Unix Storage Management

RAY A. KAMPA AND LYDIA V. BELL

Apress™

ISBN (pbk): 1-59059-029-5

Printed and bound in the United States of America 12345678910

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Technical Reviewer: Perry L. West
Editorial Directors: Dan Appleman, Gary Cornell, Jason Gilmore, Simon Hayes, Karen Watterson, John Zukowski
Managing Editor: Grace Wong
Copy Editor: Ami Knox
Production Manager: Kari Brooks
Compositor: Impressions Book and Journal Services, Inc.
Illustrator: Cara Brunk, Blue Mud Productions
Indexer: Ron Strauss
Cover Designer: Kurt Krames
Manufacturing Manager: Tom Debolski
Marketing Manager: Stephanie Rodriguez

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY, 10010 and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany. In the United States, phone 1-800-SPRINGER, email orders@springer-ny.com, or visit http://www.springer-ny.com. Outside the United States, fax +49 6221 345229, email orders@springer.de, or visit http://www.springer.de.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax: 510-549-5939, email info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the authors nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com in the Downloads section. You will need to answer questions pertaining to this book in order to successfully download the code.

CHAPTER 2

Storage Technologies Overview

Storage technologies have become a mixture of the old, the new, and the borrowed. Disk and tape are very old, with tape dating back to the 1940s and disk to the 1950s. Optical disk, automated libraries, and RAID are newer, 1980s technologies; NAS and SAN are very new 1990s technologies; and storage pooling is a borrowed concept from mainframe storage, as is centralized storage management.

Many of us have been amazed at how the old disk and tape technologies keep getting new life as engineers discover ways to pack more data onto these media and make the access of that data faster. Some trends, such as revisiting storage pooling used in mainframe environments and applying the concept to networked storage, were inevitable due to the ideas being darn good ones (picture a conga line of mainframers doing their happy dances).

Storage itself was a darn good idea. Early computers didn't use storage at all. Computations were made, answers were given, nothing was saved for future use except handwritten entries into mathematical tables. This severely limited the usefulness of old computers. Then someone came up with the idea to store data on some kind of medium. Enter the age of paper tape and cards. Punching holes into the paper medium represented and preserved the data for future use. Actually, the punched paper medium had been used before computers with textile machinery.

The next brilliant idea was to use magnetic media for data representation and preservation—first tape, and later disk. Then came optical storage. The data was set on the medium using lasers to change the reflective properties of very tiny areas of the medium. Several new ideas involving the use of individual molecules or light beams to represent and preserve data have been developed in laboratories, and these technologies may come to market within the next few decades (or so).

NOTE *The terms* memory *and* storage *are sometimes used interchangeably, for example, with CD-ROM (short for compact disk read-only memory) storage. We define* storage *as physically discrete auxiliary devices, connected either directly or through network interfaces, to a computer. We define* memory *as integrated circuit (IC) chips directly addressable by the central processing unit (CPU), or indirectly addressable via extended memory technologies. Where industry standards insist on swapping these terms, we go along despite having experienced too many muddled communications when the terms aren't well defined. We do concede that gray areas exist though, and encourage storage managers to ask enough questions to avoid confusion.*

Will there ever come a time when storage becomes obsolete? Perhaps. The only reason storage is a good idea is that data can't be kept in memory directly addressable by the CPU forever. Most of the data that today's computers work on is brought up from storage in fairly small chunks. But what if all that data could be kept in nonvolatile, inexpensive, very dense memory? There is a growing trend in the chip fabrication industry to make more nonvolatile memory, which we discuss later in this chapter.

Our CPUs are able to directly address more memory too. A 64-bit operating system can theoretically address up to 18.4 exabytes (EB). That's 18,447 petabytes (PB), or 18,446,744 terabytes (TB). Another way to look at this is that a 64-bit operating system, given enough RAM, would be able to work on around 18 *million* files and executables averaging 1 *terabyte* in size each and still have room in primary memory to spare. The potentials of this are overwhelming.

Binary Number Magic

You may have heard the joke about how to become a millionaire in just 21 days. The idea is that you put $1 into a savings account and double the balance of the savings account each subsequent day. On day 2, you add $1, day 3 it's $2, and $4 on day 4. By day 21, you will have $1,048,576. Of course, as the days go on, it becomes more difficult to make those deposits.

This is how binary numbers work, and how CPU addressing works. If you add one bit to the addressing field, you double the number of unique addresses. Thus a 32-bit operating system can address about 4 gigabytes (GB) of memory. The actual number is 4,294,967,296. Add one bit to the address field, thus making it 33 bits long, and now you can directly address about 8.6GB.

What would happen if someone built a 128-bit computer? The number of possible addresses would be approximately 3,400,000,000,000,000,000,000,000,000,000,000,000,000 (3.4E39)! Mathematicians would call this number 3.4 *duodecillion* addresses.

However, until the economics of computer memory cause it to become cheaper than storage systems, storage managers will have their hands full working with diverse storage technologies. We suspect the economics will take some decades to develop.

This chapter covers the disk and tape technologies that are commonly used in storage environments. We discuss the techniques used to allow greater data density on storage media, to make storage more reliable, and to automate physical mounts of removable storage media. The incorporation of storage and networks is touched upon at the chapter's end.

NOTE *Storage with networks is covered in Chapter 4, "Network-Connected Storage."*

Disk Storage

Storage is different from primary (or main) memory in that the data held in storage isn't directly addressable by the CPU. The data needs to be loaded into directly addressable memory before it can be used. Early storage devices provided this data sequentially, that is, one bit at a time in a long stream of bits. Programs and data that the programs worked on had to be loaded entirely into main memory. If a programmer wanted to access data or load a program that wasn't in the initial sequential stream of bits, that was just too bad.

In 1949 the Manchester Mark 1 prototype computer, developed at the University of Manchester in the United Kingdom, introduced the idea of direct-access storage, where data was kept magnetically on a spinning drum. Programs and data that were not in the initial sequential load could now be used. This allowed for longer, more complex program runs.

In the 1950s a team of IBM engineers came up with the idea of a direct-access storage device using a spinning disk rather than a spinning drum. Legend has it that the team was enjoying pizza at a local parlor and the mealtime discussion was about better ways to directly access storage. They looked at the aluminum pizza platter and brainstormed on how magnetic material could be spread on the platter, and how the platter could be made to spin. You'd need a motor to spin the platter. The parlor had a music juke box spinning 45 rpm (short for revolutions per minute) vinyl records, probably belting out the rock and country music of the era. Why not use a spindle motor from a jukebox to spin the pizza platter? The engineers were on a roll. Yes, and use a voice coil from a hi-fi speaker to move an arm back and forth across the platter, and at the tip of the arm, a read/write head! Write the data in concentric circles, sense the head position, lay down the data one bit after another as magnetic fields in a track. Then read back the data by repositioning the head over the track and sensing the magnetic fields! The concept of the direct access storage device (DASD) was born, and the computing world would never be the same.

NOTE *DASD is pronounced "daz-dee" by those in the know. Saying "dee aye ess dee" instead will bring snickers from blue hairs and gray beards (like us).*

DASD made relatively fast random data access possible through the technique of moving the read/write head to the desired data directly rather than reading all the data up to the desired part, as is true with sequentially accessed data. This in turn made virtual memory, multiprocessing, databases, word processors, and other advanced computer applications possible.

Magnetic Direct Access Storage Device

The first DASD used magnetism to store data. This technology, with many refinements, is still in common use today. The legend of the pizza parlor is reflected by the terms used to describe the details of DASD: The motor spinning the disk is called a *spindle*, the disk itself is sometimes referred to as a *platter*, and the thing that moves the read/write head assembly back and forth over the disk is called an *actuator*, part of which is a *voice coil*. Figure 2-1 shows the major parts of a magnetic disk drive.

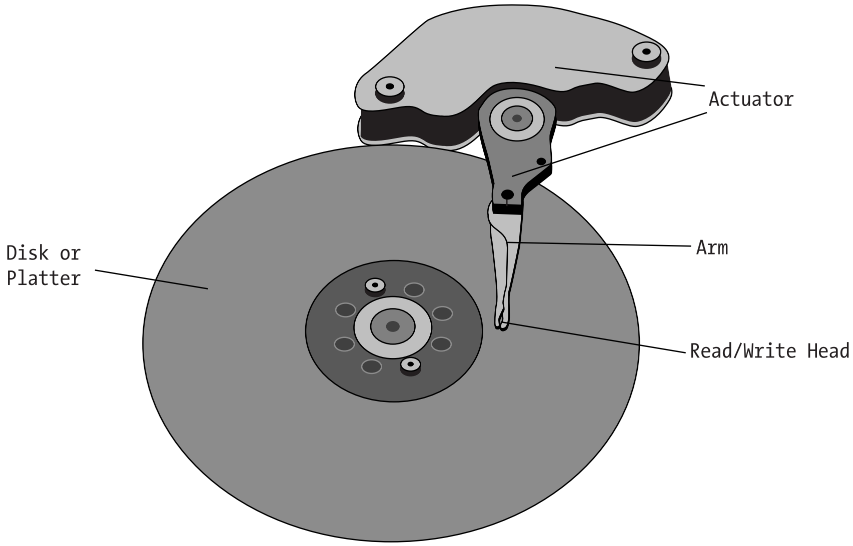

Figure 2-1. Basic DASD

The earliest commercially available DASD was the Random Access Method of Accounting and Control (RAMAC), announced by IBM in 1956. The RAMAC used a stack of 50 platters, each about 2 feet in diameter, and a single actuator arm that was poked in and out among the platters by use of rather complicated mechanics. Watching the RAMAC run reminded people of a meat slicer, and thus it was nicknamed the *baloney slicer* (see Figure 2-2).

