Introduction to iSCSI In BladeCenter

First Edition (June 2005)

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1.0 Introduction to iSCSI

The iSCSI protocol is the encapsulation of the SCSI protocol within TCPIP packets. By using TCPIP as the transport, the SCSI transactions gain the benefits of extended distance support, connectivity with potentially many more devices than SCSI or fiber channel can support, and a network infrastructure already installed and available.

iSCSI transactions consist of an iSCSI initiator transmitting a request (i.e. read/write/etc) to an iSCSI target. This iSCSI target processes the request and responds with the appropriate information (data/ack/etc). iSCSI initiators are typically application servers or even end users while iSCSI targets are typically either SAN access points or actual storage controllers. Since an iSCSI request is an encapsulation of a SCSI request, the SCSI concept of command descriptor blocks (CDBs) is applicable to iSCSI. CDBs define the type of SCSI operation, the logical block address where to start, the length of data involved, and various other control parameters. Figure 1: Overview of the iSCSI Protocol provides a conceptual view of the iSCSI layers. As Figure 1 illustrates, an iSCSI solution includes both an iSCSI Initiator and target.

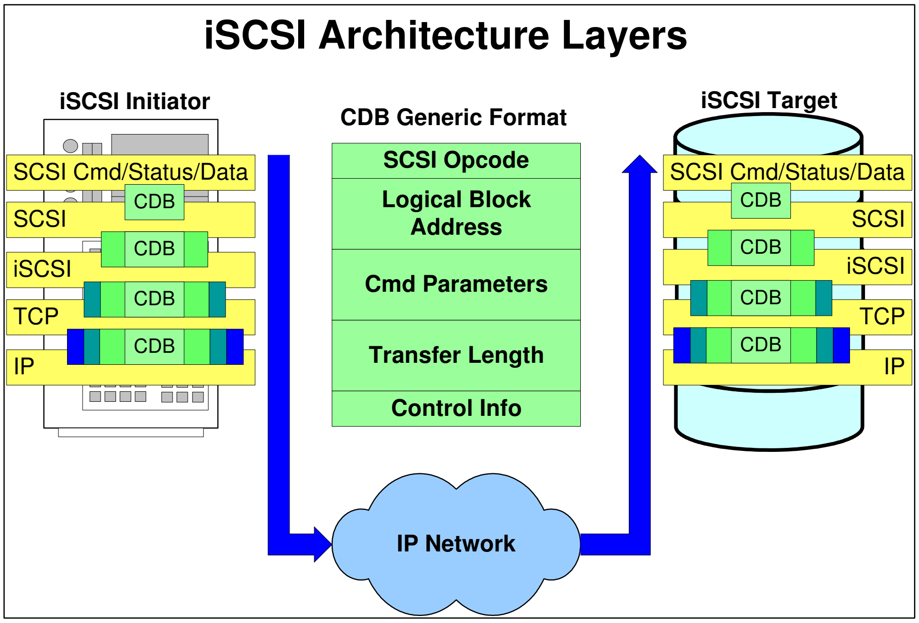

Figure 1: Overview of the iSCSI Protocol

2.0 iSCSI Components

2.1 iSCSI Initiators

The iSCSI initiator is analogous to an HBA in a fibre channel SAN or the NFS clients used in a NAS environment. However, iSCSI has advantages over both these types of remote storage. Since iSCSI runs over the existing network infrastructure, no additional equipment beyond that needed for the initiator and target is required. Fibre channel requires additional specialized, dedicated cabling and switches while iSCSI can leverage the existing Ethernet cabling and switches. For comparison, a Fibre Channel solution requires hardware based Fibre Channel HBAs, Fibre Channel switches, and Fibre Channel storage controllers while an iSCSI solution requires an iSCSI initiator (hardware or software based) and an iSCSI storage controller connected on an existing Ethernet switch fabric. IBM BladeCenter, with its integrated Ethernet fabrics, can be easily leveraged for iSCSI solutions.

An iSCSI initiator requests data blocks from the iSCSI target component. Conceptually, the iSCSI initiator component encapsulates SCSI command data blocks in TCP/IP packets to be sent over an IP network to the iSCSI target component. To the upper layers of a given stand alone or bladed server, the iSCSI initiator component appears as local storage via a standard storage software driver. The implementation of a given iSCSI initiator may be software based where the iSCSI and TCP processing is done in software running on the main processors of the server, or it may be hardware based where the iSCSI and TCP processing is embedded in hardware that appears as an adapter in the server IO topology. Since iSCSI uses the TCP/IP transport services, an iSCSI solution can use the established network fabric or an additional network fabric to transfer information between initiators and targets.

2.1.1 iSCSI Initiator Functions

An iSCSI Initiator is used to access remote storage offered by an iSCSI Target. The initiator first starts a session with the target and creates a connection to a specific LUN. It encapsulates I/O block commands in a network packet for transport over an Ethernet network. The initiator presents the remote drive to the operating system as a SCSI block device. Any applications running on the OS write to what appears to be a local SCSI device – the remote target is transparent to the OS. This ability to handle block commands from the local applications has advantages over network protocols such as NFS or SMB for applications that rely on the block view of storage rather than the file view. The initiator can either be a software initiator or a dedicated HBA. With an HBA, the entire network processing necessary to transmit the SCSI commands over the network is off-loaded to dedicated network ports. Additionally, the HBA uses its own network connection to transmit and receive the iSCSI packets from the network. If the initiator is software-based, the iSCSI driver uses the CPU to handle the iSCSI processing and the information is sent out via the common network ports.

2.1.2 iSCSI Initiator Implementations

There are three ways that an iSCSI Initiator can be implemented. Each of the three approaches differs in price, performance and capabilities.

The first is software iSCSI across a standard network interface card (NIC) This approach is the cheapest of the three but is also has some limitations on capabilities and robustness. However, there are no additional costs incurred beyond licenses (if any) for the iSCSI drivers. The initiator can be deployed on the existing network infrastructure. It depends on the CPU on the host machine to handle the iSCSI processing in addition to the normal TCP/IP processing. However, it is a good fit for casual demands for storage access. A software initiator depends on the IP stack for the OS. If that fails or is taken down by a user, access to the remote disk is lost. As a result, software initiators are not yet ideal for usage to boot the blade up. However, recognizing the cost effectiveness of a software initiator, the industry is working toward adding the capability of boot support for software initiators.

