2.1.3 Storage

The storage device is the most important component in the storage system environment. A storage device uses magnetic or solid state media. Disks, tapes, and diskettes use magnetic media. CD-ROM is an example of a storage device that uses optical media, and removable flash memory card is an example of solid state media.

Tapes are a popular storage media used for backup because of their relatively low cost. In the past, data centers hosted a large number of tape drives and processed several thousand reels of tape. However, tape has the following limitations:

- Data is stored on the tape linearly along the length of the tape. Search and retrieval of data is done sequentially, invariably taking several seconds to access the data. As a result, random data access is slow and time consuming. This limits tapes as a viable option for applications that require real-time, rapid access to data.
- In a shared computing environment, data stored on tape cannot be accessed by multiple applications simultaneously, restricting its use to one application at a time.
- On a tape drive, the read/write head touches the tape surface, so the tape degrades or wears out after repeated use.
- The storage and retrieval requirements of data from tape and the overhead associated with managing tape media are significant.

In spite of its limitations, tape is widely deployed for its cost effectiveness and mobility. Continued development of tape technology is resulting in high capacity medias and high speed drives. Modern tape libraries come with additional memory (cache) and / or disk drives to increase data throughput. With these and added intelligence, today's tapes are part of an end-to-end data management solution, especially as a low-cost solution for storing infrequently accessed data and as long-term data storage.

Optical disk storage is popular in small, single-user computing environments. It is frequently used by individuals to store photos or as a backup medium on personal/laptop computers. It is also used as a distribution medium for single applications, such as games, or as a means of transferring small amounts of data from one self-contained system to another. Optical disks have limited capacity and speed, which limits the use of optical media as a business data storage solution.

The capability to write once and read many (WORM) is one advantage of optical disk storage. A CD-ROM is an example of a WORM device. Optical disks, to some degree, guarantee that the content has not been altered, so they can be used as low-cost alternatives for long-term storage of relatively small amounts of fixed content that will not change after it is created. Collections of optical disks in an array, called *jukeboxes*, are still used as a fixed-content storage solution. Other forms of optical disks include CD-RW and variations of DVD.

Disk drives are the most popular storage medium used in modern computers for storing and accessing data for performance-intensive, online applications. Disks support rapid access to random data locations. This means that data can be written or retrieved quickly for a large number of simultaneous users or applications. In addition, disks have a large capacity. Disk storage arrays are configured with multiple disks to provide increased capacity and enhanced performance.

2.2 Disk Drive Components

A disk drive uses a rapidly moving arm to read and write data across a flat platter coated with magnetic particles. Data is transferred from the magnetic platter through the R/W head to the computer. Several platters are assembled together with the R/W head and controller, most commonly referred to as a *hard disk drive (HDD)*. Data can be recorded and erased on a magnetic disk any number of times. This section details the different components of the disk, the mechanism for organizing and storing data on disks, and the factors that affect disk performance.

Key components of a disk drive are *platter, spindle, read/write head, actuator arm assembly, and controller* (Figure 2-2):

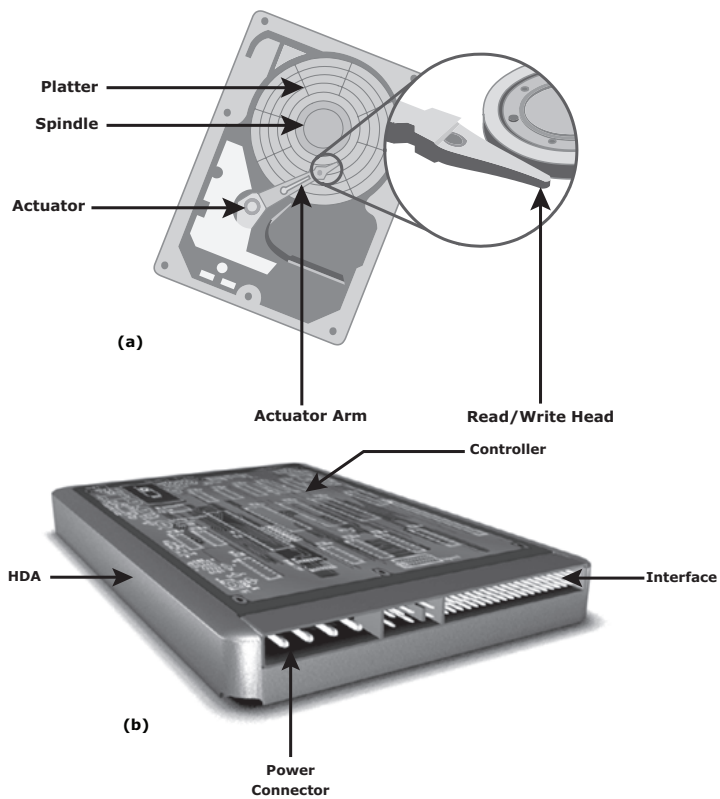

Figure 2-2: Disk Drive Components

2.3 Disk Drive Performance

A disk drive is an electromechanical device that governs the overall performance of the storage system environment. The various factors that affect the performance of disk drives are discussed in this section.

2.3.1 Disk Service Time

Disk service time is the time taken by a disk to complete an I/O request. Components that contribute to service time on a disk drive are *seek time, rotational latency,* and *data transfer rate.*

Seek Time

The *seek time* (also called *access time*) describes the time taken to position the R/W heads across the platter with a radial movement (moving along the radius of the platter). In other words, it is the time taken to reposition and settle the arm and the head over the correct track. The lower the seek time, the faster the I/O operation. Disk vendors publish the following seek time specifications:

- **Full Stroke:** The time taken by the R/W head to move across the entire width of the disk, from the innermost track to the outermost track.
- **Average:** The average time taken by the R/W head to move from one random track to another, normally listed as the time for one-third of a full stroke.
- **Track-to-Track:** The time taken by the R/W head to move between adjacent tracks.

Each of these specifications is measured in milliseconds. The average seek time on a modern disk is typically in the range of 3 to 15 milliseconds. Seek time has more impact on the read operation of random tracks rather than adjacent tracks. To minimize the seek time, data can be written to only a subset of the available cylinders. This results in lower usable capacity than the actual capacity of the drive. For example, a 500 GB disk drive is set up to use only the first 40 percent of the cylinders and is effectively treated as a 200 GB drive. This is known as *short-stroking* the drive.

Rotational Latency

To access data, the actuator arm moves the R/W head over the platter to a particular track while the platter spins to position the requested sector under the R/W head. The time taken by the platter to rotate and position the data under

the R/W head is called *rotational latency*. This latency depends on the rotation speed of the spindle and is measured in milliseconds. The average rotational latency is one-half of the time taken for a full rotation. Similar to the seek time, rotational latency has more impact on the reading/writing of random sectors on the disk than on the same operations on adjacent sectors.

Average rotational latency is around 5.5 ms for a 5,400-rpm drive, and around 2.0 ms for a 15,000-rpm drive.

Data Transfer Rate

The *data transfer rate* (also called *transfer rate*) refers to the average amount of data per unit time that the drive can deliver to the HBA. It is important to first understand the process of read and write operations in order to calculate data transfer rates. In a *read operation*, the data first moves from disk platters to R/W heads, and then it moves to the drive's internal *buffer*. Finally, data moves from the buffer through the interface to the host HBA. In a *write operation*, the data moves from the HBA to the internal buffer of the disk drive through the drive's interface. The data then moves from the buffer to the R/W heads. Finally, it moves from the R/W heads to the platters.

The data transfer rates during the R/W operations are measured in terms of internal and external transfer rates, as shown in Figure 2-8.

