume groups or disk drives, and then restart the applications.

The Cluster Manager monitors local hardware and software subsystems, tracks the availability of other nodes in the cluster, and monitors the availability of the other nodes.

As part of the takeover, the Cluster Managers delete and re-create their routes, and refresh the Address Resolution Protocol (ARP) cache of any clients. This allows clients to connect to the backup node using the same address originally used to connect to the primary node. Any client transaction in mid-flight during a node failure must be resubmitted. This does not cause data corruption since the in-flight transaction has not yet been committed.

The takeover time varies depending on the amount of resources that need to be acquired by the takeover node and the amount of application recovery processing required. In most configurations, node takeover completes within one to five minutes.

Effect on Client Applications

Node failure causes most naive client applications to hang, since the connection between the client and server is broken. After node takeover has completed, the system administrator must notify the users, who then must log back in, restart their applications, and retrace their steps to get back to where they were before the node failure occurred.

Intelligent applications, on the other hand, can maintain the application state until the server is reconfigured, and can inform users of the nature of the problem and when it should be resolved. Intelligent applications can take action; for example, they could automatically switch to a new server in case of any fallover, making the change transparent to users.

Making an Application Cluster-Aware

Application developers can use the Clinfo libraries to develop sophisticated applications. For example, the cluster status monitor included with the HACMP software was written with these libraries. Our intent in this article, however, is to demonstrate how easy it is to use the Clinfo libraries to write simple yet powerful cluster-monitoring facilities. We now discuss two C language routines, `hacmpStable` and `checkConnection`, that enable a front-end application to handle node failure while maintaining its connection to a Sybase SQL Server database.

The hacmpStable Routine

The `hacmpStable` routine queries an HACMP cluster for its status (availability for processing) and reports the status to a calling procedure (in this case, the `checkConnection` routine, described below). The status of an HACMP cluster is determined by both its state and substate. A cluster can be in one of three states:

- `CLS_UP`: At least one node in the cluster is up, and a primary is defined.
- `CLS_DOWN`: At least one node in the cluster is up, but a primary is not yet defined.
- `CLS_UNKNOWN`: The Cluster SMUX Peer daemon cannot communicate, or is not yet communicating with, an active Cluster Manager daemon.

In addition to its state, a cluster can be in one of the following substates:

- `CLSS_STABLE`: The cluster is stable (no reconfiguration is occurring).
- `CLSS_UNSTABLE`: The cluster is unstable (a change in topology is occurring).
- `CLSS_ERROR`: A script has failed; the cluster has been in the process of configuration (unstable) for too long.
- `CLSS_UNKNOWN`: The Cluster SMUX Peer daemon cannot communicate, or is not yet communicating with, an active Cluster Manager daemon.

An application running on a client machine can only communicate with a server running on a cluster node when the cluster's state is up and its substate is stable.

```
struct cl_cluster {
     int  clc_clusterid;                /* cluster id       */
     enum cls_substate clc_substate;   /* cluster substate  */
     enum cls_state clc_state;          /* cluster state     */
     char clc_primary[CL_MAXNAMELEN]; /* primary node name */
     char clc_name [CL_MAXNAMELEN];    /* cluster name      */
};
```

Figure 2. The cl_struct data structure

```
/*
 *  Routine that uses Clinfo API calls to determine
 *  the state and substate of an HACMP cluster
 */

int
hacmpStable(int clusterId)
{
   static int           firstTime = 0;

   int                   result;
   struct cl_cluster    clstr_buf;

   if (firstTime) {
      /*
       * Create initial Clinfo connection and allocate buffer arrays.
       */
      result = cl_initialize();
      if (result != CLE_OK) {
         (void)fprintf(stderr, "cl_initialize failed, %s\n",cl_errmsg(result));
         return TRUE;
      }
      firstTime++;
   }

   result = cl_getcluster(clusterId,&clstr_buf);
      if (result == CLE_OK) {
         if (clstr_buf.clc_state == CLS_UP) {
            if (clstr_buf.clc_substate == CLSS_STABLE) {
               return TRUE;
            }
         }
   } else {
      (void) fprintf(stderr, "cl_getcluster(%d) failed, %s\n",
                     clusterId, cl_errmsg(result));
   }
   return FALSE;
}
```

Figure 3. The hacmpStable source code

The hacmpStable routine uses three functions from the Clinfo C library:

- The cl_initialize routine checks to see if Clinfo is running and, if so, acquires the shared memory map. This map is managed by Clinfo using information stored in the MIB by the Cluster SMUX Peer daemon.
- The cl_getcluster routine returns information about a specified cluster in a cl_cluster data structure. The cl_cluster data structure is shown in Figure 2.
- The cl_errmsg routine takes a status code returned by Clinfo and returns the text associated with that status code.

The source code for the hacmpStable routine is listed in Figure 3.

The checkConnection Routine

The checkConnection routine determines if a client's connection to a database is still valid. If the connection is good, the checkConnection routine returns success to the calling procedure, which then continues processing. If the connection is no longer valid (for example, the node has failed), the checkConnection routine calls the hacmpStable routine to determine if the cluster is stable. If the cluster is not stable, the checkConnection routine loops, waiting for the cluster to become stable. When the cluster stabilizes, indicating that the takeover node has acquired all the resources from the failed node, the checkConnection routine then reestablishes the client application's connection to the database. The client application then resumes processing.

```
/*
 *  Routine that tests a database connection to determine
 *  if it is valid. If not, calls the hacmpStable routine to
 *  determine cluster status.
 */

int
checkConnection()
{
   char *clusterStr;

   clusterStr = getenv("CLUSTERID=");
      if (clusterStr) {
         if (!DBproc || (DBDEAD(DBproc))) {
            while (1) {
               int clusterId = atoi(clusterStr);
               if (hacmpStable(clusterId)) {
                  (*msgCallback)("Cluster stable, attempting to reconnect");
                  (void)initConnection(0);
                   break;
               }
               (*msgCallback)("Cluster unstable, waiting...");
               sleep(5);
            }
            return 0;
         }
   }
   return 0;
}
```

Figure 4. The checkConnection source code

To enable a front-end application to handle node failure transparently, an application developer would place the `checkConnection` routine before every SQL execution. The client application is then able to maintain its connection to the database server, and reconnect only if necessary.

The source code for the `checkConnection` routine is listed in Figure 4. Note that this routine was written using the Sybase SQL Server DBlib library.

Using Clinfo Routines in Client Applications

To use the `hacmpStable` and `checkConnection` routines in client applications, application developers must be sure to include the proper header files and link to the necessary library.

Header Files

Application developers must specify the following `include` directives in each source module that uses the Clinfo C library:

```
#include <sys/types.h>
#include <netinet/in.h>
#include <cluster/clinfo.h>
```

Linking the libcl.a Library

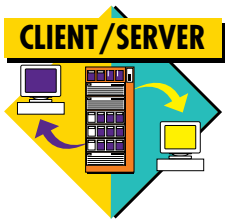
Application developers must add the following directives to the object load command of a single-threaded application that uses the Clinfo C library:

```
-lcl  -lclstr
```

The `libcl.a` library contains the routines that support the single-threaded Clinfo C library. The `libclstr.a` library holds commands and other information that may be used by some of the `libcl.a` routines.

Steven Kohler, CLAM Associates, Inc. 101 Main Street, Cambridge, MA 02142. Internet: slk@clam.com. Mr. Kohler is principal engineer in CLAM's technical support group, and is the technical lead for the HACMP for AIX change team.

Thomas Casey, CLAM Associates, Inc. 101 Main Street, Cambridge, MA 02142. Internet: tom@clam.com. Mr. Casey is the manager of CLAM's technical writing group. He has a BA from Trinity College in Hartford, Connecticut and an MA from Emerson College in Boston.



PDM Implementation Framework

By Eddie Ho, Peter Stoll, and Eric Dunn

The IBM Austin Product Data Management (PDM) implementation framework comprises several hardware and software components. Key emphasis has been placed on performance, high availability, automated backup and recovery, connection-based load balancing, system monitoring, worldwide network connectivity, shared filesystems, secured access, and development and manufacturing tool integration. The implementation framework has provided the foundation to support mission-critical Engineering Change and worldwide Release processes at IBM Austin, Texas.

For the past 20 years, the IBM Corporation has been using mainframe-based legacy Product Data Management (PDM) systems to support its mission-critical Engineering Change and worldwide Release processes. These processes facilitate and control the communication of information from development to worldwide manufacturing locations. The implementation of ProductManager/6000, PDM business software, has created a significant opportunity for the IBM Engineering Change and Release team in Austin, Texas to redefine the implementation framework using the latest in client/server technology.

To support the corporate re-engineering initiatives as well as to follow the principle of "use what we sell," IBM chose ProductManager/6000 and the RISC System/6000 (RS/6000™) platform. In addition, we had to define several hardware and software components, which focus on the following functions:

- High availability
- Automated backup and recovery
- Connection-based load balancing
- System monitoring
- Network connectivity
- Shared filesystem access
- High performance
- Secured access
- Seamless integration between development and manufacturing systems

A well-planned implementation can keep the Product Data Management installation from becoming a potential bottleneck for the Engineering Change and worldwide Release processes. Because of the diverse types of end-user workstations, various response time requirements, and multiple product integrations, a three-tiered client/server data model was chosen to support the PDM framework. Figure 1 shows this model.

Austin RS/6000 Development Environment

Both local and remote manufacturing locations require access to the PDM system to implement engineering changes. The Austin Engineering PDM implementation will support the RISC System/6000 Division, IBM Microelectronics Division, and the PC Company. It will also support the following remote manufacturing locations: Raleigh, North Carolina; Wangaratta, Australia; Vimercate, Italy; Yamato, Japan; and Sumare, Brazil.

Requirements

We identified implementation requirements before developing any formal specifications or making a capital investment. These requirements were based upon the re-engineered development process as well as current commitments to development and manufacturing customers:

- Performance equivalent to the mainframe-based legacy system
- 100 concurrent, 300+ casual users
- Scalable implementation
- Consolidated technical data repository
- Automated backup and recovery
- High availability with 7x24x365 operations
- Worldwide network connectivity
- IBM Local Area Network security compliant
- Seamless integration to CAD/CAM, part selection, and downstream Material, Requirements, and Planning (MRP) systems

Three-Tiered Architecture

IBM's ProductManager/6000 supports a flexible three-tiered client/server architecture using the Distributed Computing Environment (DCE) framework.

These tiers are client, compute, and database layers, as shown in Figure 2. The client layer executes the ProductManager/6000 Graphical User Interface (GUI) with either AIX, OS/2, or Windows workstations. The compute layer is responsible for running the ProductManager/6000 ObjectManager, which creates and submits the database transactions to the database layer. The database layer receives the database transactions and makes the necessary inserts, updates, deletes, and retrievals to the database. It is also responsible for data storage (backup and recovery).

Software Components

The software component is key to achieving the predefined requirements. Application Services Manager, Product Structure Manager, Product Change Manager, and Document Control Manager, which comprise the ProductManager/6000 solution are major components. However, other software and support infrastructure, as shown in Figure 3, is also required.

Figure 4 lists the key software support packages for Product Data Management.

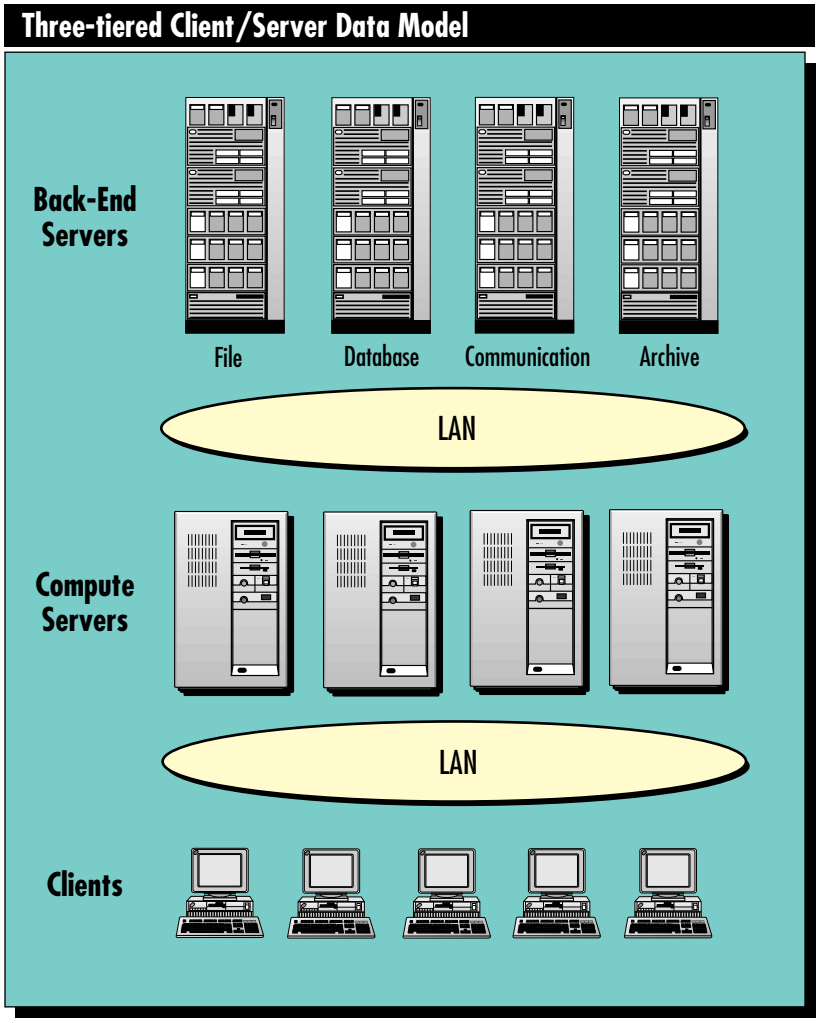


Figure 1. Three-tiered client/server data model

Performance/Capacity Planning

Performance simulation and capacity planning is an ongoing activity that supports the rollout of the implementation. The objective of the prototype environment, which initially supports this activity, is to define resource utilization and potential implementation bottlenecks. To support the performance simulations, we used preVue/X to initially define and script user workloads.

Workload

Initial workloads consisted of the following:

- Users creating and distributing folders to local and remote users
- Users reviewing and approving folders
- Multiple users creating Engineering Changes (ECs) and part numbers

- Multiple users requesting background reports
- Users browsing existing ECs and part numbers

The workload is executed using preVue/X and locally developed scripts.

Capacity Planning

BEST/1® for UNIX monitors the resource utilization during the simulation; it also reports CPU, memory, network, paging, and disk utilization. This process identifies resource utilization as well as potential performance bottlenecks. In addition, "what if" scenarios can be created using various hardware and software configurations to determine the most efficient implementation. For example, you can model and evaluate specific hardware and software configurations prior to implementation. The model analysis can be used to evaluate a specific configuration.

Network Connectivity

Network connectivity is an important consideration when implementing an enterprise PDM system client/server solution, particularly if 1,000 local and remote users will be accessing the system.

The Implementation Framework

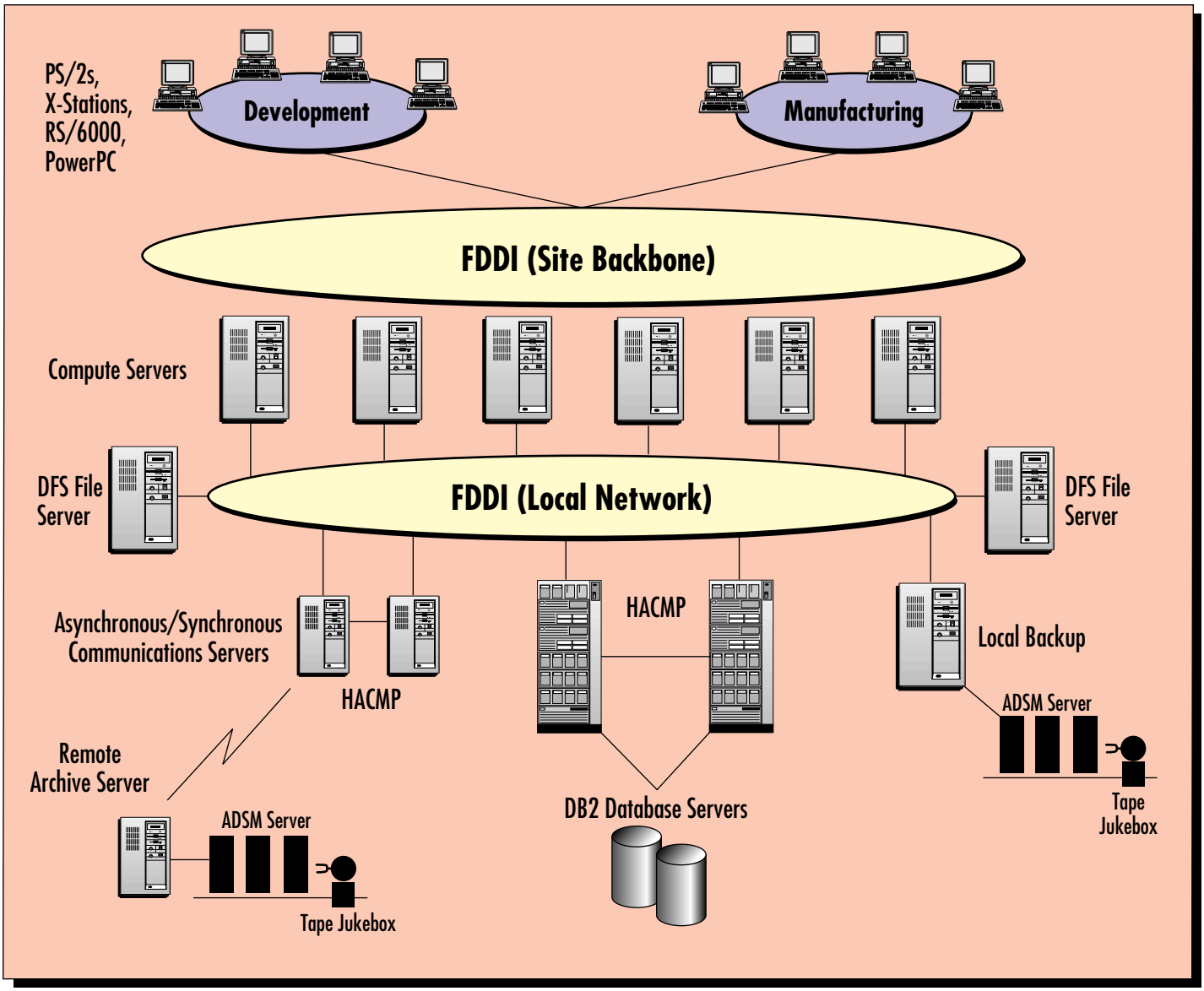


Figure 2. Implementation framework

Real-time access is required by all end users of the system. Consequently, key focus is placed on all aspects of networking.

Local Area Network

The database servers, compute servers, system backup server, and asynchronous and synchronous communication servers are connected by Fiber-optic Distributed Data Interface (FDDI) via an intelligent IBM 8250 Multi-Access Unit. This unit is connected to a router that is directly attached to the Austin site FDDI backbone.

Clients are connected to local area networks via 4/16 Mbits Token Ring. These LANs are connected to building routers, which are then directly connected to the site backbone. The standardized network protocol is the Transmission Control Protocol/Internet Protocol (TCP/IP).

Wide Area Network

With global manufacturing locations in Raleigh, Sumare, Vimercate, and Wangaratta, the PDM database must be accessible 7 days a week, 24 hours each day (7x24). Multiple routers are required to support remote connectivity. The data transfer media consists of a worldwide multiprotocol network. Connection to the PDM implementation is supported either by executing the GUI remotely or via the AIX remote login facility.