The second approach is to deploy software iSCSI on a NIC that is TCP/IP Offload Engine or TOE-enabled. TOE is a hardware-based solution that removes the burden of IP processing from the CPU on a machine and moves it down onto the NIC. Data is written directly to the NIC and it handles the IP processing necessary to transmit and receive on the network. This second implementation is faster than the first since the CPU does not have to process IP packets. It has the added benefit that the TOE functionality can be used for all network traffic, not merely the iSCSI portions of it. This approach has the potential to be more expensive than the first since it requires the purchase of TOE-capable NICS if they are not in place in the. With the TCP/IP function resident in the NIC, this second approach can evolve to support booting environments.

The third approach is to install dedicated iSCSI initiator HBAs. This is the most expensive of the three options as it requires the purchase of an iSCSI HBA, but it is the most capable and the best performer. All of the SCSI block-processing and TOE functions are integrated onto the HBA. This frees the CPU from having to do any of the iSCSI processing. SCSI block commands are written to the HBA and it does all of the required processing required for network transport. Additionally, the HBA has its own dedicated NICs. With some vendors, these NICs can also be used by the operating system for conventional network traffic. Since this HBA model for iSCSI is similar to Fibre Channel HBA models, blade booting is provided in a similar fashion.

IBM’s iSCSI initiator offering for BladeCenter is the Qlogic iSCSI Expansion for IBM eServer BladeCenter. This offering, provide by Qlogic, an implementation of the HBA approach. All the iSCSI and TCP/IP processing is offloaded to the Expansion card, thus, freeing up CPU cycles for important application processing. In addition to using the Qlogic iSCSI Expansion feature on a given blade for iSCSI processing, this feature also supports usage of the NIC ports for conventional network traffic. This combination of a complete iSCSI offload as well as conventional network traffic capabilities provides a superior solution in terms of performance and flexibility.

2.2 iSCSI Target Functions

The iSCSI target component of the iSCSI ecosystem responds to data block request from iSCSI initiator components. Conceptually, the iSCSI target component un-encapsulates SCSI command data blocks from TCP/IP packets received from the IP network and processes that given SCSI command data block resulting in a SCSI response that is encapsulated in TCP/IP packets transmitted to the requesting iSCSI initiator. The iSCSI target is a traditional storage controller containing and managing a pool of disks in a very similar way to FC or SCSI controllers. The implementation of a given iSCSI target may be software based where the iSCSI and TCP processing is done in software running on the main processors of the storage controller, or it may be hardware based where the iSCSI and TCP processing is embedded in hardware that appears as an adapter in the storage controller IO topology.

2.3 iSCSI Target Implementations

iSCSI Targets can be implemented in several different methods. The method of implementation depends on the market and demand that will be placed on the target. Additionally, the choice of hardware implementations also can depend on whether a customer has an established SAN.

As with the initiator, a target can be software-based or hardware-based. The same tradeoff of speed for cost applies to the target as it does to the initiator. The hardware initiators can be configured as an HBA in a standalone server, a dedicated disk array with an iSCSI processor or an iSCSI gateway to an existing SAN.

The dedicated disk array is similar in function to current Fibre Channel RAID controllers. It consists of a set of disks with the logic included for RAID and other data redundancy. However, customers with established SANs can benefit from the third option, the iSCSI gateway to an existing SAN. This lets users obtain the distance and cost benefits of iSCSI without losing their existing investment in Fibre Channel hardware. The gateway device processes incoming iSCSI requests from initiators and passes them on to the Fibre Channel SAN.

2.4 Overview of the Protocol and Components

This section gives an overview of the iSCSI storage protocol and the evolution of storage.

2.4.1 Evolution of Storage Protocols

There was a time when storage equated with direct-attached storage (DAS) devices. There was little controversy in defining the straightforward DAS. Products in this category include devices like vanilla IDE or SCSI hard drives and RAID arrays. However, the problem with DAS is the limitations for sharing with other machines.

Network storage options like NAS and storage area networks (SAN) solved the main issues with DAS devices by farming out data storage to dedicated machines. However, NAS and SAN both have their own limitations. NAS can be deployed over existing network infrastructure and can go anywhere the network is deployed. Protocols such as SMB or NFS do not support the block storage of DAS. This is fine for some generic file or print-sharing. However, applications such as databases favor block storage. SANs allow users to consolidate storage on block devices; however this approach requires additional and often significant investment in hardware and training. SANs are also limited to the size of the network they are deployed on based primarily on the Fibre Channel distance limits and uniqueness of the cabling and infrastructure. Multiple distant sites often require multiple SANs. iSCSI is a potential solution to address both of these limitations.

2.4.2 Brief Protocol Description

The structure used to communicate a command from an application on an initiator to a target is called a Command Descriptor Block (CDB). A CDB is an encapsulation of a task. A task is a sequence of SCSI commands. . Only one command in a task can be outstanding at any given time. When a SCSI command is executed, it results in either a data phase (read/write) or a status phase (result of the data phase). The status response terminates an SCSI command or task. The SCSI driver builds the CDB from requests issued by the application, and forwards them to the iSCSI layer. The SCSI driver also receives CDBs from the iSCSI driver and forwards them to the application. Since SCSI commands are not synchronous, the driver can issue several CDBs without getting a response from the first one issued.

2.5 iSCSI and Application Content Retrieval

Storage protocols are either block protocols (Fibre Channel, SCSI, etc) or file protocols (NAS protocols such as NFS, SMB, etc). Most organizations require a mix of file and block protocols depending on the requirements of the applications those organizations support. End-user applications and applications such as web servers for the most part run equally well on NAS or block protocols. Other applications such as databases or file streaming require more robust, low-latency block protocols such as Fibre Channel or direct-attached storage. The types of applications being deployed were a major factor in which type of storage organizations would implement. As mentioned above, most have some combination of the two deployed.

iSCSI combines the benefits of NAS (existing network infrastructure limited only by the network) with the transparency of a block protocol. The same storage technology can be used by both the back-end database applications and front-end clients. Cost and convenience may become the determining factors on what mix of iSCSI initiators and targets are deployed.

2.5.1 Brief Description of Application/Operating System Usage

All operating systems rely on storage for a persistent repository of executables and valuable content. Today, storage can be a local physical disk or a logical disk residing on a SAN whether it's a Fibre Channel (FC) based SAN or an iSCSI based SAN. Similar to FC based SANs, iSCSI SAN based storage consists of an operating system driver and network port to interface with the SAN in terms of "reading" or "writing" storage. As a result, the operating systems, and applications on top of the operating system, access the SAN driver as if it were local physical disks.