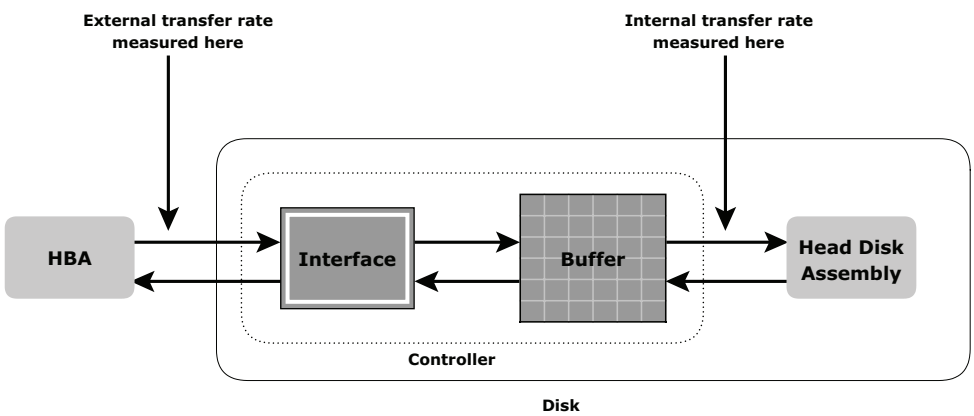

Figure 2-8: Data transfer rate

Internal transfer rate is the speed at which data moves from a single track of a platter's surface to internal buffer (cache) of the disk. Internal transfer rate takes into account factors such as the seek time. *External transfer rate* is the rate at which data can be moved through the interface to the HBA. External transfer rate is generally the advertised speed of the interface, such as 133 MB/s for ATA. The sustained external transfer rate is lower than the interface speed.

2.4 Fundamental Laws Governing Disk Performance

To understand the laws of disk performance, a disk can be viewed as a black box consisting of two elements:

- **Queue:** The location where an I/O request waits before it is processed by the I/O controller.
- **Disk I/O Controller:** Processes I/Os that are waiting in the queue one by one.

The I/O requests arrive at the controller at the rate generated by the application. This rate is also called the *arrival rate*. These requests are held in the I/O queue, and the I/O controller processes them one by one, as shown in Figure 2-9. The I/O arrival rate, the queue length, and the time taken by the I/O controller to process each request determines the performance of the disk system, which is measured in terms of response time.

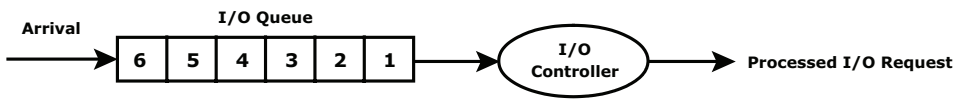

Figure 2-9: I/O processing

Little's Law is a fundamental law describing the relationship between the number of requests in a queue and the response time. The law states the following relation (numbers in parentheses indicate the equation number for cross-referencing):

$$N = a \times R \quad (1)$$

where

"N" is the total number of requests in the queuing system (requests in the queue + requests in the I/O controller)

"a" is the arrival rate, or the number of I/O requests that arrive to the system per unit of time

"R" is the average response time or the turnaround time for an I/O request — the total time from arrival to departure from the system

The *utilization law* is another important law that defines the I/O controller utilization. This law states the relation:

$$U = a \times R_S \quad (2)$$

where

"U" is the I/O controller utilization

EXERCISES

1. What are the benefits of using multiple HBAs on a host?
2. An application specifies a requirement of 200 GB to host a database and other files. It also specifies that the storage environment should support 5,000 IOPS during its peak processing cycle. The disks available for configuration provide 66 GB of usable capacity, and the manufacturer specifies that they can support a maximum of 140 IOPS. The application is response time sensitive and disk utilization beyond 60 percent will not meet the response time requirements of the application. Compute and explain the theoretical basis for the minimum number of disks that should be configured to meet the requirements of the application.
3. Which components constitute the disk service time? Which component contributes the largest percentage of the disk service time in a random I/O operation?
4. Why do formatted disks have less capacity than unformatted disks?
5. The average I/O size of an application is 64 KB. The following specifications are available from the disk manufacturer: average seek time = 5 ms, 7,200 RPM, transfer rate = 40 MB/s. Determine the maximum IOPS that could be performed with this disk for this application. Taking this case as an example, explain the relationship between disk utilization and IOPS.
6. Consider a disk I/O system in which an I/O request arrives at the rate of 80 IOPS. The disk service time is 6 ms.
   a. Compute the following:

      Utilization of I/O controller

      Total response time

      Average queue size

      Total time spent by a request in a queue

   b. Compute the preceding parameter if the service time is halved.
7. Refer to Question 6 and plot a graph showing the response time and utilization, considering 20 percent, 40 percent, 60 percent, 80 percent, and 100 percent utilization of the I/O controller. Describe the conclusion that could be derived from the graph.
8. The Storage Networking Industry Association (SNIA) shared storage model is a simple and powerful model for describing the shared storage architecture. This model is detailed at `www.snia.org/education/storage_networking_primer/shared_storage_model/SNIA-SSM-text-2003-04-13.pdf`. Study this model and prepare a report explaining how the elements detailed in this chapter are represented in the SNIA model.

Chapter 3
Data Protection: RAID

KEY CONCEPTS

- Hardware and Software RAID
- Striping, Mirroring, and Parity
- RAID Write Penalty
- Hot Spares

In the late 1980s, rapid adoption of computers for business processes stimulated the growth of new applications and databases, significantly increasing the demand for storage capacity. At that time, data was stored on a single large, expensive disk drive called *Single Large Expensive Drive (SLED)*. Use of single disks could not meet the required performance levels, due to their inherent limitations (detailed in Chapter 2, Section 2.4, "Fundamental Laws Governing Disk Performance").

HDDs are susceptible to failures due to mechanical wear and tear and other environmental factors. An HDD failure may result in data loss. The solutions available during the 1980s were not able to meet the availability and performance demands of applications.

An HDD has a projected life expectancy before it fails. *Mean Time Between Failure (MTBF)* measures (in hours) the average life expectancy of an HDD. Today, data centers deploy thousands of HDDs in their storage infrastructures. The greater the number of HDDs in a storage array, the greater the probability of a disk failure in the array. For example, consider a storage array of 100 HDDs, each with an MTBF of 750,000 hours. The MTBF of this collection of HDDs in the array, therefore, is 750,000/100 or 7,500 hours. This means that a HDD in this array is likely to fail at least once in 7,500 hours.

RAID is an enabling technology that leverages multiple disks as part of a set, which provides data protection against HDD failures. In general, RAID implementations also improve the I/O performance of storage systems by storing data across multiple HDDs.

In 1987, Patterson, Gibson, and Katz at the University of California, Berkeley, published a paper titled "A Case for Redundant Arrays of Inexpensive Disks

(RAID)." This paper described the use of small-capacity, inexpensive disk drives as an alternative to large-capacity drives common on mainframe computers. The term *RAID* has been redefined to refer to *independent* disks, to reflect advances in the storage technology. RAID storage has now grown from an academic concept to an industry standard.

This chapter details RAID technology, RAID levels, and different types of RAID implementations and their benefits.

3.1 Implementation of RAID

There are two types of RAID implementation, hardware and software. Both have their merits and demerits and are discussed in this section.

3.1.1 Software RAID

Software RAID uses host-based software to provide RAID functions. It is implemented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.

Software RAID implementations offer cost and simplicity benefits when compared with hardware RAID. However, they have the following limitations:

- **Performance:** Software RAID affects overall system performance. This is due to the additional CPU cycles required to perform RAID calculations. The performance impact is more pronounced for complex implementations of RAID, as detailed later in this chapter.
- **Supported features:** Software RAID does not support all RAID levels.
- **Operating system compatibility:** Software RAID is tied to the host operating system hence upgrades to software RAID or to the operating system should be validated for compatibility. This leads to inflexibility in the data processing environment.

3.1.2 Hardware RAID

In *hardware RAID* implementations, a specialized hardware controller is implemented either on the host or on the array. These implementations vary in the way the storage array interacts with the host.

Controller card RAID is host-based hardware RAID implementation in which a specialized RAID controller is installed in the host and HDDs are connected to it. The RAID Controller interacts with the hard disks using a PCI bus. Manufacturers also integrate RAID controllers on motherboards. This integration reduces the overall cost of the system, but does not provide the flexibility required for high-end storage systems.