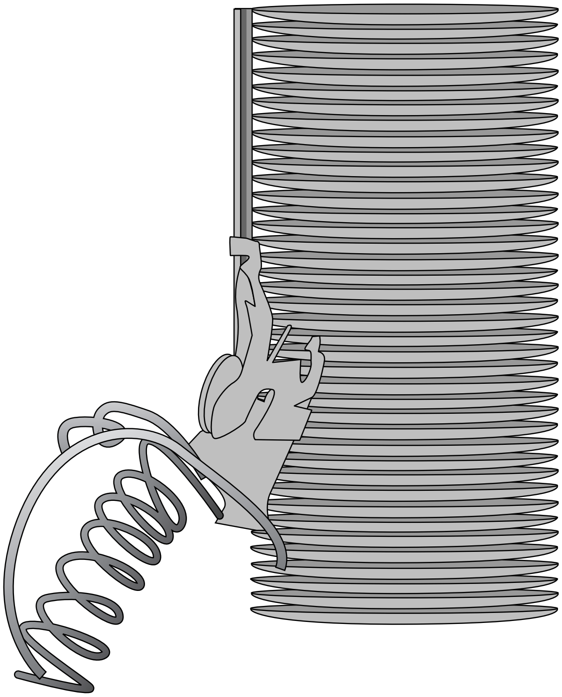

Figure 2-2. IBM RAMAC

By today's standards the RAMAC was dismally slow. It took 600 milliseconds (ms), or 1/1,000 of a second, to access data, as opposed to the current 4–10 ms average seek times. The data transfer rate was only 1 kilobits per second (Kbps). Today's common data transfer rates run from 30–200 megabytes per second (MBps).

> **NOTE** *We must point out that IBM has reincarnated the RAMAC brand name and it now represents a high capacity, fast array of disks. So don't equate the name RAMAC with slow, old DASD.*

Since the 1956 RAMAC, DASD has advanced through removable disk packs (IBM, 1963), floppy disks (IBM, 1970), and sealed Winchester disk drives (IBM, 1973). The 1973 sealed drive was called a *Winchester* because it had two spindles each with a 30MB capacity, hence a 30-30, and that reminded folks of a particular kind of popular big game hunting rifle, the .30-30 Winchester. The nickname stuck for a while to differentiate sealed drives from nonsealed. Today we just assume all magnetic disk drives are assembled in clean rooms and sealed.

Magnetic disk drives of interest to storage managers have more than one disk, with both sides of the disk being used. Figure 2-3 illustrates this concept.

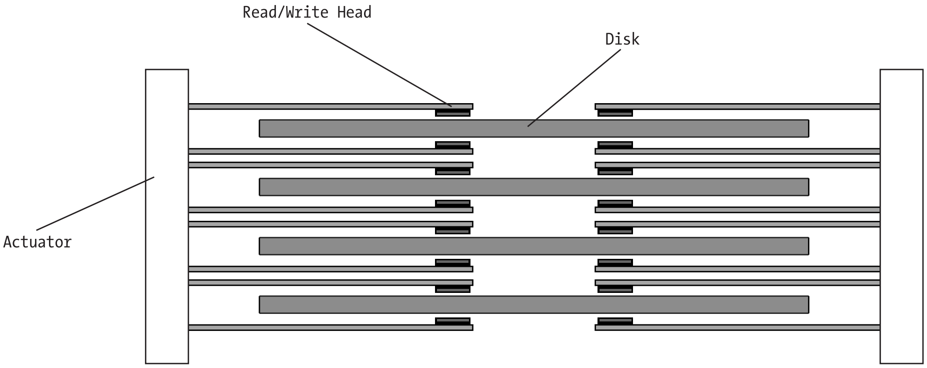

Figure 2-3. Multiple magnetic disk recording surfaces

Both sides of each disk can be read and written to by the use of two and, in this case, four heads per disk. Doubling up on actuators and heads allows for two paths to the device. This increases overall throughput by allowing two simultaneous I/O operations at a time. If one path fails, the other can still allow access to the device, thus improving fault tolerance as well.

When analyzing the relative benefits of any particular disk drive, the storage manager must be able to read and interpret the drive's data sheet and specification. A very small subset of the information is useful—that's the good news. The bad news is that it takes a little practice to decipher the information. Common data sheet/specification items are listed and explained in Table 2-1.

Table 2-1. Common Disk Data Sheet and Specification Items

DISK DATA SHEET/SPECIFICATION ITEM	DESCRIPTION
Areal density	Bits per square inch
Average latency	Average time waiting for data to spin under the read/write head
Bytes per sector	Smallest unit of capacity for the disk drive
Cylinders	Number of cylinders (stacks of tracks)
Data buffer	Size of the data buffer
Data heads	Number of read/write heads

(continued)

Table 2-1. Common Disk Data Sheet and Specification Items (continued)

DISK DATA SHEET/SPECIFICATION ITEM	DESCRIPTION
Data transfer rate buffer to host	Average rate of data transfers from and to the host
Data transfer rate to/from host	Average rate of data transfers from and to the host
Data transfer rate to/from media	Average rate of data reads and writes from and to the disk
Default buffer size	Size of the data buffer (implies variable size)
Disk media diameter	Diameter of the disks
Disks	Number of disks (or platters)
Error rate nonrecoverable real	Average number of permanent errors when trying to read data
Error rate seek	Average number of errors while seeking data
External transfer rate	Average rate of data transfers from and to the host
Form factor	Diameter of the disks (also any *size* measurement)
Formatted capacity	Maximum amount of useful data the disk drive can hold
Formatted internal transfer rate	Average rate of data reads and writes from and to the disk
Full disk seek	Average time to find the data if head travels all the way across the disk
Head recording surfaces	Number of recording surfaces
Heads	Number of read/write heads
Interface	How the disk drive connects to the host
Interface transfer rate	Average rate of data transfers from and to the host
Internal data rate	Average rate of data reads and writes from and to the disk
Internal transfer rate	Average rate of data reads and writes from and to the disk

(continued)

Table 2-1. Common Disk Data Sheet and Specification Items (continued)

DISK DATA SHEET/SPECIFICATION ITEM	DESCRIPTION
Latency	Average time waiting for data to spin under the read/write head
Logical blocks	Number of logical blocks
Media transfer rate	Average rate of data reads and writes from and to the disk
MTBF	Mean time between failures, an indication of how long the drive might last
Recording density	Bits per square inch
Recording zones	Number of recording surfaces
Rotational latency	Average time waiting for data to spin under the read/write head
Rotational speed	Disk rpm
Sector size	Smallest unit of capacity for the disk drive
Seek time average	Average time to find the data
Seek time average random	Average time to find the data
Seek time full stroke	Average time to find the data if head travels all the way across the disk
Seek time full track	Average time to find the data while skipping a whole track
Seek time maximum	Average time to find the data if head travels all the way across the disk
Seek time track to track	Average time to find the data if tracks are adjacent
Spindle speed	Disk rpm
Sustained data rate	Average rate of data transfers from and to the host
Sustained throughput	Average rate of data transfers from and to the host
Track density	Number of tracks per inch

We have left out less meaningful items such as ambient temperature range, humidity, acoustics, shock, and power because competing disk drives tend to have similar characteristics on these levels. Notice that there are often two or more ways to express the same thing, and it is more often than not that competing vendors express things differently. This can lead to annoying arguments about the ultra fine details of measuring a disk drive's performance. Don't let this marketing trick throw you off balance. Storage managers are primarily concerned with comparing the cost/performance/capacity features of disk drives at higher levels than engineers. Simply make sure there aren't huge differences in performance, capacity, and mean time between failures (MTBF) numbers. Then recommend buying the cheapest drives that meet your criteria.

TIP *Average rate of data transfers from and to the host is the best measure of performance; formatted data size is the best measure of capacity; nonrecoverable error rate is the best measure of the media reliability; MTBF is no guarantee about how long any particular drive will last.*

Permit us to add a little more on MTBF. A *mean* is a number exactly between two extremes. Therefore, if a drive has an MTBF of 1,000,000 hours, there must be an extreme low and an extreme high number too. The extreme low number may only be 1,000 hours, which would make the extreme high number 2,000,000 hours. You just don't know where any particular drive lands in that range, and woe to you should a disk array come into your shop with many low-number drives installed. Since this situation can't be foretold, the only advice we can give is to not become overly focused on MTBF claims. Sometimes disks break earlier than expected, and that's why fault-tolerant RAID and backups are so important.

NOTE *RAID is discussed later in this chapter.*

We seldom pay much attention to internal data rates just as most people don't pay attention to tachometers in automobiles. The speedometer is what counts in an automobile because it tells you how fast you and the vehicle are traveling. Most people don't care how fast the engine is spinning to make the car move. The data rate between a disk and its host is a measure of how fast data is traveling to and from the application running on the host. Most of your customers care about that, not how fast the data moves to and from the disk and its buffer.

Form factors, dimensions, interfaces, and power are only important in the context of whether the disk drive will fit into the host cabinet. Just make sure it fits, that's all.

NOTE *Direct attached storage (DAS) is covered in Chapter 3, "Direct Attached Storage (DAS)," and the discussion includes host connection technologies.*

Optical Disk

The differences between optical and magnetic disk technologies set them worlds apart. The most obvious difference is that optical disks use lasers instead of read/write heads. Another difference that isn't so obvious, until seek times are considered, is that optical disks are written in one long, continuous spiral, whereas magnetic disks are written in concentric circles (see Figure 2-4). Since the element of direct data access is lost, optical disk is considered a sequential access storage technology. As a result of poor performance in comparison to magnetic disks, optical disks have come to be treated similarly to tape—good for backup, archival, and near-line storage, as well as software distribution, but a poor substitute for magnetic disks.