With this architecture, the applications and operating system can use the iSCSI SAN storage in a variety of ways. The applications can use the iSCSI SAN storage as a container for valuable content such as email mailboxes, web content, and application preferences. Since the content is out on the SAN and, thus, can be accessed by several servers, this valuable content can be shared for either collaboration or sophisticated processing. The operating system can use the iSCSI SAN storage as a repository executables including the actual operating system image. With the potential of the valuable content and executables removed from the local server, a level of flexibility and reliability are introduced because the operation of the server can be independent of the information used by that server.

2.5.2 Brief Description of Usage for Distance

Unlike a traditional NAS, iSCSI is a block device over IP. It can be deployed across an existing network infrastructure so, depending on the performance requirements, it removes the distance limitations of a traditional SAN. It also requires less investment in new specialized hardware than a SAN, and indeed iSCSI gateways exist that can tie remote clients into existing SANs.

Distance on Fibre Channel SANS is limited to what an organization is willing to put into the Fibre Channel infrastructure to bridge the distances between SANs When Fibre Channel was designed, networks were not as robust, nor were they has fast as they are now. The designers emphasized centralized, local access. However, an iSCSI SAN is only limited by the existing network infrastructure. Light storage traffic on the order of 10 MBps can easily be handled over large distances. Heavier volumes of traffic can be handled over higher-capacity leased lines. However, this does not require the same expense that a Fibre SAN would, and with the advent of 10gig Ethernet, iSCSI can be even more considered an alternative to Fibre Channel. Additionally, companies have a major investment in training their network staff to administer the networks. From the network perspective, iSCSI does not require additional training to deploy and manage the infrastructure the way Fibre Channel does.

2.5.3 Overview of the Value Proposition

2.5.3.1 Benefits of Using EN Transport

As mentioned earlier in this paper, the benefits of using the Ethernet transport are cost and in most cases distance. Depending on the planned usage, an iSCSI SAN can go anywhere the existing network infrastructure is deployed. Currently most networks run at 1GB speeds and Fibre Channel is capable of 2GB. However, with the advent of TOE and 10GB Ethernet, this limitation may disappear.

Another additional benefit is reliability. Deploying iSCSI over Ethernet makes use of the inherent reliability of a transmission medium that has been in service for multiple decades. TCP/IP is an existing standard with its own controls for transmission control and error recovery. Lastly, customers can choose to have storage and network traffic on the same network infrastructure. With a traditional SAN, storage traffic must be on a different fabric.

2.5.3.2 Benefits of Storage Management

In today’s ever-more complex storage networks, good storage management practices are crucial to realizing the full benefits of these storage networks. These practices include issues such as LUN creation – how many and how to assign access. Access control is also critical. This includes which users receive access to which LUNs , and what files on a LUN a user can access. Having the storage available for users when needed is also required. RAID can be used on the disk controllers at the target to maintain availability even with failures at the disk level. Regular backups of data are an essential element of any disaster recovery protocol. Snapshots of a running system should also be considered. These snapshots capture the real-time state of a running system and can make it much easier to recover a system that has failed. These issues are all ones that apply equally to any storage network, be it iSCSI, a Fibre SAN or a NAS config.

2.5.3.3 Benefits of Cost Effectiveness (acquisition cost and TCO)

iSCSI can provide a cost-effective alternative to Fibre Channel for an organization wishing to deploy a SAN or add devices to an existing one. There are two types of costs included in TCO associated with deploying a SAN – hard costs which include acquisition of hardware, training and setup. Soft costs include hardware utilization (over/under capacity) and downtime. iSCSI hard costs are cheaper than Fibre Channel. There is much less equipment to buy to deploy a new SAN, and what equipment is required is cheaper than the corresponding Fibre equipment. Special training is not required for iSCSI to the degree it is with Fibre Channel. iSCSI runs on the existing network infrastructure. Organizations that deploy iSCSI are likely to have a staff already trained. iSCSI also provides the cost benefits of administering centralized storage vs. DAS.

2.6 iSCSI Solution Topologies

This section gives a brief introduction to the iSCSI solution topologies from IBM. Using iSCSI as a system boot technology involve an iSCSI initiator service located on a server of some type using the iSCSI protocol over Ethernet to an iSCSI target service located on a storage appliance to retrieve executables required to boot up the server. Additionally, once the components are in place to support booting using iSCSI, the usage of iSCSI for moderate server paging can be implemented as well. Below in Figure 2: Conceptual View of iSCSI Solution Components, the iSCSI ecosystem