Chapter 4
Intelligent Storage System

Business-critical applications require high levels of performance, availability, security, and scalability. A hard disk drive is a core element of storage that governs the performance of any storage system. Some of the older disk array technologies could not overcome performance constraints due to the limitations of a hard disk and its mechanical components. RAID technology made an important contribution to enhancing storage performance and reliability, but hard disk drives even with a RAID implementation could not meet performance requirements of today's applications.

KEY CONCEPTS

- Intelligent Storage System
- Front-End Command Queuing
- Cache Mirroring and Vaulting
- Logical Unit Number
- LUN Masking
- High-end Storage System
- Midrange Storage System

With advancements in technology, a new breed of storage solutions known as an *intelligent storage system* has evolved. The intelligent storage systems detailed in this chapter are the feature-rich RAID arrays that provide highly optimized I/O processing capabilities. These arrays have an operating environment that controls the management, allocation, and utilization of storage resources. These storage systems are configured with large amounts of memory called *cache* and use sophisticated algorithms to meet the I/O requirements of performance-sensitive applications.

4.1 Components of an Intelligent Storage System

An intelligent storage system consists of four key components: *front end, cache, back end,* and *physical disks*. Figure 4-1 illustrates these components and their interconnections. An I/O request received from the host at the front-end port is processed through cache and the back end, to enable storage and retrieval of data from the physical disk. A read request can be serviced directly from cache if the requested data is found in cache.

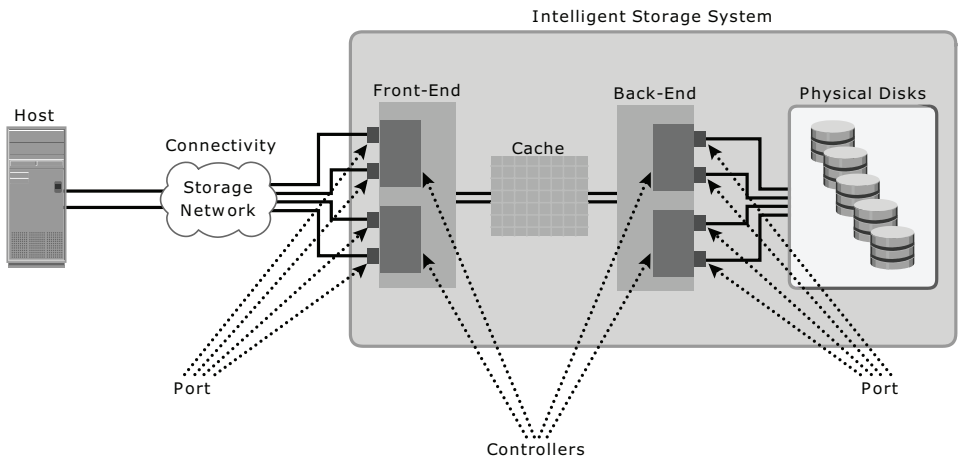

Figure 4-1: Components of an intelligent storage system

4.1.1 Front End

The front end provides the interface between the storage system and the host. It consists of two components: front-end ports and front-end controllers. The *front-end ports* enable hosts to connect to the intelligent storage system. Each front-end port has processing logic that executes the appropriate transport protocol, such as SCSI, Fibre Channel, or iSCSI, for storage connections. Redundant ports are provided on the front end for high availability.

Front-end controllers route data to and from cache via the internal data bus. When cache receives write data, the controller sends an acknowledgment message back to the host. Controllers optimize I/O processing by using command queuing algorithms.

Front-End Command Queuing

Command queuing is a technique implemented on front-end controllers. It determines the execution order of received commands and can reduce unnecessary drive head movements and improve disk performance. When a command is received for execution, the command queuing algorithms assigns

a tag that defines a sequence in which commands should be executed. With command queuing, multiple commands can be executed concurrently based on the organization of data on the disk, regardless of the order in which the commands were received.

The most commonly used command queuing algorithms are as follows:

- **First In First Out** (FIFO)**:** This is the default algorithm where commands are executed in the order in which they are received (Figure 4-2 [a]). There is no reordering of requests for optimization; therefore, it is inefficient in terms of performance.
- **Seek Time Optimization:** Commands are executed based on optimizing read/write head movements, which may result in reordering of commands. Without seek time optimization, the commands are executed in the order they are received. For example, as shown in Figure 4-2(a), the commands are executed in the order A, B, C and D. The radial movement required by the head to execute C immediately after A is less than what would be required to execute B. With seek time optimization, the command execution sequence would be A, C, B and D, as shown in Figure 4-2(b).

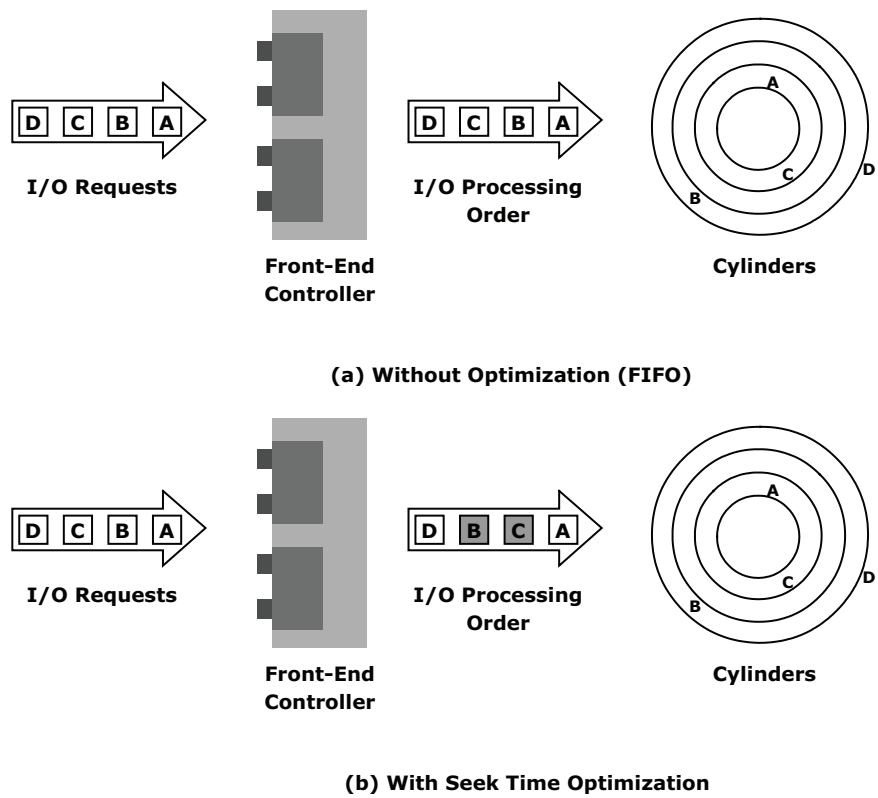

Figure 4-2: Front-end command queuing

- **Access Time Optimization:** Commands are executed based on the combination of seek time optimization and an analysis of rotational latency for optimal performance.

Command queuing can also be implemented on disk controllers and this may further supplement the command queuing implemented on the front-end controllers. Some models of SCSI and Fibre Channel drives have command queuing implemented on their controllers.

4.1.2 Cache

Cache is an important component that enhances the I/O performance in an intelligent storage system. Cache is semiconductor memory where data is placed temporarily to reduce the time required to service I/O requests from the host.

Cache improves storage system performance by isolating hosts from the mechanical delays associated with physical disks, which are the slowest components of an intelligent storage system. Accessing data from a physical disk usually takes a few milliseconds because of seek times and rotational latency. If a disk has to be accessed by the host for every I/O operation, requests are queued, which results in a delayed response. Accessing data from cache takes less than a millisecond. Write data is placed in cache and then written to disk. After the data is securely placed in cache, the host is acknowledged immediately.

Structure of Cache

Cache is organized into pages or slots, which is the smallest unit of cache allocation. The size of a cache page is configured according to the application I/O size. Cache consists of the *data store* and *tag RAM.* The data store holds the data while tag RAM tracks the location of the data in the data store (see Figure 4-3) and in disk.