High Availability

To support a 7x24 mission-critical PDM implementation, we have placed significant emphasis on limiting any single point of hardware failure. To limit downtime, we added additional hardware such as a standby system, dual communication adapters, and shared DASD.

In addition, we installed IBM HACMP/6000™ and LoadLeveler™ to support high availability. HACMP/6000 facilitates the takeover of all resources by eliminating a single point of failure. LoadLeveler monitors and manages work loads. Together, they extend the overall system availability of the implementation.

Communication Monitors

The ProductManager/6000 asynchronous and synchronous communication monitors are required to support real-time and batch communications to the database layer. These interfaces support CAD, MRP integration, batch reports, and Product Data Interface (PDI) communications.

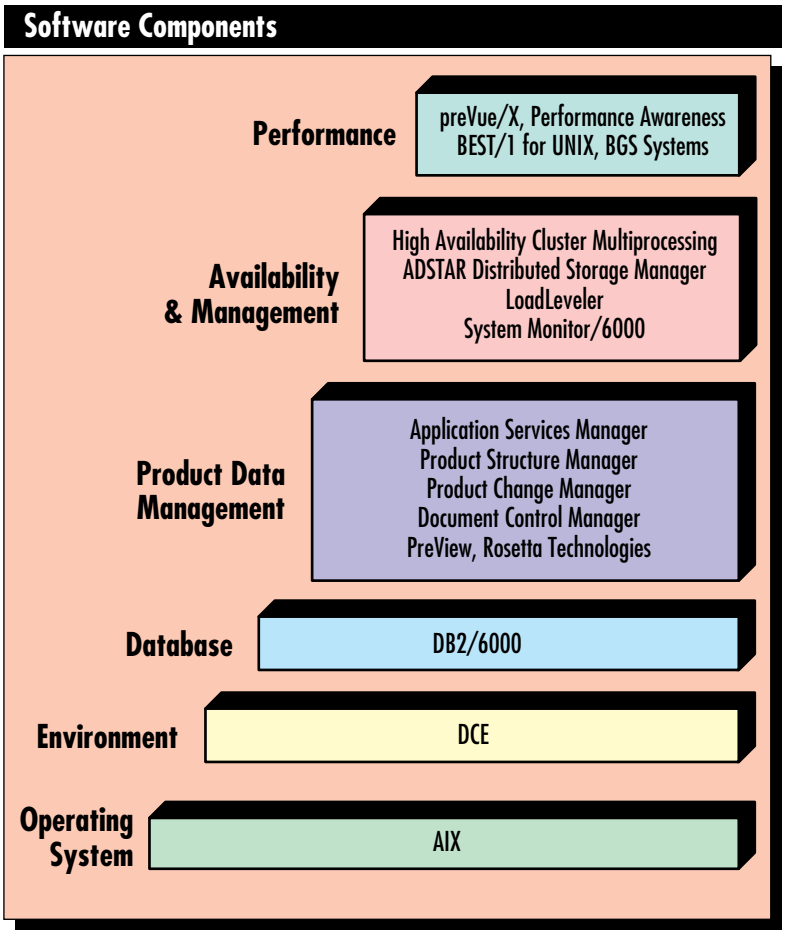


Figure 3. Software components

Software Support Packages for PDM

Software	Function
preVue/X	Application simulation in a user environment
BEST/1® for UNIX	Capacity planning and management
High Availability Cluster Multiprocessing (HACMP)	24-hour availability support
ADSTAR™ Distributed Storage Management (ADSM)	Backup and recovery
LoadLeveler	Load balancing
System Monitor/6000	Resource monitoring
DB2/6000™	Mission-critical database support
DCE	Distributed Computing Environment infrastructure support
AIX	Operating system

Figure 4. Software support packages for PDM

Each communication monitor has a dedicated server.

To support a 7x24 availability of these processes, we defined a highly available cluster. In the event of a single point of failure, the failed process is initiated on the standby node. The migrated processes will be reinitiated on the primary node when it is activated. This highly available configuration is called *shared*.

Database Servers

The implementation requires high availability of the database. The database must be available 24 hours a day to support both real-time and batch communication.

To support this requirement, we defined a highly available cluster using the hot-standby configuration. If a single point of failure occurs, DB2® is accessed on the standby system. The data is readily accessible since it has been mirrored to the backup node.

DCE provides the client/server infrastructure that allows distributed applications to interact in a heterogeneous environment.

Connection-Based Load Balancing

To provide end users with the best performance on the compute servers, we defined and implemented connection-based load leveling to support an enterprise ProductManager/6000 implementation. A common front end was developed to support centralized access to the PDM system and its associated products. When you access the PDM system using the customized front end, IBM's LoadLeveler software component that executes on the asynchronous communication server assumes control. LoadLeveler's interactive session support identifies which compute server has the least load based upon customized metrics. Once the least-loaded compute server has been identified, a remote procedure call is executed to pass the user resources to the specified compute server.

Backup and Recovery

We initially defined backup and recovery objectives to support the enterprise implementation. Since IBM's disaster-recovery program requires up to two years of data to be supported, the objectives established for the Austin Engineering Release backup and recovery consist of the following:

- To provide both local and remote (archive) backups
- To recover all data that has been backed up within two hours
- To recover the database from any point in time over two years

Automated backup is necessary to support a mission-critical implementation. Nightly, the database and data residing on the compute server are incrementally backed up to an 8mm Exabyte® 60 tape library. Weekly, the database is automatically archived off-site using the DB2 backup facility. The data is backed up over the Wide Area Network to an Exabyte 120 8mm tape library that resides off-site.

Both local and remote backups are automated via the ADSTAR Distributed Storage Management (ADSM) software. ADSM can support automated backup based upon a set of predefined parameters that are created between the ADSM client and server processes. Specific parameters consist of establishing the policy domain, policy set, management class, and copy groups. Recovery is a manual process invoked through the ADSM client interface.

Distributed Computing Environment

DCE software is a major component of the Enterprise ProductManager/6000 implementation. DCE provides the client/server infrastructure that allows distributed applications to interact in a heterogeneous environment. Several components make up DCE:

- Distributed Time Service (DTS)
- Cell Directory Service (CDS)
- Security Service
- Distributed File System (DFS)
- Global Directory Agent (GDA)

High availability of the DCE implementation consists of three DTS processes, two CDS processes, two security processes, two DFS processes, and two GDA processes. The entire DCE cell comprises five servers with one or more active DCE processes per server.

DCE's DFS is used extensively in the ProductManager/6000 implementation. DFS, a distributed filesystem, enables cooperating hosts (clients and servers) to efficiently share filesystem resources across both LANs and WANs. Within ProductManager/6000, DFS supports the centralized access of ProductManager/6000 and associated binary files, as well as electrical and mechanical pre-release technical data. The DFS repository is accessible via the PDM compute servers and

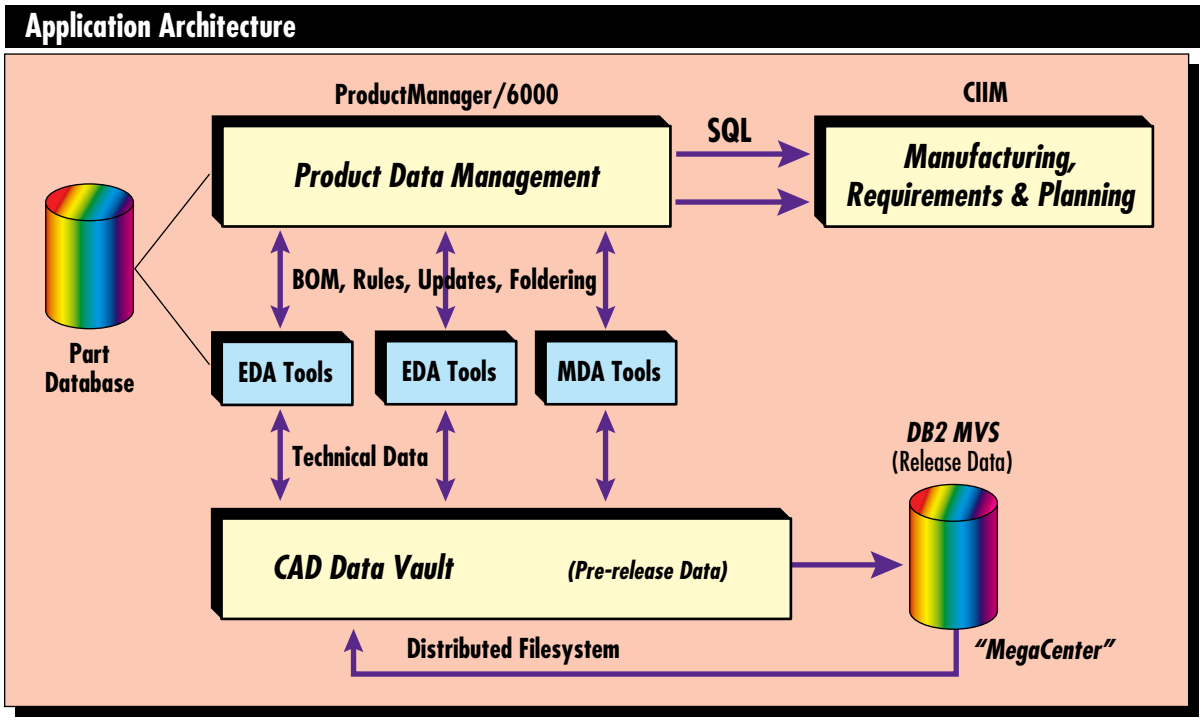


Figure 5. PDM product integrations and interfaces

local clients. Once data is updated on the DFS server, the local client caches are refreshed.

System Monitoring

IBM System Monitor/6000 monitors each server that comprises the PDM implementation. This software provides online notification of predetermined resource conditions. If a condition is triggered, notification is sent to PDM Information System support via E-mail or digital pager, depending on the resource and severity of the condition. In addition, a World Wide Web page will be updated to reflect the status of each server and resource every 15 minutes.

Seamless Integration

The Product Data Management system eases the flow of product information from development to manufacturing. Consequently, the PDM implementation framework must support several key product integrations and interfaces, shown in Figure 5.

The Austin Product Data Management implementation framework supports both batch interfaces and seamless integrations between ProductManager/6000, MDA/EDA design applications, CAD data vault, parts database, and the Material, Requirements, and Planning system.

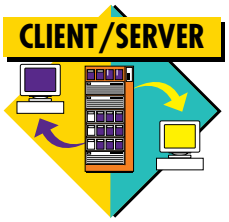
The batch interface is based upon using the ProductManager/6000 product data interface. This interface supports remote data communications via the Asynchronous Communication Monitor (ACM). Seamless integration is supported via the ProductManager/6000 Synchronous Communication Monitor (SCM). The SCM provides a real-time interface to the electrical and mechanical design applications as well as the MRP system.

Eddie Ho, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Ho is a senior programmer in the AIX Executive Briefing Center. He has a BS in Computer Science from the University of Wisconsin and an MS in Computer Science from North Dakota State University.

Peter Stoll, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Stoll is a senior manufacturing analyst in the RS/6000 Manufacturing area. He has a BA from the University of California at Los Angeles.

Eric Dunn, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Dunn is an advisory development programmer in the RS/6000 Technical Services area. He has a BS in Management Information Systems and an MBA from Old Dominion University in Norfolk, Virginia.

DCE DFS Interoperability in Data Sharing Environments



By Jean E. Pehkonen

As enterprises migrate to the Distributed Computing Environment (DCE) and Distributed File Services (DFS), existing environments may use legacy data or systems, which can make moving to a pure DFS environment impossible or impractical. In these cases, DFS may need to work in environments that contain Network File System (NFS®) or Andrew File System (AFS®). Fortunately, DFS can operate in most of these situations. This article discusses possibilities for coexistence and interoperability of DCE DFS in both NFS and AFS environments.

As DCE environments become the enterprise solution of choice, many DCE applications can provide distributed services such as data sharing, printing services, and database access. As part of the Open Software Foundation's (OSF) DCE, DFS provides filesystem services for these environments. DFS uses Remote Procedure Calls (RPCs) for data transfer, Cell Directory Services (CDS) for naming, and DCE Security for authentication services to provide data sharing services.

In addition to its use of DCE services, DFS itself is rich in features. It provides a uniform global filespace that allows all DFS client users to see the same view of the filespace. It caches filesystem data at the client for improved scalability and performance by reducing network traffic to file servers. DFS also supports advisory file locking.

One DFS feature is the ability to export the operating systems's native filesystem. In AIX, the native filesystem is the Journaled File System (JFS). In addition, DFS also provides its own physical filesystem—the DCE Local File System (LFS). The DCE LFS supports DCE Access Control Lists (ACLs) on files and directories for securing access to data and advanced data management capabilities such as replication and load balancing.

NFS and DFS Differences

Although DCE DFS and NFS are both distributed filesystem products, the two products differ in their data-sharing models and semantics. NFS relies on a peer-to-peer client/server model between machines, while DFS operates within an autonomous administrative unit called a *cell*. Semantically, NFS is a stateless system implying that the NFS server does not maintain information about the client requests. DFS maintains the state of file and directory information to provide UNIX single-site semantics by using an internal token manager.

Administration of a DFS environment is centralized—a system administrator can complete the majority of filesystem administration from a single system within the cell. Administration of an NFS environment, on the other hand, often involves updating information on each NFS client system in the environment.

Given these differences, the two products seem unlikely to coexist or interoperate in one environment. However, there are several scenarios where this may be desirable and indeed possible.

Coexistence with NFS

Both DFS and NFS maintain the concept of "exporting" data. In both contexts, a filesystem is

Jean E. Pehkonen

made available to which client systems can remotely gain access. Clients then access this data by mounting the filesystem remotely.

One difference between NFS and DFS becomes apparent in this area. Once the data is exported, each NFS client must remotely mount that filesystem in order to gain remote access to it. Because DFS uses a uniform global filespace, a DFS client only needs to mount the root of the filespace (/...) to get access to all data exported through DFS. The DFS system administrator is responsible for organizing the filespace and determining where the mount points for the data will be located. Once the administrator has set up the tree structure, all clients see the same view.

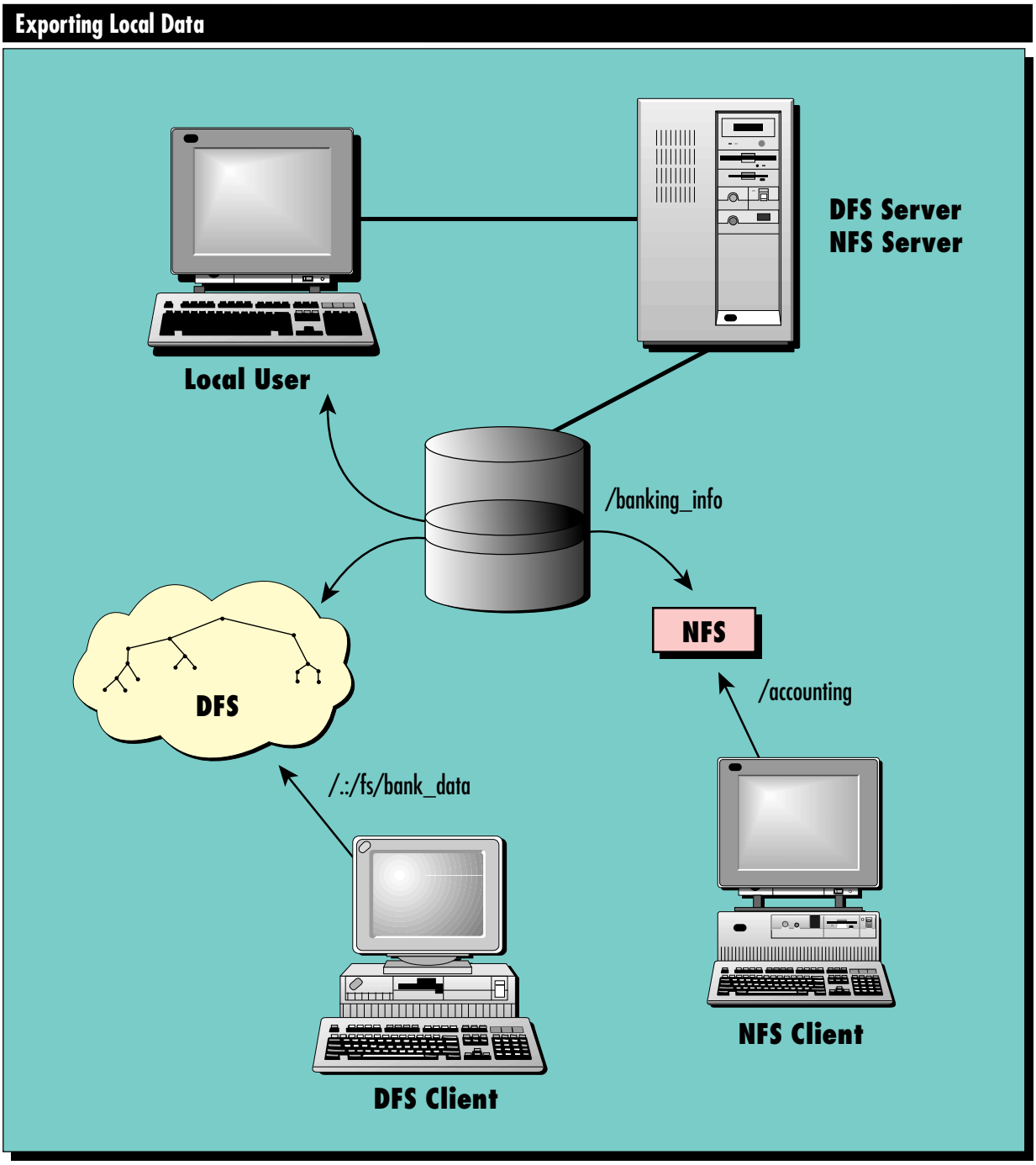


Figure 1. Exporting local data

On the server system, native JFS filesystems can be exported to NFS for use by NFS clients. At the same time, the filesystem can also be exported to DFS for use by DFS clients. Figure 1 shows this scenario. A RISC System/6000 (RS/6000) system is providing data through NFS and DFS simultaneously. The local filesystem, `/banking_info`, has been exported to both DFS and NFS so client systems in the environment can access the data. If the system is running the NFS client software, the user can use the data through the NFS mount point, `/accounting`. DFS clients running DCE and DFS software can use the data by accessing it through the path `/.:/fs/bank_data`.

In addition to being accessible to NFS and DFS client users, the filesystem still remains accessible to any local users who may have access to the RS/6000 through a local account login. These users can access the data using the local path `/banking_info`.

Because NFS is a stateless system, NFS clients may not necessarily see data changes from the server immediately. Data will be maintained consistently among DFS clients and local users due to UNIX single-site semantics.