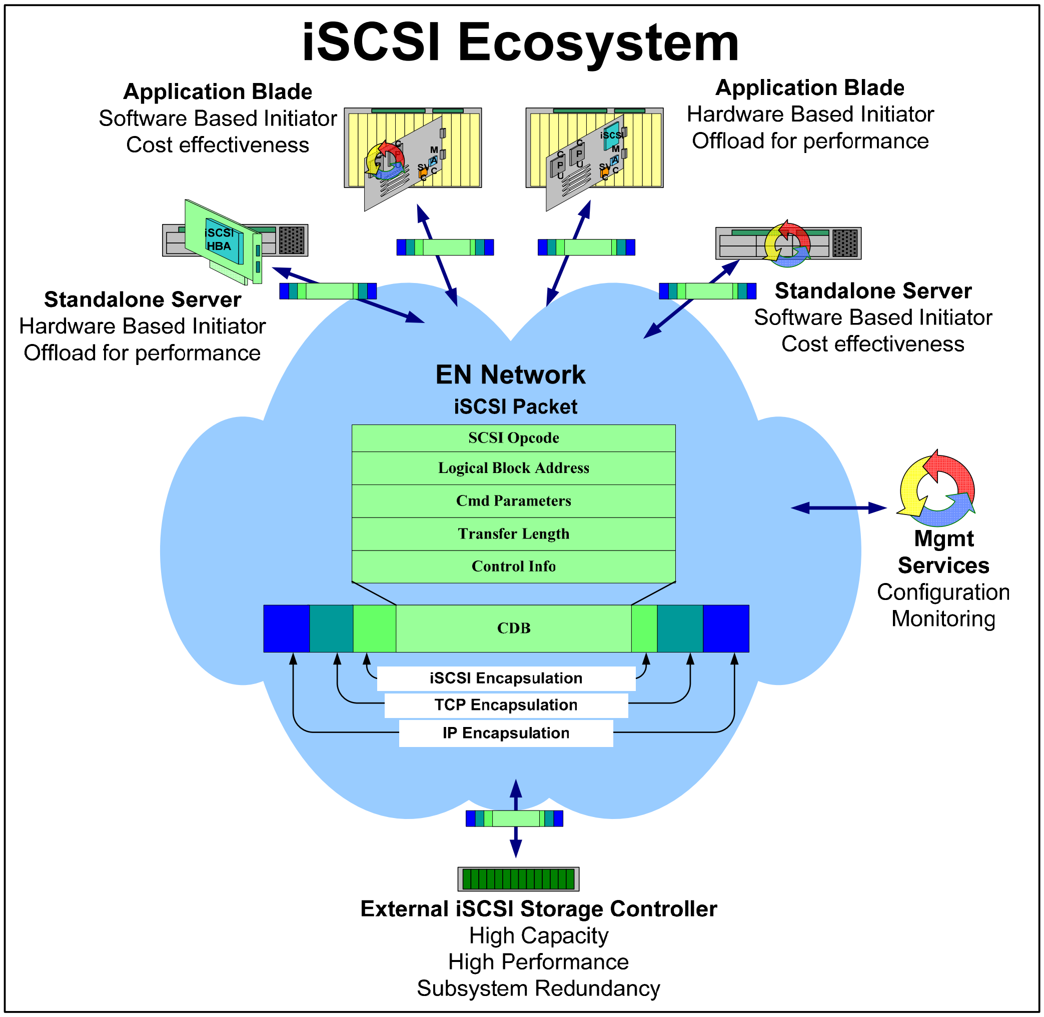

Figure 2: Conceptual View of iSCSI Solution Components

2.6.1 BladeCenter Centric

Figure 2 above shows a mixture of Blades, Standalone servers and dedicated iSCSI Storage Controllers making up the iSCSI Ecosystem. However, an iSCSI can be deployed as a Blade-centric system. Blades can be configured as iSCSI initiators and can use a dedicated external controller or a gateway to an existing Fibre SAN.

Blades could be deployed as initiators using either the iSCSI Software Initiator or the Qlogic iSCSI Expansion adapter for IBM eServer BladeCenter. In a blade-centric deployment, blades might be deployed as diskless with boot and all other disk storage on a remote iSCSI storage controller. From a target perspective, an external iSCSI target can be used to replace or supplement the local disks installed in each blade. It is possible (and desirable) to configure the switch modules installed in the BladeCenter to provide access to the iSCSI target.

2.6.2 Rack Server Centric

In a Rack Server-centric configuration, a target could either be the standalone iSCSI Controller or an iSCSI gateway to an existing Fibre SAN. A rack server would be configured either with a software initiator or a dedicated iSCSI HBA.

2.7 iSCSI Solution Management

Figure 2 above shows the entire ecosystem being managed by a combination of IBM Director, Remote Deployment Manager (RDM) and DHCP/TFTP services. IBM Director would be used to configure the standalone iSCSI targets, Rack Servers and Blades. Director can also be used to remotely push iSCSI configurations down to the assorted iSCSI initiators in the ecosystem. RDM would be used to manage and deploy the assorted boot images (if any) that the iSCSI initiators would boot to. Additional piece of management software that is not shown is the management module software on the management module in a BladeCenter Chassis. This can be accessed either directly or through IBM Director. It is also used to provide finer control over the blades within a specific chassis, but the scope is limited to that particular chassis. Director is used to manage all of the chassis and iSCSI initiators in a given ecosystem.

2.7.1 iSCSI Initiator ⇔ iSCSI Target Nexus

The iSCSI target responds to data block requests from iSCSI initiator components. Conceptually, the iSCSI target component un-encapsulates SCSI command data blocks from TCP/IP packets received from the IP network and processes that given SCSI command data block resulting in a SCSI response that is encapsulated in TCP/IP packets transmitted to the requesting iSCSI initiator. The target does not being conversations with the initiator, instead it responds to session and connection requests and CDBs that come in from remote initiators.

3.0 iSCSI Protocol Discussion

This next section gives a brief overview of the iSCSI Protocol and how it functions.

3.1 iSCSI Discovery

iSCSI discovery is managed by the iSNS, or Internet Storage Name Service. This is a lightweight protocol developed for iSCSI SANs. The iSNS server maintains a database of the iSCSI devices on a network that contains the type of each device (disk controller, initiator, tape, etc) and the IP addresses for each. Devices are added when that device registers with the server. An iSNS server can also zone the devices in much the same way that Fibre Channel fabrics are zoned.

3.2 iSCSI Security Model

iSCSI provides block-level access to storage. As such it is handled by the host OS as if it were local storage. This means that often the root or Administrator user on the initiator OS is used to configure and initiate access. However, an iSCSI target has no way of knowing which user on the host system is requesting access to the storage. All user-level authentication and authorization for access to data within the LUN must be delegated to the initiator operating system. This is the same model used by direct-attached or SAN-attached storage.

The target can still perform some authentication by restricting which initiators are allowed to access each LUN. Additionally, if an iSNS is being used, the ecosystem can be zoned to restrict initiator access to certain targets. Each LUN may have access restricted to a specified group of initiator identifiers. Initiators may be required to authenticate using the Challenge Handshake Authentication Protocol (CHAP).

For data security, IPSec may be used to authenticate initiators or to encrypt data on the network. IPSec is a network-level security protocol used to secure traffic between two networks or between two hosts on a network. It is not restricted exclusively to iSCSI. It is also recommended practice to configure a network topology that minimizes risk of unauthorized access to or modification of data. This can be accomplished through good network design practice and proper configuration of switches and other network hardware.

3.3 iSCSI Login

The iSCSI login is used to establish an iSCSI session between an iSCSI initiator and an iSCSI target. The process begins when the initiator opens a connection to the TCP port that the target is listening on. The TCP connection has to be marked as belonging to an iSCSI session. After this connection is opened, parameters such as security authentication are exchanged and agreed upon. The process is analogous to a Fibre Channel connection being opened. Other operational parameters are agreed upon at this stage. After assigning a connection ID (CID) the initiator then sends a login request that includes the protocol version supported by the initiator, a session ID, the CID, and the negotiation phase the initiator is ready to enter into. The login request may also contain security parameters such as CHAP or IPSec keys. Once all parameters are agreed upon and the initiator is logged in, a connection is opened and data transfer can begin.

3.4 iSCSI Data Transfer

As mentioned earlier, data transfer is accomplished through the use of CDBs, or command descriptor blocks. The CDBs are themselves encapsulated in IP packets for transmission across the network.

3.5 iSCSI Logout

The logout process provides a method for a clean shutdown to close an iSCSI connection or session. The initiator is responsible for commencing the logout procedure. The target may prompt this by sending an iSCSI message that indicates an internal error. Whether prompted or not, initiator requests a logout. After this is sent, no additional requests may be sent. The response from the target indicates that cleanup is complete and no additional responses will be sent on the connection. Additionally, the response also contains information pertaining to recovering the connection should the initiator wish to do that.