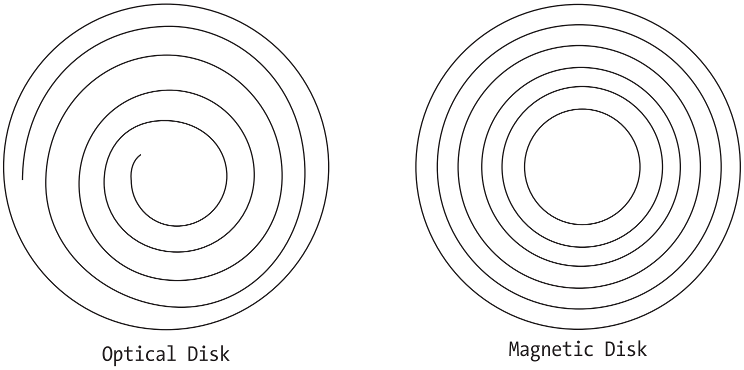

Figure 2-4. Optical versus magnetic writing techniques

NOTE *Near-line storage is discussed in more detail later in this chapter, in the section, "Automated Tape Libraries."*

There are three ways to write data to an optical disk: Burn a hole in the substrate, raise a dimple in the substrate, or crystallize the substrate. All three methods change the reflective characteristics of the disk substrate so that the read laser can tell an *on* bit from an *off* bit. Burning holes in aluminum substrate was the first attempt at recording on an optical disk, but this proved to be a bad way of doing it. Seems that atoms of aluminum, a basic metallic element, drift around unless cooled to absolute zero. The aluminum atoms making up the edges of the burned holes liked to drift over to the neighboring holes, and so data error rates were too high. Dimples and crystallization became the standard ways of writing data to optical disk, along with melting areas in polymer dyes (similar to burning holes).

The trouble with dimples is that you can't erase them. Raising them is easy using the heat of the laser. Pushing them back down is problematic, and so the dimple writing technique is only good for write once, read many (WORM) optical disks. The same is true for the melting and hole-burning techniques.

Crystallization of the substrate can be reversed though, using laser light, and this constitutes compact disk, rewritable (CD-RW) technology.

The promise of optical disk was a compelling reason for manufacturers like StorageTek and Memorex to shovel mountains of money into its development in the 1980s. The two primary attractions were higher data densities per square inch and the ability to handle optical disks like records in a jukebox. The common term for an optical library that automatically mounts and dismounts disks is still a *jukebox*. The story for StorageTek has a sad ending—the optical disk project failed. However, the robotics technology developed for optical was later used in the very successful line of StorageTek automated tape libraries. Memorex was more successful bringing optical disk to market. The primary uses of the Memorex disks were to replace microfiche archiving systems and to hold vast amounts of data such as seismic readings for the oil and gas industries. Memorex has since opted out of the large storage marketplace.

By the mid-1990s, optical disks had become consumer items on PCs. Today there are few manufacturers of industrial strength optical disk jukeboxes due to improvements in tape speeds and densities. Tape has overshadowed optical disk. Optical disk technology may be fading from the storage manager's radar screen, but it is enjoying high interest in the consumer video markets. High-density, multilayered digital versatile disks (DVDs), which can hold up to 17GB per disk, are common items in video stores these days. With this attractively high capacity per disk, and with DVD read/write technology coming to market, the storage manager may see the return of industrial strength optical jukeboxes on the data center floor.

A hybrid of magnetic and optical technologies is the magneto-optical (MO) disk. An MO disk is written using magnetism and read using laser light. This is possible because a magnetically written medium records data by assigning the north and south polarity of the magnetic fluxes as 0 and 1, or 1 and 0. Either way works. It just so happens that the light-reflecting angle of a north polarity flux varies just a little bit from a south polarity flux. The read laser is sensitive enough to differentiate between a north flux and a south flux reflection. MO disks are better performers than straight optical disks because the element of random access (data in concentric circles) is reintroduced. Industrial-strength MO jukeboxes are available on the market now.

RAID

RAID once was an acronym that was derived from *redundant array of inexpensive disks*. As it turned out, RAID was not less expensive on a cost per megabyte basis due to the need for multiple controllers, advanced microcode, large caches, and other supporting hardware.

The acronym's translation became *redundant array of independent disks*, as asserted by the RAID Advisory Board (RAB). This translation doesn't seem to fit either because RAID disks, in terms of fault tolerance, are dependent upon one another as you will see shortly. A translation we've heard that seems to fit better is *redundant array of intelligent disks*. We like this because RAID implementations require intelligent controllers and many have impressive algorithms that approach a complexity near to artificial intelligence. Some RAID levels are implemented in operating system software too, most notably RAID-1 and RAID-5.

Although several storage manufacturers were actively developing RAID before, credit for RAID's start is often given to a paper published at the 1988 ACM (short for Association for Computing Machinery) SIGMOD (or Special Interest Group on Management of Data) Conference. The paper's title was "The Case for Redundant Arrays of Inexpensive Disks" by Patterson, Gibson, and Katz, University of California, Berkeley. (We will refer to this paper as the Berkeley paper from here on.) The idea was to take several small, relatively inexpensive disk drives commonly used in the PCs of the day, and combine them to do three things:

- *Make a number of small drives look like one big drive, or Single Large Expensive Disk (SLED), a term used by the paper's authors:* In 1988 the common form factor for mainframe SLEDs was a 14-inch diameter platter. A single mainframe disk drive took up multiple square feet of floor space, had a maximum capacity of around 800MB, and stood about 4 feet high. The authors contrasted this with a typical 100MB PC drive with a 3.5-inch disk form factor. The cost per megabyte for the PC drive was 30 to 40 percent lower than that of a SLED.

- *Increase the speed of storage:* The paper's authors identified the following trends in the computer industry: CPU speed was increasing rapidly, primary memory density was increasing rapidly—but disk I/O rate was increasing at a snail's pace by comparison. Something needed to be done to avoid a crisis (wasted CPU and memory power) in the computing industry, so the Berkeley paper declared. The solution was this: Instead of a SLED serving only one I/O at a time, some of the RAID levels described in the paper would allow multiple I/Os to be serviced at once, which is also called *parallel I/O*. The authors acknowledged that this technique would need some super charging, and so adding cache to the proposed arrays was suggested.

NOTE *Cache improves disk performance and is discussed later in this chapter.*

- *Increase the fault tolerance of storage:* The paper's authors described five levels of raid: RAID-1, RAID-2, RAID-3, RAID-4, and RAID-5. All of these levels are fault tolerant using mirroring, error detection and correction, or parity. (Fault-tolerance schemes are explained in detail later.) Initially, the fault-tolerance schemes were introduced to raise the MTBF of an array because without the fault tolerance, MTBF reduces dramatically when multiple small disks are used to make one big disk image. The authors pointed out that if you use 100 small disks with an MTBF of 30,000 hours each, the MTBF of the array becomes 30,000/100 = 300 hours. As RAID became more accepted, fault tolerance became a big selling point and lost its stigma as a necessary added level of complexity. For seasoned mainframe and Unix storage managers, the ability to lose part of a disk image without losing the whole disk image, or any of the data in that image, while a bad array disk was replaced was a major breakthrough in system reliability. Prior to RAID-2, RAID-3, RAID-4, and RAID-5, the only way to protect data against a disk crash was to mirror (RAID-1).

CAUTION *RAID does not necessarily mean that the array has* hot-swappable *disks. A hot-swappable disk is one that can be removed from the RAID array and replaced without taking the server down. The array hardware must support hot-swappable disks—otherwise, you must take the server down before replacing the bad array disk. If this is not done, data can be lost.*

The RAID ideas were welcomed into the industry with a great deal of excitement over the potential cost savings for storage. The excitement faded for a while as early products appeared with price tags that dashed any hope of saving money. At the same time, the price for SLED storage was dropping rapidly, and that further delayed RAID acceptance until 1994 when EMC announced the Symmetrix 5500 array. RAID then took the storage market by storm, and the use of SLEDs became history. The Symmetrix 5500 could hold an entire terabyte of data in the amount of floor space that one SLED took up (about 6 square feet), had faster I/O due to heavy caching, and enabled remote mirroring.

NOTE *Remote mirroring is discussed later in this chapter and also in Chapter 4, "Network-Connected Storage."*

The finer distinctions among RAID levels have been points of confusion over the years, due in part to vendor marketing trying to offer better RAID than competitors. We have revisited the original paper, available from the ACM Web site (`http://www.acm.org`), and when other information sources were not clear, we went with the original definitions while composing the following summary.

RAID-0 features *striping,* in which data is spread across multiple disk drives on a block basis. Each stripe resides on a separate drive (see Figure 2-5). This RAID level was not described in the paper; however, RAID levels 2 through 5 do use striping along with fault tolerance schemes. Although striping alone gives no fault tolerance because losing any one disk in the array results in lost data, it is useful for increasing throughput. Multiple data transfers can be initiated at once, using multiple controllers and data paths. RAID-0 is *fast on reads and writes but not fault tolerant.*

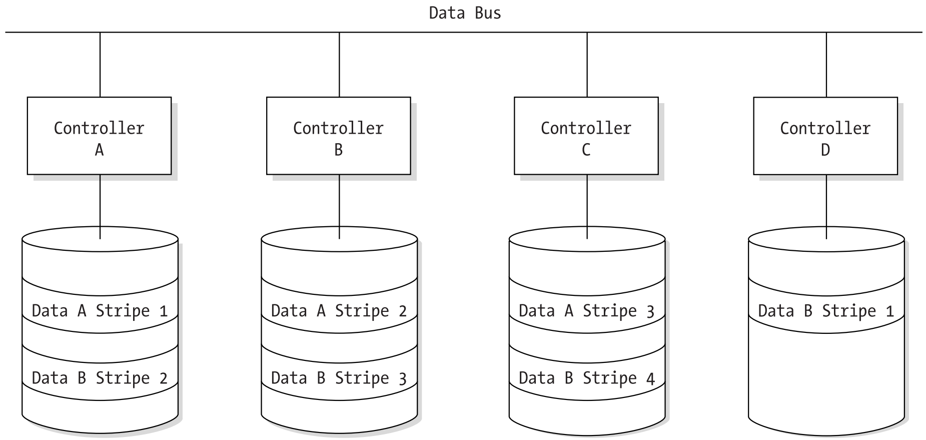

Figure 2-5. RAID-0, striping

In RAID-1, *mirroring* occurs, which is when one or more disk drives are active copies of another. If one drive is lost, the other(s) retain the data (see Figure 2-6). The probability of both drives in a copy pair failing at once is low. The probability falls even further if two drives are copies of another. If each disk in a mirror has its own controller, known as *disk duplexing*, read/write performance can be better than a single disk.