Entries in tag RAM indicate where data is found in cache and where the data belongs on the disk. Tag RAM includes a *dirty bit* flag, which indicates whether the data in cache has been committed to the disk or not. It also contains time-based information, such as the time of last access, which is used to identify cached information that has not been accessed for a long period and may be freed up.

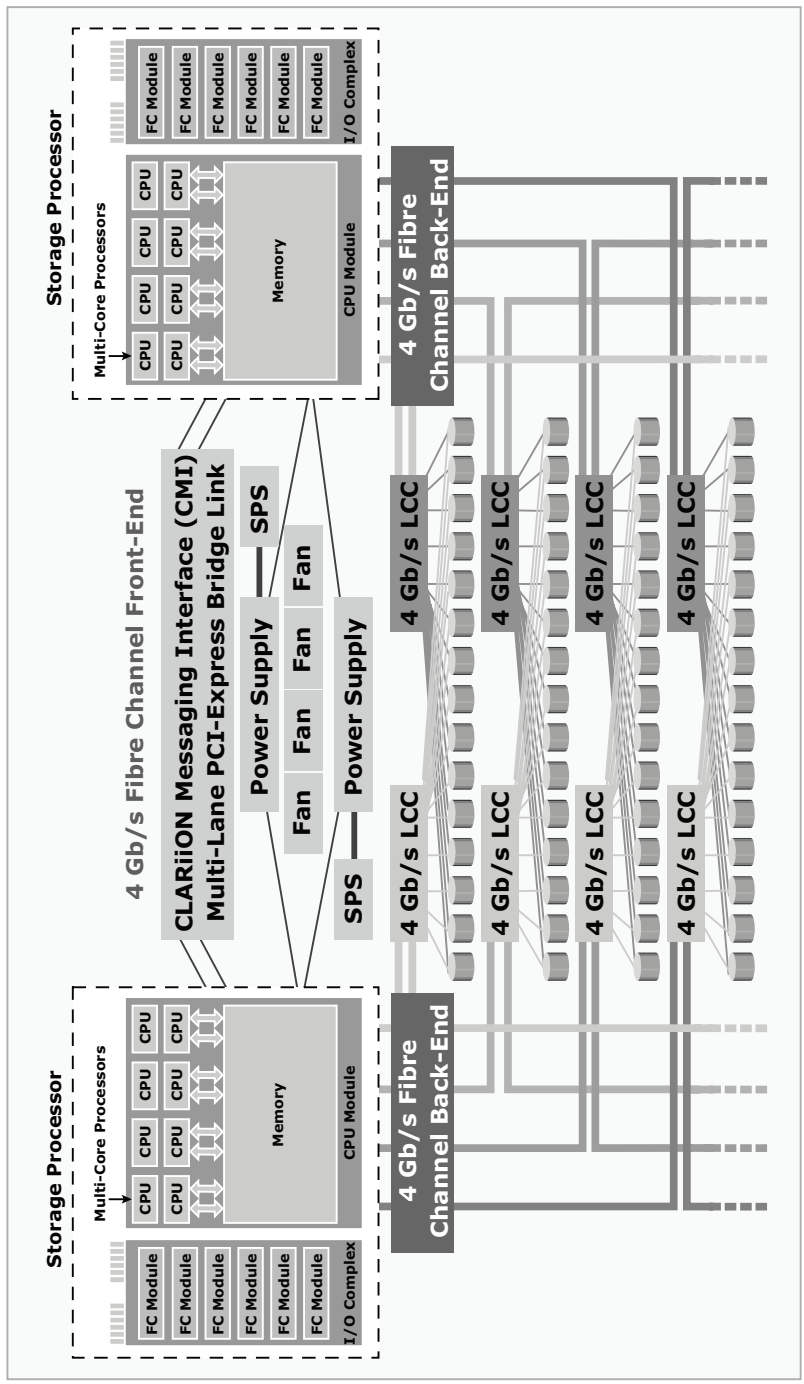

Figure 4-10: CLARiiON architecture

The CLARiiON architecture supports fully redundant, hot swappable components. This means that the system can survive with a failed component, which can be replaced without powering down the system. The important components of the CLARiiON storage system include the following:

- **Intelligent storage processor (SP):** Intelligent SP is the main component of the CLARiiON architecture. SP are configured in pairs for maximum availability. SP provide both front-end and back-end connectivity to the host and the physical disk, respectively. SP also include memory, most of which is used for cache. Depending on the model, each SP includes one or two CPUs.
- **CLARiiON Messaging Interface (CMI):** The SPs communicate to each other over the CLARiiON Messaging Interface, which transports commands, status information, and data for write cache mirroring between the SPs. CLARiiON uses PCI-Express as the high-speed CMI. The PCI Express architecture delivers high bandwidth per pin, has superior routing characteristics, and provides improved reliability.
- **Standby Power Supply (SPS):** In the event of a power failure, the SPS maintains a power supply to the cache for long enough to allow the content to be copied to the vault.
- **Link Control Card (LCC):** The LCC provides services to the drive enclosure, which includes the capability to control enclosure functionalities and monitor environmental status. Each drive enclosure has two LCCs. The other functions performed by LCCs are loop configuration control, failover control, marker LED control, individual disk port control, drive presence detection, and voltage status information.
- **FLARE Storage Operating Environment:** FLARE is a special software designed for EMC CLARiiON. Each storage system ships with a complete copy of the FLARE operating system installed on its first four disks. When CLARiiON is powered up, each SP boots and runs the FLARE operating system. FLARE performs resource allocation and other management tasks in the array.

4.3.3 Managing the CLARiiON

CLARiiON supports both command-line interface (CLI) and graphical user interface (GUI) for management. *Naviseccli* is a CLI-based management tool. Commands can be entered from the connected host system or from a remote server through Telnet/SSH to perform all management functions.

Navisphere management software is a GUI-based suite of tools that enables centralized management of CLARiiON storage systems. These tools are used to monitor, configure, and manage CLARiiON storage arrays. The Navisphere management suite includes the following:

- **Navisphere Manager:** A GUI-based tool for centralized storage system management that is used to configure and manage CLARiiON. It is a web-based user interface that helps to securely manage CLARiiON storage systems locally or remotely over the IP connection, using a common browser. Navisphere Manager provides the flexibility to manage single or multiple systems.
- **Navisphere Analyzer:** A performance analysis tool for CLARiiON hardware components.
- **Navisphere Agent:** A host-residing tool that provides a management communication path to the system and enables CLI access.

4.3.4 Symmetrix Storage Array

The EMC Symmetrix establishes the highest standards for performance and capacity for an enterprise information storage solution and is recognized as the industry's most trusted storage platform. Figure 4-11 shows the EMC Symmetrix DMX-4 storage array.

EMC Symmetrix uses the Direct Matrix Architecture and incorporates a fault-tolerant design. Other features of the Symmetrix are as follows:

- Incrementally scalable up to 2,400 disks
- Supports Flash-based solid-state drives
- Dynamic global cache memory (16 GB–512 GB)
- Advanced processing power (up to 130 PowerPC)
- Large number of concurrent data paths available (32–128 data paths) for I/O processing
- High data processing bandwidth (up to 128 GB/s)
- Data protection with RAID 1, 1+0 (also known as 10 for mainframe), 5, and 6
- Storage-based local and remote replication for business continuity through TimeFinder and SRDF software

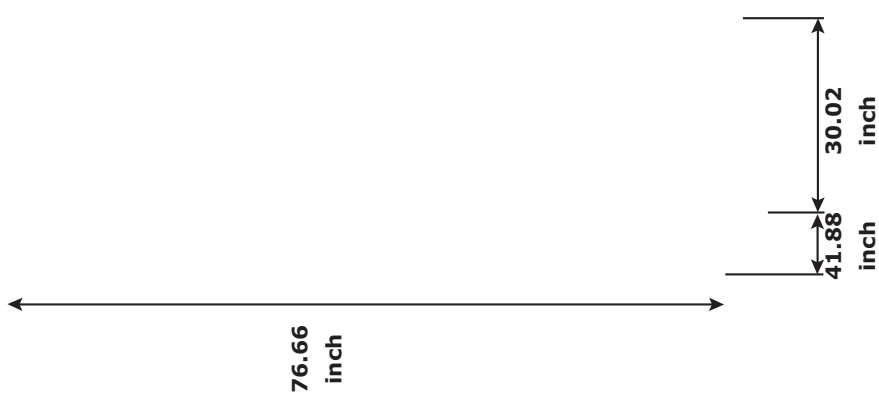

Figure 4-11: EMC Symmetrix

- **Symmetrix Enginuity:** This is the operating environment for EMC Symmetrix. Enginuity manages and ensures the optimal flow and integrity of information through the various hardware components of the Symmetrix system. It manages all Symmetrix operations and system resources to optimize performance intelligently. Enginuity ensures system availability through advanced fault monitoring, detection, and correction capabilities and provides concurrent maintenance and serviceability features. It also offers a foundation for specific software features for disaster recovery, business continuance, and storage management.

