Chapter 6

Device Overviews

Magnetic disks form the foundation of online storage systems. These devices are augmented in storage architectures with magnetic tape, enabling an economical and enduring media for storing historical information. Additional storage technologies such as solid-state disks and optical devices play both specialized and enabling roles that support these two fundamental storage devices. Although the concepts in this chapter may seem elementary in nature, the mastery of these storage principles are an integral part of understanding how to apply these basic storage components optimally when designing and implementing a storage network.

Storage architectures must support various, and at times, disparate business applications. These applications demonstrate various I/O requirements that place a demand on the storage devices. The specific I/O requirements for the application are referred to as an I/O workload (more information about I/O workloads can be found in Chapter 17). The understanding of storage at the device level will contribute to developing an entire storage system that forms the critical basis for all application performance.

Application systems designed with the supporting storage devices in context with workloads begin to develop into responsive infrastructures. Without device knowledge and consideration, storage systems often form monolithic storage farms that dictate a one size fits all approach. Such conditions set the stage for unresponsive and costly application performance. With today's Plug and Play implementation frenzy, the thought and analysis given aligning compatible storage devices with storage workloads is generally an afterthought (more information regarding workload planning for storage networks are contained in Part V and Chapters 21 and 22).

Peripheral Connectivity Components and Concepts

Peripheral devices are conceptually defined as those computer elements that are not part of the computer's motherboard but must attach to the periphery of this self-contained system. Storage peripheral devices are connected using two distinct functions, the host adapter and the controller. Although at times they are integrated into one device, the distinctions between the two, though often fuzzy, are required for all storage attachment. These devices are depicted in Figure 6-1, which illustrates their proximity to the main computer system. The basic functionality of the host adapter allows peripheral devices to connect to a server's internal bus system. The controller's basic function, although it has many, allows multiple devices to communicate through one adapter path. Together they form the connectivity structure of a server's I/O system.

This encompasses all devices that process I/O from the server to include storage, but also printers, plotters, graphic displays, keyboards, and other devices that interact and are directly connected to the system. For our discussion, we will focus on storage devices. However, we will devote a portion of our discussion to network connectivity adapters given their integration and importance in developing storage networks.

A short discussion about network connectivity can provide an important set of principles that encompass the functions of direct device connectivity and how networks interface with the server. As shown in Figure 6-2, the Network Interface Card (NIC)

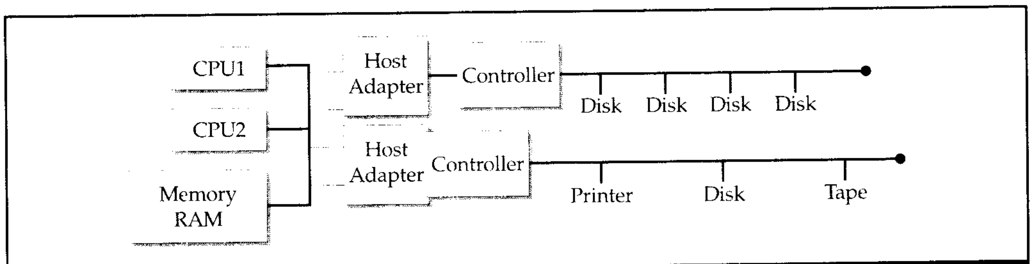

Figure 6-1. *Peripheral connections to the main computer elements*

connects the system bus to the external network, be it a local area network (LAN) or a wide area network (WAN). The components are the same as with any host adapter; it consists of an electronics card that connects to the server motherboard or a backplane.

This is supported by a set of related software that interacts with the low-level hardware protocols and the operating system, sometimes referred to as firmware or microcode. Firmware refers to software instructions that interact directly with microprocessor components. These instructions are specific to the type of microprocessor being used and generally not accessible to users.

Note *A backplane is a type of connector circuit board where the board has no other circuitry than to connect together bus slots or connectors. Thus, an entire computer system can reside on a motherboard that can connect into a backplane for common I/O attachment. Refer to blade computing, high-end SMP, and shared I/O configurations in Chapter 7 and Appendix D.*

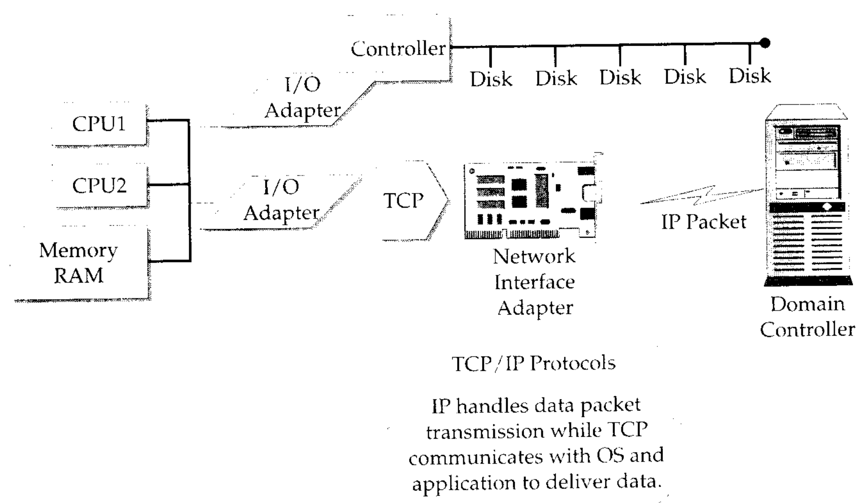
Controller
Disk
Disk
Disk
Disk
Disk
I/O Adapter
CPU1
CPU2
Memory RAM
I/O Adapter
TCP
Network Interface Adapter
IP Packet
Domain Controller
TCP/IP Protocols
IP handles data packet transmission while TCP communicates with OS and application to deliver data.

Figure 6-2. *Connecting to the network*

The functions of the Network Interface Card (NIC) are to transmit the identity of the server to the network, receive messages addressed to the server, and translate the network protocols to pass on to the operating system components. The user or application data and instructions encapsulated within the networking protocols can now be used by the server's applications.