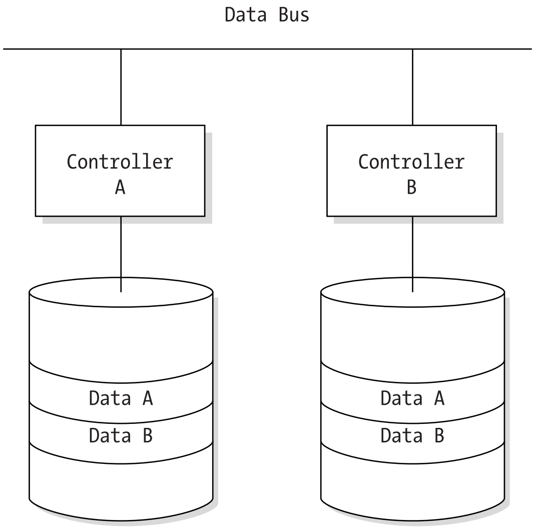

Figure 2-6. RAID-1, mirroring

NOTE *Disk duplexing not only improves performance through the splitting of I/O operations, it adds more redundancy to the RAID-1 array. If one controller fails, the other keeps on working. However, the server will need to be brought down to replace the failed controller.*

If all disks in a single mirror operate off the same controller, write performance degrades (the write action must be performed repeatedly), while read performance remains the same. As you would expect, doubling or tripling hardware for RAID-1 makes the implementation expensive. RAID-1 is *very safe but expensive.*

RAID-2 is the *error-correction code (ECC)* level, in which stored data includes extra bits that describe the data, and if an error occurs while reading the data, the data can be rebuilt on the fly with the ECC information (see Figure 2-7). The original RAID-2 definition described the ECC information as being stored on separate disks. The scheme was so expensive that no vendor actually implemented RAID-2 as described in the Berkeley paper; however, most disk drives now have ECC built into them anyway. On today's disks, the ECC bits are laid down along with the data on the same disk. ECC differs from parity in that parity doesn't allow rebuilding data on the fly *after a read error,* but in other RAID levels, parity does allow rebuilding *lost data before* a read. RAID-2 also involves striping, but since most drives have ECC now, RAID-2 *has become the same as RAID-0.*

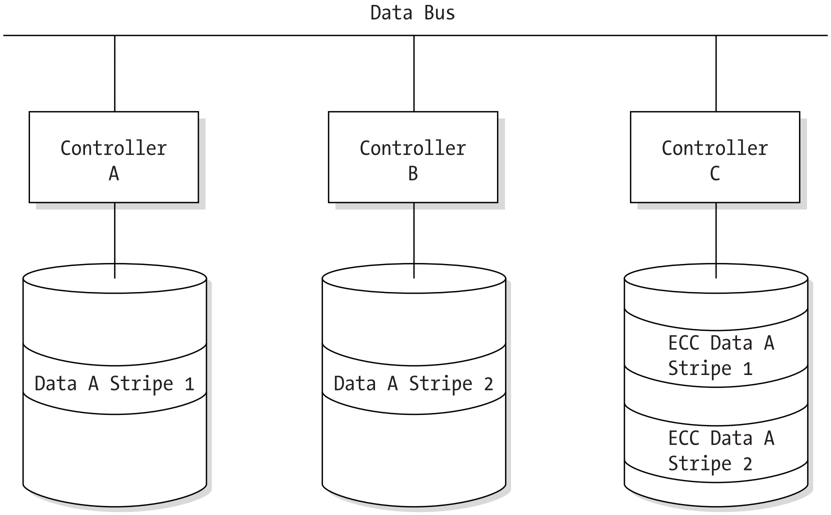

Figure 2-7. RAID-2, error-correction code and striping

RAID-3 features *striping with dedicated parity drive*, where *data is striped on a bit level* (both big and small data is striped) and one drive of the array is dedicated to parity (see Figure 2-8). The Berkeley paper limits RAID-3 to one I/O at a time, but this is not necessarily the case (see the note that follows). Parity is data calculated off the original data, which can be used to re-create the original data should a data drive fail and be replaced. The parity calculation results in slower write times. Read times are not affected because parity is kept on a separate drive, therefore the read process can go straight to the data. The minimum number of drives is three, two data drives and one parity drive. No data is lost if either or both data drives (or however many are in the array set) fail, or if only the parity drive fails. If the data drives are lost, the data can be re-created off the parity drive. If only the parity drive is lost, parity can be recalculated off the data. If one data drive and the parity drive fail, data is lost. RAID-3 is *safe but slow on writes.* However, please read the following disclaimer.

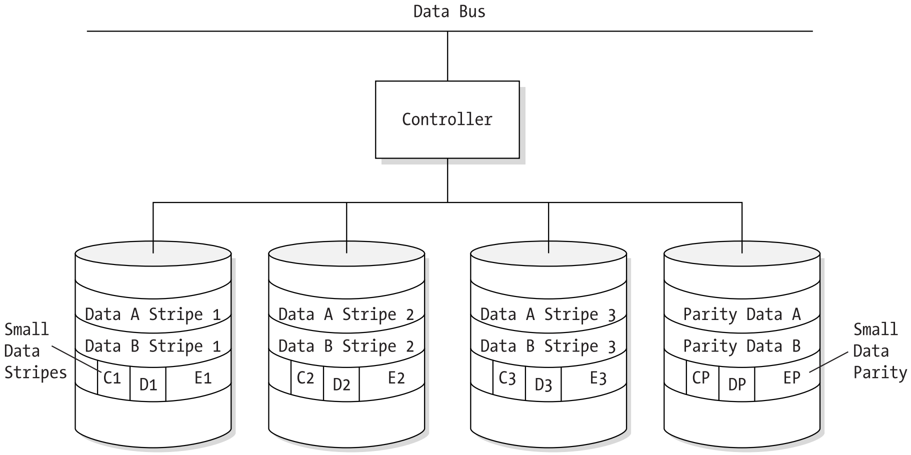

Figure 2-8. RAID-3, striping with dedicated parity drive

NOTE *Most current RAID-3 implementations have increased performance by incorporating parallel I/O and other performance enhancements.*

RAID-4 features *striping with dedicated parity drive*, where *data is striped on a block level* (large data still striped across multiple drives but small data isn't striped) across synchronized drives, and one drive of the array is dedicated to parity (see Figure 2-9). Block striping and synchronization of the drives results in the ability to do parallel I/O across the data drives. RAID-3 allows only one I/O to be performed at a time, so RAID-4 has better performance, at least by the Berkeley paper's definition (see the note that follows). Parity calculations benefit from the parallel I/O also. RAID-4 is *safe and performs better than RAID-3, but not as well as RAID-1*. As with RAID-3, we present the following caveat.

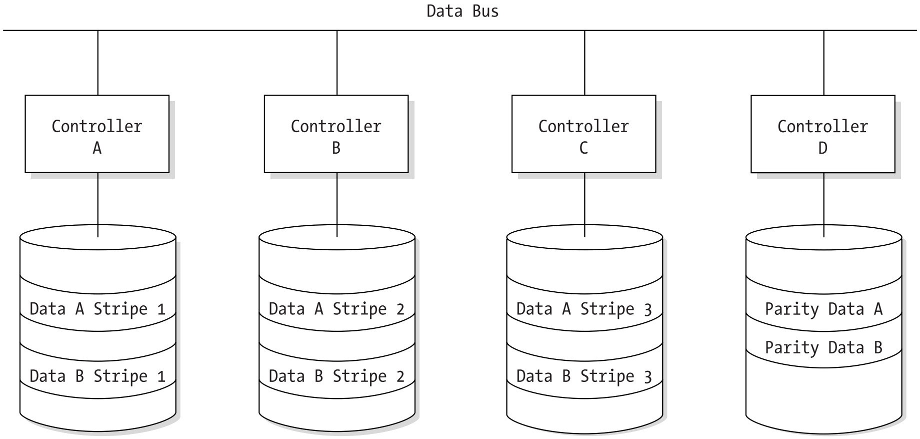

Figure 2-9. RAID-4, striping with dedicated parity drive, parallel I/O

NOTE *RAID-4 is seldom, if ever, seen in the marketplace. RAID-3 is more common, and through the use of performance enhancements would perform as well or better than RAID-4—if there were any RAID-4 arrays, that is.*

In RAID-5, the *striping both data and parity* level, the parity is striped across all drives in the array along with data (see Figure 2-10). This has an advantage over RAID-3 and RAID-4 in that the performance hit on writes is smaller because more disks are available for recording the parity, allowing the parity writes to be made in parallel. The trade-off is that reads take a performance hit due to the read process coming across parity while seeking data, and the parity must be passed over. RAID-5 is *safe, faster than RAID-3 and -4 on writes, but slower on reads.*