Summary

This chapter detailed the features and components of the intelligent storage system — front end, cache, back end, and physical disks. The active-active and active-passive implementations of intelligent storage systems were also described. An intelligent storage system provides the following benefits to an organization:

- Increased capacity
- Improved performance
- Easier storage management
- Improved data availability
- Improved scalability and flexibility
- Improved business continuity
- Improved security and access control

An intelligent storage system is now an integral part of every mission-critical data center. Although a high-end intelligent storage system addresses information storage requirements, it poses a challenge for administrators to share information easily and securely across the enterprise.

Storage networking is a flexible information-centric strategy that extends the reach of intelligent storage systems throughout an enterprise. It provides a common way to manage, share, and protect information. Storage networking is detailed in the next section.

EXERCISES

1. Consider a scenario in which an I/O request from track 1 is followed by an I/O request from track 2 on a sector that is 180 degrees away from the first request. A third request is from a sector on track 3, which is adjacent to the sector on which the first request is made. Discuss the advantages and disadvantages of using the command queuing algorithm in this scenario.
2. Which type of application benefits the most by bypassing write cache? Why?
3. An Oracle database uses a block size of 4 KB for its I/O operation. The application that uses this database primarily performs a sequential read operation. Suggest and explain the appropriate values for the following cache parameters: cache page size, cache allocation (read versus write), pre-fetch type, and write aside cache.
4. Download Navisphere Simulator and the lab guide from `http://education.EMC.com/ismbook` and perform the tasks listed.

Direct-attached storage (DAS) is often referred to as a stovepiped storage environment. Hosts "own" the storage and it is difficult to manage and share resources on these isolated storage devices. Efforts to organize this dispersed data led to the emergence of the storage area network (SAN). SAN is a high-speed, dedicated network of servers and shared storage devices. Traditionally connected over Fibre Channel (FC) networks, a SAN forms a single-storage pool and facilitates data centralization and consolidation. SAN meets the storage demands efficiently with better economies of scale. A SAN also provides effective maintenance and protection of data.

This chapter provides detailed insight into the FC technology on which a SAN is deployed and also reviews SAN design and management fundamentals.

6.1 Fibre Channel: Overview

The FC architecture forms the fundamental construct of the SAN infrastructure. *Fibre Channel* is a high-speed network technology that runs on high-speed optical fiber cables (preferred for front-end SAN connectivity) and serial copper cables (preferred for back-end disk connectivity). The FC technology was created to meet the demand for increased speeds of data transfer among computers, servers, and mass storage subsystems. Although FC networking was introduced in 1988, the FC standardization process began when the American National Standards Institute (ANSI) chartered the Fibre Channel Working Group (FCWG). By 1994, the new high-speed computer interconnection standard was developed and the Fibre Channel Association (FCA) was founded with 70 charter member companies. Technical Committee T11, which is the committee within INCITS (International Committee for Information Technology Standards), is responsible for Fibre Channel interfaces. T11 (previously known as X3T9.3) has been producing interface standards for high performance and mass storage applications since the 1970s.

Higher data transmission speeds are an important feature of the FC networking technology. The initial implementation offered throughput of 100 MB/s (equivalent to raw bit rate of 1Gb/s i.e. 1062.5 Mb/s in Fibre Channel), which was greater than the speeds of Ultra SCSI (20 MB/s) commonly used in DAS environments. FC in full-duplex mode could sustain throughput of 200 MB/s. In comparison with Ultra-SCSI, FC is a significant leap in storage networking technology. Latest FC implementations of 8 GFC (Fibre Channel) offers throughput of 1600 MB/s (raw bit rates of 8.5 Gb/s), whereas Ultra320 SCSI is available with a throughput of 320 MB/s. The FC architecture is highly scalable and theoretically a single FC network can accommodate approximately 15 million nodes.

6.2 The SAN and Its Evolution

A *storage area network (SAN)* carries data between servers (also known as *hosts*) and storage devices through fibre channel switches (see Figure 6-1). A SAN enables storage consolidation and allows storage to be shared across multiple servers. It enables organizations to connect geographically dispersed servers and storage.

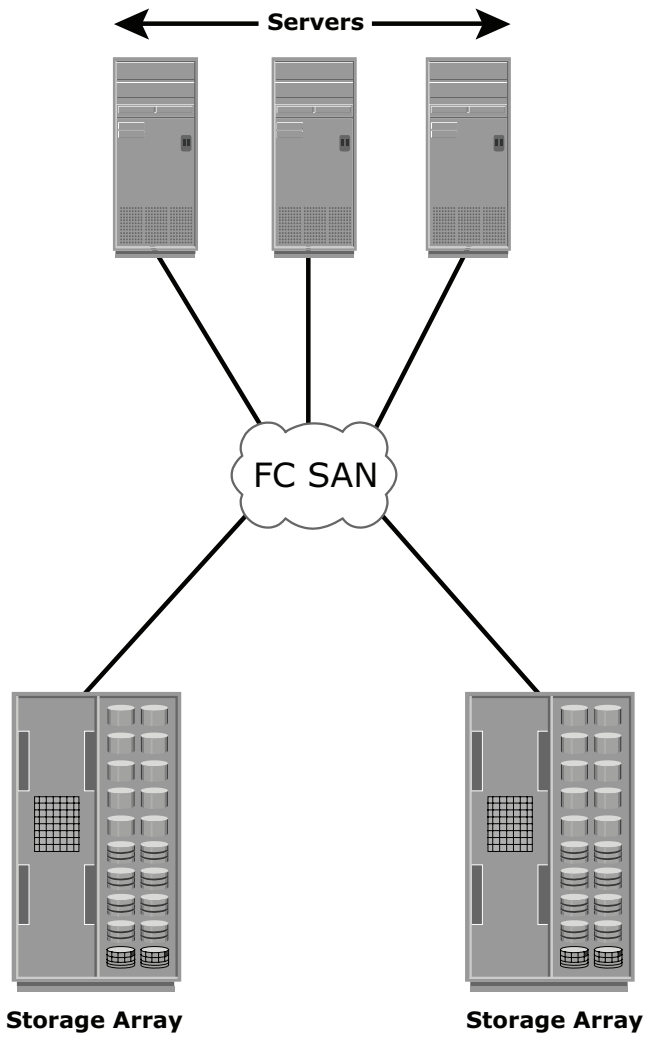

Figure 6-1: SAN implementation

A SAN provides the physical communication infrastructure and enables secure and robust communication between host and storage devices. The SAN management interface organizes connections and manages storage elements and hosts.

In its earliest implementation, the SAN was a simple grouping of hosts and the associated storage that was connected to a network using a hub as a connectivity device. This configuration of a SAN is known as a *Fibre Channel Arbitrated Loop (FC-AL)*, which is detailed later in the chapter. Use of hubs resulted in isolated FC-AL SAN islands because hubs provide limited connectivity and bandwidth.

The inherent limitations associated with hubs gave way to high-performance FC *switches*. The switched fabric topologies improved connectivity and performance, which enabled SANs to be highly scalable. This enhanced data accessibility to applications across the enterprise. FC-AL has been abandoned for SANs due to its limitations, but still survives as a disk-drive interface. Figure 6-2 illustrates the FC SAN evolution from FC-AL to enterprise SANs.