Interoperability with NFS

The DFS filespace (`/...`) can also be exported through NFS, which allows NFS clients to mount and access DFS even though they are not running a DFS client. This may be advantageous in environments where DFS client software is not available for all hardware platforms

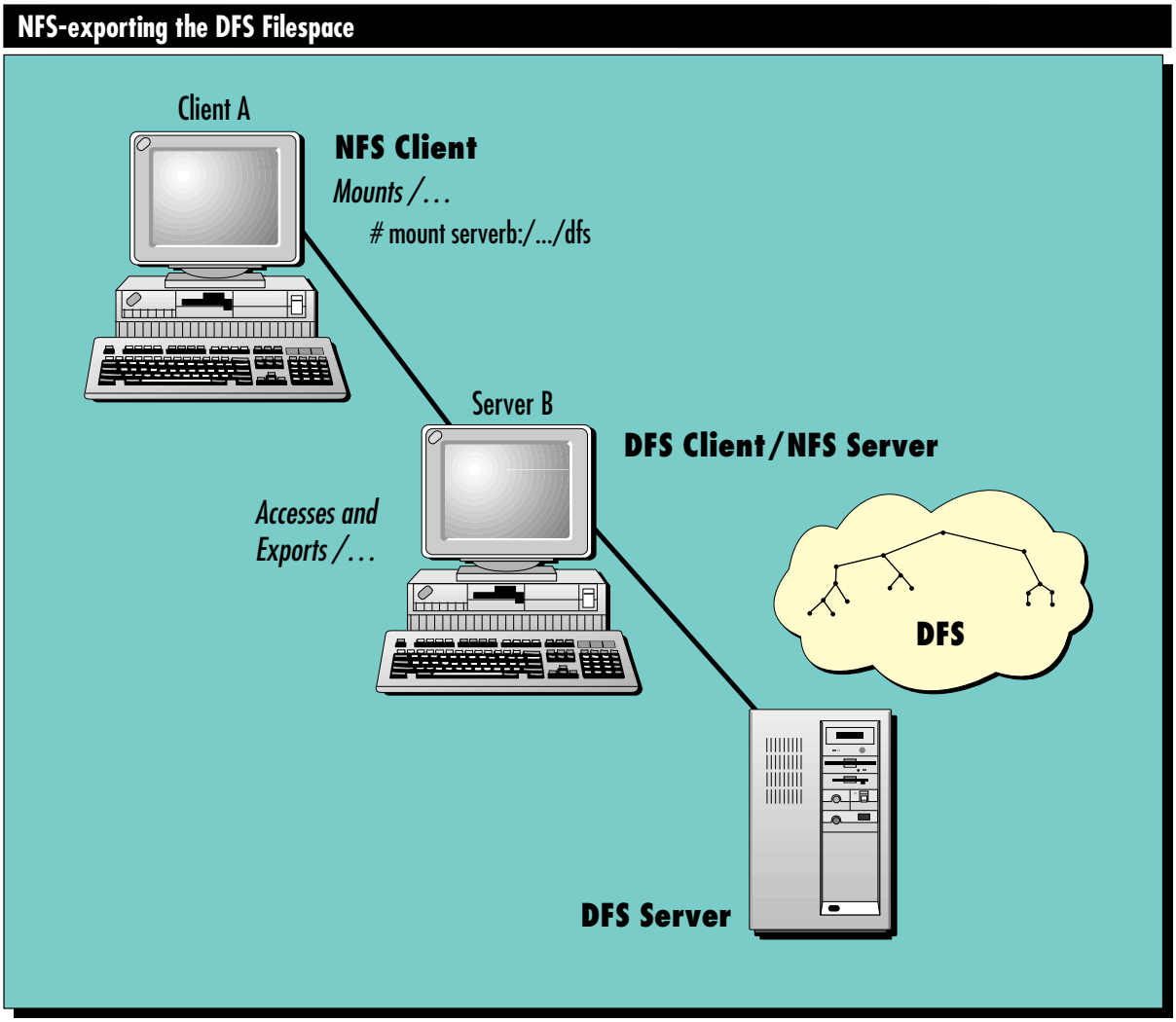


Figure 2. NFS-exporting the DFS filespace

in the enterprise, such as DOS-based PCs. Of course, the platform does need an NFS client software package available.

Figure 2 shows this scenario. A DFS client system within the cell can access the DFS filespace. In addition, the DFS client can run NFS server software and make the DFS filespace available for mounting by adding it to the `/etc/exports` file. An entry such as `/...` will allow this. NFS clients may, in turn, mount the filesystem the same way they mount other remote filesystems. The NFS client will then have access through the path, `/dfs`. True DFS clients will have access to the same information through the path `/.:/fs`.

DFS provides a higher level of data protection by using ACLs on both files and directories for data stored within DCE LFS filesets. (DCE ACL support is not available for JFS filesystems exported to DFS.) This level of protection usually means that a user must be authenticated as a DCE principal to gain access to the DFS filespace. NFS clients cannot be DCE-authenticated, so any access they gain will be as an unauthenticated user. Figure 3 shows an example of an ACL set to protect the DFS filespace to allow access only by authenticated users.

```
# SEC_ACL for /.:/fs:
# Default cell = /.../dfsfvt.cell.austin.ibm.com
user_obj:rwxcid
group_obj:r-x--
other_obj:r-x--
```

Figure 3. ACLs on the DFS filespace

```
# SEC_ACL for /.:/fs:
# Default cell = /.../dfsfvt.cell.austin.ibm.com
mask_obj:rwx-id
user_obj:rwxcid
group_obj:r-x--
other_obj:r-x--
any_other:rwx-id
```

Figure 4. ACLs allowing write permission for NFS client

```
$ dfsiauth -add -r blizzard.austin.ibm.com -i 1022 -u jean
Enter Password:
dfsiauth: <blizzard.austin.ibm.com, 1022> mapping added
DCE principal: jean
```

Figure 5. Establishing an authentication mapping

In this situation, any NFS clients wishing to gain access to the DFS filespace will be denied access since they do not have any DCE authentication associated with them. For NFS clients to gain access, the `any_other` ACL can be added to the DFS objects. Figure 4 shows an `any_other` ACL on a DFS directory, which allows an NFS client to have write permission to the data in that directory. The `any_other` ACL allows unauthenticated users—those who do not match any of the other ACLs—access to the file or directory. Adding the `any_other` ACL opens the filespace for NFS clients, but it also allows any unauthenticated DCE user to access the data. This may not be desirable, depending on the confidentiality of the data being stored.

NFS/DFS Gateway

An additional product introduced in AIX DCE Version 1.3 allows NFS users authenticated access to DFS. The NFS/DFS Authenticating Gateway allows NFS users to maintain an authentication mapping to DFS so they can access data without using the `any_other` ACL. Data in DFS can be maintained under ACL control, but NFS clients can still access the data.

The NFS protocol maintains security by using a hostname, userid pair for each client. For example, user jean with userid 1022 on the system, blizzard.austin.ibm.com will be sent to an NFS server. However, this pair means nothing to DFS, which maintains accessibility based on DCE principals and DCE ACLs. The NFS/DFS Gateway enables a mapping to be established between the NFS information and the DCE information. Figure 5 shows a mapping being established using the `dfsiauth` command.

Once this mapping has been established, requests from the NFS client, blizzard.austin.ibm.com, and the user 1022 on that system will be authenticated as the DCE principal, jean. Any files or directories for which the DCE principal, jean, has accessibility can be accessed by that particular NFS client.

AFS/DFS Differences

Conceptually, DFS is similar to AFS. Like DFS, AFS maintains a uniform global filespace, centralized administration, and information caching at the client systems. However, AFS

does not offer all the same features as DFS. DFS features such as ACLs on files and directories (AFS maintains ACLs on directories only), UNIX single-site semantics, and file locking are among some features that are not available for AFS. AFS cannot export the operating system's native filesystem. This limitation may inhibit the interoperability allowed for NFS and DFS clients; however, some accessibility is still possible in these types of data sharing environments.

Coexistence with AFS

On client systems, it is possible to run the software for the DFS client and the AFS client simultaneously. Because each product uses a local disk cache (or memory cache) for storing information brought across the network, two separate caches must be available on the system. Each product maintains information in its cache differently, so it is not possible for both products to share one filesystem for the local cache. This coexistence may be advantageous in situations where data needs to be migrated from an existing AFS cell to a new DCE DFS cell. Since the data from both cells is accessible at one system, data can be moved from one cell to another by using AIX `cp` and `mv` commands. Once all data from AFS has moved to DFS, the AFS cell may be unconfigured.

mounted	mounted over	vfs
AFS	/afs	afs
DFS	/...	dfs

Figure 6. DFS and AFS clients

Figure 6 shows the partial results of an AIX `mount` command where both DFS and AFS clients are running.

It should also be noted that both AFS and DCE maintain system time by keeping the system clock in sync against a server system in the cell. However, if the servers are running on different machines, they may not be maintaining the same clock time. Therefore, AFS on the client system will attempt to synchronize its time while DCE using DTS time services is synchronizing its time to another server. This scenario may lead to confusion at the client system. So it is recommended that for systems running DFS and AFS clients, the `afsd` process should be invoked with the `-nosettime` option. This option will stop AFS from maintaining time and allow the DCE time services to provide one-time service for the client system.

DFS and AFS fileserver system software may be run on one system. However, because AFS does not allow exportation of native filesystems, it is not possible for AFS and DFS to serve the same data in a single filesystem. Data can exist in the JFS or DCE LFS format for DFS, and may also be duplicated in AFS partitions to be served to AFS clients on one server system.

Due to their heredity from Transarc® several administrative commands in AFS and DFS have maintained the same names. This may cause confusion for system administrators if they do not know which command they are using. This problem can be avoided by specifying the command using its full pathname. The DFS commands reside in the `/usr/bin` directory on AIX systems. The AFS commands reside in the `/usr/afs/bin` directory. Those commands with overlapping names include the `bos` command, the `bosserver`, `upclient`, and `upserver` daemons, the `udebug` command, and the `scout` monitoring utility.

The NFS/DFS Authenticating Gateway allows NFS users to maintain an authentication mapping to DFS.

Interoperability with AFS

With their AFS 3.3 client software, Transarc added support to allow AFS clients access to DFS cells as well as to AFS cells. Accessing a DFS cell is nearly transparent, from the user's point of view. The DFS cell represented as `/.../<cellname>/fs`, for example, may be accessed as `/afs/<cellname>`. Figure 7 shows an AFS 3.3 client running on a system. Users on this system may access the AFS cell, `/afs/xyz.com`. They may also access the DFS cell using the path `/afs/dcexyz.com`. DFS users will access the DCE DFS cell using the path, `/.../dcexyz.com/fs`. AFS client users must use a special login command called `dlog`, which allows them to authenticate to DCE and access the DFS filespace. The AFS `fs` command has also been extended to allow AFS clients to list and modify DCE ACLs on files and directories in the DFS filespace.

In order to provide interoperability for the AFS 3.3 clients on the server side, the DFS servers must run a daemon named `adapt`, a protocol translator. This daemon intercepts requests from AFS clients and redirects them to DFS. Without this daemon, it is not possible for AFS

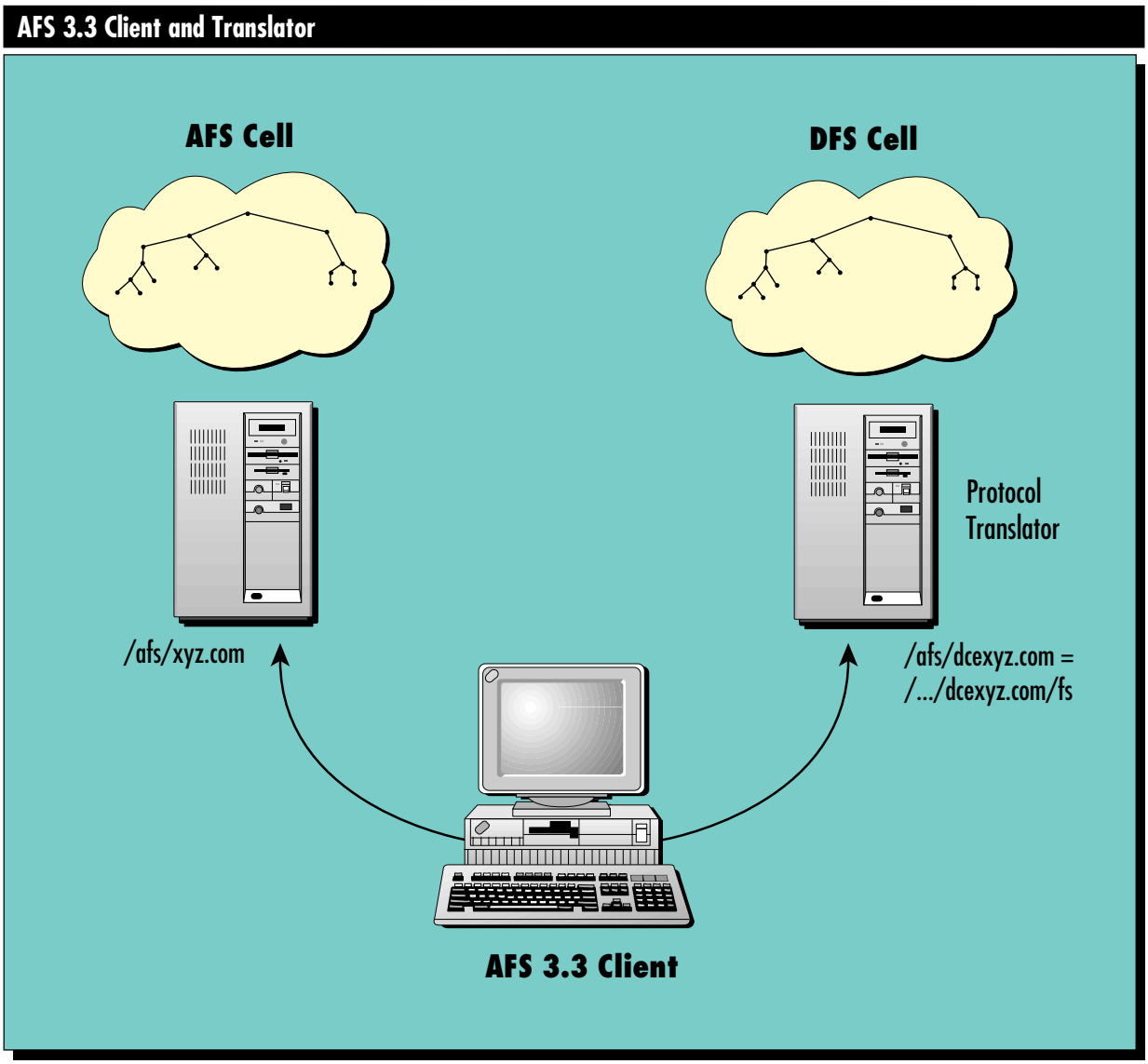


Figure 7. AFS client and translator

clients to access the DFS filespace. It is not possible to run this daemon on a system that is running an AFS server because it will attempt to send AFS requests to DFS, not allowing the AFS server to service the request. This translator is part of Transarc's AFS/DFS Migration Toolkit. In addition to running the translator, the AFS cell administrator must make the DCE DFS cell available by mounting it in the AFS filespace. This is done by executing an `fs mkmount` command for the DFS filespace root.

Summary

The interoperability options described make it possible for AFS and NFS users to use DFS. Legacy data in JFS filesystems can be easily shared between NFS and DFS. With the addition of gateways and translators, it is possible to further smooth the transition to DFS and allow a variety of clients to access DFS data.

Jean Pehkonen, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Internet: jean@austin.ibm.com. Ms. Pehkonen is a programmer in the DCE DFS Development department of IBM's LAN Systems Division. She has worked on IBM's AIX DFS products for the past five years. She has a BS in Computer Science from the University of Minnesota.

HACMP for AIX: Version 4.1.1 Update

By Daniel P. Cox

The newly announced HACMP for AIX, Version 4.1.1, has expanded support for the entire RISC System/6000 family—from the C20 to Scalable POWERparallel systems. It also supports AIX Version 4.1.4. The new features including the Visual System Manager, Cluster Node Snapshot utility, and Quick Configuration utility make installation, administration, and management easier than ever.

IBM's High Availability Cluster Multiprocessing for AIX (HACMP for AIX) now supports up to eight RISC System/6000 (RS/6000) uniprocessors, SMP processors, and Scalable Parallel (SP2™) thin or wide nodes in a highly available cluster. This function provides horizontal scalable growth and high availability in an open systems environment.

High availability, a computing configuration that recovers automatically from a single or multiple points of failure, provides better assurance against application and system downtime than standard hardware and software alone. Together, the HACMP software and a cluster of loosely coupled processors called *nodes* provide application availability by transferring control from a failed processor to a backup that has redundant capabilities. When a failure occurs or when components are restored to operation, HACMP executes test and recovery scripts tailored by the system administrator to specify the actions to be taken. HACMP software detects and recovers from failures of disks, disk adapters, networks, network adapters, and processors.

HACMP relies on the application, however, to make any failure recovery or fallover transparent to external users and client machines. If a node fails, nominal recovery time is approximately 30 to 300 seconds. Actual recovery time is a function of the system configuration, the application configuration, the size of the user's databases, and the user's recovery script (if any).

HACMP can be an alternative to upgrading a processor in the RS/6000 product line, such as an RS/6000 POWERserver™ 570 to 591. It provides horizontal cluster performance scalability with applications distributed across RS/6000 processors, sharing the disk and/or CPU resource of the cluster as independent or concurrent operating machines.

Figure 1 shows the HACMP system architecture.

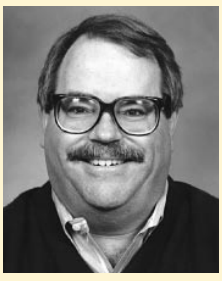
HACMP for AIX Version 4

The latest version of HACMP for AIX provides complete high availability for the RISC System/6000 from the entry-level Model 7009 C10 to the 7015 R21 models, as well as the Scalable POWERparallel (SP) thin and wide node models. It supports the RS/6000 SMP Models G30, J30, and R30 from 2-way to 8-way processors. Today, IBM is the only vendor offering this range of system support in the UNIX industry.

This new version, HACMP 4.1.1, provides enhanced concurrent access support for serial-link and Serial Storage Architecture (SSA) disk subsystems, ease-of-configuration enhancements for HACMP installations, a new High Availability for Network File System for AIX (HANFS for AIX) feature, and support for new IBM processors, disk subsystems, and devices.

Key features of HACMP include the following:

- RS/6000 hardware processor models can be mixed and matched.
- RS/6000 SP thin or wide node models can be used together.
- Disk subsystems from SCSI differential fast-wide, high-speed 9333 Serial Disk and Serial Storage Architecture 7133 can be used in the same cluster.