On the out-bound side, the reverse takes place. The application turns over its data and target location to the operating system, which works with the NIC software drivers to encapsulate the data and ensure it's transmitted in the correct way to the network.

The network software that enables this communicates and works with the NIC in TCP (Transmission Control Protocol) or IP (Internet Protocol). Together they form what is called the TCP/IP software layers, or stacks—the open systems standard used by all computer manufacturers to communicate within networks. As previously illustrated, we can view TCP/IP simply as the message format and envelope, where TCP allows the application and operating system to encapsulate the user data while the IP portion contains the envelope that transports the data to its network target locations.

As discussed previously, all peripheral devices have a host adapter and controller component. This concept, although viewed somewhat more abstractly in networks, is the same. However, to place this in its proper perspective we must view the other devices in the network as being externally connected, be they servers or clients. That means the controller functionality in networks consists of switches and routers that interconnect the multiple devices—in this case, other computers that are attached to the network.

Figure 6-3 extends the network diagram into the area of controllers, with switches, routers, telephone exchange equipment, and other WAN-related communication devices.

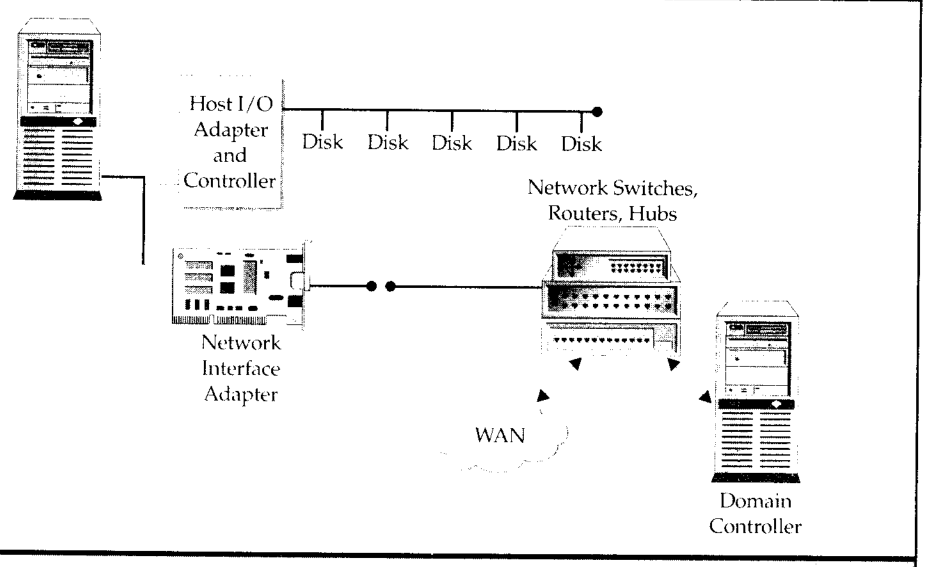
Host I/O Adapter and Controller
Disk
Disk
Disk
Disk
Disk
Network Switches, Routers, Hubs
Network Interface Adapter
WAN
Domain Controller

Figure 6-3. *Overview of connectivity inside the network*

As indicated in this illustration, the equipment and software that make up the network are numerous; however, none of it needs to be known by the server or client, other than the network address. We have seen the high-level benefits and configurations when this concept is extended into storage components. Unlike traditional transient network traffic, storage I/O operations require a more defined protocol using the same host adapter and controller concepts.

The Host Adapter

The host adapter connects the computer's system bus to the external pathway where various devices are attached. In many cases, this is another bus technology, one that specializes in communicating with specific device types and which requires unique commands to perform I/O operations with the server. Host adapters consist of two main components. First is a hardware component, generally in the form of electronic cards that attach to the server motherboard or backplane (more about bus connectivity in Chapter 7). Just as important is the related software component that allows the specific devices to drive the underlying hardware bus protocols, the operating system, and, ultimately, the application.

The combined components, as shown in Figure 6-4, consist of microcode that functions within the electronic card circuitry, microcode of the basic input/output (BIOS) that exists on the computer's motherboard, and high-level software that executes on the computer. The high-level software (commonly called drivers) works with the hardware microcode to provide the device command set (native I/O calls), services for the device, the operating system interface, and error recovery and checking routines. All of these functions communicate with the operating system through the BIOS microcode embedded in the computer's motherboard. Any device utility software running on the computer will use the same native software.

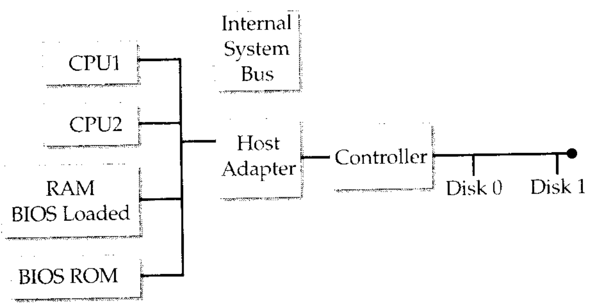

Figure 6-4. *Host adapter functions*

Host adapters are provided as bundled solutions from storage vendors, or piecemeal through a host of hardware vendors. This requires the configuration activities to be both challenging and creative. Regardless of how the solution is acquired, the configurations are pretty standard, albeit limited. First, there are only so many host adapters you can attach to a server. This is a function of the server itself and, more specifically, the bus technology within the server. Second is the effect of multiple software drivers that can operate within a single server system—which means that having multiple vendor software drivers can become risky, given the different implementation that occurs within the division between the hardware (for example, the microcode) and the high-level software driver executing within the server.

This requires a homogeneous approach to implementation—in other words, a common vendor solution to host adapters running on a single server. In addition, the number of adapters available for an individual server—more specifically, the server's bus—defines the high end of paths that can be physically configured for directly connected I/O peripherals. This presents a theoretical limiting factor in storage sizes given the type of workload.

The number of paths, as depicted by Figure 6-5, shows the limitations of single path configurations when dealing with even medium-sized databases. Other components within the storage hierarchy will also be affected.