Today, Internet Protocol (IP) has become an option to interconnect geographically separated SANs. Two popular protocols that extend block-level access to applications over IP are iSCSI and Fibre Channel over IP (FCIP). These protocols are detailed in Chapter 8.

6.3 Components of SAN

A SAN consists of three basic components: servers, network infrastructure, and storage. These components can be further broken down into the following key elements: node ports, cabling, interconnecting devices (such as FC switches or hubs), storage arrays, and SAN management software.

6.3.1 Node Ports

In fibre channel, devices such as hosts, storage and tape libraries are all referred to as *nodes.* Each node is a source or destination of information for one or more nodes. Each node requires one or more ports to provide a physical interface for communicating with other nodes. These ports are integral components of an HBA and the storage front-end adapters. A port operates in full-duplex data transmission mode with a *transmit (Tx)* link and a *receive (Rx)* link (see Figure 6-3).

6.9.1 Core-Edge Fabric

In the *core-edge fabric* topology, there are two types of switch tiers in this fabric. The *edge tier* usually comprises switches and offers an inexpensive approach to adding more hosts in a fabric. The tier at the edge fans out from the tier at the core. The nodes on the edge can communicate with each other.

The *core tier* usually comprises enterprise directors that ensure high fabric availability. Additionally all traffic has to either traverse through or terminate at this tier. In a two-tier configuration, all storage devices are connected to the core tier, facilitating fan-out. The host-to-storage traffic has to traverse one and two ISLs in a two-tier and three-tier configuration, respectively. Hosts used for mission-critical applications can be connected directly to the core tier and consequently avoid traveling through the ISLs to process I/O requests from these hosts.

The core-edge fabric topology increases connectivity within the SAN while conserving overall port utilization. If expansion is required, an additional edge switch can be connected to the core. This topology can have different variations. In a *single-core topology*, all hosts are connected to the edge tier and all storage is connected to the core tier. Figure 6-21 depicts the core and edge switches in a single-core topology.

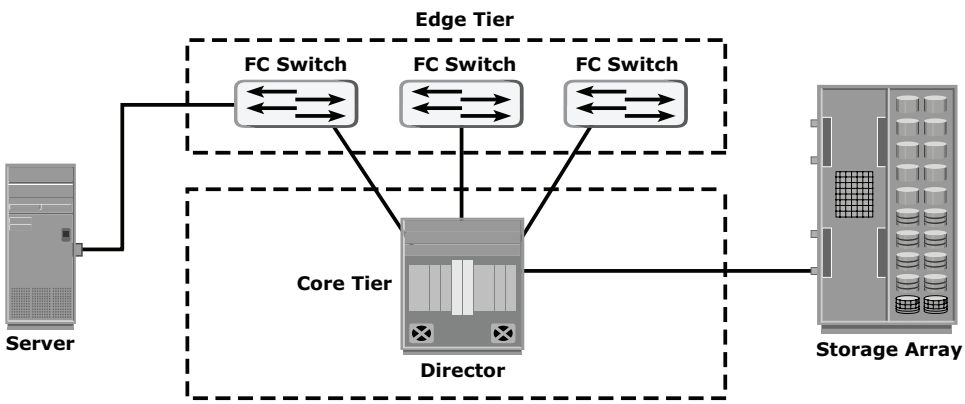

Figure 6-21: Single core topology

A *dual-core topology* can be expanded to include more core switches. However, to maintain the topology, it is essential that new ISLs are created to connect each edge switch to the new core switch that is added. Figure 6-22 illustrates the core and edge switches in a dual-core topology.

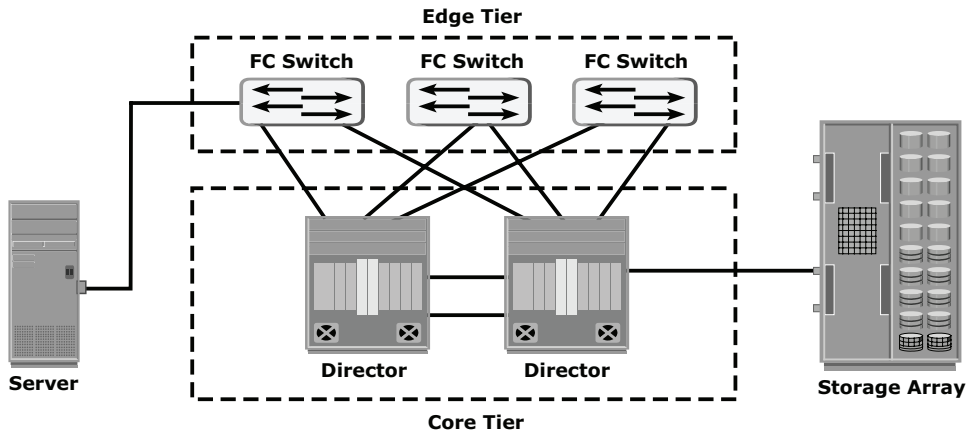

Figure 6-22: Dual-core topology

Benefits and Limitations of Core-Edge Fabric

The core-edge fabric provides one-hop storage access to all storage in the system. Because traffic travels in a deterministic pattern (from the edge to the core), a core-edge provides easier calculation of ISL loading and traffic patterns. Because each tier's switch is used for either storage or hosts, one can easily identify which resources are approaching their capacity, making it easier to develop a set of rules for scaling and apportioning.

A well-defined, easily reproducible building-block approach makes rolling out new fabrics easier. Core-edge fabrics can be scaled to larger environments by linking core switches, adding more core switches, or adding more edge switches. This method can be used to extend the existing simple core-edge model or to expand the fabric into a compound or complex core-edge model.

However, the core-edge fabric may lead to some performance-related problems because scaling a core-edge topology involves increasing the number of ISLs in the fabric. As more edge switches are added, the domain count in the fabric increases. A common best practice is to keep the number of host-to-storage hops unchanged, at one hop, in a core-edge. Hop count represents the total number of devices a given piece of data (packet) passes through. Generally a large hop count means greater the transmission delay between data traverse from its source to destination.

Command-line utilities such as Telnet and SSH may be used to log on to the switch over IP and issue CLI commands. The primary purpose of the CLI is to automate the management of a large number of switches or directors with the use of scripts. The third option is to use browser-based tools that provide GUIs. These Java-based tools can also display the topology map.

Fabricwide management and monitoring is accomplished by using vendor-specific tools and Simple Network Management Protocol (SNMP)-based, third-party software.

EMC ControlCenter SAN Manager provides a single interface for managing Storage Area Network. With SAN Manager one can discover, monitor, manage, and configure complex heterogeneous SAN environments faster and easier. It streamlines and centralizes SAN management operations across multi-vendor storage networks and storage devices. It enables storage administrators to manage SAN zones and LUN masking consistently across multi-vendor SAN arrays and switches. EMC ControlCenter SAN Manager also supports virtual environments including VMware, Symmetrix Virtual Provisioning, and Virtual SANs.

Figure 6-25 illustrates EMC ControlCenter SAN Manager interface.