Daniel P. Cox

HACMP System Architecture

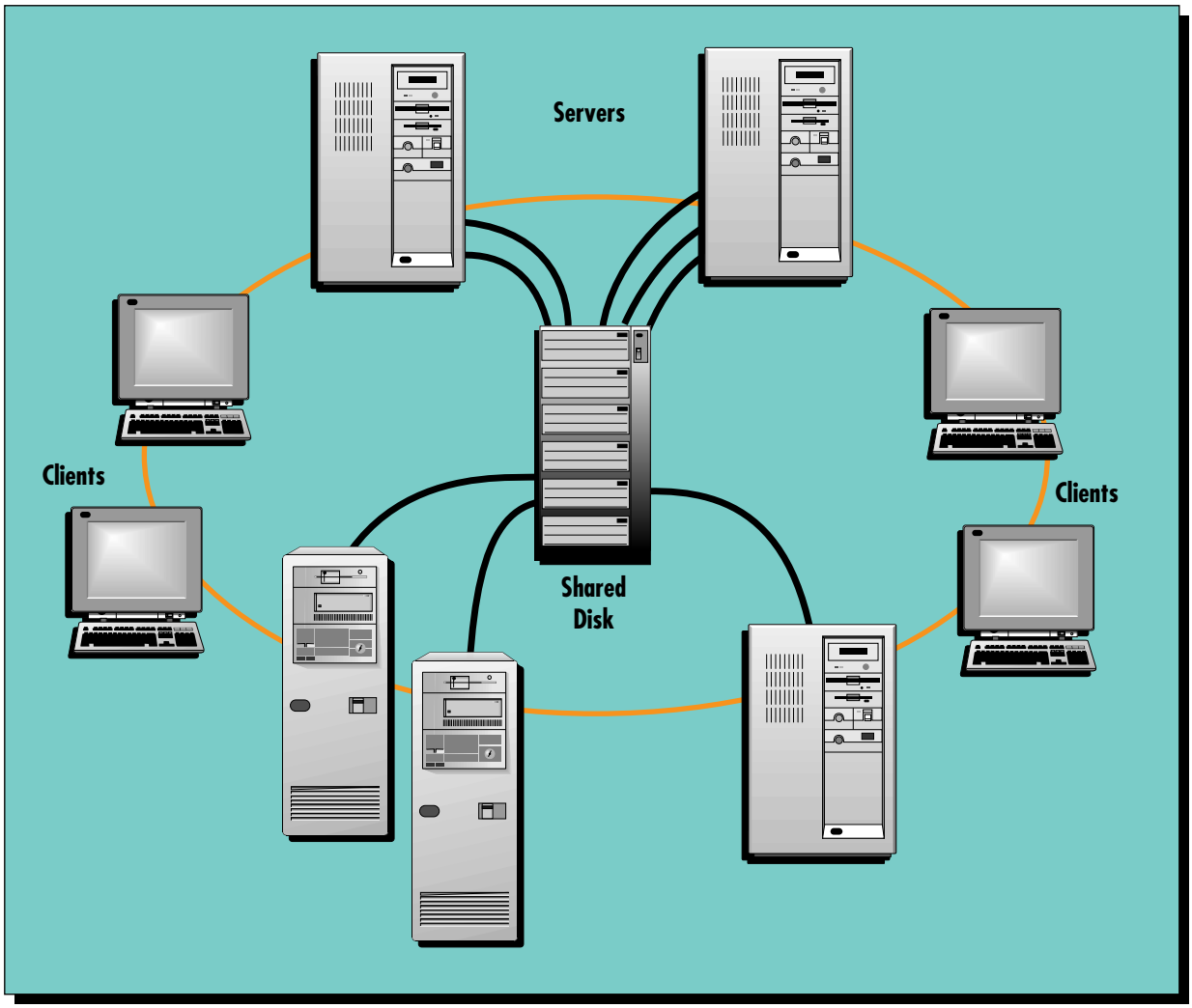


Figure 1. HACMP system architecture

- The hardware system configuration is flexible. Various LAN networks with either Ethernet, Token-Ring, Fiber-optic Data Distribution Interface (FDDI), or Fibre Channel Standard adapters with the same level of IP Address Take-Over (IP@TO) support can be used together.
- Users can operate a cluster of mixed AIX system levels, which optimizes configuration and application flexibility in a mixed cluster. This provides the flexibility to seamlessly migrate from previous HACMP Version 3 clusters on AIX Version 3 to HACMP for AIX Version 4 without taking the HACMP cluster out of service to upgrade the system and application software.
- System administrators can install, customize, and manage the HACMP cluster using the Visual System Manager (VSM), a drag-and-drop interface based on an AIX industry-standard from Common Open Software Environment (COSE).
- The Cluster Node Snapshot utility extends the HACMP cluster environment.
- Concurrent resource management for the 9333 High Performance Serial and the 7133 SSA disk subsystems now provides the horizontal scalability to produce up to an estimated 18,000 TPMs on an 8-node cluster of 8-way 7015-R30 SMP machines using the PowerPC 601™. This scalable performance has multisecond application recovery—unmatched in the industry for performance and availability.

HACMP for AIX Version 4 Highlights

Several highlights of this new version are described in the following sections.

High Availability Subsystem

HACMP 4.1.1 marks a significant advance in ease-of-configuration and system management enhancements for high-availability clusters. This subsystem provides the base services for cluster membership, system management, configuration integrity and control, and base services for fallover and recovery. It also provides programmers and system administrators with cluster status and monitoring facilities. The High Availability Subsystem contains all functions described in the remainder of this section except the Concurrent Resource Manager (CRM) and HANFS for AIX, which are separate features.

Visual System Manager

HACMP 4.1.1 makes it easy to install and configure an HACMP cluster. With AIX Common Desktop Environment (CDE) and a graphics display, the new Visual System Manager with its intuitive drag-and-drop interface allows system administrators to visualize the relationships between hardware and software resources that define the high-availability cluster, and drag screen objects from one location to another within a cluster.

Drag-and-Drop Graphical Interface

In addition to the standard SMIT interface for configuring HACMP clusters, HACMP 4.1.1 now provides a Graphical User Interface (GUI) for drag-and-drop configuration. The AIX VSM icons make it easier to visualize relationships between resources within the cluster while configuring new systems. The VSM template objects can be used to create multiple copies of the same configuration for multiple clusters. Users can load a saved configuration and make minor cluster-specific modifications to the image. VSM also makes migration and reconfiguration tasks easier.

The icon-based interface in HACMP 4.1.1 enables the user to filter subsets of data about the cluster configuration. By using a filter, users can focus on specific resources within a cluster, deliberately limiting the display to specific highly available resources, networks, or subsystems within a cluster.

Quick Configuration

The Quick Configuration utility, another new function in HACMP 4.1.1, enables installation and configuration of HACMP clusters using a few keystrokes. Selecting from a menu of hardware configuration options, users can quickly and easily select an appropriate hardware configuration, letting HACMP execute the logic necessary for configuring the new cluster, including set up of the initial AIX environment.

Common hardware configurations included in HACMP 4.1.1 are those using Redundant Array of Independent Disks (RAID), Small Computer Systems Interface (SCSI), and serial devices. With the Cluster Node Snapshot utility, users can create and save additional customized "quick configurations."

Cluster Node Snapshot Utility

The Cluster Node Snapshot utility can "capture" the configuration of an existing cluster in an ASCII representation or restore a captured configuration to a cluster, providing an invaluable tool for quick analysis of a cluster during migration or cloning of additional clusters.

Support organizations can use this facility to track changes to a particular cluster; system administrators might use it to maintain multiple cluster configurations, swapping configurations as needed.

The Cluster Node Snapshot utility can also be used with the HACMP GUI: any snapshot can be viewed as a simple text file or through its graphical representation in VSM, enabling quick analysis and diagnosis.

HACMP 4.1.1 now provides a Graphical User Interface (GUI) for drag-and-drop configuration.

Concurrent Resource Manager

HACMP 4.1.1 supports concurrent access to RAID, serial-link and SSA disk subsystems. This allows multiple RS/6000 systems (up to eight are supported for a serial-link-attached disk and up to two are supported for an SSA-attached disk) to access a shared disk simultaneously. HACMP has distributed locking facilities that support this function. For example, software such as the HACMP Distributed Lock Manager, can coordinate the use of the shared disk, which allows the workload to be spread across the cluster.

HANFS for AIX

High Availability for Network File System for AIX (HANFS for AIX) provides high availability for data accessed via the Network File System (NFS). It replaces HANFS Version 3 for those who are using AIX Version 4.1.4 or later. HANFS for AIX supports the configurations, the takeover, and the functionality of HANFS Version 3. The HANFS for AIX interface for administrators closely resembles the interface for the corresponding components of HACMP.

Two AIX systems running HANFS for AIX form a single highly available NFS server. This server can survive hardware and software outages—plus certain types of planned outages—that would make a single system NFS server unavailable. A

properly configured HANFS for AIX system that has no single points of failure can continue to serve data to NFS clients despite the loss of a single processor, disk, adapter, or network.

Additionally, HANFS for AIX allows one processor to take over for another with no disruption of service to the NFS clients—the NFS clients see only a temporary delay in the response by the server. HANFS for AIX supports all RS/6000 server systems in two node clusters along with a wide range of disk and network choices. One system can be the active server and the other an active standby. The workload can also be split between the two systems, in which each RS/6000 functions as the backup for the other. HANFS for AIX supports all NFS clients that are supported by AIX 4.1.4.

Cluster Manager monitors the state of the nodes, interfaces, and networks that comprise a cluster.

System Management

HACMP Version 4.1.1 provides enhancements for system administrators such as the following:

- Installation verification services allow the system administrator to confirm the installation procedures.
- Conversion tools ease the migration from a previous version of HACMP directly to HACMP for AIX Version 4.1.1.
- Comprehensive, data-driven scripts in HACMP minimize the need to modify the HACMP fallover scripts, simplifying cluster management and configuration.
- Cluster status and monitoring tools allow the system administrator to pass cluster status information to a single node in the cluster.
- Upgrade functions include version compatibility, a new function in this release. An existing cluster running HACMP for AIX Version 3.1 can be upgraded to HACMP Version 4.1 without taking the entire cluster off-line. During the upgrade process, individual nodes in the cluster can be removed from the cluster, upgraded one at a time, then reintegrated into the cluster. Nodes running both HACMP for AIX Version 3.1 and Version 4.1 can coexist while the rest of the nodes are upgraded.

For greater scalability, different instances of an application such as order entry can be run on multiple HACMP cluster nodes. CRM provides access controls and locking mechanisms for a database manager and an application in different nodes to share common disk storage. In addition, CRM provides several operator commands to control a shared disk environment.[1]

How Does HACMP Work?

HACMP 4.1.1 software automatically reconfigures the available replicated resources when hardware failures or outages occur, while allowing for hardware and software resources to be fully utilized when there are no hardware malfunctions in place. This fallover and recovery/restart is end-user generated through the supplied configuration and administrative scripts that can be modified to meet application configuration requirements.

In addition to providing high availability, HACMP 4.1.1 can also be configured to provide loosely coupled multiprocessing services. These configurations, whether concurrent access or parallel distributed access, allow a workload to be spread across multiple RISC System/6000 processors, sharing the disk and/or CPU resource of clustered processors.

HACMP for AIX focuses on a cluster as a set of RS/6000 and AIX resources. The cluster can include one or more of the following: disks, volume groups, filesystems, network addresses, or applications. This clustered approach, together with the capability of applications fallover and recovery/restart of HACMP 4.1.1, provides additional levels of high-availability processing for mission-critical applications, such as retail point-of-sale systems, banking credit verification, travel reservation, order-entry/inquiry, or health and hospital care systems.

Cluster Manager

The Cluster Manager (CM) component of HACMP monitors the state of the nodes, interfaces, and networks that comprise a cluster. It also provides highly available access to these resources and to critical disk data and software resources running on the cluster. To perform these functions, CM consists of three functions:

- **Heartbeat**. The CM monitors the nodes and network interfaces associated with a cluster using a heartbeat protocol. Heartbeat information received from other nodes in the cluster is interpreted and used to identify the failure of nodes, networks, and communications interfaces.

[1] CRM configurations commonly require an application that is designed to take advantage of concurrent access configurations. Not all disk subsystems sold with the RISC System/6000 servers operate in CRM environments. Consult your IBM Marketing Representative or Business Partner for current supported disk subsystems.

- **Membership**. The nodes in a cluster vote to admit or expel member nodes when changes in cluster state are detected. A node's membership status in the cluster determines its availability to manage cluster resources and to participate in failure recovery scenarios.
- **Synchronization**. When nodes are admitted or expelled, resources associated with those nodes can be moved from or added to other nodes, and these changes are made in a controlled, synchronized way. This ensures that the cluster resources are unavailable for the smallest possible time.

The CM and related services run on each of the nodes within the cluster. When the cluster is online, the CM monitors its health by sending heartbeat data over the interfaces that connect the nodes. If the CM detects a failure in part of the cluster, it reconfigures the cluster to enable it to direct data traffic through a path that avoids the failed cluster component.

The CM also triggers events based on changes in the state of the cluster. These events cause AIX shell scripts to run on the cluster, providing both system- and user-level responses to the event.

Cluster Configurations

An HACMP cluster can operate with up to eight nodes in numerous configurations, including hot standby, rotating standby, mutual takeover, and concurrent access. The following examples describe some configuration possibilities using the simplest 2-node cluster, but larger numbers of nodes can be readily clustered by defining the resources accordingly.

Hot standby: One node owns all resources; if it fails, the standby node takes over the resources. When the failed node rejoins the cluster, the resources are returned to the original node that owned them.

Rotating standby: This is identical to a hot standby, except that when the failed node rejoins the cluster, the resources are not returned to the node until the standby node fails.

Mutual takeover: Resources are divided among the nodes; some are owned by each node. If either node fails, the other node takes over all the resources. When the failed node rejoins the cluster, the resources are returned to the original owning node.

Concurrent access: Two or more nodes are active simultaneously, sharing the same physical disk resources. Both nodes own the disk resources. Other resources are divided between the two nodes, each owning some of them.

HACMP for AIX, Version 4.1.1

Some features of the latest version of HACMP include the following:

- Full concurrent access support for all IBM types of disk subsystems, including RAID subsystems, Serial-link disks, and Serial Storage Architecture disks
- Common support across the entire RISC System/6000 family, from C20 to Scalable POWERparallel systems
- Additional system management functions that make it easy to use for cluster creation, administration, and replication
- Visual System Manager with a graphical interface for defining and customizing clusters
- Cluster Node Snapshot utility that captures a specific cluster configuration and makes replication of clusters easier
- Quick Configuration utility that allows installation and configuration of HACMP clusters using just a few keystrokes

Together these features provide more flexibility and extendability as well as enhanced installation, administration, and management features.

Resources not owned by each node are designated as takeover. If either node fails, the other node takes over all resources. When the failed node rejoins the cluster, the resources are returned to the original owning node.

Clusters of up to eight nodes offer great configuration flexibility:

- One node can back up seven nodes.
- Although eight nodes can operate during peak operations, that number can be less during non-peak hours with minimal interruption of service.
- All resources could go to one node upon failure or be split among several nodes.
- Using cascading takeover, resources could be passed to either the primary standby or to its secondary standby node.

It is important to remember that an HACMP configuration is simply a definition of how the resources are defined on each node within the

cluster. The number of resources, combined with how they can be assigned, gives the cluster designer nearly unlimited flexibility in laying out an HACMP configuration.

Alternate Data Access Configurations

HACMP supports alternate data access configurations, addressing the need to run applications that share the same data on multiple nodes within a cluster.

In a concurrent access configuration, each multiple system has its own path to the disks holding the data. Any system in the cluster can physically access the data, but the systems must cooperate to ensure coordinated accesses to the data and to preserve data integrity. For these configurations, the HACMP Concurrent Resource Manager provides the necessary access controls and locking mechanisms for a database manager and an application on different nodes to share access to the common storage. Such configurations provide a high degree of scalability, limited only by the number of systems that can simultaneously attach to the shared disks.

HACMP provides the event management and application fallover in 30–300 seconds for simple applications.

Additionally, HACMP provides the mechanisms to allow a node to take over the function of another that fails. Since the shared disks can be accessed from any system, HACMP can restart the application that was running on the failed system on one of the surviving nodes.

Another solution to the problem of multiple systems accessing the same data is to have the data logically partitioned by the system within the cluster with a Database Manager (DBM) providing access to both partitions. Each system within the cluster has sole access to a dedicated partition of the total data set. The system nodes themselves are interconnected through a fast communications link. When a request is made to a particular system, the system decides whether or not it can locally access the data. If not, either the request is forwarded by the DBM to the owning system, or the data is retrieved by the DBM from the owning system.

This configuration is called *parallel distributed access* because the individual systems can process requests in parallel and access the distributed data across the cluster. The cluster scales to the degree that the communications path between the system does not become a bottleneck.

HACMP provides two advantages to a cluster built around parallel distributed access. First, if the disks can be physically connected to two or more systems, then HACMP can restart an application from a failed system on another system that can access the same disks. Second, HACMP's Distributed Lock Manager provides locking mechanisms for a database manager running on multiple nodes, which allows coordinated access to the shared data.

In both types of data access configurations, HACMP uses the mirroring capabilities of the base AIX operating system to make the data highly available and accessible. The configuration flexibility of HACMP allows users to choose the cluster topology and database manager that best suits the requirements of their computing environment. In fact, HACMP can support both types of data access within a common cluster.

Why Use HACMP?

HACMP is designed to maintain a highly available application and data server for those applications that cannot fail for any length of time. On average, a system outage is approximately 4.5 hours and will cost approximately $200,000 in lost revenue.

HACMP provides the event management and application fallover in 30–300 seconds for simple applications. It may take longer if complex database processor transactions or large database files must be re-created because of a system catastrophe caused by a natural disaster such as a flood, fire, earthquake, or a man-made disaster such as a power outage caused by terrorist acts.

The HACMP control system provides flexibility for users to design their own system for event control and recovery/restart that meets their processing and management needs across a wide spectrum of hardware and software offerings both by IBM and other OEM hardware and software providers. Some key points below address the HACMP benefits.

Improves Application Availability. HACMP for AIX improves systems reliability of RS/6000 systems in high-availability application environments. HACMP for AIX can be added to RS/6000 and AIX application configurations. The incremental costs to implement HACMP for AIX compared to reinvesting in new and additional hardware provides an inexpensive solution to increased system availability that may be required in certain business-critical application areas.

Provides Scalability. HACMP for AIX can extend the current system configuration. By using the clustering facility, the number of users on the system can be expanded. HACMP provides scalability without replacing current hardware and software solutions if processing capability exceeds the current system capacity.

Scalability is an important use of HACMP for AIX in a mutual takeover configuration, which extends

system capacity by splitting the workload on up to two to eight systems. The control system provides increased system availability as the number of users and applications increase, and system outages become more critical to the business process.

The Concurrent Resource Manager subsystem also provides availability and true horizontal scalability. This is apparent when a single image of the data is required for the application to execute and the data cannot be partitioned as in the mutual takeover configurations.

Provides Backup/Recovery. HACMP for AIX provides the basic service when business-critical applications require availability and recovery/restart. This service can be defined using any HACMP for AIX environments with multiple disk and network attachments available. In single-system environments, no shared backup is available when a processor failure occurs. Based on the user's definition of the configuration, an AIX HACMP for AIX cluster can operate in many different configurations, such as an Idle Standby or Simple Fallover to an idle machine.

HACMP for AIX provides additional capability in its inter-LAN network fallover. When a primary network fails, HACMP for AIX will fall over to a backup LAN on a separate LAN topology. For example, the primary LAN is Token Ring and the fallover backup LAN is Ethernet. This applies to disk subsystems as well, where the primary disk device is 7135-110 RAIDiant Array and the backup mirrored device is the 9333 High Performance Serial Disk subsystem.

Provides Investment Protection. HACMP for AIX can provide users with a no single-point-of-failure in an HACMP for AIX server configuration, without the extensive investment required for fault-tolerant hardware. HACMP for AIX and RS/6000 have many hardware redundant features. HACMP for AIX requires only replicated devices to eliminate a single-point-of-failure.

Enables Horizontal Growth. HACMP for AIX allows the total application performance to expand without removing or replacing the current processor base. It provides scalable growth by increasing the number of RS/6000 processors as an alternative to vertical upgrades from one model to another. When an RS/6000 model is reaching capacity with the current workload, adding a second or even third processor can be almost transparent in an HACMP for AIX Mutual Takeover configuration.

Applications and data can be transferred from one RS/6000 to another, without the need to migrate data or port applications. A variety of RS/6000 models can be attached in an HACMP for AIX cluster with the disk subsystem as the common link.

Provides An Alternative to SMP. HACMP for AIX provides an alternative to stand-alone SMP configurations.

When an application or a business processing environment grows and the number of users increases, the requirement for availability of the processing system can outweigh the increased speed of the application offered in today's symmetric machines. HACMP for AIX provides a solution for both requirements: application scalability and higher availability.

Application scalability can be handled through Mutual Takeover or Partitioned Workload, where one measure of scalable performance can be achieved. It can also be handled in a Concurrent I/O Access environment that uses a single-system data image. These HACMP for AIX configurations provide availability and recovery/restart for any number of system outage conditions, such as disk, disk adapter, LAN network, LAN adapter, or processor node failure.