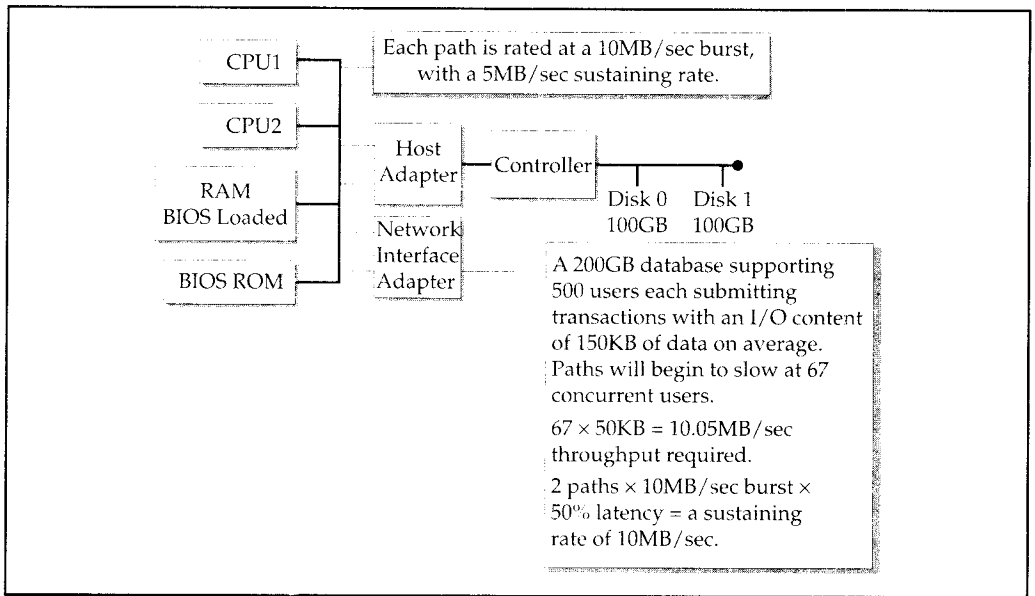
CPU1
CPU2
RAM BIOS Loaded
BIOS ROM
Each path is rated at a 10MB/sec burst, with a 5MB/sec sustaining rate.
Host Adapter
Controller
Disk 0 100GB
Disk 1 100GB
Network Interface Adapter
A 200GB database supporting 500 users each submitting transactions with an I/O content of 150KB of data on average. Paths will begin to slow at 67 concurrent users.
67 x 50KB = 10.05MB/sec throughput required.
2 paths x 10MB/sec burst x 50% latency = a sustaining rate of 10MB/sec.

Figure 6-5. *Path limitations through single adapter strategies*

The Controller

The controller allows multiple devices to be connected to the adapter. As the adapter forms the primary path to and from the server, the controller provides the connectivity to the actual devices that communicate with the server. As indicated in Figure 6-6, the controller functions can be manifold but primarily control the flow of data to and from the devices. In doing so, it provides an addressing scheme for the devices, shields the server from the operational complexities of the device, and provides another temporary location for data (prior to it being written or cached) for faster access.

Like adapters, controllers consist of both hardware and software components. However, their primary functionality is driven by hardware microcode, as indicated in Figure 6-6.

By keeping track of a disk's position and being the initiator of the drive's read/write electronics (see the next section for more details on magnetic head disk assemblies, referred to as *HDA* mechanisms), the controller provides data buffering between the disk and the host interface or adapter with its error detection, correction, retry, and recovery functions. Both of these functions are susceptible to changes in connectivity due to their communications with the operating system.

The controller also offers the system's attached disks as an acceptable storage location even though the actual physical mapping may be very different. This allows the controller to translate I/O system requests into actual physical commands that can be executed on the real disk drive. Typical in most systems, the controllers and operating system view the disk drive locations as a linear pool of logical locations. This level of abstraction translation, though having existed in I/O configurations for many years, is necessary to translate logical blocks of data into physical addresses on the storage media.

This is a key concept within the controller functionality. It demonstrates how the operating system communicates within a logical set of storage locations that are considered virtual to the controller, which translates those locations into the physical realities the controller must manage. Figure 6-7 represents an overview of how logical views are mapped into physical views.

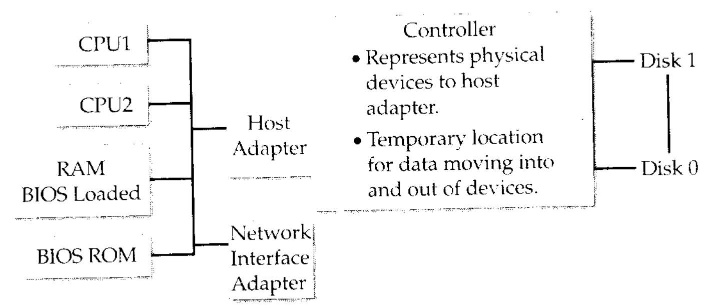
CPU1
CPU2
RAM
BIOS Loaded
BIOS ROM
Host Adapter
Network Interface Adapter
Controller
• Represents physical devices to host adapter.
• Temporary location for data moving into and out of devices.
Disk 1
Disk 0

Figure 6-6. *Controller functions*

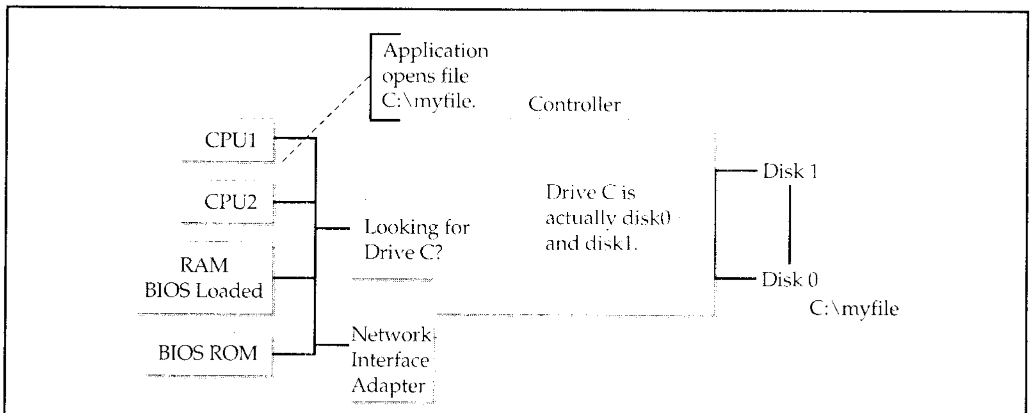
Application opens file C:\myfile.
Controller
CPU1
CPU2
RAM BIOS Loaded
BIOS ROM
Looking for Drive C?
Network Interface Adapter
Drive C is actually disk0 and disk1.
Disk 1
Disk 0
C:\myfile