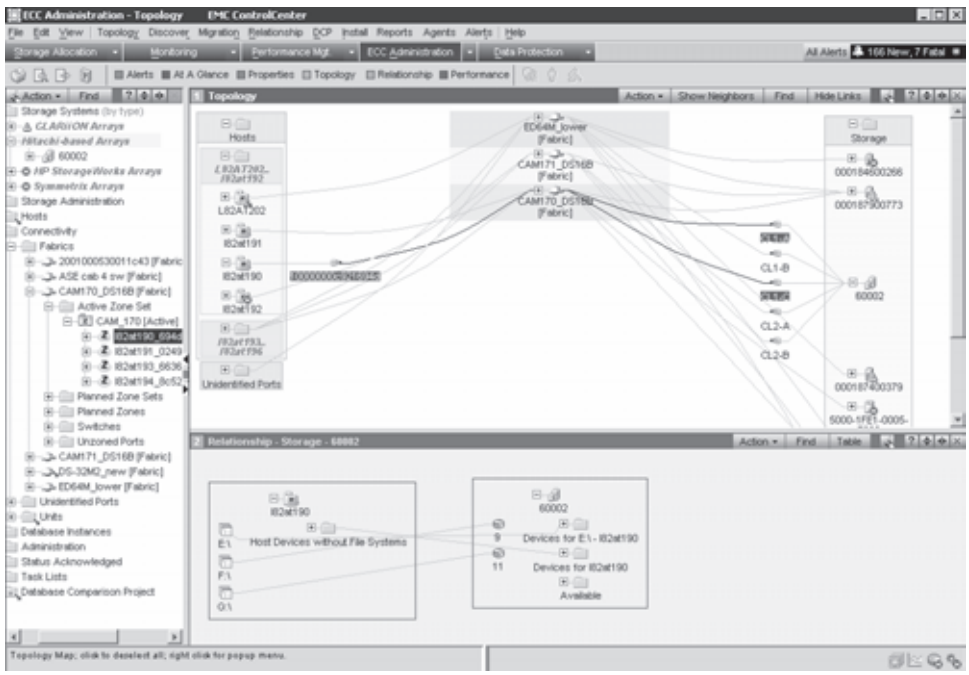

Figure 6-25: Managing FC switches through SAN Manager

Summary

The SAN has enabled the consolidation of storage and benefited organizations by lowering the cost of storage service delivery. SAN reduces overall operational cost and downtime and enables faster application deployment.

SANs and tools that have emerged for SANs enable data centers to allocate storage to an application and migrate workloads between different servers and storage devices dynamically. This significantly increases server utilization.

SANs simplify the business-continuity process because organizations are able to logically connect different data centers over long distances and provide cost-effective, disaster recovery services that can be effectively tested.

The adoption of SANs has increased with the decline of hardware prices and has enhanced the maturity of storage network standards. Small and medium-size enterprises and departments that initially resisted shared storage pools have now begun to adopt SANs.

This chapter detailed the components of a SAN and the FC technology that forms its backbone. FC meets today's demands for reliable, high-performance, and low-cost applications.

The interoperability between FC switches from different vendors has enhanced significantly compared to early SAN deployments. The standards published by a dedicated study group within T11 on SAN routing, and the new product offerings from vendors, are now revolutionizing the way SANs are deployed and operated.

Although SANs have eliminated islands of storage, their initial implementation created islands of SANs in an enterprise. The emergence of the iSCSI and FCIP technologies, detailed in Chapter 8, has pushed the convergence of the SAN with IP technology, providing more benefits to using storage technologies.

EXERCISES

1. **What is zoning? Discuss a scenario,**
   - **(i) where soft zoning is preferred over hard zoning.**
   - **(ii) where hard zoning is preferred over soft zoning.**
2. **Describe the process of assigning FC address to a node when logging in to the network for the first time.**
3. **Seventeen switches, with 16 ports each, are connected in a mesh topology. How many ports are available for host and storage connectivity if you create a high-availability solution?**
4. **Discuss the advantage of FC-SW over FC-AL.**
5. **How flow control works in FC network.**
6. **Why is class 3 service most preferred for FC communication?**

Chapter 7
Network-Attached Storage

Network-attached storage (NAS) is an IP-based file-sharing device attached to a local area network. NAS provides the advantages of server consolidation by eliminating the need for multiple file servers. It provides storage consolidation through file-level data access and sharing. NAS is a preferred storage solution that enables clients to share files quickly and directly with minimum storage management overhead. NAS also helps to eliminate bottlenecks that users face when accessing files from a general-purpose server.

KEY CONCEPTS

- NAS Device
- Remote File Sharing
- NAS Connectivity and Protocols
- NAS Performance and Availability
- MTU and Jumbo Frames

NAS uses network and file-sharing protocols to perform filing and storage functions. These protocols include TCP/IP for data transfer and CIFS and NFS for remote file service. NAS enables both UNIX and Microsoft Windows users to share the same data seamlessly. To enable data sharing, NAS typically uses NFS for UNIX, CIFS for Windows, and File Transfer Protocol (FTP) and other protocols for both environments. Recent advancements in networking technology have enabled NAS to scale up to enterprise requirements for improved performance and reliability in accessing data.

A NAS device is a dedicated, high-performance, high-speed, single-purpose file serving and storage system. NAS serves a mix of clients and servers over an IP network. Most NAS devices support multiple interfaces and networks.

A NAS device uses its own operating system and integrated hardware, software components to meet specific file service needs. Its operating system is optimized for file I/O and, therefore, performs file I/O better than a general-purpose server. As a result, a NAS device can serve more clients than traditional file servers, providing the benefit of server consolidation.

This chapter describes the components of NAS, different types of NAS implementations, the file-sharing protocols, and the transport and network layer protocols used in NAS implementations. The chapter also explains NAS design considerations and factors that affect NAS performance.

7.1 General-Purpose Servers vs. NAS Devices

A NAS device is optimized for file-serving functions such as storing, retrieving, and accessing files for applications and clients. As shown in Figure 7-1, a general-purpose server can be used to host any application, as it runs a generic operating system. Unlike a general-purpose server, a NAS device is dedicated to file-serving. It has a real-time operating system dedicated to file serving by using open-standard protocols. Some NAS vendors support features such as native clustering for high availability.

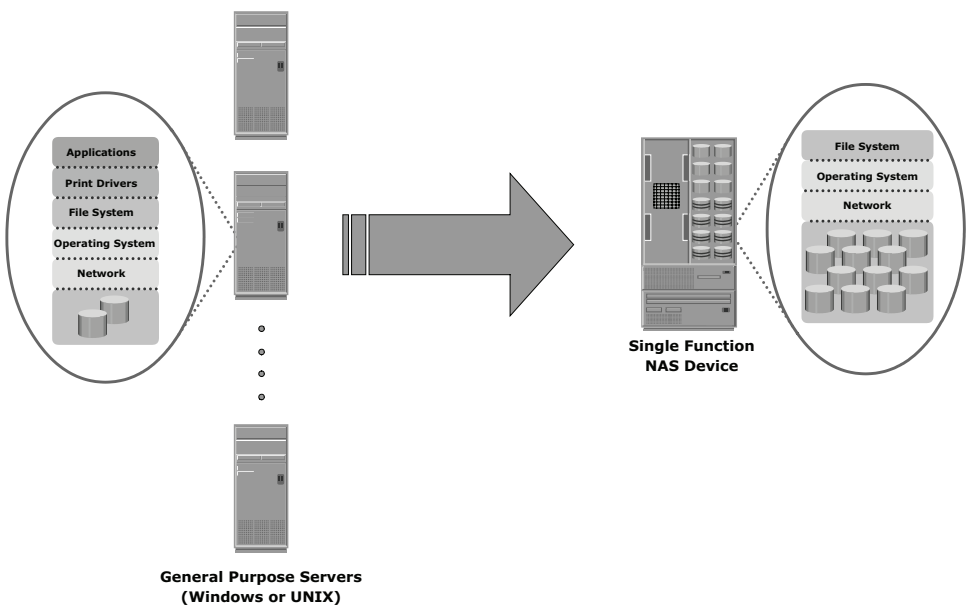

Figure 7-1: General purpose server vs. NAS device

7.2 Benefits of NAS

NAS offers the following benefits:

- **Supports comprehensive access to information:** Enables efficient file sharing and supports many-to-one and one-to-many configurations. The

many-to-one configuration enables a NAS device to serve many clients simultaneously. The one-to-many configuration enables one client to connect with many NAS devices simultaneously.

- **Improved efficiency:** Eliminates bottlenecks that occur during file access from a general-purpose file server because NAS uses an operating system specialized for file serving. It improves the utilization of general-purpose servers by relieving them of file-server operations.
- **Improved flexibility:** Compatible for clients on both UNIX and Windows platforms using industry-standard protocols. NAS is flexible and can serve requests from different types of clients from the same source.
- **Centralized storage:** Centralizes data storage to minimize data duplication on client workstations, simplify data management, and ensures greater data protection.
- **Simplified management:** Provides a centralized console that makes it possible to manage file systems efficiently.
- **Scalability:** Scales well in accordance with different utilization profiles and types of business applications because of the high performance and low-latency design.
- **High availability:** Offers efficient replication and recovery options, enabling high data availability. NAS uses redundant networking components that provide maximum connectivity options. A NAS device can use clustering technology for failover.
- **Security:** Ensures security, user authentication, and file locking in conjunction with industry-standard security schemas.