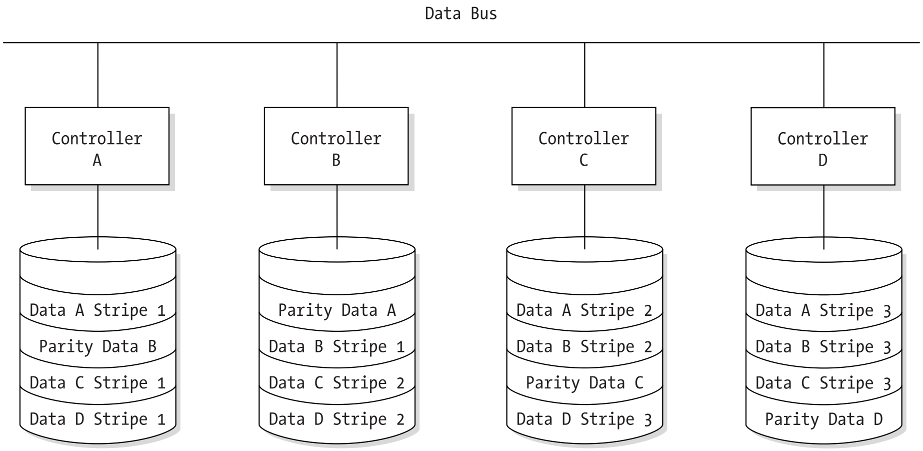

Figure 2-10. RAID-5, striping both data and parity, parallel I/O

The original RAID paper attempted to project what applications would perform best with the various levels, and what levels give more capacity. RAID-1 is the worst for capacity because disk drives are doubled or tripled. RAID-3, RAID-4, and RAID-5 are the best because they use several data drives to one parity drive, or stripe the data and parity across all drives.

We summarize the performance thinking of the Berkeley paper in the next paragraph for those interested in historical trivia. Other readers can skip down to the paragraph after that and not miss a thing.

RAID-2, RAID-3, and RAID-4 were ranked low for performance on small I/O due to ECC and parity calculation overhead, and the ECC/parity drive bottleneck. RAID-5 was rated better on the strength of eliminating the dedicated ECC or parity drive bottleneck. RAID-1 was rated the best for small I/O because no ECC or parity needed to be calculated and written. On large I/O, RAID-3, RAID-4, and RAID-5 were given the highest ratings.

All this performance thinking became moot with the addition of large cache and intelligent controllers to the commercially available RAID arrays. Figure 2-11 illustrates a cached RAID array.

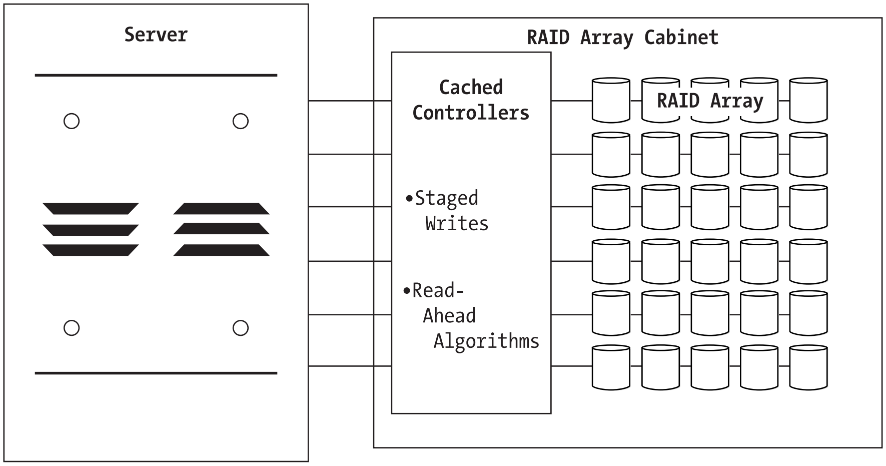

Figure 2-11. Cached RAID array

Cache is a set of memory chips on the disk controller that stages reads and writes. It is an old trick to reduce the mechanical and electrical delays associated with spinning disks, but a useful one. RAID arrays achieve high I/O rates using cache, although there is a potential danger. If the cache loses power, the staged writes disappear unless battery power is supplied to the cache memory. Memory of this sort is called nonvolatile storage (NVS) or nonvolatile random access memory (NVRAM).

Nonvolatile, No Battery RAM

One of the holy grails of computing is the search for NVRAM that doesn't need batteries. Intel introduced flash memory in 1988 as a commercially viable way of satisfying this yearning. Although flash memory has proven useful in various computer components, digital photography storage, and small mobile communication devices, it isn't quite right for cache NVRAM.

Another type of NVRAM is being developed jointly by IBM and Infineon Corporation. Magnetic random access memory (MRAM) uses magnetism to store bits rather than electricity. This development started at the end of the year 2000, and the first commercially available chips are scheduled to appear in 2004. Because MRAM uses magnetism to store bits rather than electricity, it thus achieves nonvolatility. It also promises to be less expensive to manufacture than flash memory and to provide the kind of random access that cache needs.

Should these promises become reality, the implications to storage management are significant. Not only would cache benefit, but we may see solid-state disk (SSD, explained later in this chapter) gaining a more significant place in storage management.

Caching can also bring added intelligence to disk controllers. Read-ahead algorithms anticipate what data will be needed next and bring the data up off the disk ahead of demand, into the cache, all set to go.

As mentioned previously, the original RAID paper described only five levels (1 through 5). RAID-0 was tacked on because vendors had striped disks before 1988, and so they hopped on the RAID bandwagon with a new name for old technology. It was cool to have RAID. RAID-1 was also available before the paper, referred to as *dual copy* or simply *mirrored disk*. Having RAID-*big number* became cool too, even though there's no hierarchical relationship where bigger is necessarily better. Inevitably the waters were clouded by the introduction of RAID-6, RAID-7, RAID-10, RAID 1+0, RAID-53, RAID-S, and Software Tape RAID.

RAID-6 is RAID-5 on steroids; in this level, the parity scheme is doubled. It is overkill and not commercially available. Some vendors have tried to sell logical disks (splitting big disks into logical small disks, or combining small disks into logical big disks) as RAID-6+. Ah well, whatever. It is still RAID-5.

RAID-7 is RAID-5 distributed, where an embedded computer, caching, high-speed bus, and other components that make up a stand-alone computer are included. The notion of offloading I/O processing away from the CPU is fairly old, first done on mainframe I/O subsystem channels. (One of your authors remembers an attempt to put a small mainframe between the host mainframe and DASD [or SLED] devices, back in the early 1980s, with the virtual storage subsystem [VSS]—ah, memories!) RAID-7 provides the premium expression of this idea with its stand-alone computing power. The term *RAID 7* is actually a registered trademark of the Storage Computer Corporation (`http://www.storage.com`).

RAID-10 represents RAID-1 and RAID-0 combined, where mirroring gives high fault tolerance and striping yields high performance.

RAID-1+0 is "lazy" RAID-10, where if any one disk in the array is lost, all mirroring is lost too. The entire array becomes RAID-0 until the bad disk is replaced.

RAID-53 is RAID-0, but with each data stripe being held across a RAID-3 array of disks, thus giving the fault tolerance of RAID-3 to RAID-0. Why call this level 53 instead of 03? We don't know, it probably sounded cooler to call it 53.

RAID-S is RAID-3 with faster writes due to XOR gating technology in the disk controllers. This is an EMC Corporation proprietary implementation.

Software Tape RAID is RAID-1 for tape, or basically an auto-copy facility provided in some automated tape libraries, where two tapes held in separate drives are written to with the same data simultaneously. We prefer the term *auto copy* since the RAID acronym does happen to mention disks.

NOTE *The most common marketplace RAID levels are RAID-0, RAID-1, RAID-3, and RAID-5.*

Eventually the confusing noise surrounding RAID levels caused the RAB organization to create the Disk System and Array Controller Classification Program. The focus of the program is fault tolerance since performance has become more of an issue of caching and controller intelligence. Who cares about the finer performance differences of RAID levels if the competing products are all heavily cached and highly intelligent? The classification program also includes other fault tolerance issues such as redundant power supplies.

Storage managers are always putting up with hocus-pocus dog-and-pony shows, and the introduction of RAID into the markets seemed to have brought quite a circus to town. We have worked hard and long to clear away the clouds of confusion surrounding RAID levels and hope this helps you to focus more on what is really important: the performance and reliability of storage systems.

Remote Mirroring

One of the more intriguing developments that paralleled RAID was remote mirroring of disk drives. The fundamental idea is that you have a disk array in some city, say Chicago, and another somewhere else, say Los Angeles. Now, if you get a high-speed dedicated telephony line between the two (DS3 and above), and establish communication between the two arrays through their controllers as a RAID-1 mirror, the data written in Chicago is also written to the array in Los Angeles.

But what good is that? Well, if you are wanting immediate system disaster recovery (DR), there you have it. Of course, all the other computer hardware and software running in Chicago has to be running in Los Angeles as well. The overall system in Los Angeles has to be smart enough to take over the Chicago workload. The same is true for the Chicago system. It too must be able to take over the workload should the Los Angeles system experience a disaster failure.

Remote mirroring is the deluxe, meaning the most expensive, form of DR and should be used only for systems that absolutely cannot tolerate any downtime; however, it is a sweet solution for very large, complex data centers that are otherwise nearly impossible to protect from disaster. Figure 2-12 gives a graphical representation of remote mirroring for disaster recovery.