Figure 6-7. *The controller's view of reality*

Magnetic Disk Storage

Magnetic disk storage is the packaging of magnetic media disks, packaged read/write head electronics on a spinning electro-mechanical device. Disks are made up of round platters coated with a magnetic material that can be written and read electronically through the encoding of the magnetic media with binary data.

Figure 6-8 shows the platters that make up the magnetic media and the read/write electronics that surround the disks, all of which resembles old analogue record players. The read/write electronics are recording heads that both read and write data. As discussed previously, the head assembly can house data buffers that compensate for delays in reading or writing data.

The platters are organized into tracks, concentric rings that form the writable surface of the platter. These are further divided into sectors which form the smallest readable/writable units on the disk. As the disk rotates, portions of the sectors become visible as the read/write heads are positioned to that track.

Drives are made up of more than one disk, with each storing data on both sides. Each surface must have its own read/write head and a set of tracks which can be accessed through a single position (for instance, when vertically aligned). This is called a cylinder. Given this organization, every sector on the drive can be uniquely addressed by its cylinder, head, and sector numbers.

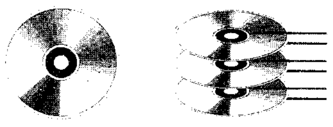

Figure 6-8. *The disk drive anatomy*

Sectors are important because they are the fundamental metric that determines the size of data stored within the disk. The number of bytes in a sector corresponds to the disk's formatted capacity. As indicated in Figure 6-8, the sector must contain information about addresses and synchronizing HDA error corrections in the header fields. Given that the header and associated synchronization gaps are not usable for data, the formatted capacity is always less. Formatting a drive writes the header information and arbitrary data pattern to verify correct error checking. Although entire drives are normally formatted at one time, a technique called *soft sectoring* allows a single track or hard sector to be formatted.

Technical specifications that surround disks include the capacity, transfer rate, and average access time. Each metric is important when considering the given workload supported. While capacity may be an issue, the number of transactions necessary to read data from the disk may compromise seek time. On the other hand, large transfers of data (like that used by datacentric applications in data warehousing) may push capacity and transfer rates to their optimum while sacrificing average seek time.

Capacity, as mentioned previously, comes in two forms: formatted and unformatted. As you've probably guessed, the larger the disk, the greater the amount of addressing and error correction code (ECC) overhead, thereby reducing space for actual user data.

Transfer rate and throughput refers to the speed at which the HDA processes read and write operations. Throughput is the amount of data the drive can deliver or accept at the interface on a sustained basis. This is important because the term "interface" can have many connotations and implementations.

Average access time is the time needed to position the HDA to a specific cylinder, as well as and the time it takes for the requested sector to rotate under the heads. Thus, disk rotation speeds are related to capacity.

Disk Systems

By integrating controllers into the disk assemblies, we begin to have complete storage systems. These come in an amazing amount of configurations and offerings, but the basic model is the attached controller that uses multiple disks drives within the enclosure.

As disk drives are linked together, they form an array. This is used for basic capacity and performance enhancements. Disks configured in this manner can be used individually as Disk1 through Disk4, as shown in Figure 6-9, or they can be used in an integrated fashion by combing the capacities of the array and using them as one large disk. For example, in Figure 6-9, Disk1 through 4 can be combined to form a virtual disk *V* which can use the entire capacity of the array. In this scenario, the application I/O sees disk *V* as one large disk drive even though it's only a virtual representation of the four physical drives. The application and the operating system understands that data can be stored on disk drive *V* and lets the controller take care of where the data is actually written using the pool of disks 1 through 4.

This forms the basis for advanced functionality in storage systems such as Redundant Arrays of Independent Disks (RAID) and the concept of storage virtualization. The functionality of RAID provides disk redundancy, fault tolerant configurations, and data

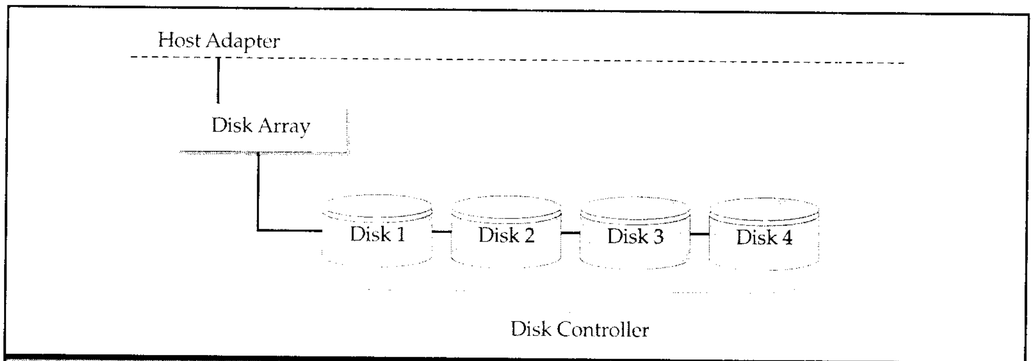

Figure 6-9. *Disk arrays*

protection resiliency. All this is facilitated through the controller mechanisms discussed previously and illustrated in Figure 6-6. The RAID decoupling of the physical drives from the application software I/O requests has driven the usage and advancement of storage virtualization technologies and products.