3.6 iSCSI Error Processing

iSCSI has several methods for handling errors, depending on where in the process the error occurred. At the lowest level, the SCSI task can result in an error. This is no different than a SCSI error on DASD. It is also handled in much the same way. Because of this possibility, both the target and initiator buffer commands until they are confirmed. This lets the commands be resent if necessary to recover from an error.

At the next level, the transmission of the IP packet containing the iSCSI command can have an error. An error at this level is no different from an error on any other network data. This can often occur in IP networks, especially WANs. Any errors here would be detected by the TCP/IP protocol. TCP/IP is a robust protocol and has excellent self-healing capabilities. After a packet is sent, the sender must wait for an acknowledgement from the receiver that the packet was sent correctly. The receiver does this checking by examining a checksum contained in the packet. If the data was damaged in transmission, the sender resends the data. Only after the sender receives that acknowledgement will it send another packet.

The last level is an iSCSI session-level error. In this instance, the network infrastructure itself may have problems. If this happens, the initiator and target will both make an attempt to recover the connection. This error recovery can be aided by having redundant paths from an initiator to the target, or having redundant targets for an initiator to connect to.

4.0 Applications Leveraging iSCSI

This next section provides a brief introduction to some of the ways that customers can leverage an iSCSI SAN.

4.1 Accessing Application Content

Most of the focus of iSCSI to date has been on its potential for remote boot. However, iSCSI can also be used exclusively for access to remote application content. Depending on the application, iSCSI can be preferable to a network protocol such as NFS. Because iSCSI is a storage block protocol, applications such as database that prefer this can use iSCSI much more effectively than they can a network protocol. There can also be less work for the OS to do, since the iSCSI initiator is responsible for connections to the target. The OS does not have to create and maintain the network connections.

4.2 System Boot

Traditionally, network storage protocols such as NFS require an OS kernel with at least the capabilities to load a network stack. This in turn requires alternative boot methods such as PXE for a remote boot solution. Fibre Channel provides a true remote boot, however distances from the fibre SAN can be limited.

iSCSI gives the benefit of a true self-contained in-band remote boot solution without the limitations of Fibre. The boot iSCSI initiator provides a simplified set of capabilities required to provide the BIOS with the disk blocks needed for boot. The boot initiator resides in the Blade BIOS flash. Acting as an Int 13 service, it presents the remote iSCSI LUN to the Blade BIOS as a local disk. The initiator is presented to the user as a boot option in the BIOS configuration utility.. Once the initiator has established a connection to the target and has started the operating system successfully, the operating system loads its own iSCSI driver and finishes the boot process. For redundancy, the BIOS will maintain a connection to the target. However, the OS will use the connection set up by its driver. If that connection fails for any reason, the OS driver will redirect the iSCSI traffic across the connection set up by the Blade BIOS.

4.3 Content Replication and Disaster Recovery

Content Replication is the practice of deploying and managing multiple copies of a server image to different iSCSI Targets on a network. This can be done using IBM's Remote Deployment Manager in conjunction with IBM Director. Once this is done, a Server Blade that is using iSCSI to boot remotely now has several redundant targets to connect to should it not be able to establish a connection to the primary target. Additionally, a Server Blade that is booting remotely can also be redeployed on demand to run a different OS image should a customer require it. The blade can simply be shut down and configured to connect to a different target and LUN. Similarly, the blade can be reconfigured back to its original LUN at any time.

Content Replication also makes disaster recovery easier and lowers recovery time dramatically. The blade can be configured to attempt a connection to the primary target. Should this fail, if configured, the blade will automatically attempt a connection to a secondary target. Content replication ensures that both targets will contain the same image for that blade. Likewise, should the blade itself fail, either the replacement blade or another existing blade can be redeployed in its place.

5.0 IBM BladeCenter iSCSI Solutions

Leveraging iSCSI solutions for IBM BladeCenter can provide value to the customer in terms of SAN benefits at a more cost effective level. In looking at the usage of iSCSI with IBM BladeCenter, several solution models are available. At the entry level where there are casual storage demands, a solution might be comprised of several blades using an industry standard software initiator accessing a cost effective iSCSI target such as the IBM DS300 over the base Ethernet fabrics found in BladeCenter. At the high end where there are more significant storage demands, a solution might be comprised of several blades using a hardware initiator accessing a performance oriented iSCSI target such as a Netapp 9xx or an iSCSI⇔FC gateway over additional Ethernet fabrics available in BladeCenter.

From a configuration perspective, there are several choices available. One choice is to use network services such as Dynamic Host Configuration Protocol (DHCP) or iSCSI Name Services (iSNS) that defines a protocol and network components that provide configuration information to the respective iSCSI initiator when requested. Another choice is to use the configuration interfaces of the given initiator to define the iSCSI configuration. In either case, the objective is to inform the initiator where the selected targets are in the network

Figure 3: Conceptual View of an IBM BladeCenter iSCSI Entry Solution presents an entry solution for BladeCenter while Figure 4: Conceptual View of IBM BladeCenter iSCSI High End Solution presents a high end solution.