HACMP for AIX configurations require minimal customization when failures occur in either the operating or application system.

HACMP/6000 configurations require minimal customization when failures occur in either the operating or application system.

I**ncreases User Productivity**. High availability allows business-critical processes to operate when components or subsystems fail or during maintenance downtime.

HACMP for AIX allows users to shutdown a system for scheduled maintenance and restart the system with automatic resynchronization of applications and data from the backup or fallover system. When a system or subsystem fails, HACMP for AIX provides automatic fallover with application and data restart. In this case, the failing system can be repaired and put back into service, and will relink into the HACMP for AIX cluster without loss of data.

HACMP for AIX also can detect multiple events. Depending on the event (such as a network failure), a unique fallover scenario can be executed, such as reinitiating service on a replicated network adapter that does not abort the cluster processor, but shifts service to the replicated or backup network adapter.

Allows Minimal Impact on End Users. The high-availability features of HACMP for AIX allow system administrators to make changes, apply upgrades, and provide scheduled hardware and/or software maintenance and backup services with minimal impact to the end users.

A backup system allows end users who are normally assigned to the machine that is out of service to continue their work. HACMP for AIX automatically integrates the processor back into the cluster and resychronizes the applications and data onto the repaired machine.

Enables Horizontal Growth. HACMP for AIX enables non-disruptive horizontal growth in the server complex through a loosely coupled cluster implementation. It provides scalable growth through its cluster configuration environments.

For scalable growth, Mutual Takeover configurations can be created that result in almost 100% CPU efficiency in a cluster of two or more processors. These configurations deal with applications that can split data independently between machines—sharing only data that is common for all applications.

HACMP/6000 uses UNIX interfaces that make it easy to port to future UNIX-based systems and subsystems.

In data sharing, LAN and filesystem overhead can reduce CPU efficiency. If data must be shared, the Concurrent I/O Access environment can be used, but it is less efficient than a single system because distributed locks must be maintained between cluster processors.

Provides Easier Portability. HACMP for AIX uses UNIX interfaces that make it easy to port to future UNIX-based systems and subsystems. HACMP for AIX uses industry-standard TCP/IP communication protocols as the transport mechanism among machines in the cluster as well as clients that may be attached to one or more cluster processors. HACMP for AIX can use multiple TCP/IP interfaces on the current RISC System/6000 adapters, such as Ethernet, Token-Ring, Fiber-optic Data Distribution Interface (FDDI), and Fibre Channel adapters and switches.

Provides Incremental Systems Facilities. HACMP for AIX provides incremental systems facilities—for adding new disk subsystems and adding new processors without stopping the cluster—to the existing application base for 8-way scalability through clustering as well as high-availability fallover.

HACMP for AIX allows different RS/6000 models to be clustered together to meet users' requirements for processing capacity and availability. Users can upgrade a current machine vertically (for example, 7013-591 to 7013-591) or purchase a second machine (for example, an additional 7013-580) and cluster the application and data workload onto both machines. This option increases CPU efficiency, improves system performance, and provides a higher level of availability than can be achieved in a stand-alone configuration.

Provides Common Interface. HACMP for AIX uses System Management Interface Tool (SMIT) for the basic installation, configuration, and administration in an HACMP for AIX cluster environment. It also uses SMIT to configure and update a cluster from a single node, providing a common management focal point from previous HACMP for AIX configurations.

Provides Cluster Monitoring. If HACMP for AIX detects a server failure, it provides cluster monitoring and automatic fallover to backup processors. It alerts either a local or a remote services console if a cluster fails. HACMP for AIX can use the NetView/6000 control system, sending a Simple Network Management Protocol (SNMP) alert record to the NetView/6000 host reporting on cluster status if an out-of-service condition occurs. If NetView/6000 is not used, this can also be monitored in the HACMP for AIX cluster console service.

Allows Script Customization. System administrators and application programmers can extend and customize HACMP for AIX scripts for use in configuration and applications to meet specific availability and/or scalability requirements. Although scripts are provided with the system, it is necessary to customize them to meet any specific requirements. HACMP for AIX now includes comprehensive scripts to ease the customization process. Users can define pre- and post-scripts that will minimize modifications to the HACMP for AIX-provided scripts.

Summary

HACMP for AIX provides flexibility and scalability across all AIX systems. It runs on all AIX versions and supports all the RISC System/6000 servers including the 9076 SP models. It supports the largest number of active processor nodes, most disk subsystem variety, all LAN topologies, and provides the richest set of Relational Database Management System (RDBMS) support by the RDBMS vendor community.

Daniel P. Cox, IBM Corporation 11400 Burnet Road, Austin, TX 78758. Mr. Cox is the brand manager for clustered systems in the RISC System/6000 Division. After joining IBM in 1968 as a programmer in the IBM San Jose Laboratory, he has held positions in development programming, systems engineering, product planning, and management. He has been involved with HACMP for AIX since its beginning—in both planning and implementation. Mr. Cox has a BS and an MS from San Jose State University.

Using PCMS to Control AIX Software Development

By Sohail Haque

One of the challenges in managing medium to large software projects is the inability to retrieve real-time, "up to the minute" information on the current status of the project. Use of team meetings, paper-based status reports and E-mail are typical methods that project managers use to gather new information. This article shows how SQL Software's Process Configuration Management Systems (PCMS) can be used in software development on IBM's RISC System/6000 using AIX.

Software engineering is constantly being challenged to improve itself. Several panaceas have been proposed over the last ten years. First was the development of rigid requirements, design, and coding phases, combined with upper-Computer-Aided Software Engineering (CASE) technology. Now, object-oriented analysis, design, and programming languages, plus the concepts of rapid application development and prototyping are currently in vogue to improve software productivity.

Throughout these methodology wars, one very basic problem has not been addressed: the lack of centralized information. Software engineering organizations lack centralized data that provides both the engineering team and management with the status of all changes and activities related to software.

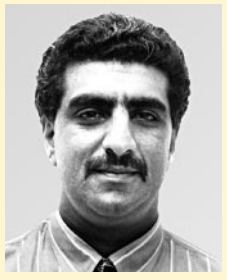
This real-world engineering information contains the status of all defects, enhancement requests, and customer calls, incorporated with the physical changes of various design models, requirements documents, and physical source code. It is gathered via the approaches found within Process Configuration Management Systems (PCMS). PCMS collects real-world engineering information by combining process workflow behavior; change initiators such as defects, enhancement requests, and new requirements; and physical change activity on the corresponding application objects with help desks and customer support databases.

What is PCMS?

PCMS is a set of integrated products developed by SQL Software to enable enterprises to leverage their financial investments in engineering technology and customer service. Although SQL Software offers a scalable set of products that are fully integrated within themselves, they are also open and can be woven into customer environments and frameworks. They are built using industry-standard technologies, such as relational databases, to protect customer investments.

PCMS is an active system that combines a process workflow engine with comprehensive change and problem management, version management, build automation, and help desk products. These components enable PCMS to support the entire life cycle of both software and system development including software production and manufacturing, documentation, and hardware. It enables the resultant systems to be enhanced and maintained, while providing comprehensive customer support services.

PCMS represents a new breed of industrial-strength configuration management systems that provide concurrent engineering and build


Sohail Haque

support for the developer as well as important status, cycle time, and customer service reporting for management. Unlike traditional tools that passively log events without providing any real control or anticipation of problems, PCMS actively supports project cycles including development, production, maintenance, and customer support.

PCMS enables managers, developers, and support staff to achieve higher quality, and reduced development time and costs.

PCMS Process Engine

An advanced process engine, common to all PCMS products, enables project processes and their interrelationships to be modeled. Because of these relationships, objects (such as source files) and composite objects (such as executables and documents) can be managed, cross-related, audited, and reported on. PCMS adjusts effortlessly to different project life cycles and new methods and is readily adaptable to corporate processes. PCMS process models can be used and replicated across a large enterprise, dispersed projects, and between contractors.

Figure 1 shows the components of the PCMS product set.

PCMS*VMB:	Comprehensive version management, software build, release and distribution product to support developers and management
PCMS*CTS:	Enterprise-wide change and problem management product
PCMS*Helpbench:	Solution-based help desk
PCMS*PCwin:	Seamless access to PCMS (with security controls) from a PC Windows and Windows NT environment
PCMS*ART:	Project-level archive, retrieval, and transfer system
PCMS*NLS:	Network library support for mixed UNIX, VMS®, Microsoft, and Windows NT solutions; provides networking to the process engine
PCMS*SII:	System integration interface including API and pre- and post-event trigger callouts

Figure 1. PCMS product set

Optimizing PCMS Performance

A PCMS network can be divided into library nodes (those on which PCMS item libraries reside) and non-library nodes (the remainder of the nodes). The network can be configured to take advantage of the available computing resources. PCMS*NLS provides networking facilities to permit operations across both homogeneous and heterogeneous environments. It can also be used to spread the processing load.

Using Oracle® as an example, Oracle processes should execute on the fastest node in the network, and if possible, have no PCMS logins on it. Operating system parameters should be optimized with as much RAM as possible for each library node in the network. If a single-user workstation is used on the network, the working set sizes can be significantly increased to reduce paging.

Creating UNIX Accounts for PCMS and Oracle

Certain accounts must be created before installing any PCMS and Oracle products.

Two special accounts, pcms and oracle, should be created—each in a group by itself with no other accounts in the same group. These two accounts will own the pcms and oracle files. Apart from this, they should only be used for Oracle and PCMS administration functions as described in *PCMS Database Administration Guide*, which comes with the system. An additional directory may be required to hold the Oracle database files. Before creating a PCMS base database, the correct Oracle runtime must be installed.

After the installation is complete, the next step is to load the Flight Simulator (FS) demo, which has the process model for software development and change. This process model will be used in the rest of this article.

Configuring PCMS for Projects

PCMS has a screen-based fully configurable process model that is known as a *control plan* when it is completed. This demo product's control plan includes documentation. This article will discuss a portion of the full control plan.

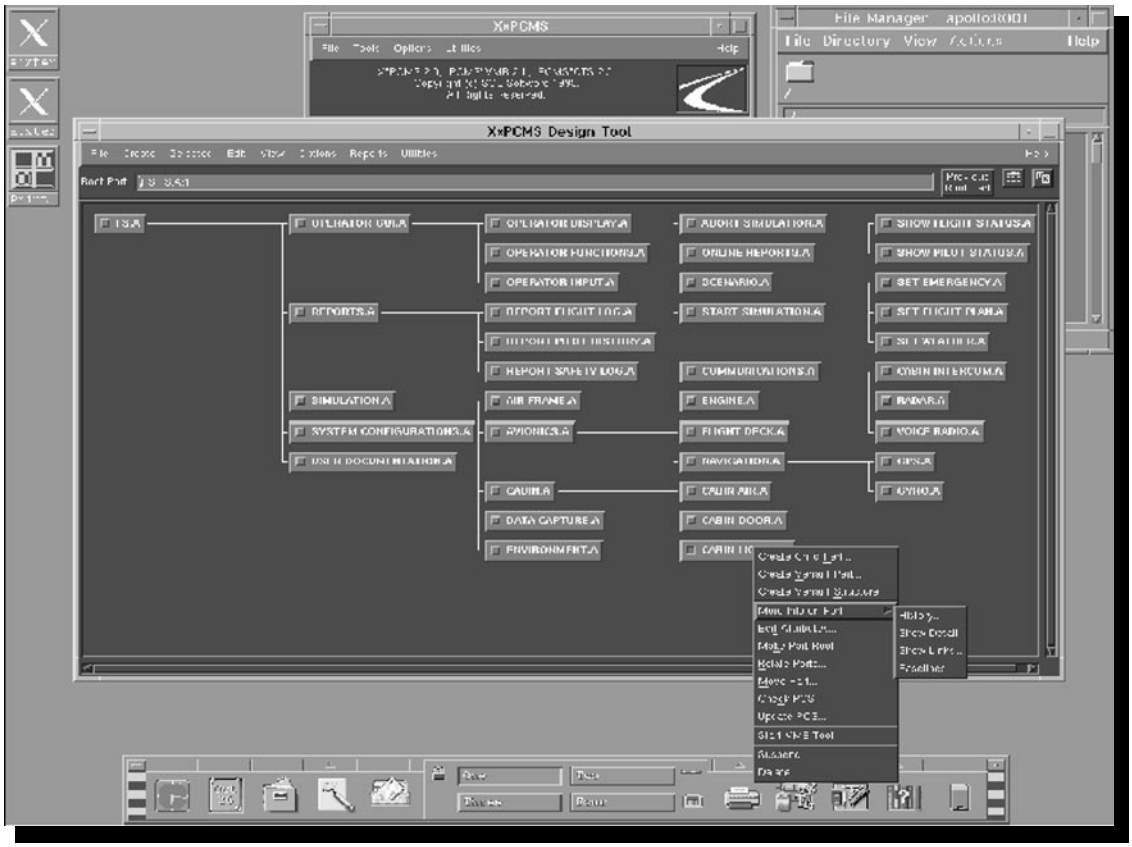


Figure 2. Product structure for Flight Simulator

Product Breakdown Structure

Product is the root node at the top of the PCMS product structure and the highest-level design component. The product is usually a family of products, a single product, a project, a system, or an application. A system can be structured into subsystems; systems and subsystems can be managed in a similar way.

In the Flight Simulator product, each subsystem consists of *modules*, the smallest units to be programmed or engineered.

Figure 2 shows the breakdown of the product structure for a Flight Simulator.

Design Parts and Product Items

Within the project, many product items will be the same type. For example, a project will have many source files and module specifications that are classified for control purposes into item types. For example, source files could be defined as type SRC; design specifications could be defined as type DS. Physical items could have life cycle states such as created, reviewed, audited; they could also be modified during product development.

In the product breakdown structure, design parts can have classifications similar to items. A documentation plan can be constructed using the design part categories and item types. The mandatory and optional item types for each category of design part can be defined. For example, a module must have a module specification and source code. The documentation plan provides a mechanism to ensure that a product represented by a product baseline is complete, as shown in Figure 3.

Activities and Responsibilities

When a set of development activities is completed, the result is a deliverable item or a modified product item. An activity may not result only in a new or modified product item, but it could also be a change in its life cycle state.

The life cycle of a product item is a set of state transitions. For example, a source file has the following states:

State 1: Under work. The product item is planned or it is being worked on under the control of the engineer. The engineer completes the work and moves the item revision to the next state.

State 2: In review. From the engineer's point of view, the product item is completed and

ready for review. The item revision is mailed to the lead engineer, where it goes into the lead engineer's "To Do List" or pending tray. The lead engineer looks at the source and related documentation, then moves the item revision to the next state.

State 3: Approved. The product item has been checked and can be used or released.

The person who performs an activity on the life cycle has a role on the project. Development work is performed by those who have a role within the product sub-tree being developed; for example, engineer for "Operator GUI."

The role is the part you play within the development process. For example, you may be the engineer for your own Show Flight Status module and a tester for the PC Controls subsystem. Your user role is the "hat you are wearing" when you perform an activity on a physical item.

Since personnel and circumstances may change, PCMS provides the necessary administrative functions to assign and maintain users' roles within the design part structure, and to delegate responsibilities for functions such as role assignment and design part creation.

The development process includes the following:

- Analysis and design until the product is ready to be built
- Change
- Build and test loop
- Component assembly into product release sets

Product Item Type	Description	Comments
DOC	Any project documents that are not defined explicitly in the control plan, such as progress reports, reference information, interface definitions	
MIN	Minutes from project meetings, such as technical, review, or quality meetings	Major actions from these meetings may be held as change requests or internal change documents to track progress
RS	Requirements specification, usually associated with the product or subsystem	For interface to requirement tools define item types `RQT_RDR, RQT_RSD`
UG	User Guide(s)	
FS	Functional specification	
DS	Design specification	
MS	Module specification to describe the module being developed	Could be generated from CASE tool
SRC	Source files, C++, ADA, BASIC	Define formats for source items in the CM plan
OBJ	Object code files compiled from SRC	By keeping this item under control the build process can do minimal rebuilds by recompiling only those items that have changed
TPL	Test plan	
TPR	Test procedures	
PBL	Product baseline item	Used to track the details and handover of a baseline during the product development, such as a system baseline from development to systems test to customer acceptance testing; it can exist at any level and will reside at the top-level design part of the product baseline
DI	Development item	For engineers and developers to keep notes and supporting project documentation

Figure 3. Product items with a workset for Flight Simulator

The supporting processes may be used for activities that are repeatedly used during the development activities.

Role names that describe those involved in the processes are generic. In smaller or less formal organizations, one individual may cover more than one role as defined in this section. The development process breaks down into phases, each with its own set of deliverables: some to the end customer and some to the next stage in the development process. Deliverables from each stage in the process are the main objects to be controlled and managed by PCMS as product items. After each phase, a product baseline should be taken to capture a snapshot of the documents and project files. This baseline can serve as a reference point for the next phase, and management reports can show the progress.

Figure 4 shows the roles involved in the development processes.

Figure 5 lists the tasks in the development process and the roles that are either involved or responsible.

Development Process Roles

TM	Test Manager
DM	Development Manager
LE	Lead Engineer
LA	Lead Analyst
RM	Release Manager
ENG	Engineer
BUI	Builder
AN	Analyst
DE	Design Engineer
AUT	Author
TE	Tester
CM	Configuration Manager
ITM	Integration Test Manager
STM	System Test Manager

Figure 4. Roles in the development process

DEVELOPMENT TASK	RESPONSIBILITY*							
	AN	LA	ENG	LE	TM	DM	RM	CM
Analysis								
Prepare Analysis docs	R	I						
Review Analysis docs	I	R		I				
Prepare model	I	R		I				
Review analysis deliverables	I	I		I				
Make analysis Baseline	I	R				I		
Verify Baseline contents				I				
Develop product acceptance criteria	I	I		I	R	I		
Review product acceptance criteria	I	I		I	I		R	I
Design								
Prepare Design docs	I	I		I				
Prepare Int and System Test docs		I		I	R	I		
Prepare Design model	I	I		R				
Review Design model	I	I		I				
Review Design docs				R	I	I		
Review Int and System Test docs		I		I	I	I	R	
Extend PCMS design structure				R				I
Prepare data model						R		
Make functional Baseline	I		I	R				
Verify Baseline contents			I	I		R		

*I=Involved, R=Responsible

Figure 5. Development process tasks

The product life cycle represents the state of a product at any point in time. After a subsystem or the whole product (for example, Flight Simulator) is completed, a product baseline file is created and included in the Product Baseline (PBL). The item representing the PBL documents the transition from the development team to the release manager and the test teams.

The Product Baseline uses the life cycle shown in Figure 6. As the baseline is released into the various environments, the PBL file moves through the life cycle. The product baseline should be created at the product or subsystem level of the design part structure.