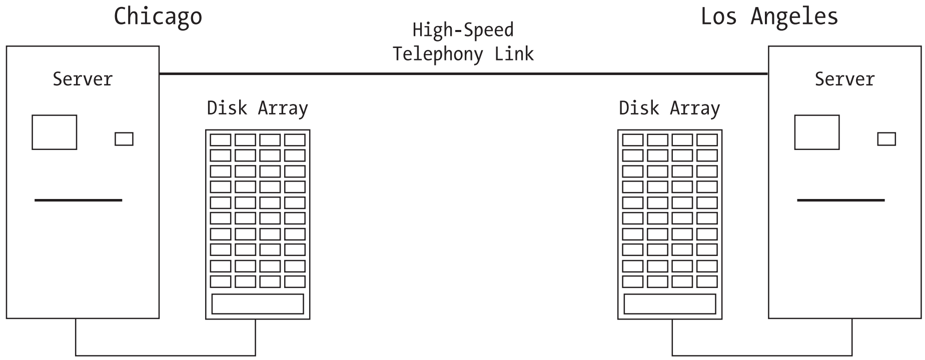

Figure 2-12. Remote mirroring disaster recovery

NOTE *Disaster recovery is covered in more detail in Chapter 6, "Disaster Recovery."*

If disaster strikes Chicago, Los Angeles can take over the workload. If disaster strikes Los Angeles, Chicago can take over the workload. The telephony link is dedicated.

There's another use for remote mirroring, and that is moving large data centers. If ever you have had to move a large data center the old way, you know what a major effort and risk this is. All the data has to be frozen and rolled to tape. The machines need to be shut down and loaded onto trucks, then shipped cross-country to the new location. Then everything has to be hooked back up the way it was, and hopefully the systems will power up without problems. This hardly ever happens. Something got jiggled too much, something fell off the truck, something went by strong magnets. Some darn thing always seems to happen. Sometimes the data itself comes up useless and restores from tape need to be done. What? You mean the tapes were lost? Oh boy. Meanwhile, the organization is in DR mode. The systems are not available and the business is dying, and so is your career. Better find those tapes quickly.

A safer—albeit very expensive—way to do this is through remote mirroring (see Figure 2-13). This involves installing new equipment at the target data center. The new equipment will support all the applications of the old center, plus remote mirroring of the disk arrays. Once the mirrors have been established and the target site's systems have all stabilized, the move is accomplished by simply stopping any active databases at both data centers for a small amount of time while the mirrors synchronize, then break the mirrors. Start the databases at the target data center and there you are, all moved. You can now do whatever is necessary with the equipment at the old data center while revenue happily flows into the organization, uninterrupted but for a very brief time.

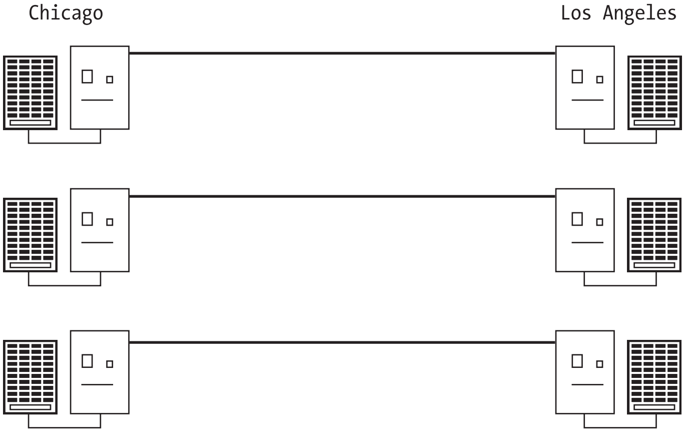

Figure 2-13. Moving a data center using remote mirroring

Figure 2-13 depicts a data center in Chicago being moved to a new data center in Los Angeles. Los Angeles has all-new equipment. The high-speed telephony lines are dedicated. Once all the systems have stabilized, databases stopped, and data synchronization completed, the move is finished by disconnecting the telephony links and bringing up the databases in Los Angeles.

Data synchronization means that even though the databases have been requested to stop, some I/O may still be in transit between the mirrored sites. These I/O operations must be completed in order for the data to be exactly the same (synchronized) at each site. Breaking a mirror in this situation means disconnecting the telephony link. If the mirrors are broken by disconnecting the telephony links, and this is done while the unfinished I/O is still being processed, database transactions could be lost. Although it is true that any database worth its salt can recover from lost transactions, it is better to avoid this situation whenever possible because managers and DBAs will have more work imposed upon them. They will also probably know who brought this avoidable work to them, and that is never good for your reputation.

We don't want to oversimplify the processes involved in establishing and breaking remote mirrors. You will need to take into consideration the distance between the data centers and the speed of the available link. Are you going to mirror in a synchronous mode (I/O isn't complete until both disks in a mirrored pair finish the I/O)? If so, the link speed must be fast enough not to slow down processing to an intolerable degree. Along the same line of thinking, if enough miles separate data centers, there may not be a fast enough link available to support synchronous mirrors. In this situation, you may want to consider suspending database processing while the mirrors synchronize.

There are also situations where you might be able to use asynchronous mirroring (I/O completed locally is treated as complete even when the remote mirror hasn't finished its I/O) or a third mirror at the local site that can be broken off and then synchronized with the remote site. The nature of the applications running on the servers to be moved will determine what mirroring techniques will work best in your particular case.

Telephony line speeds have increased to the point where remote mirroring is a more reasonable option for DR and data center moves (see the sidebar "Telephony Line Speeds"). Remote mirroring technology will likely become more visible as organizations seek the deluxe DR level and the low-risk data center move.

Telephony Line Speeds

DS0 = 64 Kbps
ISDN (basic) = 128 Kbps
ISDN (primary) = 1.544 Mbps
DS1 (T1) = 1.544 Mbps
DS2 = 6.312 Mbps
DS3 (T3) = 44.736 Mbps
OC 1 = 51.840 Mbps
OC 3 (STM-1) = 155.52 Mbps
OC 9 = 466.56 Mbps
OC 12 (STM-4) = 622.08 Mbps
OC 18 = 933.12 Mbps
OC 24 (STM-8) = 1.244 Gbps (gigabits per second)
OC 36 (STM-12) = 1.866 Gbps
OC 48 (STM-16) = 2.488 Gbps
OC 96 = 4.900 Gbps
OC 192 = 9.950 Gbps
OC 256 = 13.271 Gbps
OC 768 = 39.813 Gbps
OC 3072 = 159.252 Gbps
DWDM = 10.900 Tbps (terabits per second)

In case you're wondering what the preceding line types are, here's a short description:

- *DS:* Digital Signal

- *DWDM:* Dense Wave Division Multiplexing

- *ISDN:* Integrated Services Digital Network

- *OC:* Optical Carrier
- *STM:* Synchronous Transport Module
- *T:* Short for TDM, Time-Division Multiplexing

Solid-State Disk

SSD is one of those technologies that makes you slap your forehead and say, "Sure! Why don't we always do that?" SSD is a bunch of RAM chips in a box with a controller that makes the memory look like some particular form of disk. Data is written to and read from the SSD memory. There is no spinning disk to cause seek and latency delays. No physical actuator has to move an arm about. Don't look for any magnetic fluxes to read and write. Because of this, there is no read/write head to crash. SSD only has beautifully fast electrons, lounging about in their native silicon transistors, ready to deliver their data straight down the wire with no physical delay other than electronic resistance. Figure 2-14 shows the components that make up SSD.

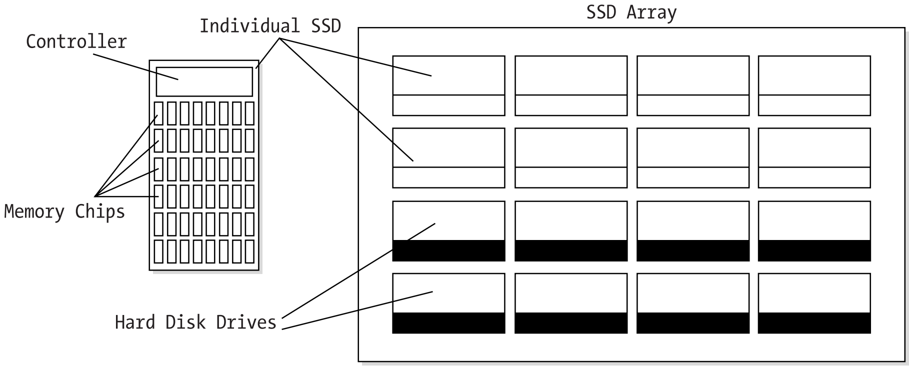

Figure 2-14. Solid-state disk components

Each SSD is made up of enough memory chips to yield the desired capacity and a controller to emulate the desired disk type. The individual SSDs can be put together into a larger array, and RAID fault tolerance can be incorporated too. Hard disk drives are used on some SSD types to increase the reliability of the array.

SSDs have been on the market for decades. They were first used in virtual memory environments where fast paging devices were needed to boost mainframe system performance. Later, with the advent of relational databases with indexes that get hammered by intensive I/O, SSD found a market niche there.

The reason SSD isn't used everywhere is that it costs magnitudes of order more than spinning disk drives. Now the reason for this cost difference, as is often the case with new technologies, is twofold: The components are expensive and the market demand is low. SSD isn't new technology, but the components are still relatively expensive and the demand hasn't grown.

SSD usefulness as a paging device went out the window when extended memory came to the mainframe CPU. Extended memory isn't directly addressable by the CPU, but it can be used for paging. It can also be used for buffering database indexes and the transactions themselves, so the demand for SSD suffered.

One of the big drawbacks of SSD is that once the power goes away, so does the data. Traditionally, batteries (bulky, heavy ones) have been included in SSD to keep the data if the primary power drops, or a power cable gets kicked out, or someone's kids visiting the data center decide to flip the red switches (not kidding, it really happened). Some current SSDs use a regular disk drive to back up the data, which is a good idea. This reminds us of a cached disk array, but with dedicated RAM for each drive and up-front disk emulation.