Letting the server's operating system believe it has a pool of storage capacity frees up the I/O manager services, allowing it to perform other tasks, and facilitates the development of applications without the low-level coding of specific I/O calls. Remember, there is no such thing as a free lunch and the translation of calls to the virtual pool of data has to be processed either in the storage controller or in the third-party storage software running on the server. It may even be a combination of both. Either way, the overhead continues to exist, although it may be offloaded and rely on RAID processed within the controller.

Disk Arrays

The first level of storage array configuration is JBOD (Just a Bunch Of Disks), depicted in Figure 6-9. JBOD links a set of disk drives together to form an array where each drive becomes an addressable unit. These provide additional capacity; however, they do not provide any fault resiliency in the event of an inoperable drive. Partitioning data throughout the disks and providing a layer of virtualization services can be done through software, which is generally part of the I/O management of the operating system. These functions are also available through third-party software applications.

Although these functions offer a level of data redundancy (depending on the sophistication of the software), they do not provide any method of fault tolerance for continuous operations in the event of a lost disk drive. Consequently, while these guarantee some data loss protection, they do not guarantee that the data remains available even if disk drives become inoperable within the array. Functions that provide continuous operations and fault resiliency are provided by RAID.

RAID Storage Arrays

RAID is, as its name implies, a redundant array of independent disks that becomes an addressable unit made up of separate and independent disk drives (shown in Figure 6-10). The main difference from JBOD arrays is that RAID partitions data throughout the array and recovery functions. The recovery functions are developed using disk parity information that is calculated to reassemble missing data from a failed drive to the remaining drives within the array. The data is distributed throughout the array in a manner most effective to the recovery and protection strategy. RAID has several levels in which these recovery and protection strategies can be implemented.

The various RAID levels of implementation provide recovery and data protection that is appropriate to the application's and user's continuity requirements. This allows for performance, data protection, and automatic recovery from drive failures within the array. RAID has become the de facto standard for disk hardware fault tolerance functions. As an accepted standard, all storage manufacturers offer it; however, all storage vendors have also evolved into a diversity of proprietary functions that define the type of fault tolerant protection offered. These are encapsulated within the storage firmware run on the controller and are referred to as both RAID software functions and RAID firmware.

To make things more confusing, some RAID functions can be provided through software that runs on the server. This differentiation becomes distinct as we look at the standard RAID levels as defined throughout the industry. As such, RAID functionality can be selected either through hardware or software components, or a combination of both.

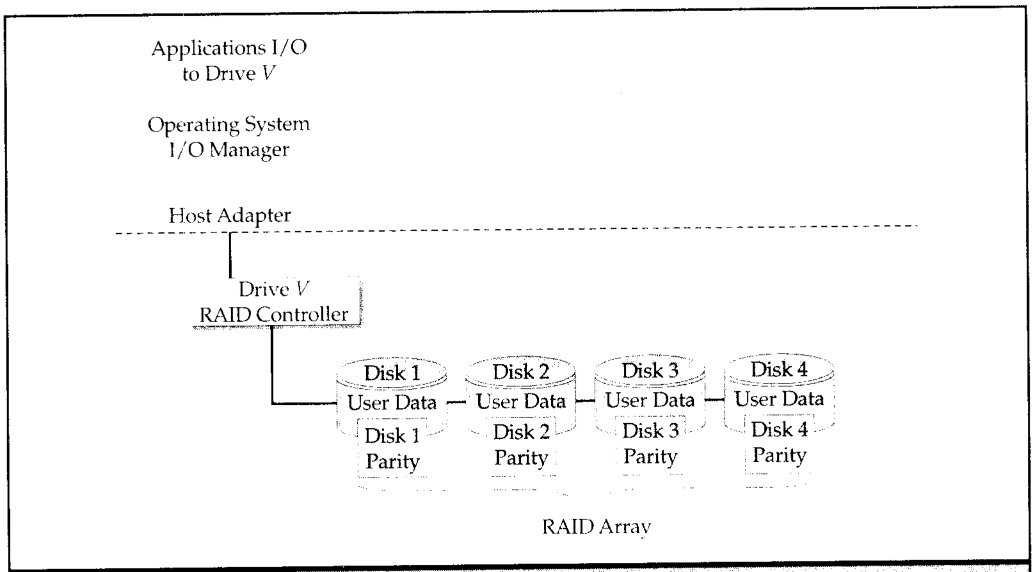
Applications I/O to Drive V
Operating System I/O Manager
Host Adapter
Drive V RAID Controller
Disk 1 User Data Disk 1 Parity
Disk 2 User Data Disk 2 Parity
Disk 3 User Data Disk 3 Parity
Disk 4 User Data Disk 4 Parity
RAID Array

Figure 6-10. *The basic RAID architecture for a storage array*

RAID configuration levels range from numbers 0 through 5 for basic RAID services; however, extended RAID levels, although proprietary to vendor implementations, have gone beyond basic configurations. Given the numerous options for partitioning data and the diversity of ways that parity information can be calculated and partitioned, the options for RAID can be numerous. Through the evolution of real experiences and exposure to typical applications and I/O workloads, two of the five have proven the most valuable, and therefore are the most popular. These are RAID levels 1 and 5. RAID levels 1 and 5 will most likely be the ones implemented within the data center. However, we will pay particular attention to RAID level 4 as an example of how a vendor can use a proprietary and bundled level.

RAID level 0 simply uses the storage array for partitioning of data without any disk parity information. This provides for data protection without a recovery mechanism should any of the data become unavailable from a failed drive. Many software-only data protection mechanisms use a RAID level 0 configuration, where the data is written to primary and secondary files. Similar to RAID level 1 (mirroring) without any disk recovery mechanism, this is sometimes called software mirroring. In other cases, the data is striped across the disk array for performance reasons. This allows the disk write process to take advantage of several head and disk assembly mechanisms for multiple files with high disk write requirements.

RAID level 1 offers the ability to provide a primary and secondary set of data. As one file is updated, its secondary is also updated, keeping a safe copy for data protection just like RAID level 0. The major difference is the calculation of parity information, as shown in Figure 6-11. This is used in the event a disk drive fails within the RAID array. In that case, the available mirrored copy of the data continues processing. When the drive failure is corrected, the unavailable copy of the mirror is reassembled through the parity information contained within the RAID controller. Theoretically, the data is never unavailable and remains online for the application use.