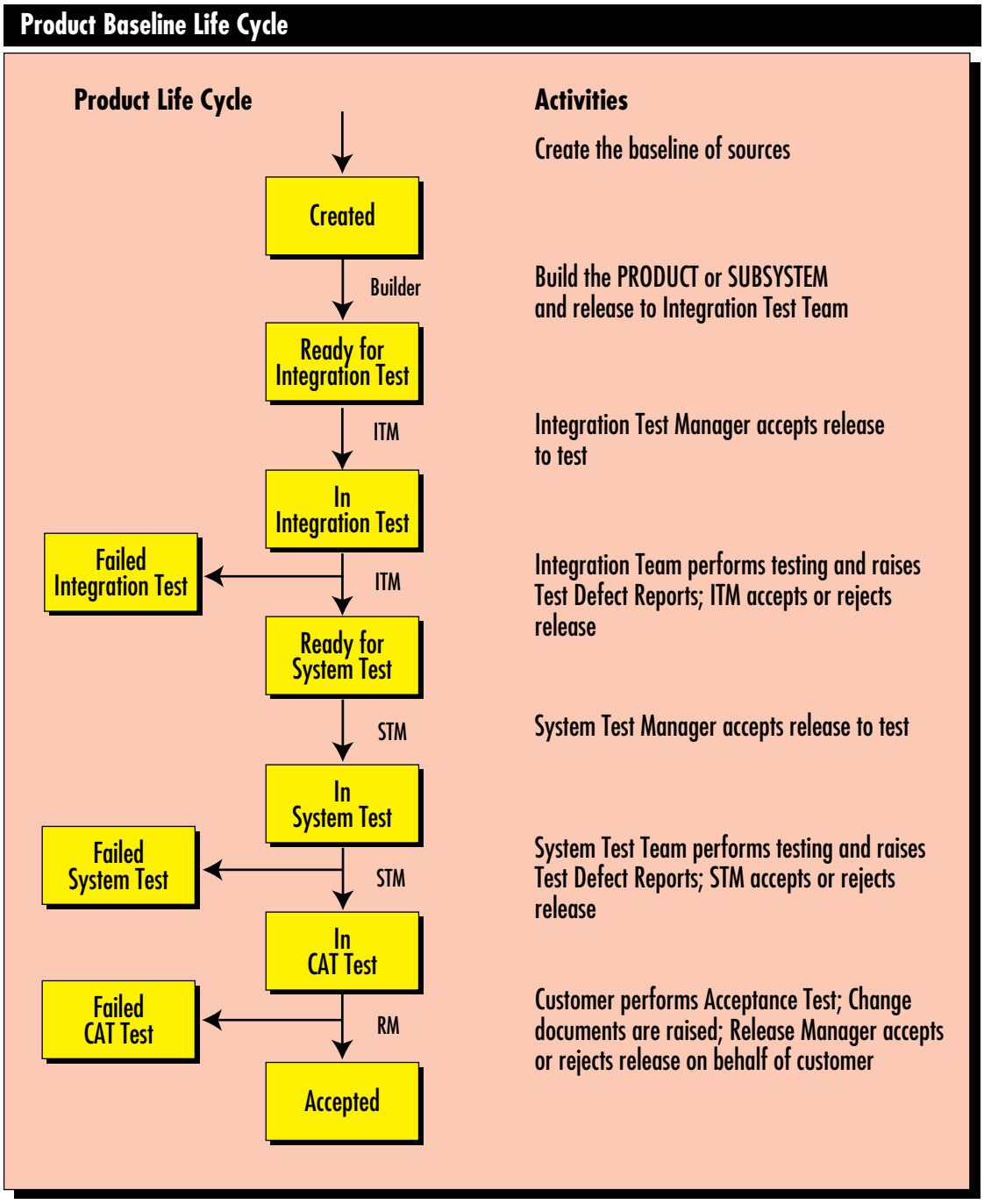


Figure 6. Product baseline

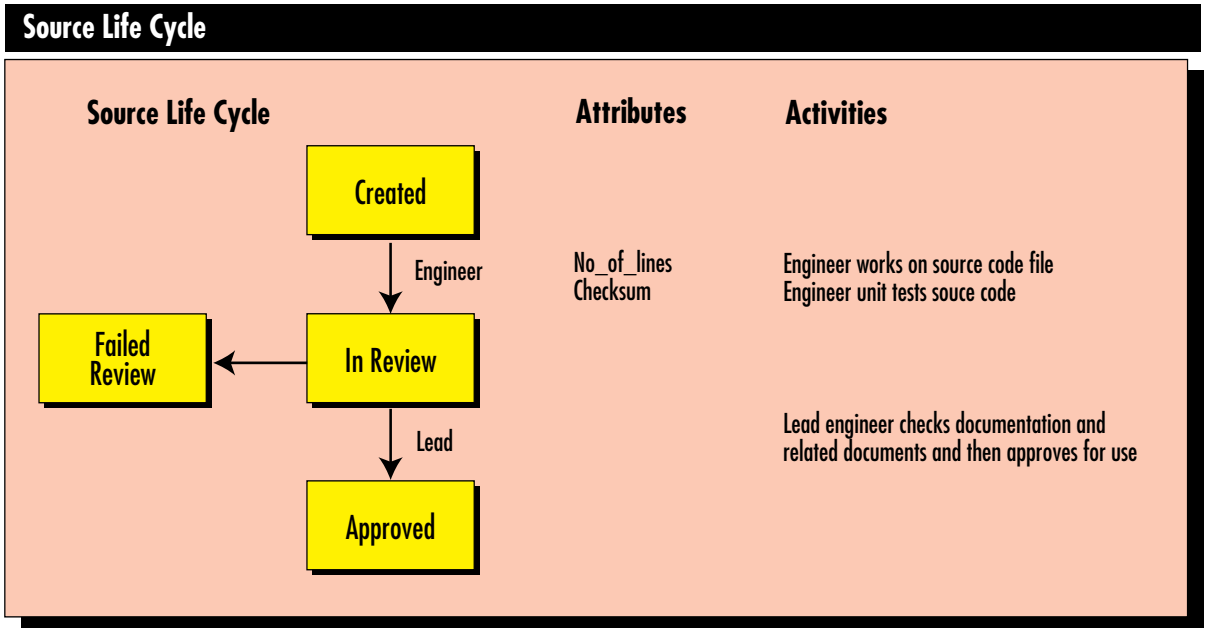


Figure 7. Source life cycle

Source Life Cycle

Source files are used for building software products. For PCMS build purposes, a single PCMS item type may differ in item formats. For example, one SRC item may have a C format, another C++, and another ADA. Using the PCMS browse and edit scripts, specific editors can be associated with PCMS items, based upon their item type and format. The following are associated item types:

SRC	Source files
INP	Source screen files
DAT	Data files
DBS	Database script files

A product item will have a life cycle that is a set of state transitions, such as a source file (SRC), shown in Figure 7.

A PCMS*API event may be used to derive the attribute values `no_of_lines` and `checksum`. Other metrics attributes can be added.

The Maintenance Process

Typically, maintenance is concerned with maintaining development releases and performing bug fix and patch releases. In PCMS, maintenance can be effectively modeled by using the release baseline and capturing all amendments within the change documents. Then amendments can be applied to the original release baseline via the set of change documents using Create Revised Baseline (CRB).

Figure 6 shows this scenario within the development process. Maintenance, however, must address issues arising from the help desk and support environment. Collating changes into appropriate work packages provides a convenient mechanism for planning maintenance activities. It enables both problems and requests to be processed and placed in a work package that represents the work to be done and the time frame.

The change management process and the use of work packages is important to maintenance. An overview of the change management process and an example of a PCMS life cycle is shown in Figure 8. The roles in the change control process are as follows:

CRV	Change Reviewer
CA	Change Assessor
CL	Change Lead Engineer
CE	Change Engineer
CB	Change Board
CM	Change Manager

Reporting

SQL Reporting generates comprehensive standard reports. These reports allow you to track

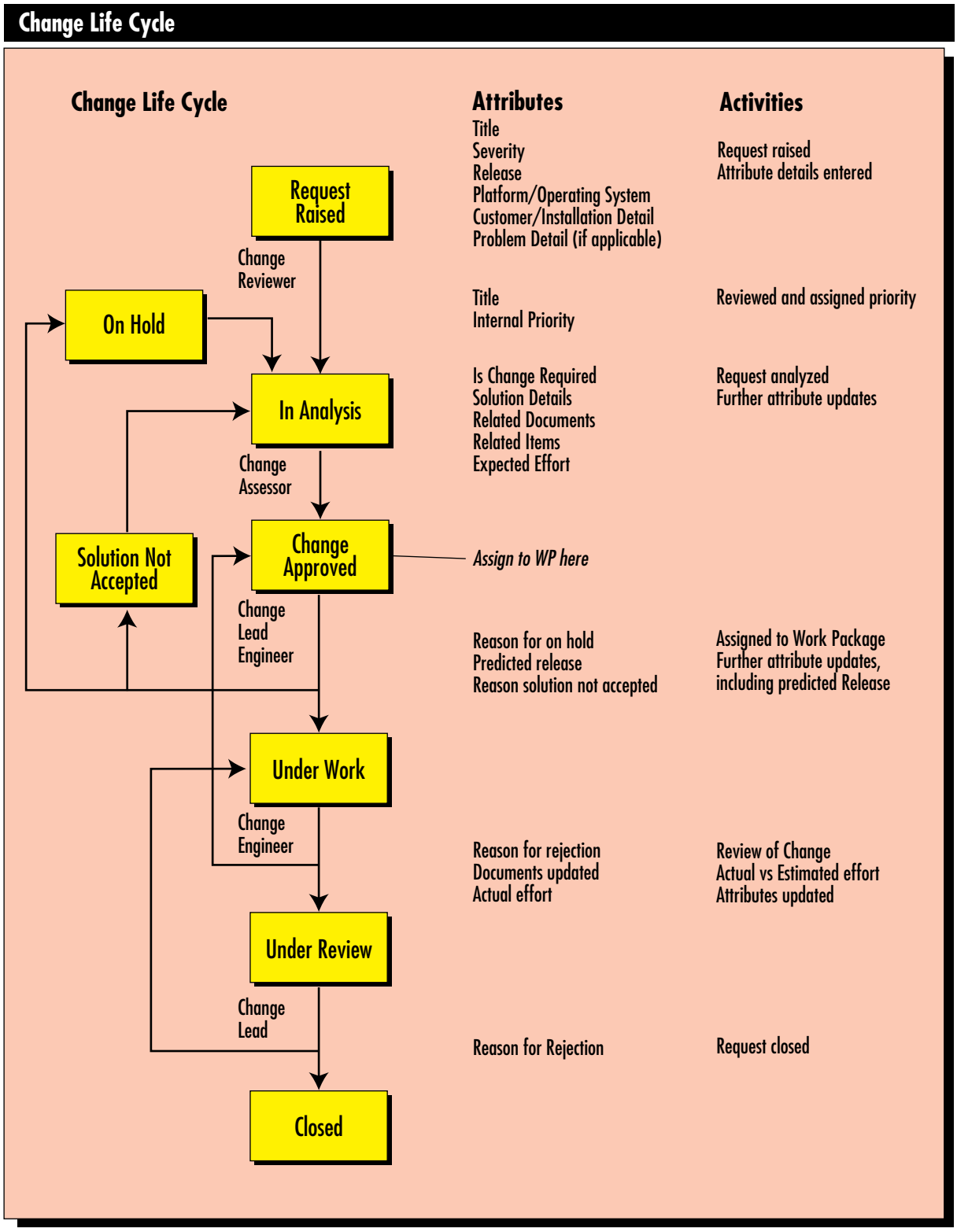


Figure 8. The change management process and PCMS life cycle

the history of development objects and change documents, such as defect reports and other statistics from the relational database used by PCMS. Since PCMS is an active system, it provides a dynamic rather than a static view of your change process. Therefore, managers can review the status of any part of the project at any time in order to assess impact, take actions, and plan ahead.

Benefits

Usually when a change request reaches a particular state, change activity begins against various types of information to correct that change request. This could be a combination of design documentation, source code, user documentation, test plans, and test suites, as shown in Figure 9.

Connecting and packaging this changed information against the change controls automates the correction process. Creating a change package, grouping together multiple physical changes as a change package, and tying this change to a particular state in the change life cycle, can offer important benefits including the following:

- Automation of changes from development to test to release
- Faster testing turnaround times
- Reduced compilation and linkage cycles
- Improved quality over the contents of a release
- Faster generation of release notes and product errata

Migration often fails because of poor change packaging. Testing builds fail because the correct set of changed objects cannot be created. Changes are left behind in development directories, and engineering hours are lost trying to diagnose the reason that test builds fail. This increases the impact of the compile on the machine and adds effort to each testing cycle. Much wasted time is eliminated by repository tracking and controlling these packaged changes.

If the dependencies between the programming, quality assurance, and documentation are tracked and packaged, overall quality improves with each change activity. In order to create the test suite, the test group must have a solid understanding of the changes to the design and the code. Likewise, the documentation group must know what design changes will occur so that the documentation matches the operation of the product.

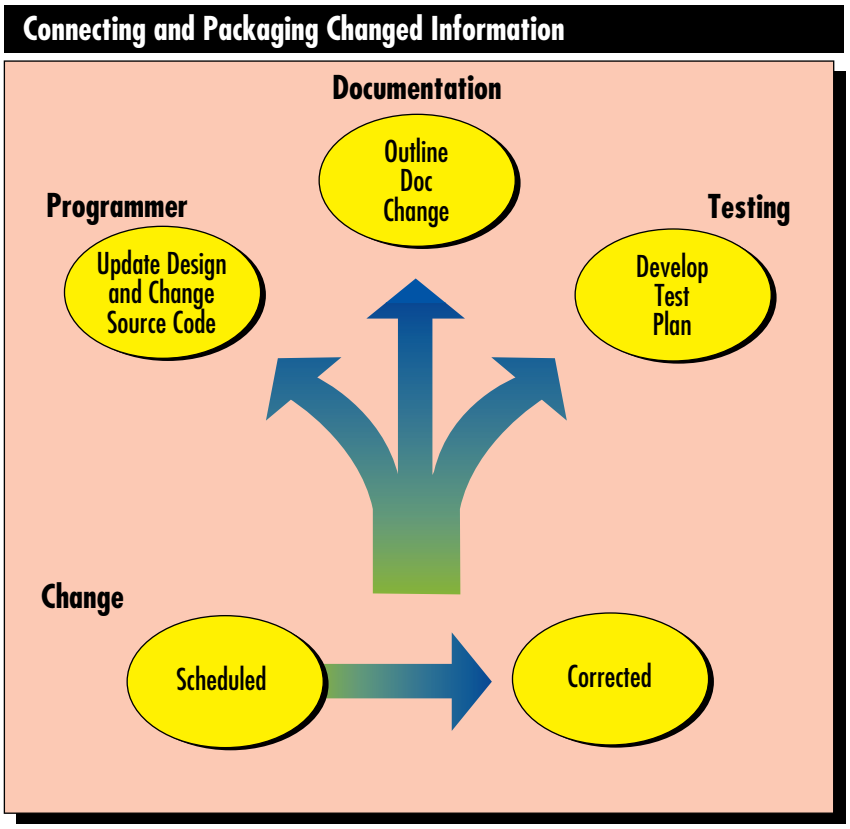


Figure 9. Changed information

As changes are packaged and rolled out into the next release, summarizing the physical changes included in that release accelerates release note documentation and provides valuable knowledge transfer to customer support/help desk personnel who must support the application. If the application is a commercial product, this transfer of knowledge can extend to the field sales force and technical consulting staff—providing great benefits to the organization.

Sohail Haque, SQL Software, 8500 Leesburg Pike, Vienna, VA 22181. 703-760-0448. Internet: info@sql.com. Mr. Haque is director of Technical Services. He has over nine years of experience in Process Configuration Management.

IBM's Porting Center of the East

By Valerie Paul

The Solution Partnership Center—East recently opened in Waltham, Massachusetts, a suburb of Boston, to provide training and technical support for solution developers on the East Coast. The 8,000 square foot porting and marketing center provides 24-hour access with total security for developers to perform their ports and validation functions.

Amidst dust, noise, and construction workers everywhere, the Solution Partnership Center—East (SPC—East) opened its doors for the first time in May 1995. Developers could not wait; they wanted to port their products and they needed support to make it happen. Although construction was not complete until August, developers began checking out the center.

The first Solution Partnership Center was established in San Mateo, California to serve the developer community in the West. The new center was set up in Waltham because that area is considered the "Silicon Valley of the East." The Route 128 area is often billed as "Technology Highway" because in Massachusetts alone there are over 1,900 software companies. And those companies generate over $7 billion in sales—more than 8% of the U.S. software market.

Developer Support

The Solution Partnership Center offers several types of support to solution developers:

- **Porting center**. SPC serves as a porting center, with seven rooms available 24 hours a day in a totally secure environment. Each room can be configured to meet the needs of developers. For example, one day a room could be equipped with a Symmetric Multiprocessor (SMP) RISC system, and the next day it can have an entry-level Power Personal system.
- **Executive briefing room**. This room can accommodate 70 people with up-to-date electronic media. IBM presents technology briefings in this room, but as a solution developer you can also use this room for your own customer presentations.
- **Customized lab**. The lab has PS/2s® for up to 32 people. OS/2, AS/400®, RISC System/6000, and client/server labs are driven off the servers.
- **Technology lobby**. The technology tables contain the latest ThinkPads® and provide demonstrations of IBM's entrance into the Internet, demos of Lotus Notes, and games on OS/2 Warp™.
- **Solution Developer Hall of Fame**. As a solution developer, once you have ported to IBM technology, you are eligible for a "mini commercial" to appear in this corridor for all to see. SPC—East is located next to the IBM customer center in Waltham, so customers often see these glossies and request more information about developer packages and offerings.

Special Offerings for AIX Developers

Mike Sheets, program manager for RISC and AIX, said, "What we have found with solution developers is that they are very interested in doing business in the same accounts where IBM is successful. We are well-positioned at the Solution Partnership Center—East to answer the question 'How do I do business with IBM?' Solution developers need to stay ahead of the technology curve to be able to compete successfully in their markets. We help them do that by providing con-

venient, no-charge access to the latest IBM software and hardware products."

One of the most important offerings to AIX developers is the availability of the center for porting. The center provides full scalability with three Power Personal stations, 42Ts and 43Ps, two J30 4-ways that can be made into one 8-way SMP, plus front ends to the J30. It also has a C10 and expects a 39H soon. By the end of 1995 the center will have an SP2 8-node machine as well. This variety of equipment makes porting less complex and almost painless.

In addition to the 24-hour availability of equipment, the center has technology consultants dedicated to RISC System/6000 porting. These consultants have systems engineering backgrounds and work closely to help you understand IBM's RISC technology and the latest product offerings.

A typical scenario for doing an AIX port begins with a pre-port meeting of the technology consultants and the technical staff from the solution developer organization. This meeting is to ensure that the center has all the equipment necessary to support any porting needs, networking requirements, and software. The dates for the actual porting process result from this meeting. The solution developer actually does the port, with the technology consultants working hand-in-hand to ensure that the process proceeds smoothly.

Technology consultants also ensure that the developer takes advantage of IBM's latest technology. Since this technical resource is on-site, it helps increase the developer's productivity and keeps the process moving. Since technology consultants have a direct link to the development labs, they can get immediate response to any issues that arise during porting. There is no waiting for answers—problems are resolved on the spot without losing productive time.

The center also offers marketing, education, and detailed technical lab events for AIX developers. Seminars on AIX are often presented by IBM technical staff from Austin. According to Glenn Rigby, marketing program team leader, "The response to our marketing and education events has been overwhelmingly positive. It gives us a great feeling to bring leading-edge technology to independent software vendors at a price that is hard to beat—free."