7.3 NAS File I/O

NAS uses file-level access for all of its I/O operations. File I/O is a high-level request that specifies the file to be accessed, but does not specify its logical block address. For example, a file I/O request from a client may specify reading 256 bytes from byte number 1152 onward in a specific file. Unlike block I/O, there is no disk volume or disk sector information in a file I/O request. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrieve data.

The NAS operating system issues a block I/O request to fulfill the file read and write requests that it receives. The retrieved data is again converted to file-level I/O for applications and clients.

7.3.1 File Systems and Remote File Sharing

A file system is a structured way of storing and organizing data files. Many file systems maintain a file access table to simplify the process of finding and accessing files.

7.3.2 Accessing a File System

A file system must be mounted before it can be used. In most cases, the operating system mounts a local file system during the boot process. The mount process creates a link between the file system and the operating system. When mounting a file system, the operating system organizes files and directories in a tree-like structure and grants the user the privilege of accessing this structure. The tree is rooted at a mount point that is named using operating system conventions. Users and applications can traverse the entire tree from the root to the leaf nodes. Files are located at leaf nodes, and directories and subdirectories are located at intermediate roots. The relationship between the user and the file system terminates when the file system is unmounted. Figure 7-2 shows an example of the UNIX directory structure under UNIX operating environments.

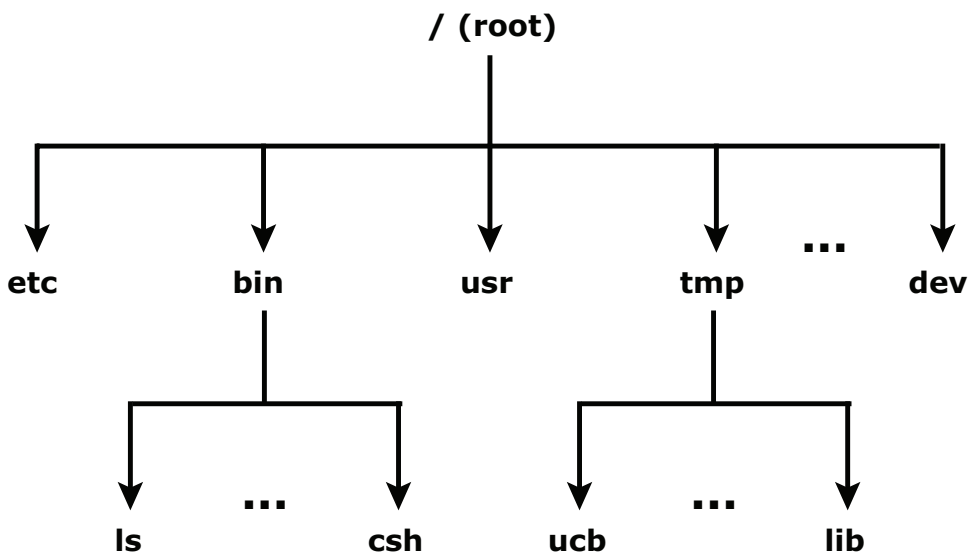

Figure 7-2: UNIX directory structure

7.3.3 File Sharing

File sharing refers to storing and accessing data files over a network. In a file-sharing environment, a user who creates the file (the creator or owner of a file)

determines the type of access to be given to other users (read, write, execute, append, delete, and list) and controls changes to the file. When multiple users try to access a shared file at the same time, a protection scheme is required to maintain data integrity and, at the same time, make this sharing possible.

File Transfer Protocol (FTP), distributed file systems, and a client/server model that uses a file-sharing protocol are some examples of implementations of file-sharing environments.

FTP is a client/server protocol that enables data transfer over a network. An FTP server and an FTP client communicate with each other using TCP as the transport protocol. FTP, as defined by the standard, is not a secure method of data transfer because it uses unencrypted data transfer over a network. FTP over Secure Shell (SSH) adds security to the original FTP specification.

A *distributed file system (DFS)* is a file system that is distributed across several hosts. A DFS can provide hosts with direct access to the entire file system, while ensuring efficient management and data security.

The traditional *client/server model*, which is implemented with file-sharing protocols, is another mechanism for remote file sharing. In this model, the clients mount remote file systems that are available on dedicated file servers. The standard client/server file-sharing protocols are NFS for UNIX and CIFS for Windows. NFS and CIFS enable the owner of a file to set the required type of access, such as read-only or read-write, for a particular user or group of users.

In both of these implementations, users are unaware of the location of the file system. In addition, a *name service*, such as Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), and Network Information Services (NIS), helps users identify and access a unique resource over the network. A *naming service protocol* creates a namespace, which holds the unique name of every network resource and helps recognize resources on the network.

7.4 Components of NAS

A NAS device has the following components (see Figure 7-3):

- NAS head (CPU and Memory)
- One or more network interface cards (NICs), which provide connectivity to the network. Examples of NICs include Gigabit Ethernet, Fast Ethernet, ATM, and Fiber Distributed Data Interface (FDDI).
- An optimized operating system for managing NAS functionality
- NFS and CIFS protocols for file sharing
- Industry-standard storage protocols to connect and manage physical disk resources, such as ATA, SCSI, or FC

The NAS environment includes clients accessing a NAS device over an IP network using standard protocols.

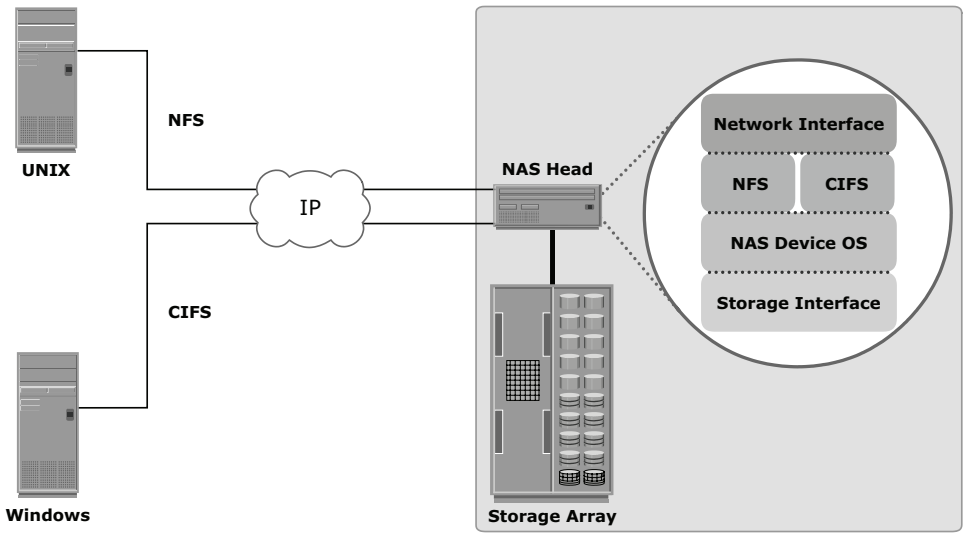

Figure 7-3: Components of NAS

7.5 NAS Implementations

As mentioned earlier, there are two types of NAS implementations: integrated and gateway. The *integrated NAS* device has all of its components and storage system in a single enclosure. In *gateway* implementation, NAS head shares its storage with SAN environment.

7.5.1 Integrated NAS

An integrated NAS device has all the components of NAS, such as the NAS head and storage, in a single enclosure, or frame. This makes the integrated NAS a self-contained environment. The NAS head connects to the IP network to provide connectivity to the clients and service the file I/O requests. The storage consists of a number of disks that can range from low-cost ATA to high-throughput FC disk drives. Management software manages the NAS head and storage configurations.

An integrated NAS solution ranges from a low-end device, which is a single enclosure, to a high-end solution that can have an externally connected storage array.