RAID level 5 is a partitioning of the data across the RAID array for performance purposes. This is depicted in Figure 6-12 along with the complement of disk parity information also striped across the array. In the event of a disk failure, the missing data is reassembled through the parity information processed by the RAID controller. Once the failed disk drive is corrected, the data is reassembled back onto the failed drive and the entire array is back in full operation. Theoretically, the data is never unavailable as in RAID level 1.

RAID levels 2 and 3 are derivatives of mirroring and level 5, and use different strategies for storing the parity information. Although RAID level 4 is also a derivative of level 5, it provides striping across the array of user data, and reserves a disk within the array for processing of parity data (shown in Figure 6-13). This is used in some NAS solutions where the storage is bundled and the RAID array processing is hardcoded within the package—in other words, you don't have the option to move to other RAID levels.

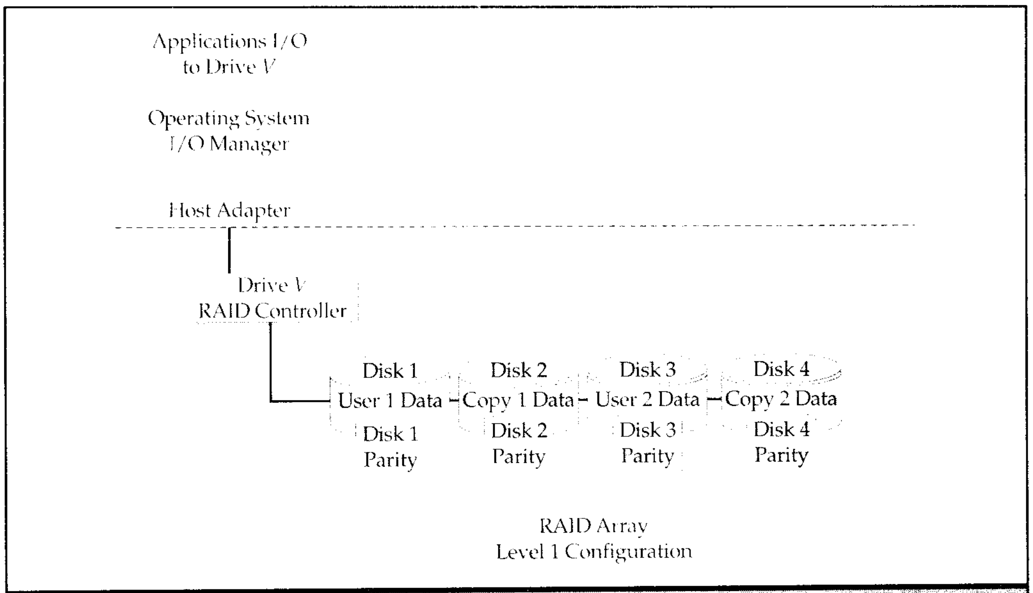

Figure 6-11. *The RAID level 1 configuration*

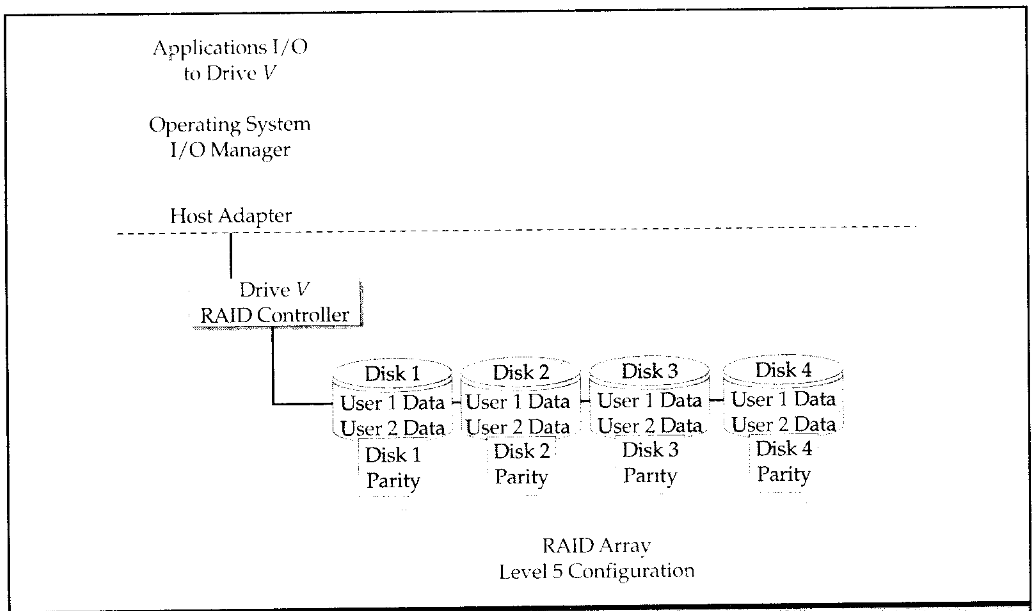

Figure 6-12. *The RAID level 5 configuration*

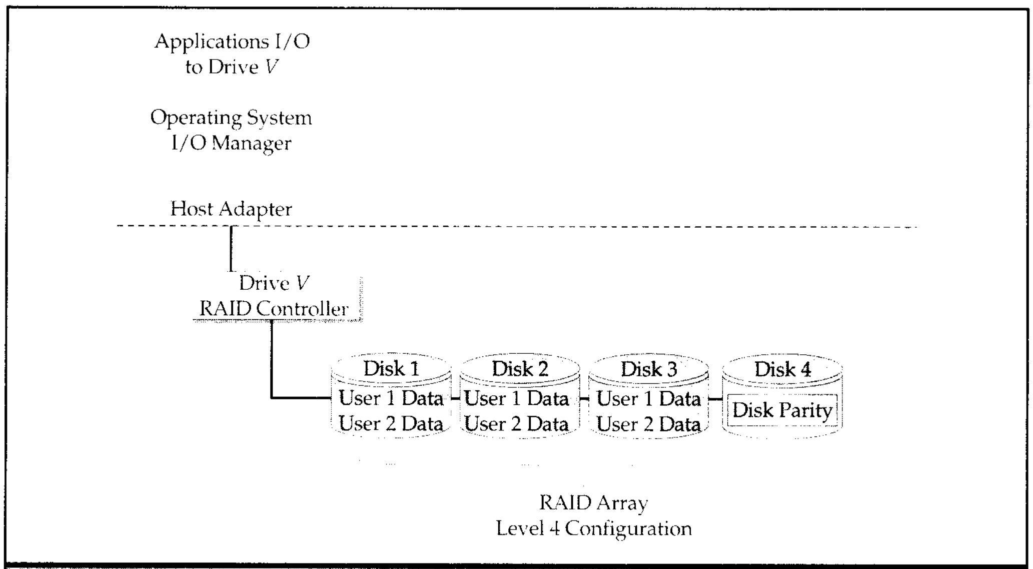

Figure 6-13. *The RAID level 4 configuration*

Magnetic Tape Storage

Tape is the most ubiquitous form of storage today, with more data being stored on tape than any other media. Although tape storage started as the earliest form of mass storage, it survives today due to its cost resiliency and continued capability to store large amounts of systems and application data for safekeeping.

Tape Drives

The most important aspect of tape systems is their sequential nature. Even though a good deal of innovation has been applied to tape technology, write mechanisms, and optimization, the information remains sequentially bound. This is important because of throughput of write processes and speed of access. The ability to find a specific location with tape media requires the transport system to sequentially move throughout the tape to a specific sector.

Once the location is found, it can read the information and reposition itself for additional operations. Within the writing operation, the tape must be positioned in the proper location and written sequentially. Although this has been enhanced through software utilities that provide tape catalogue information that speeds location search for data that has been written to tape, the random nature of access that is inherent with magnetic disks does not exist within tape.