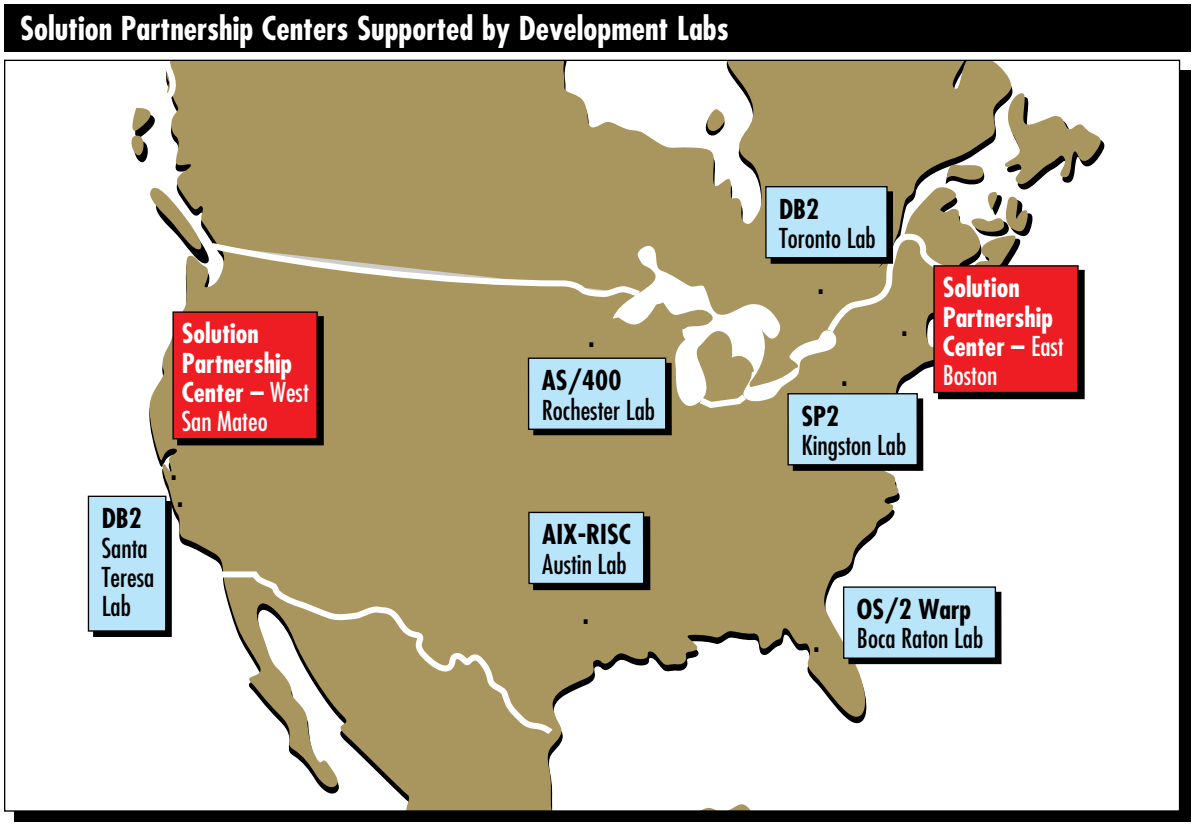


Figure 1. Solution Partnership Centers in the U.S. and Canada

When To Call The Center

SPC—East is the ideal solution if you need to port your product to a more extensive platform, such as moving from a uniprocessor to an SMP processor or from an SMP to a POWERparallel processor. Bring your application to the center and validate that your software does what you are committing that it does on the platform of choice. You can validate your port and establish a prototype, often needed to close business in a bid situation.

Or simply use the center to take advantage of the latest technology, such as moving from AIX 3.2.5 to AIX 4.1. The center has many machines that have the latest technology capabilities. You can fine-tune applications in a dedicated test environment that may not be possible on-site at your own location.

And if your application does not run on IBM technology, the center provides education technology presentations to introduce you to the platforms and the technology available. This can be followed up with ports, validation and prototype tests on IBM platforms, and then conclude with help in defining a strategy after your applications have been ported.

Although the center provides a full support service, you can choose any portion of the process that benefits you. For example, one developer whose market was about 90% Sun® and only 5% IBM was not comfortable in really knowing from experience how their application ran on IBM. The center provided a test bed of multiple machines so they could determine how their application equates on IBM equipment and how to increase the IBM market share for their product.

How to Contact Solution Partnership Center—East

Solution Partnership Center—East
IBM Corporation
404 Wyman Street
Waltham, MA 02154-1280

Phone: 1-800-678-4249
Fax: 617-890-7587

World Wide Web: http:www.austin.ibm.com\developer\ (list of events)

For AIX developers: Mike Sheets—program manager for RISC
Internet: msheets@boston.vnet.ibm.com

Upcoming Events at SPC—East

AIX Multi-vendor Platform Open House	November 28-30
Solution Development with Lotus Notes	December 6
OpenDoc Seminar	December 13

Why Call the Center

The SPC—East can offer significant benefits to solution developers:

- Provides access to the latest IBM technology for AIX, OS/2, AS/400, or client/server
- Provides flexibility and customization based on needs, whether they are ports, validation, prototypes, or benchmarks
- Increases productivity with on-site resources to answer questions and resolve issues
- Reduces costs by serving the local community and providing the latest equipment
- Increases awareness of new technology through marketing and education programs

Now that the dust has settled and the center is established, it's time to investigate the opportunities that await. For more information about how you can take advantage of the center, contact the center.

Valerie Paul, Solution Partnership Center—East, IBM Corporation, 404 Wyman Street, Waltham, MA 02254. Ms. Paul is manager of the Solution Partnership Center. She has a BA in Mathematics and Engineering from Rutgers University and has spent the last eight years dedicated to working with solution developers.

FlowMark—A Workflow Manager

By Carolyn Cummiskey

IBM FlowMark is a workflow manager that enables enterprises to control their business activities—regardless of the nature of their business. IBM is offering solution developers the opportunity to use FlowMark free of charge.

As business processes become increasingly complex, planning and managing diverse activities and resources become a greater challenge. Controlling the execution and movement of work within a company can take as much time and resource as doing the work itself.

Workflow technology is particularly important as organizations re-engineer their business processes. Why? Because a thorough knowledge and understanding of how work is currently being done is necessary in order to redesign and improve the way a large enterprise operates.

A workflow management tool such as IBM FlowMark enables re-engineering because it forces companies to fully describe their business procedures. Workflow technology is ideal for pointing out problems in a process. It can also be a diagnostic tool for assessing the quality of the business process. In essence, it enables organizations to constantly redefine their business by streamlining processes and increasing productivity.

Automated workflow systems can help speed tasks and processes. With a workflow system in place, the flow of information-based activities and tasks can be automated, compressed, and analyzed. Once a process is computerized, the software will assist in re-engineering, by allowing specific procedures to be monitored and measured.

Workflow software is used to monitor operations. Once a process is controlled by a workflow system, a company can measure how long the process takes, how much time the average employee needs to perform a task, and the number of rejects and approvals.

IBM FlowMark

IBM FlowMark is a workflow manager that gives an enterprise control of its business activities— regardless of the type or nature of the business. It can automate complex business processes that people do every day—often in volume. It is a state-of-the-art, object-oriented, distributed application that can run as stand-alone software or as part of a client/server system. Current implementations run on AIX and OS/2 servers with support for OS/2, AIX, and Windows clients. Multiple servers can also share a single database.

FlowMark can help you precisely define, document, test, and control your company's business processes—opening the way to improved productivity and better teamwork. It can automate routine tasks, which reduces errors.

FlowMark's graphical editor can help you build a workflow model. You can define who performs each task, which computer programs are to be used with each task, what other tasks must be completed, what data is required before starting each task, and how information flows between task activities. All this information is stored in a FlowMark database.

When business conditions change, FlowMark enables you to design new processes or update existing processes quickly and efficiently; you can add or remove activities, change their order, or assign them to different people. FlowMark can improve daily operations and keep processes moving.

FlowMark for MVS™, now in limited availability, enables users to integrate mainframe- and LAN-based applications, using IBM's object technology. It will ultimately provide MVS Server capability. FlowMark can be easily integrated with other products such as IBM's ImagePlus®, VisualInfo™, or Lotus Notes.

FlowMark Partners in Development Program

The FlowMark Partners in Development Program provides an opportunity for professional developers to try IBM FlowMark. There are no fees to participate in the program.

The program provides technical support and help to solution providers to integrate IBM FlowMark into their products and services. Professional developers, integrators, and resellers who sell commercial software and/or services are eligible to participate in the program.

FlowMark Partners in Development provides participants with the following services:

- Free use of FlowMark code, documentation, and tools
- Technical support
  - —No charge
  - —Toll-free 800 number
  - —Possible provision of higher-level support, if warranted
  - —Broad skills base, high customer satisfaction
- Vendor advocacy provides answers to your nontechnical questions
- Membership in complementary IBM Developer Assistance Programs, such as IBM Image PID Program
- Participation in IBM Partnership programs and other marketing activities
- Listing of your product or service in online catalogs

IBM encourages participants in the program to incorporate elements of FlowMark into their solutions offerings as well as to present features, benefits, and strengths of IBM FlowMark to their clients.

See the box on this page for requesting more information about the program.

FlowMark Partners in Development Program

For membership information about FlowMark Partners in Development, send your name, company name, address, plus phone and fax numbers to the following:

Mail: Carolyn Cummiskey
IBM Corporation
MD 241
150 Kettletown Road
Southbury, CT 06488
USA

Voice: 1-203-262-4767

Fax: 1-203-262-2141

Internet: Carolyn Cummiskey at cummisk@vnet.ibm.com

Carolyn Cummiskey, IBM Corporation, 150 Kettletown Road, Southbury, CT 06488. Internet: cummisk@vnet.ibm.com. Ms Cummiskey is program manager for IBM FlowMark Partners in Development.

Global Networking Using X.25

By Eddie Ho and John Ellis

The RISC System/6000 (RS/6000) is the preferred platform for global networking. NAFTA and the democratization of the Eastern European nations have created significant international business opportunities for many types of companies and industries. And communications is an important component of these trade opportunities. The primary networking infrastructure of these developing areas is based on X.25. The AIXLink/X.25 package (AIXLink/X.25 on AIX 3.25 and 4.1) can bridge the transformation from a host-centric model to a wide-area client/server application model.

Despite the widespread availability of Integrated Services Digital Network (ISDN) and frame relay in the U.S. and Europe, they are still not available in many areas of the world. But X.25-based packet switching is available worldwide and is the only stable, reliable service in some countries. Each country has its own Postal Telephone and Telegraph (PTT) that provides the communications within its borders, including the X.25 technology.

The International Telegraph and Telephone Consultative Committee (CCITT) defined the X.25 standard for attaching computer equipment to a Packet-Switched Data Network (PSDN). The data is carried in packets over circuits that are shared by many users. The packet can vary in size from 16 to 4096 bytes. Each connection—the point-to-point communication between two computers—is called a *virtual circuit.* Tariffs are typically based on a monthly subscription charge plus usage charges based on the number of packets transmitted.

X.25 has several advantages compared to a leased-line point-to-point connection.

- **Global standards**: Since X.25 is available in all countries, there are standards that apply to all locations.
- **Vendor independence**: The technology and equipment comes from one source—generally the PTT; therefore, equipment and technology are always compatible.
- **Security**: Security is available at the access level to the network and also at the destination location.
- **Ownership of your network without managing one**: The PTT is usually the network provider, which manages the networking technology.
- **Available in almost every country**: Because of its widespread acceptance, X.25 is available internationally.

Figure 1 shows the X.25 network in an enterprise environment in which workgroup users in a LAN environment are accessing remote data/source using the PSDN.

X.25 Architecture

The Open Systems Interconnection (OSI) reference model has three separate layers in its network:

Physical layer: This level of function is implemented at the AIX device-driver level. It consists of functions such as maintaining interface consistency, providing error recovery for High-level Data Link Control (HDLC) frames, and supporting auto call units. CCITT recommends the V.24, V.35, and X.21 physical interfaces.

Frame layer: The frame level uses a link access procedure to ensure that data and control information are accurately exchanged over the circuit between the computer and the network. Its error recovery procedure is based on `LAP_B`, which is a subset of HDLC.

Packet layer: X.25 is a connection-oriented protocol. The primary function of this layer is to give users access to the virtual circuit. The packet-

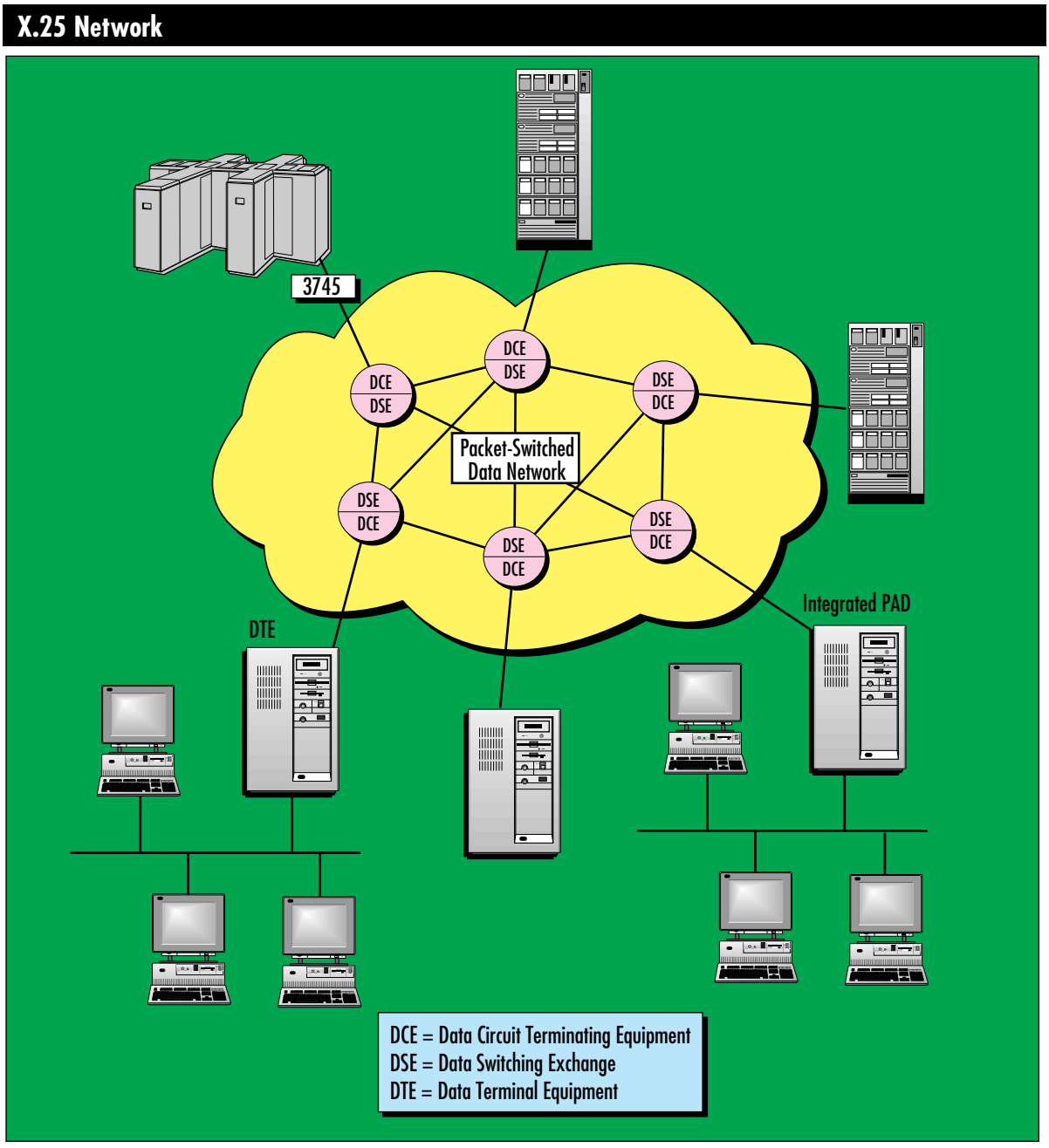


Figure 1. X.25 network in an enterprise environment

level protocol specifies how virtual circuits are established, maintained, and cleared.

The AIXLink/X.25 package implements these three layers. Transport protocols such as Systems Network Architecture (SNA) or TCP/IP use the packet layer. Figure 2 illustrates this layered network model.

Communication to the X.25 network uses logical channels assigned by the carrier. Each channel is mapped into a *virtual circuit*—the point-to-point communication between two computers. A virtual circuit exists only for the duration of the call for Switched Virtual Circuits (SVCs). After that, the *logical channel*—a communication path between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE)—can be reused for other applications. Another type of circuit is a Permanent Virtual Circuit, which provides a long-term connection between DTE and DCE.

The network carrier determines the number of concurrent channels—the RS/6000 hardware can

support up to 1024 channels, depending on the adapter type. The administrator predetermines and configures the number of circuits for incoming, outgoing, and two-way circuits. Figure 3 illustrates the relationship of the logical channel and virtual circuit.

AIXLink/X.25

AIXLink/X.25 is available on both AIX Version 3.2.5 and 4.1. Some important new features are listed below:

- Complies with the CCITT 1988 X.25 standard
- Supports X.3, X.28, and X.29 Packet Assembler/Disassembler (PAD)
- Supports Simple Network Management Protocol (SNMP) for Management Information Bases (MIBs) for the packet (RFC 1382) and frame (RFC 1381) layers
- Supports dedicated or switched circuits
- Supports automatic or user-defined point-to-point DTE/DCE configuration
- Supports up to 512 logical channels per line
- Supports X.21, V.24, and V.35 interfaces
- Supports speeds up to 64 Kbits/second

The networking protocols used with AIXLink/X.25 include TCP/IP, which supports all TCP/IP applications and interfaces, and SNA, which supports all SNA architectures including sub-area, Low-Entry Networking (LEN), and Advanced Peer-to-Peer Networking® (APPN®).

AIXLink/X.25 has three types of Application Programming Interfaces (APIs) supporting programming at the packet or frame level. These APIs are usually protocol suites, such as TCP/IP and SNA, or special commands. The APIs are as follows:

- **Network Provider Interface (NPI)**: This is an API at the packet layer that provides a connection-oriented interface based on the UNIX International Standard NPI Version 2.0. The API is based on the STREAMS model. The PAD implementation uses this interface.
- **Data Link Provider Interface (DLPI)**: This API, at the frame layer, is based on Version 2.0 of DLPI from UNIX International. The interface model is STREAMS.
- **Common Input/Output (COMMIO)**: These are compatible interfaces for AIX 3.2.5 applications, used by the SNA protocol suite.

Commands for the Administrator

Since AIXLink/X.25 has new features and functions, the commands listed in Figure 4 provide administrators with some help in setting up and managing the X.25 network.

Packet Assembler/Disassembler

In many regions, the PAD protocol converter is used for protocol support. It enables you to attach a low-cost terminal device, such as an ASCII terminal, to the X.25 network. The X.3 standard defines PAD protocol.