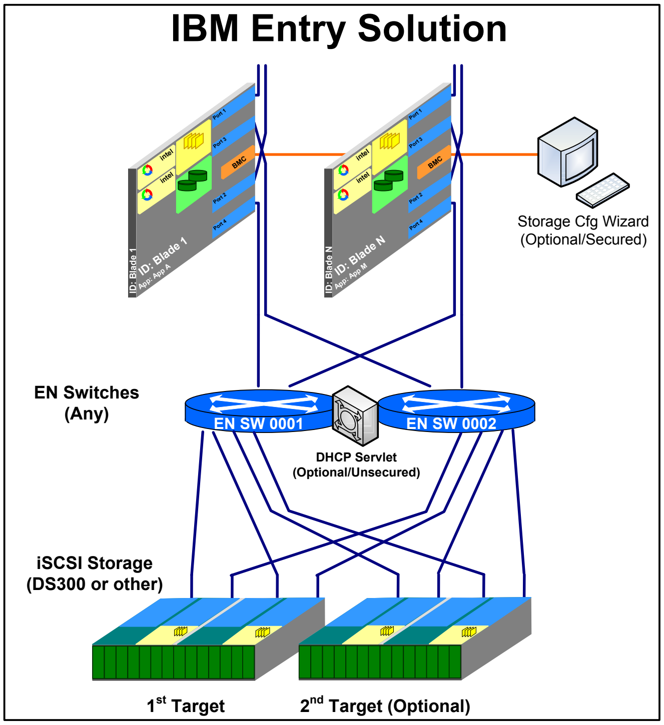

Figure 3: Conceptual View of an IBM BladeCenter iSCSI Entry Solution

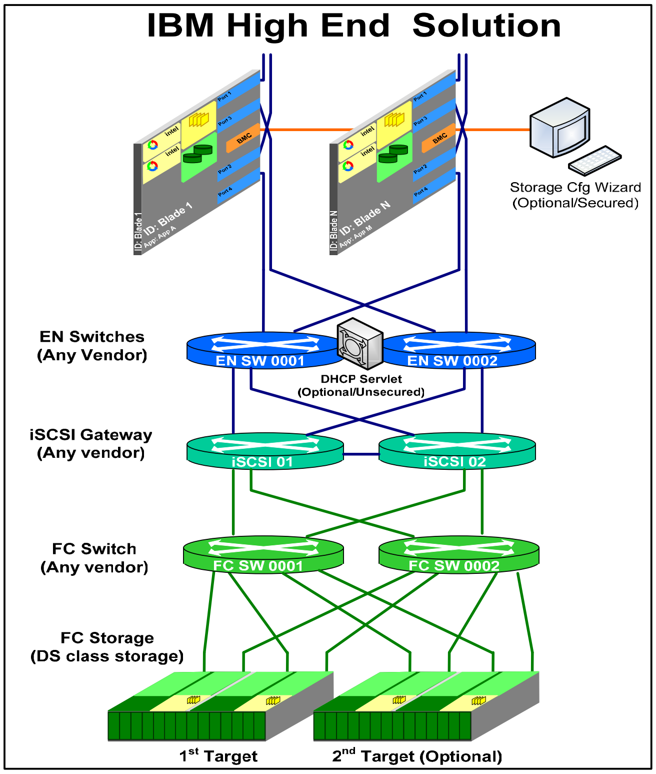

Figure 4: Conceptual View of IBM BladeCenter iSCSI High End Solution

6.0 IBM iSCSI Components for BladeCenter

The components involved in creating an iSCSI solution for BladeCenter consist of the initiator used on the BladeCenter blades, the iSCSI targets, and configuration services if used. The choice of iSCSI initiator, iSCSI target, and configuration method is dependent on the requirements of an intended solution.

6.1 IBM iSCSI Initiator: Qlogic iSCSI Expansion Option for Blades

The Qlogic iSCSI Expansion option for blades provides 2 ports of connectivity. Each port is capable of providing completely offloaded iSCSI initiator capability as well as generic Network Interface Controller (NIC) capability. Recall that in BladeCenter, a given expansion card has connectivity access to switch fabrics 3 and 4. In the iSCSI context, switch fabrics 3 and 4 are standard Ethernet fabrics provided by Ethernet Switch Modules (ESMs) in switch bays 3 and 4. From the external ports of an ESM, the iSCSI traffic flows through the external Ethernet / IP fabric to an iSCSI target such as the IBM DS300 iSCSI Storage Controller. Similarly, all Ethernet / IP traffic flow stemming from a given blade through the iSCSI Expansion Card and through the Ethernet ESMs flows through the external Ethernet / IP fabric to the respective destination.

In the hierarchy of systems and software, the layers above the iSCSI Expansion option for blades view the iSCSI card as a Host Bus Adapter (HBA) for iSCSI storage traffic and view the iSCSI card as a NIC adapter for other Ethernet traffic. With support for Microsoft Windows 2000 and 2003, the windows environment is well represented. With support for Red Hat and SUSE Linux distributions, the Linux environment is similarly represented.

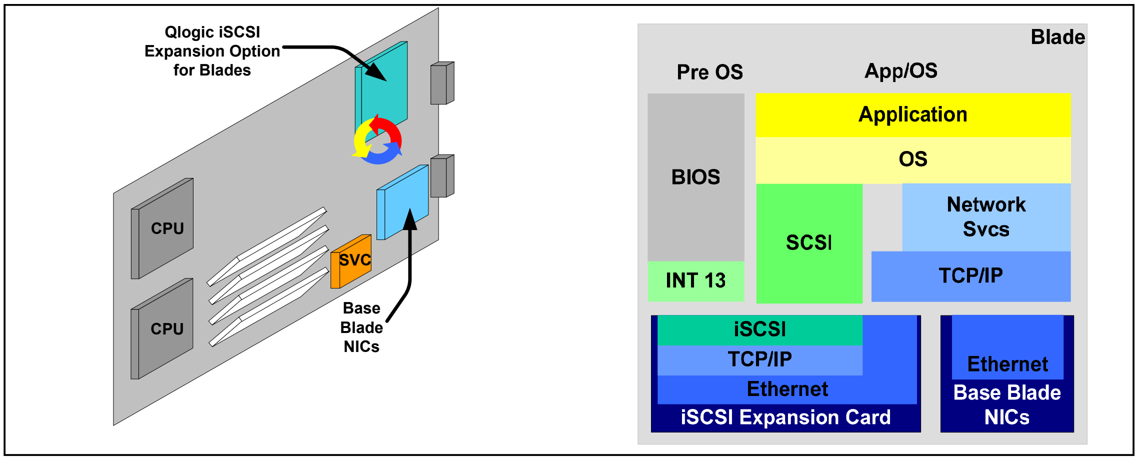

Figure 5: Conceptual View of IBM iSCSI Expansion Card and Corresponding Software Architecture

With the network resources and software architecture in place for the Qlogic iSCSI Expansion option for blades, a variety of applications can leverage this solution. For example, applications such as Email or databases can be processed on the blade while the valuable content is located remotely in a shared storage pool. The value proposition of placing the content remotely includes more efficient usage of storage capacity, simplified content management, and storage allocated flexibility. Another example to consider is blade boot up where the actual disk image resides remotely and is accessed during the boot process. The value proposition of using remote storage to boot include the benefits of stateless blades where content and applications are not tied to the blade, better storage protection, and the foundations for provisioning.

6.2 Industry Provided Software Initiators

The industry has provided a variety of software initiators, usually tied to the operating systems. A given software initiator is a software service running on the blade processors that can access any available NIC interface. Typically, the iSCSI software service relies on the host operating system TCP/IP layer on down to provide the actual conduit to the desired NIC. The benefit of iSCSI software initiators is the relatively low solution cost since the software is typically free and the hardware (blade, NICs, switches) is already in place with the exception of the actual iSCSI target. The limitation of iSCSI software initiators is that they are intended for casual storage demands since the service burdens the host processors to the extent storage is accessed.