Flash memory does not lose data if the power goes away. This is called NVS, but it doesn't fit into the kind of random access technology that SSD requires. NVRAM chips that aren't dependent on batteries and offer the kind of random access that SSD needs aren't available yet, but may be within the next few years (see the sidebar "Nonvolatile, No Battery RAM" earlier). If the cost per MB of NVRAM decreases below that of disk, SSD will have a brand new place under the sun. Remember the case the Berkeley paper made about CPU and memory speeds increasing fast, but not I/O? If SSD becomes the replacement for magnetic disk, I/O speed will not be such a big issue.

Thinking along these lines, the potential of transferring the I/O bottleneck away from the disk to the DAS interface—or the network, if this is network-connected storage—is strong. We discuss DAS interfaces more in Chapter 3, "Direct-Attached Storage (DAS)" and networks more in Chapter 4, "Network-Connected Storage." You will find from these discussions that the I/O bottleneck does indeed move among disks, interfaces, the network, and even the CPUs.

NOTE *I/O writes can also be cached in the server's memory for some Unix systems. The writes are staged to be performed to the actual disk device at a later time, but the applications keep on running as if the writes had been actually performed. The penalty in using this technique to achieve higher I/O performance comes in the form of used processor memory and CPU cycles; however, if storage is attached with a slow technology, this technique is better than using SSD on the slow attachment.*

Tape Storage

Magnetic tape has been around since the 1940s. Until the 1980s, tape was wound around open reels and mounted by hand. The process was terribly slow until a little bit of automation was introduced using vacuum columns to thread the tape from one reel to the other. Then in 1986, IBM released the first tape cartridge, the 3480, shown in Figure 2-15. The cartridge held a single reel of tape and was about the size of a thin paperback novel.

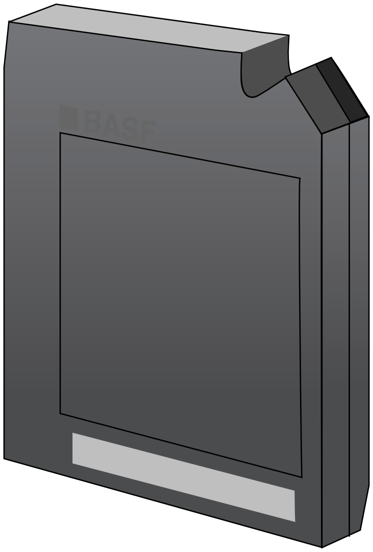

Figure 2-15. 3480 tape cartridge

The tape cartridge lent itself well to handling in automated tape libraries since the rectangular plastic housing of the cartridge could be easily picked up by a robotic two-fingered hand. Bar code labels fit easily on the end of the cartridges too, allowing the robotic eye to scan information about the tape volume. Robotics and the low cost of tape storage have given the technology a longer life than some of us have ever expected. Higher densities and more reliable tape technologies are constantly being developed and advanced, leading us to believe tape will be around for a long time to come.

Tape Writing and Handling Standards

The tape storage industry has developed several “standards” of writing and handling tape. All of these methods strive to accomplish three things: 1) increase the amount of data that can be written to a single tape, 2) increase the speed that the data can be read from and written to the tape, and 3) increase the reliability of tape storage.

Increasing the amount of data that a single tape can hold can be done by squeezing more bits per inch onto the tape or extending the tape's length. The use of multiple tracks on a tape serves to extend its length. For example, if you have a tape that is 100 meters long, adding 100 tracks to the tape makes it seem to be 10,000 meters long. Another way of adding tape length is to use thinner plastic for the tape itself.

Increasing the speed that data can be read from and written to the tape is more tricky, and can involve starting the read/write operations from the middle of the tape. This has the effect of shortening the tape's physical length by half, and so getting to data that is toward the middle, going either direction, takes less time. Reading data at the ends of the tape takes more time, but less than if the whole length of the tape needed to be rolled.

Increasing the reliability of tape storage involves more closely controlling the read/write head alignment with the data tracks and error detection/correction techniques. Tape material is another consideration, as some materials are better than others for holding the data longer, or reducing the write error rate, or both. Tape is a very physical storage technology regarding its recording medium due to how the tape transport mechanisms and read/write heads physically contact the tape. Add all this up and you may wonder how tape has ever survived this long. We think it is because engineers have so much fun working with the mechanical challenges, but in reality it is the low cost of tape storage that keeps it going.

Linear Tape-Open (LTO) technology is a tape standard developed jointly by Hewlett-Packard, IBM, and Seagate. Since this is an open technology, any vendor can be licensed to develop and distribute LTO tape. Generation 1 LTO licenses became available in 1998 and products began appearing in 2000. Three more generations of LTO are slated to be released, with Generation 2 due in 2002. The capacity and data transfer speed of each generation doubles the previous. Generation 1 has a 100GB/cartridge (uncompressed) capacity with a 10–20 MBps transfer rate. When Generation 4 comes out, the top capacity will be 800GB/cartridge with a 80–160 MBps transfer rate. The release schedule for LTO additional generations is Generation 3, 2004, and Generation 4, 2006.

LTO comes in two flavors, a two-reel cartridge called Accelis, designed for speed, and a single-reel cartridge called Ultrium, designed for capacity. Accelis tapes load to the middle of the tape and thus attain better performance. Ultrium single-reel cartridges can hold more linear meters of tape, so they can hold more capacity.

LTO writes multiple tracks both forward and backward, which is common among other tape formats. Data compression built into the tape drive yields more capacity per cartridge. ECC skips by bad parts of the tape when writing. While reading, ECC rebuilds data found on a bad part of tape and writes it elsewhere on the tape. This is all well and good—other tape technologies support what LTO does so far. The unique things are how the read/write heads move and the technique used to position them.

Figure 2-16 shows an LTO tape drive's read/write heads with eight read/write elements per head. The head assembly moves up and down. The servo bands contain information to accurately position the heads. These bands were laid down during the tape's manufacturing. A direction buffer is maintained between the sets of tracks and at the tape edges to avoid magnetic interference and problems with recording too close to the tape edges. The figure is vastly simplified in that the actual number of total tracks on an Ultrium tape is 384, and each read/write pass covers eight tracks at once.

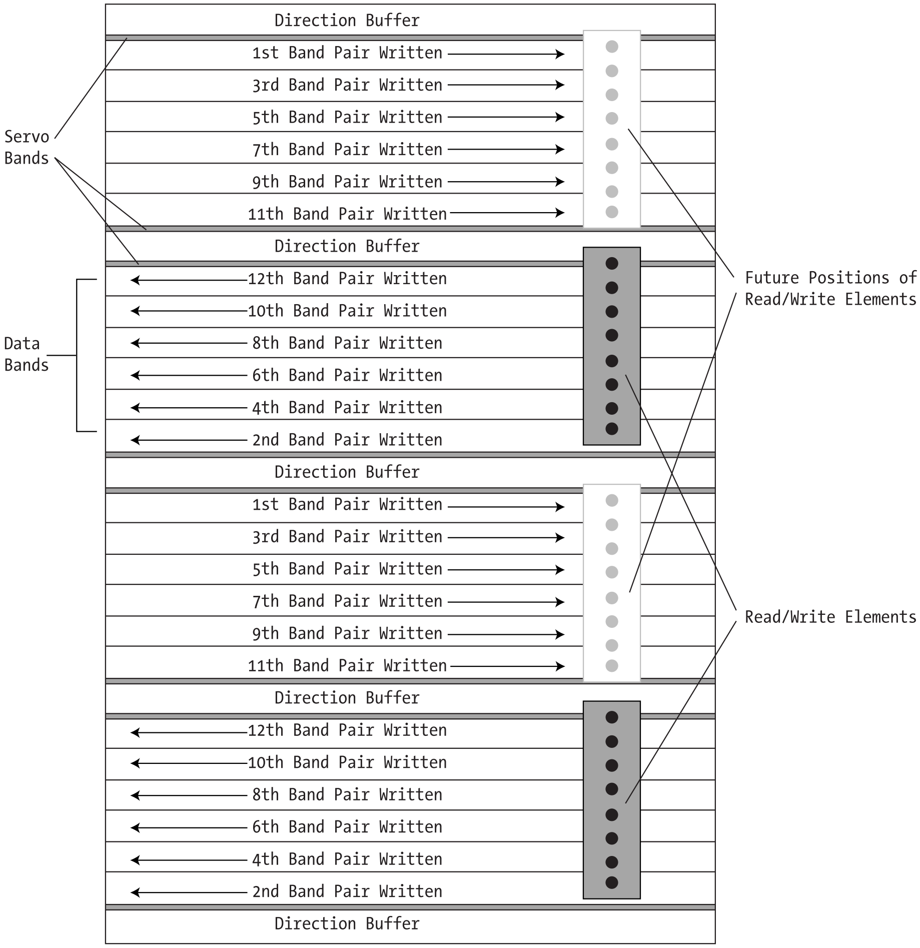

Figure 2-16. LTO read/write technology

The claimed advantages of LTO technology are as follows:

- The open, nonproprietary standard can be implemented by many vendors.
- Each cartridge has very high capacities.
- The servo bands provide high reliability for head positioning.

The primary disadvantage with LTO is that a migration from older tape technologies becomes necessary. Although LTO Generation 2, 3, and 4 drives will be able to read older LTO tapes, none of the LTO drives can read anything but LTO tapes. One suggested method is to use automated tape libraries with both old and new technology drives. The library management software will need to be smart enough to tell the difference between the technologies in order to pull this off. Another suggested method is to hire a third party to perform the migration. Either way, migration isn't cheap. Your organization will need to believe that the advantages of LTO far outweigh the cost of migration.

Of the older tape standards, there are two fundamental kinds: linear track and helical scan. Linear track tape drives write and read parallel tracks along the tape. Helical scan tape drives read and write short tracks angled between the tape edges. Figure 2-17 illustrates the differences.