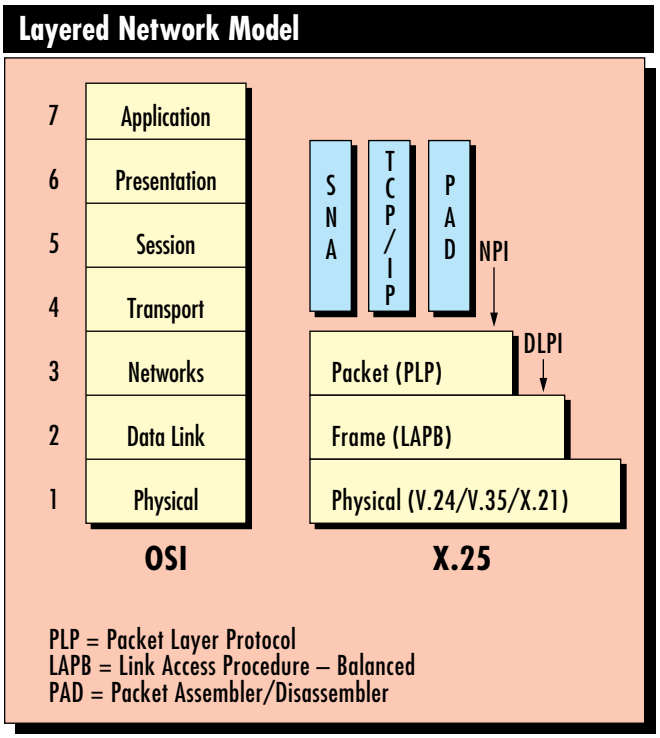


Figure 2. Layered network model

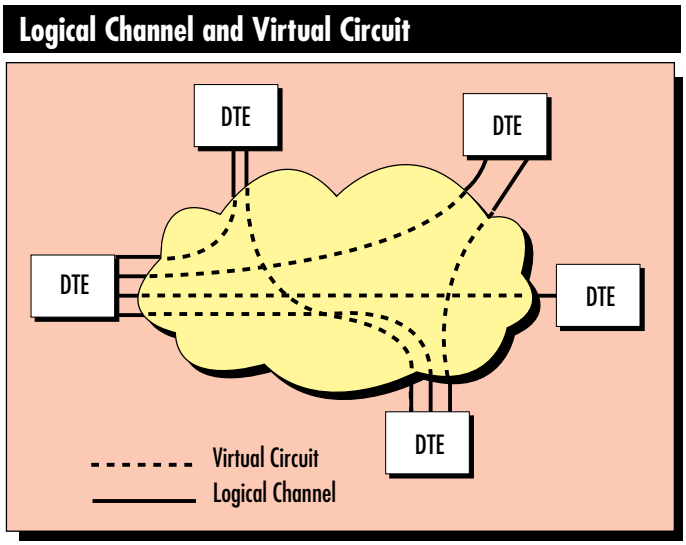


Figure 3. Relationship of logical channel and virtual circuit

AIXLink/X.25 Commands

Troubleshooting Commands

xtalk	Communicates with other DTE; manages address lists for outgoing calls
x25mon	Monitors the X.25 port activity
x25status	Shows all the link status

Operation Commands

chsx25	Re-initializes the attributes of an X.25 port
lspvc	Lists the non-default Permanent Virtual Circuit (PVC) attributes for an X.25 port
lsx25	Lists the configuration of the X.25 support on the system
mkpvc	Creates or modifies a non-default PVC on an X.25 port
mksx25	Adds an X.25 port
rmsx25	Removes an X.25 port
chdev	Allows modification for the X.25 adapter attributes
lsattr	Shows the X.25 adapter and port attributes

Networking Protocol Commands

x25ip	Manages a translation table from IP addresses into X.25 Network User Addresses (NUAs)
xspad	Starts a terminal PAD session
x29d	Starts the X.29 daemon

Figure 4. AIXLink/X.25 Commands

PAD Standard and Application Environment

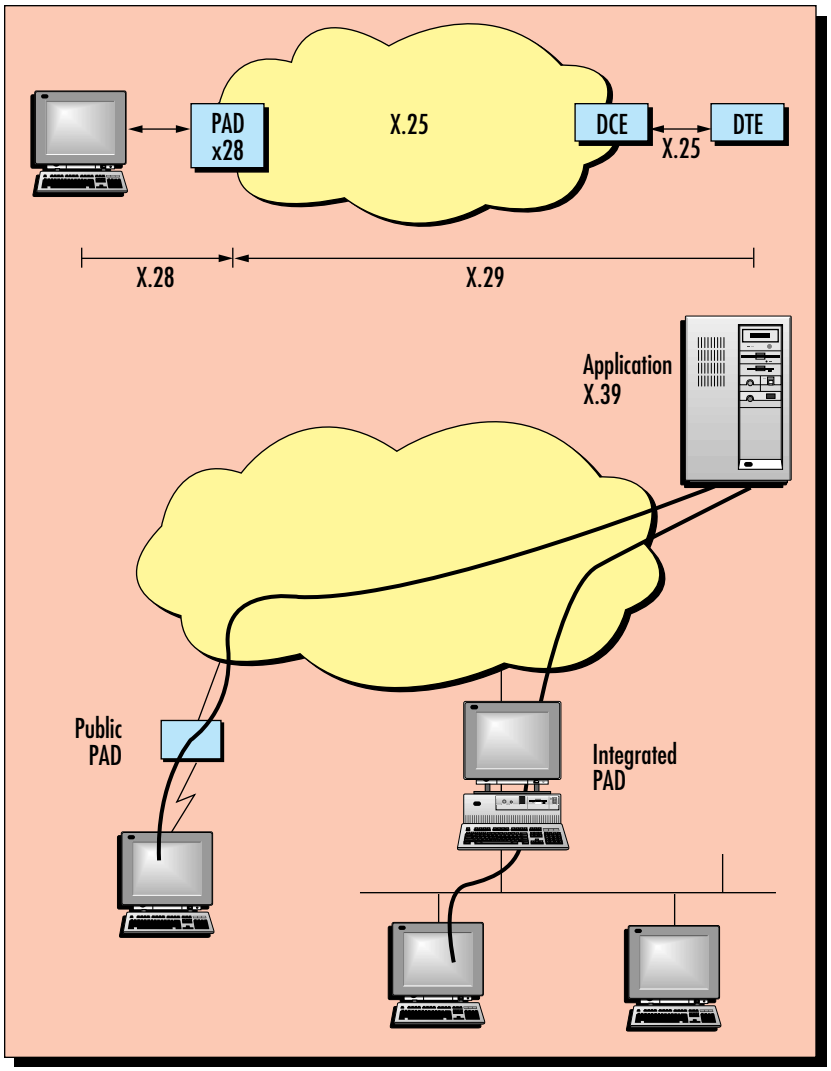


Figure 5. PAD standard and application environment

The PAD is usually provided by the PTT, carrier, or telco. The communication between the PAD and an ASCII terminal is X.28, which defines the ASCII terminal control behavior and characteristics. The communication between the remote DTE and the PAD is X.29 through X.25. X.29 defines the way in which a PAD and a remote DTE exchange control messages.

AIXLink/X.25 provides an integrated solution using both PAD and the ASCII terminal. Attaching an RS/6000 to a public PAD does not require AIXLink/X.25 software because the operating system emulates an ASCII terminal using AIX utilities such as `cu` or `ate`. Figure 5 illustrates both the communication standards and the application environments.

Communication Adapters

There are two families of adapter types based on the number of ports required. Each adapter family has multiple feature codes with different accessories. Figure 6 shows the features for each adapter to help you choose the right accessory for your adapter. AIXLink/X.25 can support two adapter types:

- **ARCTIC Portmaster Adapter/A**
  - —8 ports (V.24 interface) or 6 ports (V.35 or X.21 interface)
  - —Up to 64 Kbits/second at full duplex concurrently for each port
  - —Up to 512 virtual circuits per port with a maximum of 1024 per adapter

- **X.25 Interface Co-Processor**
  - —Single-port support with a V.24, V.35, or X.21 interface; either a Micro Channel® or ISA interface
  - —Supports up to 512 virtual circuits per port

A WAN Services Primer

Some commonly available communication services are listed below. Services not listed are T3 and SONET, high-bandwidth circuits usually used for private networks in a campus environment.

56-Kbits/second: The traditional low-bandwidth leased line has been the point-to-point 56-Kbits/second service. The customer specifies the locations to be connected and installs a Customer Service Unit/Data Services Unit (CSU/DSU) at either end of the link; the carrier establishes the connection between the sites. This is also known as Digital Data Service (DDS).

Switched 56: This is the switched version of 56-Kbits/second link. Charges are typically less than dedicated 56-Kbits/second circuits, but users also pay usage charges by the minute for the flexibility of connecting to multiple sites.

ISDN: Integrated Services Digital Network is an alternative to leased-line connections. Like Plain Old Telephone Service (POTS), ISDN is a switched service with a digital interface that eliminates the need for a modem. An ISDN terminal adapter is used in place of a modem. ISDN includes three types of services:

- B-channel service runs at 64 Kbits/second and is used for voice, circuit-switched data, or packet-switched data.
- D-channel, typically used for sideband signaling information, runs at 16 Kbits/second.
- H-channel service, sometimes referred to as ISDN 384, runs from 384 Kbits/second to 2 Mbits/second. It is intended for multimedia applications.

ISDN lines are provided in two configurations:

- Basic Rate Interface (BRI) = 2 B + 1 D
- Primary Rate Interface (PRI) = 23 B + 1 D

Frame Relay: Often characterized as "fast X.25", frame relay has little of the overhead associated with X.25, omitting packet sequencing and error checking. Frame relay is designed to accommodate bursty LAN traffic. In the U.S., carriers offer speeds from 56 Kbits/second to 1.544 Mbits/second. Frame relay can support both Permanent Virtual Circuits (PVCs) and Switched Virtual Circuits (SVCs). Few carriers today offer SVCs, so frame relay connections are typically point-to-point. Users generally need a leased line to the carrier's location to support frame relay connections.

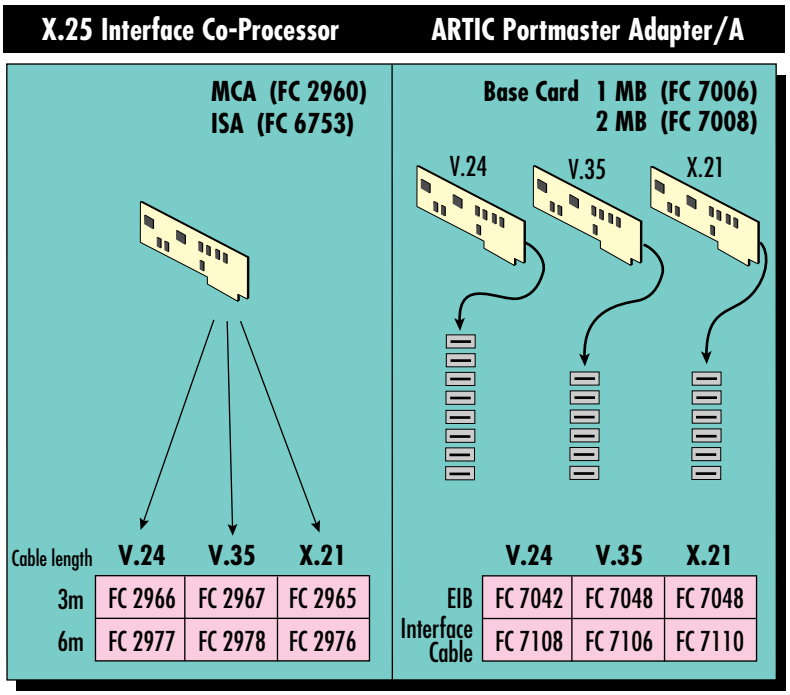


Figure 6. ARTIC and X.25 adapter accessories

T1: This is a digital service with a bandwidth of 1.544 Mbits/second. Also referred to as DS1 channel, T1 service has been the traditional high-bandwidth communications service of choice for private networks.

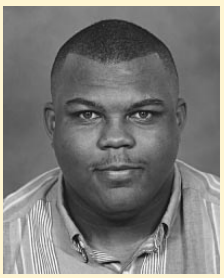
SMDS: Switched Multimegabit Data Service offers high-bandwidth, packet-switched circuits. Because SMDS supports bandwidth up to 45 Mbits/second, high-performance communications equipment (such as DSU/CSUs and routers) is required. SMDS is offered only within the coverage area of certain local carriers.

Eddie Ho, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Ho is a senior programmer in the AIX Executive Briefing Center. He has a BS in Computer Science from the University of Wisconsin and an MS in Computer Science from North Dakota State University.

John Ellis, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Ellis is a staff programmer in the AIX Communication Development group. He has a BSc from the University of Kent in the United Kingdom.

AIX Questions

Compiled by Daryl Green



Carl Senegal

Fred Arnold

The AIX Solution Provider Technical Support Group in Austin, Texas, supports software vendors who are developing or porting applications to AIX. This article is a compilation of questions that are frequently asked by vendors. The name of the responding Technical Support Group staff member appears after each response.

The output from the `ipcs` command seems to be garbage. How can I correct this problem?

Run the `bosboot` command and reboot the system. Several commands, including `ipcs`, use a copy of `/unix` kept in the boot logical volume. System changes can cause this copy to become outdated or corrupted. Running the `bosboot` command and rebooting the system synchronizes the copy in the boot logical volume, which allows the correct values to be restored.

—Carl Senegal

```
# crash           (bring up the crash utility)
> nm seminfo      (dump the address of the seminfo struct)

     000E7F70 000004  TC SD    <seminfo>
     000DDA60 00001C  RW SD    seminfo

> od 000DDA60 16  (the line in the above output that)
                  (contains 'RW' gives the address of)
                  (the seminfo struct)

000dda60: 00001000 0000ffff 00000400 00000400
000dda70: 00002010 00007fff 00004000 00000000
000dda80: 10000000 00000001 00001000 00000000
000dda90: 00000001 00000000 40282329 31380931
```

Figure 1. Identifying the number of semaphores per ID

How can I determine the maximum number of semaphores per ID?

You can find the maximum number of semaphores per ID by using the `seminfo` struct defined in the `/usr/include/sys/sem.h` file and the `crash` command. Figure 1 shows what to run as root.

Check the `/usr/include/sys/sem.h` file for the format of the `seminfo` struct.

Since the second variable of the struct is `semmsl`, the second word from the `od` output is the maximum number of semaphores per ID (`semmsl`). This value is in hex and must be converted to decimal. In Figure 1, the `semmsl` value is 0xffff, which is decimal 65536.

Using this method can also identify other system values, such as message queue values (`/usr/include/sys/msg.h`) and shared memory values (`/usr/include/sys/shm.h`).

—Carl Senegal

How do I install from tape to a 40P machine?

You cannot install from tape to a 40P; it will not access tape during the boot process. If you are installing a `mksysb`, you must first boot from a CD-ROM, then change the installation device to tape.

—Fred Arnold

How can I do traces with smit?

Simply enter `smit trace`; then choose the item you want to trace and select the criteria you want to use.

—Fred Arnold

When creating a `mksysb` to load onto another system, should I always include device drivers for the new system?

AIX Version 4.1.X only installs the drivers required by the current system. If other drivers are needed for another system, they must be installed before the `mksysb` is created so that the other system can load the `mksysb`.

—Fred Arnold

What is a D5 processor and does AIX 4.1.3 have a special version for these processors?

The D5 processor is called an Entry-Level System—a Model 7006 (4xx), 7009 (Cxx), or 7011 (2xx). These systems require a special version of AIX 4.1.3 to be an entry-level server.

—Fred Arnold

How do I configure a 601 keyboard for kanji input and output?

Enter the following commands:

```
% cp /usr/lpp/X11/defaults/xmodmap/
   En_US/keyboard /tmp/JP_keyboard
% vi /tmp/JP_keyboard
```

Figure 2 shows the changes that need to be made to `/tmp/JP_keyboard`.

Next, enter the following commands at the prompt:

```
% xmodmap /tmp/JP_keyboard
% aixterm -lang Ja_JP
```

To test the keyboard configuration, do the following in a new `aixterm` window:

1. Enter `abc`. It should be displayed in Roman characters.
2. Press <F9> (the Romanji key) to set Romanji entry when hiragana is selected.
3. Press <F11> (the Hiragana key) to start hiragana phonetic input.
4. Enter some text that you wish to convert to kanji.
5. Press <F12> (the Kanji key) to convert hiragana-romanji text into kanji text.

```
. . .
!keycode 124 =  F5              NoSymbol        NoSymbol
!keycode 125 =  F6              NoSymbol        NoSymbol
!keycode 126 =  F7              NoSymbol        NoSymbol
keycode 126 =   Eisu_toggle     NoSymbol        NoSymbol
!keycode 127 =  F8              NoSymbol        NoSymbol
keycode 127 =   Henkan          NoSymbol        NoSymbol
!keycode 128 =  F9              NoSymbol        NoSymbol
keycode 128 =   Romanji         NoSymbol        NoSymbol
!keycode 129 =  F10             NoSymbol        NoSymbol
keycode 129 =   Katakana        NoSymbol        NoSymbol
!keycode 130 =  F11             NoSymbol        NoSymbol
keycode 130 =   Hiragana        NoSymbol        NoSymbol
!keycode 131 =  F12             NoSymbol        NoSymbol
keycode 131 =   Kanji           MaeKouho        NoSymbol
!keycode 132 =  Print           NoSymbol        NoSymbol
!keycode 133 =  Cancel          NoSymbol        NoSymbol
!keycode 134 =  Pause           NoSymbol        NoSymbol
. . .
```

Figure 2. Changes to the /tmp/JP_keyboard

6. Press <F7> (the Eisu_toggle key) to shut off hiragana input.
7. Enter `def`. It should be displayed in Roman characters.

—Michael Nicholas

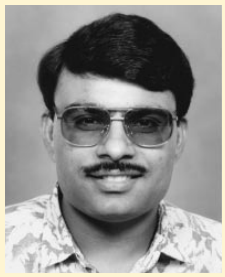
Michael Nicholas

I do not want the desktop login to come up automatically. How do I configure my system for a command-line startup?

Comment out the `dt` entry in the `/etc/inittab` file to prevent the automatic desktop login. To start the desktop from the command line, enter the following:

```
xinit /usr/dt/bin/Xsession . . (extensions)
```

—Michael Nicholas

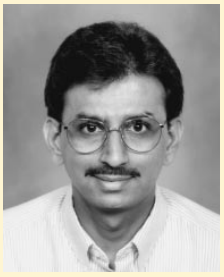


Viral Shah

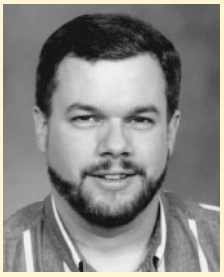
What is the maximum number of sockets that a process can open?

Any process can open up to 2000 sockets. This limit is hard-coded and cannot be changed.

—Viral Shah



Darshan Patel



Daryl Green

I have iFOR/LS installed. The `netls` daemon will not start when I boot the system. When I tried starting up manually using the `startsrc` command, it died after a few seconds. How can I get this daemon working?

1. Perform the following commands to ensure that `llbd`, `glbd`, and `netlsd` are down.

```
stopsrc -s llbd
stopsrc -s glbd
stopsrc -s netlsd
```

2. Remove `/usr/lib/netls/conf/log_file`.
3. Run `/usr/lib/netls/conf/netls_config`. Answer "no" to the questions and select the default cell.
4. Run `/usr/lib/netls/conf/netls_first_time`.
5. Perform the following to ensure daemons are active:

```
lssrc -s llbd
lssrc -s glbd
lssrc -s netlsd
```

—Darshan Patel

Can I mount everything an NFS server exports without knowing the filesystem names and mounting them separately?

Use the automounter to mount all filesystems that a given NFS server exports and that your client is authorized to access. Enter `automount -m /net -hosts` to run the automounter. For example, after issuing this command, if you want to access all filesystems exported from a server named `aix6k`, change directories to `/net/aix6k`. The automounter would then mount all available filesystems from that server below the `/net/aix6k` mount point. Calling any executable program from this mount tree using an explicit path would accomplish the same result. Because these are automounted filesystems, they will be unmounted automatically by default if no process accesses that tree for five minutes.

—Daryl Green

Can I specify the order in which the various hostname resolution databases (such as `NIS`, `DNS`, `/etc/hosts`) are accessed?

In AIX 3.2.5, the system always checks `NIS` first, then `DNS/BIND`, and finally the local `/etc/hosts` file. In AIX 4.1, however, you can use the `/etc/netsvc.conf` file or the `NSORDER` environment variable to configure the order of access. Place a line in the `netsvc.conf` file using the syntax `hosts=value,value,value` (for example, `hosts=bind,local,nis`) or issue `export NSORDER=value,value,value` to override the default sequence. If both `NSORDER` and `netsvc.conf` are used, the `NSORDER` environment variable takes precedence.

—Daryl Green