From a solution perspective, Figure 3: Conceptual View of an IBM BladeCenter iSCSI Entry Solution is ideal with the blade able to access cost effective storage like the DS300. Figure 4: Conceptual View of IBM BladeCenter iSCSI High End Solution, while supportable is less attractive since the investment in higher end higher performance SAN is counter to the low cost of a software initiation.

From a software solution perspective, Microsoft has provided an iSCSI software initiator for Windows 2000 and Windows 2003. Similarly, the Linux community has provided a software initiator of iSCSI as well through several distributions and through the source forge Linux community.

From an architectural perspective, the software architecture for an iSCSI software initiator is much different, In essence, the iSCSI service is running in the operating system and utilizing the host processors and standard NICs available on the blade. Figure 6: Conceptual View of Software Initiator on a Blade and Corresponding Software Architecture presents a conceptual view of an iSCSI software initiator.

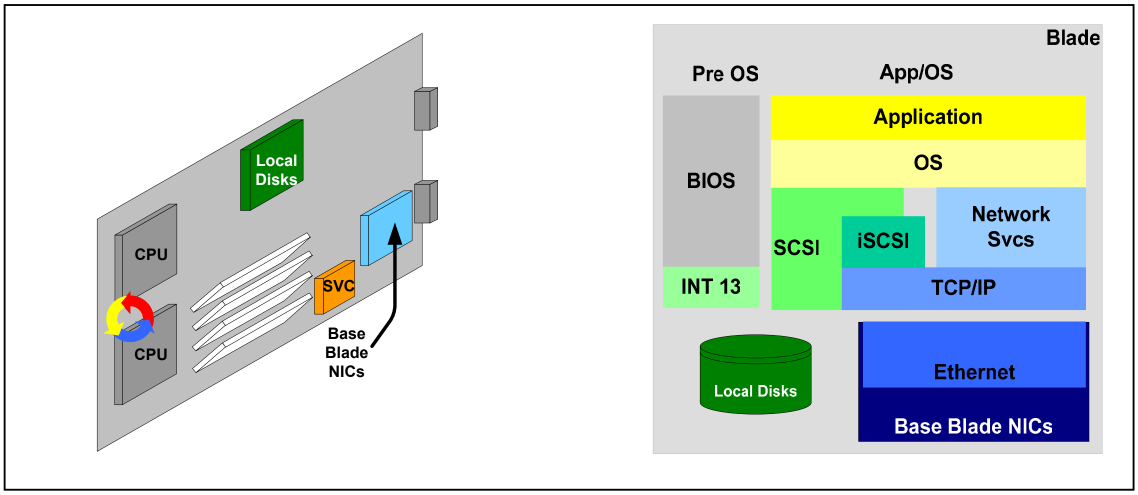

Figure 6: Conceptual View of Software Initiator on a Blade and Corresponding Software Architecture

With the difference in architecture, differences in usage develop. An iSCSI software initiator can be used for Email and entry level databases but not for real intense workloads such as clustered databases or media streaming. In addition, at the present time, software initiators present some challenges in the blade boot up applications. Specifically, Microsoft does not current condone the usage of any software initiator for boot up while Linux has various approaches that support boot up scenarios, although most or very complex. As a result, currently, iSCSI software initiators are ideal for small entry level solutions with casual storage demands.

6.3 So Which iSCSI Initiator is Best for Me?

While each solution implementation requires careful analysis before any purchasing decisions are made, there are some "rules of thumb" that provide general guidance.

In general, if the host processors have roughly 50% of their respective MIPs or cycles free, a software initiator is worthy of consideration. Note that most software initiatives do not consume all the 50% available, rather, one must consider the implications of the application load and iSCSI initiator load surpassing 80% of the host processor cycles in terms of bottlenecks and delayed transactions.

For email applications, if the user pool for the given server is over several thousand users, a hardware initiator may be more appropriate. The industry, through several case studies from various vendors, has coalesced to a consistent view of storage demand per user at about ½ storage IO per user.

For database applications, if the number of clients is small and the transaction load is low and can linger (i.e. business intelligence or the occasional "look up" operation), then a software initiator is worthy of consideration. Any intense database activity such as real time transactions like product purchases or inventory updates is better satisfied by the hardware initiator.

For file/print serving, file serving has a different behavior than print serving. In file serving, the blade acts as the file server while using iSCSI (or FC for higher end implementations) to access storage. Although file serving is, in essence, based on human activity, the storage demands for file serving can be significant for a large user pool. A large user pool, on the order of a thousand users, would better be served by a hardware initiator while a smaller user pool can make the software initiator a worthy option. Print serving is far less demanding, especially as the world moves toward more and more electronic collaboration. As a result, print serving is an ideal application for software initiators.

For messaging and logging where a steady stream of text entries needs to be stored for audit reasons, a software initiator is a worthy option.

In general, for more demanding workloads and storage demands, the hardware initiator is the better option. But if the solution requirements can be met by a software initiator, the cost effectiveness is worth exploiting. When assessing the solution requirements and potential solution implementations, consult the solution component providers such as IBM for a more quantitative view of the initiator options.

6.4 IBM iSCSI Target: DS300

The IBM DS300 is an iSCSI target that provides many of the features found in FC storage controllers at a cost effective price. The DS300 is architected as a dual controller, where 2 RAID controllers reside in the DS300 chassis and coordinate activities between them. The DS300 provides 14 3.5" SCSI drive bays providing the actual stable storage used by a given operating system or application. With 146 GB drives, the total capacity of the DS300 is 2TB of raw storage capacity. The DS300 supports RAID 0,1,5,10 as well as write thru and write back storage caching. To round out the availability capabilities, the DS300 supports redundant power and cooling as well. These features, at its cost effective price point, make the DS300 an attractive entry storage controller.

6.5 3rd Party iSCSI Targets

For higher end solutions, other technologies and iSCSI targets may be of interest. These higher end solutions require more demanding performance, more capacity, and richer features. With these higher requirements on the solution, other technologies such as iSCSI⇔FC gateways or high performance file server/storage controller can be more appropriate.

An iSCSI⇔FC gateway provides an iSCSI target resource to servers using iSCSI initiators and provides a FC interface to a FC SAN. The intended market for such devices is the class of solutions that want to leverage their FC SANs further by adding iSCSI capabilities to it. In this regard, an iSCSI⇔FC gateway allows the solution architect to leverage and extend the FC SAN in place today provide iSCSI support tomorrow. These solutions can be provided via feature enhancements on a FC switch, such as Cisco MDS9xxx family, or can be provided via a stand alone appliance such as Cisco 5428 Storage Router or FalconStor IPStor appliances.