Another important operational note that should not be overlooked is the effect of tape controller functionality. The tape controller, as shown in Figure 6-14, is necessary

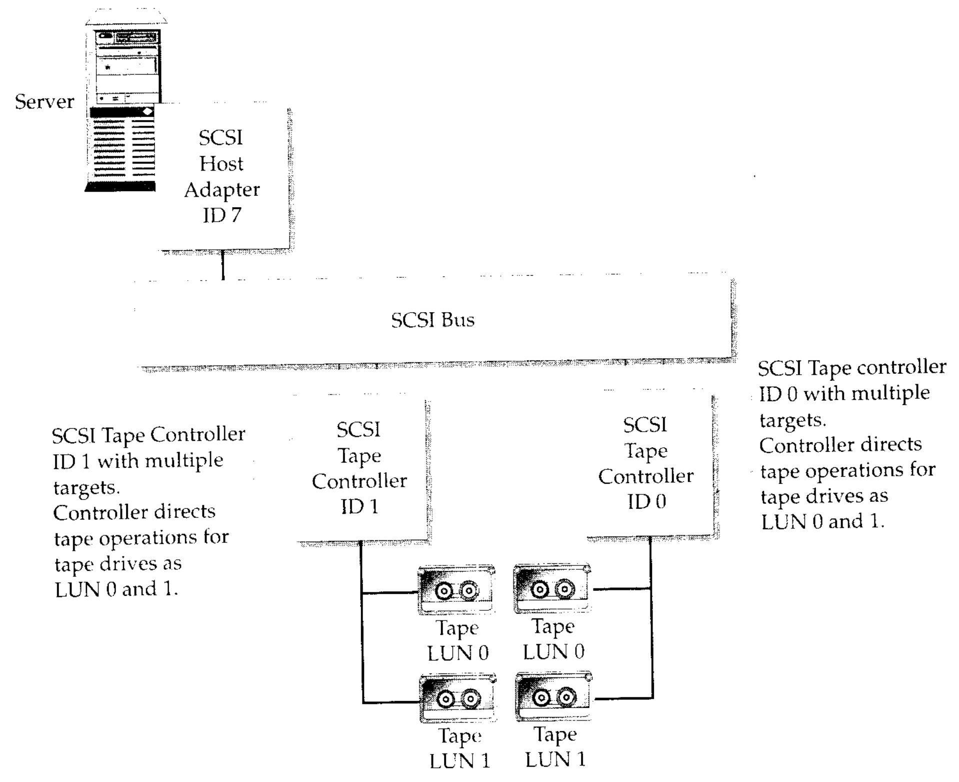
Server
SCSI Host Adapter ID 7
SCSI Bus
SCSI Tape Controller ID 1 with multiple targets. Controller directs tape operations for tape drives as LUN 0 and 1.
SCSI Tape Controller ID 1
SCSI Tape Controller ID 0
SCSI Tape controller ID 0 with multiple targets. Controller directs tape operations for tape drives as LUN 0 and 1.
Tape LUN 0
Tape LUN 0
Tape LUN 1
Tape LUN 1

Figure 6-14. *An overview of the SCSI tape drive*

for many control functions. Among the most important is the capability to address multiple drives within the tape system. Just as with disk controllers, tape controllers provide a level of transparency to the operating system and applications. In addition, and most importantly, the majority of controllers will have some form of error recovery and correction when dealing with data as it comes from the system bus, generally from the RAM buffers. If the tape system has error recovery code (ERC) in its controller, the system RAM will dump its contents to the tape controller buffers and refresh its buffer locations for another load. If not, there is a long wait as the RAM buffers slowly transfer their locations to the read/write head and wait for confirmation.

The effects of tape devices and operations within a storage network environment cannot be overlooked. We will discuss tape systems associated with SAN and NAS in Parts III, IV, and V. However, it's important to point out that fast system resources, such as RAM, become further decoupled from slower devices like tape media through the separation of network devices. As a result, the timing and throughput differences can become magnified and problematic.

Formats

Like disks, a distinguishing characteristic of tape drives is the format used to write and read data to and from the media. Popular formats in today's data center environments range from Digital Linear Tape (DLT), the most popular for commercial enterprises, to new formats such as Linear Tape Open (LTO). The general format specifies the type of media and the technique the tape heads use to write the data, most of which are supported by an open standard so that multiple vendors can provide an approximation of an interoperable solution.

The tape media is divided into parallel tracks. The number of tracks, the configuration of the read/write heads, and the mechanisms surrounding tape transport all make up the uniqueness of each format. Regardless of format, the tape media is formatted into blocks. As discussed, the format of the tape depends on the geometry of the media, the layout of the drive heads, and the transport control mechanisms, which all go together to define the size of the block in bytes that can be written. Blocks make up segments that can be referenced within the location of the media for faster access. Each segment has certain blocks that contain error correction codes and space for directories used by software utilities reserved within track 0 or a special directory track.

Read/Write Mechanisms

The read/write head mechanisms determine not only the format of the tape (tracks, parity, density, and so on), but more importantly the throughput and density of the write process and the speed of the read process. In most cases, today's tape drives contain read/write heads that perform dual roles. Surrounded by a read head, a write can take place as the read function verifies the operation, and vice versa as the tape transport changes direction. As mentioned earlier in our discussion of controllers, today's tape systems support increased levels of data buffering and, in some cases, caching, file systems, and fault-tolerant configurations similar to RAID technologies with disk systems.

Tape Systems

The utilization of tape systems predates all other mass storage strategies used for commercial and scientific computing. The importance of tape in data centers, both present and future, cannot be overlooked. In understanding this legacy form of storage, there are two important points to consider. First is the natural extension to tape systems (which is unlike online disk media): the Tape Library. As data centers create sets of information for recovery, archival, and legal purposes, the tapes must be stored in environments that are secure, environmentally protected, and accessible through some form of catalogue structure. Second, is the level of sophistication vendors have developed to support the offline, nearline, and online strategies that encompass tape usage.

The Tape Library

Let's first examine the tape library. Tape libraries provide both logical and physical archival processes to support the data center. The most important aspect of the library is rapid access to archived information. This requires some type of index, catalogue, and structure whereby the tape hardware working with the catalogue and archival software can locate a tape, mount in a drive, verify (yes, it's very important to verify it's the right tape), and read.

Provisioning and preparation follows this to support all the direct copying that is done. Generally performed during an off-shift, production information is written to tapes for a multitude of purposes. Although in many cases this will be the backup/recovery process, the sophistication of application jobs will utilize several levels of backup to ensure user data integrity is maintained. Also important is the type and form of copying that is performed in moving data to and from an online media. This means that some processes copy all data from the disk, regardless of the application, while others copy data to tape using specific file information, and yet others provide sophisticated update processes to archived data only changed if the online data has changed. Some of the more sophisticated processes take periodic "snapshots" of production data and copy these to other online or offline media.

Auto-loader libraries are highly sophisticated mechanical hardware that automate much of the provisioning and preparation in administrating tape media. Auto-loader development was driven largely by the need to provide a cost-effective way to eliminate the mundane and manual job of mounting tapes. Another major value is the ability to provide a larger number of pre-stage tapes for tasks like nightly backup and archival processing. Auto-loader libraries are integrated with tape drives and controller mechanisms and provide various additional levels of functionality in terms of buffering, error recovery, compression, and catalogue structure. Enterprise-level media libraries use mega-libraries, called *silos,* to provide an almost fully automated set of tape activities.

As you would suspect, many enterprise-level tape library systems are integrated into storage area networks. These SANs enable tape library systems to translate the SCSI-based tape drive operations with the Fibre Channel–based SAN network. NAS, on the other hand, has only recently integrated tape systems into its solutions. These can become more problematic given the direct attached internal storage characteristics of the NAS device.

How these processes and resources are managed requires a good deal of planning and execution within a generally tight time schedule. In order for this to happen, the correct amount of blank media should be available, formatted, and mounted in a timely fashion, the scheduling and processing of jobs needs to proceed according to a predefined schedule, and any exceptions must be handled in a timely manner.

Optical Storage

Optical storage serves as a media where laser light technologies write information in a fashion similar to disk architectures. Although logically they share similar characteristics, such as random access of the stored data and have head disk assemblies (HDAs) to read the media, these similarities disappear with the media used and operation of the read/write assembly. Optical storage compact discs (CD-ROMs) and DVDs (digital video disks) are driven by laser technology. As we all know, CDs and DVDs are removable storage with much higher densities than traditional diskette technology. As such, they have become a very successful product distribution media, replacing tapes.

Optical has achieved notoriety as a new distribution format in the fields of software, entertainment, and education. However, only recently have they been able to function as a reusable media with digital read/write capabilities. Unfortunately, their speed and throughput have not met requirements for use within the data center for commercial backup/recovery, or other archival media.

CDs and DVD optical media have never been favored as a replacement for magnetic disks or tapes in larger enterprise data centers. However, they has been slowly gaining a niche in the support of special applications that deal with unstructured data, the distribution of data, and delivery of information. Examples are datacentric objects like videos, integrated audio/video application data, and entertainment and educational integrated video, image, audio, and text information. The basic configurations for optical media are very much like tape drives and libraries. In commercial applications, they are supported through optical libraries which include multiple drives and auto-loader slots for selecting media.

The major difference between magnetic disks and tape is the use of lasers to write information on the optical media. This requires modifying a type of plastic media through the use of the laser beam, either by bleaching it, distorting it, or creating a bubble. The data is written in tracks but uses a different geometry that employs a spiraling technique to format the tracks from the center of the disk to its outer areas. Regardless of how it forms the modification, this is the method of writing data to the optical disk. When accessed for read operations, the optical device turns at a variable rate so that access throughout the disk is close to linear.

However, even with optical attributes and increased data capacities, the optical media has yet to achieve the throughput necessary for online transactional multiuser systems, or the necessary controller sophistication for data protection and recovery operations. Consequently, optical libraries will not proliferate within storage network installations even though they are heavily used in applications that favor imaging, videos, and audio. Systems that are SCSI-based can participate with SANs through a bridge/router device (see Chapter 14) and are available as NAS devices using a NAS microkernel front end with optical drives and autoloader hardware attached.