A combination of a high performance file server and storage controller in one appliance provides the capability of supporting both file IO such as NFS or CIFS as well as block IO such as iSCSI. The intended market for such devices is the classes of solutions that want present the storage over a variety of the protocols in order to support flexibility. Usually, these solutions are provided in an integrated appliance such as the Netapp 2xx or Netapp 9xx family of file servers or, again, Falconstor IPStor appliances.

6.6 So Which iSCSI Target is Best for Me?

While each solution implementation requires careful analysis before any purchasing decisions are made, there are some "rules of thumb" that provide general guidance.

In general, if the workload is demanding, then many of these 3rd party options may be more appropriate. Generally, demanding can be indicative of the size and locality of the transaction as well as the number of transactions per second. As an example of transaction size and locality, streaming video is indicative of large transactions (1MB or more per transaction) and a high degree of locality (its quite likely that the next request is after the next set of disk blocks following the disk blocks used for the current request). As an example of the number of transactions per second, database demands are indicative of large numbers of transactions per second (1000s).

In general, if the workload is casual, then the DS300 may be more appropriate. Generally, casual can be indicative of number of transactions or that the usage is driven by a specific event. As an example, operating system paging should be optimized to be casual since it has a direct affect on the server performance. In terms of paging, casual refers to 10s of 4KB page access to disk per server. As an example of event driven usage, casual can refer to boot scenarios where they occur infrequently.

6.7 iSCSI Configuration Methods

Currently, there are two general approaches to configuring an iSCSI SAN. The first approach is to use network services such as DHCP to provide configuration information to the initiator when requested. The second approach is to use the configuration interface of the given devices to configure the solution.

Using network services entails configuring DHCP, iSNS, SLP, or other similar services to provide a given initiator the information it needs to access the iSCSI target. Namely, the configuration information includes the IP address, the iSCSI name (iqn), and any network information required to get to the target such as gateway address to get out of the subnet. The administrator configures the network service to provide the information to a given initiator once the initiator identifies itself to the network services.

Using the configuration interface entails the administrator to configure the initiator specifically with the information needed to find the target. Namely, again, the IP address, iSCSI name, and network information. The key difference is that using the configuration interface removes the need for any network services to be present in the network.

On the target side, due to the complexity of setting up storage in general (creating LUN, defining the size/RAID/cache behavior, configuring access), the configuration of the iSCSI target is typically done using a configuration interface. It should be noted that there are emerging standards such as SMI-S and IMA that focused on standardizing the configuration process for storage. As these standards efforts mature, much better configuration models can be leveraged.

6.8 So Which Configuration Method is Best for Me?

While each configuration method can be useful, the key to the choice is the security model perspective. In essence, the security model addresses how best to restrict access to the storage to the designated servers. For example, if only 2 servers should see specific content on the storage resource available on an iSCSI storage controller because it contains confidential information, then a security model is necessary. On the other hand, if the content on the storage resource, while important, is not confidential, then no security model is really needed.

In the context of a security model, network services provide, at best, a weak security model while using the configuration interface of the iSCSI initiator and target provides a stronger security model of authentication or even encryption.

6.9 Summary of iSCSI Component Choices

To best summarize the choices, Table 1: Summary of iSCSI Component Choices, presents several key question that may be an aid to in determining the right solution components and approachs.

Decision Parametric	**"YES"**	**"NO"**
Are Storage demands casual in nature? (see "rules of thumb" for guidance)	Software initiator (OS vendor) DS300 iSCSI target	Qlogic iSCSI Expansion Option 3rd party iSCSI target
Is boot support required for windows?	Qlogic iSCSI Expansion Option)	Software initiator (OS vendor)
Is boot support required for Linux?	Qlogic iSCSI Expansion Option Note: manual software approach possible (contact IBM for details)	Software initiator (OS vendor)
Intend to use base Ethernet fabrics in BladeCenter?	Software initiator (on base NICs)	Note: add'l EN ESMs required Qlogic iSCSI Expansion Option or Gigabit Ethernet Expansion Option
Is security(authentication) a concern?	Use config I/F on initiator	Use network services (DHCP)

Table 1: Summary of iSCSI Component Choices

7.0 References and Further Readings

7.1 General iSCSI References

"iSCSI The Universal Storage Connection", ISBN 0-201-78419-X, John L Hufferd, Oct 2002

"internet small computer systems interface", RFC3720, IETF, April 2004

IBM Redbook Background on iSCSI
http://www.redbooks.ibm.com/redbooks/pdfs/sg246291.pdf

7.2 General iSCSI Initiator References

Description of QLogic QLA 4010/4010C HBA
http://download.qlogic.com/manual/22751/SANblade_4010_Users_Guide.pdf

IBM documentation on Qlogic iSCSI Initiator Solutions
http://pc.ibm.com/support

7.3 General iSCSI Target References

Description of IBM's DS300 iSCSI Storage Controller
http://www-1.ibm.com/servers/storage/support/disk/ds300/installing.html

Description of Netapp's iSCSI Storage Solutions
http://www.netapp.com/solutions/iscsi/

7.4 iSCSI Oriented Operating System References

Description of Microsoft iSCSI Solutions
http://www.microsoft.com/WindowsServer2003/technologies/storage/iscsi/default.mspx

Description of Linux iSCSI Solutions
http://linux-iscsi.sourceforge.net/

7.5 Boot Oriented Operating System References

Configuring iSCSI SAN Booting with the QLogic QLA 4010/4010C HBA
http://now.netapp.com/NOW/knowledge/docs/hba/iscsi/qlogic/pdfs/sanboot.pdf

White Paper: Boot from SAN in Windows Server 2003 and Windows 2000 Server
http://www.microsoft.com/windowsserversystem/storage/solutions/sanintegration/bootfromsaninwindows.mspx

Storport in Windows Server 2003: Improving Manageability and Performance in Hardware RAID and Storage Area Networks
http://www.microsoft.com/windowsserversystem/ storage/technologies/storport/storportwp.mspx

Server Clusters: Storage Area Networks - For Windows 2000 and Windows Server 2003
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/clustering/starenet.mspx

Microsoft Storage Technologies – Boot from SAN
http://www.microsoft.com/windowsserversystem/storage/technologies/bootfromsan/default.mspx

Support for Booting from a Storage Area Network (SAN)
http://support.microsoft.com/default.aspx?scid=kb;en-us;305547