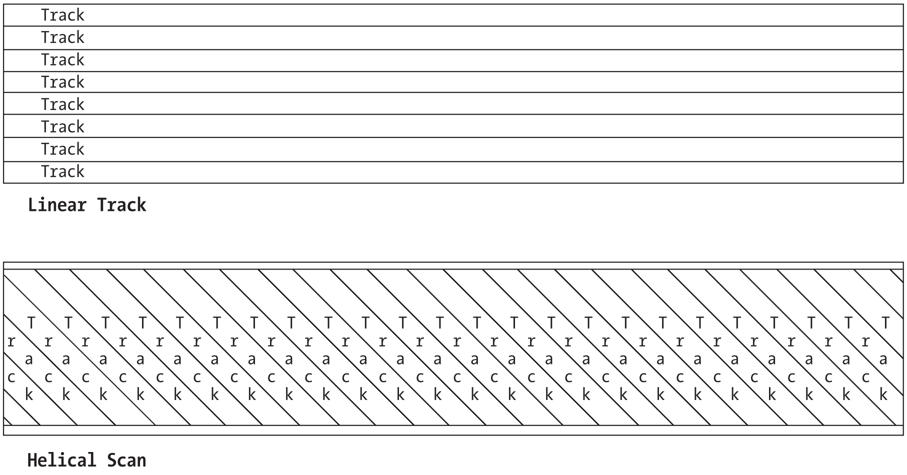

Figure 2-17. Linear track and helical scan technologies

With linear track tape, tracks can be read or written from beginning of tape (BOT) to end of tape (EOT). Then other tracks can be read or written from EOT to BOT. The tape drive does a lot of winding and rewinding when in use. Helical scan tape drives read and write from BOT to EOT without any rewinding except after the read or write operation has completed.

Digital linear tape (DLT) is an example of linear track tape. The number of tracks vary between 128 and 208, and capacity, after data compression, can reach up to 70GB. SuperDLT is a recent enhancement that bumps the capacity upward toward the 100GB level. DLT is commonly found in storage environments, along with other proprietary linear track tape (3490, 3590, and so on). The technique of writing linear track tape from center to edges is called *serpentine* recording. This technique takes advantage of the physical fact that the tape center is more stable, while in motion, than the tape edges.

Advanced intelligent tape (AIT) and Mammoth are examples of helical scan tapes. AIT capacity reaches up to 65GB of compressed data, and Mammoth up to 40GB, compressed.

The competition among tape drive manufacturers is an ongoing situation, so the previous capacity numbers for linear and helical scan tapes will likely grow with time.

TIP *Keep in mind that the numbers for compressed data are soft because they depend on how compressible the data actually is. Hardware compression built into the tape drive is more efficient than software compression because it saves CPU cycles and memory usage at the server. Hardware compression ratios may or may not be better than software compression.*

Semiautomation with Tape Loaders

Some degree of tape handling automation can be achieved by using tape loaders. A *tape loader* is a fixture that attaches to a tape drive (see Figure 2-18). The fixture holds several tapes to be written, often referred to as *scratch tapes*. The scratch tapes must be put into the loader and removed from the loader manually. This fact limits the value of tape loaders to systems requiring tape for not much more than backup. They are especially useful for systems that have backups spanning multiple tapes since manually loading the tapes would cause greater delays.

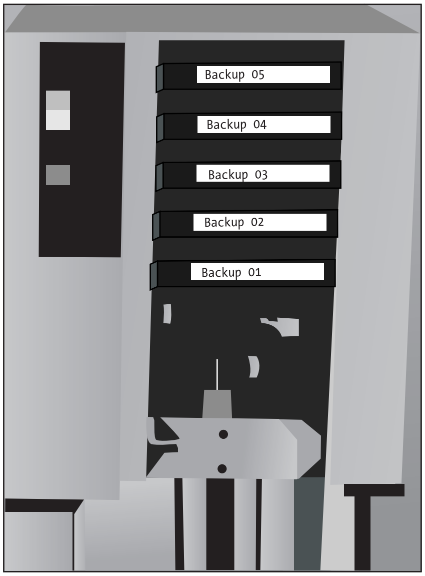

Figure 2-18. Tape loader

Tape loaders are incapable of selecting a random tape to be read. The tapes get loaded one after another in lockstep fashion. The only intelligence in a tape loader is knowing when a tape needs to be loaded, taking the tape that was just processed out of the drive, putting it in a stack of output tapes, and loading the next available tape from the input stack. Once all the tapes have been read or written, an operator has to take the output stack out of the loader and put fresh input tapes into it. The tape loader has to be told by the operating system when a tape mount is being requested and when the system is finished with the tape volume. If the tape loader puts the wrong volume into the drive, the operating system cries for help from a human operator, who must then fix things manually.

All this manual work implies that storage managers will be burdened some of the time with handling the human error factor for tape loads. Larger data centers can't afford squandering time on this old issue, and so automated tape libraries have become the tape technology of choice. In addition, faster mount times brought greater potentials to these data centers, which we will now talk about.

Automated Tape Libraries

Have you ever waited for a tape mount done by human hands? Even the fastest tape person takes a few minutes to find the tape and stick it into the drive. If your tape person is very busy, you may wait many, many frustrating minutes.

Automated tape silos reduced this time to a few seconds, and these are reliable seconds. No phone calls are required to remind someone to do the task.

StorageTek introduced the first automated tape library in 1987 (see Figure 2-19). IBM followed with a pieced-together competitor, the 3495 Tape Library Data Server, nicknamed Conan the Librarian. The nickname is a reference to a book/movie about an ancient warrior, *Conan the Barbarian.* As the story goes, Conan was an imposing guy with a strong sword arm. The 3495 library was large and imposing too, and sported a robotic arm that put the barbarian's to shame, being an original engineering manufacturer (OEM) piece meant for heavier industrial applications.

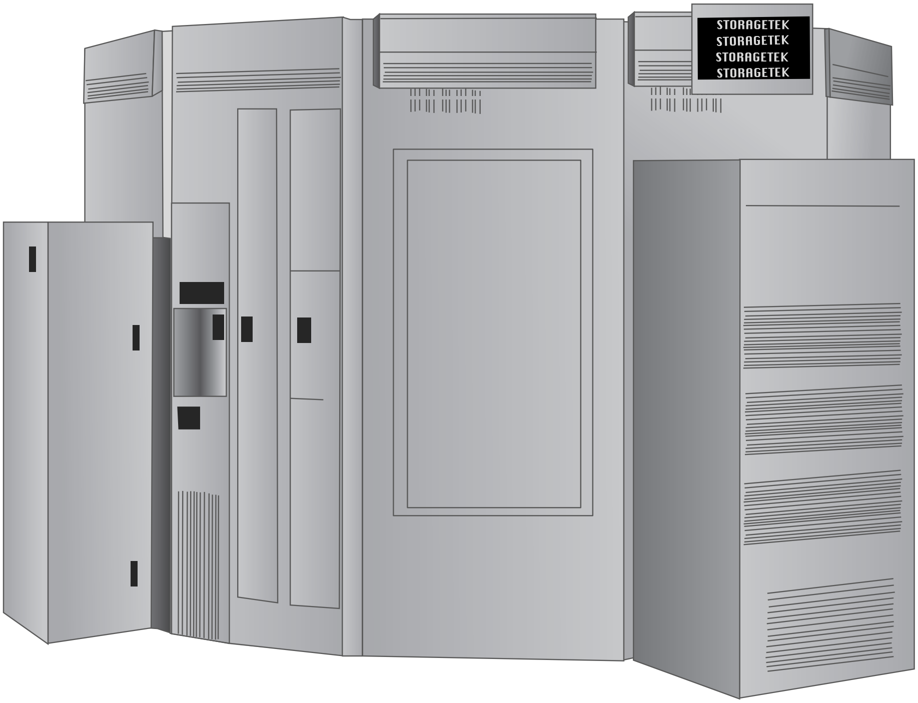

Figure 2-19. StorageTek Automated Tape Library

NOTE *OEM is a business shorthand way of saying, "We bought the thing from someone else. We did not develop this from scratch. We may have added or modified the functionality to fit our needs."*

Conan didn't sell very well due to the amount of floor space it required, and so IBM developed the 3494 Magstar automated libraries. These libraries vastly improve upon Conan's inefficiencies and take up much less floor space.

Today the storage manager often works with automated tape libraries. The libraries may be small rack-mounted affairs, cabinet-sized, or the more massive models that can be combined using cartridge pass-through ports. Some cabinet-sized libraries can be joined together with pass-through ports too. Pass-through allows tapes held in one library to be mounted on tape drives in another. It is a way of increasing the tape cartridge capacity of the overall library past the capacity of a single component of the library.

When disk storage in an installation is measured in terabytes, tape storage tends to break the petabyte level due to multiple backups and near-line storage.

Near-line storage refers to data that isn't used often being put onto tape in an automated tape library. The data is offline, that is, it can't be accessed until put back onto disk. Meanwhile, applications and users think the data is still on disk. With the fast automated tape loads, the data can be restored to disk relatively quickly where applications and users experience short, tolerable (theoretically) delays. The added speed that a robotic tape library provides has led to the coinage of the term *near-line*, or almost online but not quite. Of course, if the tape library is under heavy demand, the wait for data will elongate. Near-line storage grew out of Hierarchical Storage Management (HSM).

HSM was first developed in the mainframe world to better use expensive disk space. The idea was that if data on disk isn't used for a certain amount of time, it can be automatically rolled off to some cheaper medium, which usually means tape. If the user of the data needed it back, the HSM system would automatically restore the data to disk. If the data wasn't recalled from tape for a very long period of time, it could be automatically expired, freeing up tapes. HSM was first implemented without the benefit of automated tape libraries, and because of this it wasn't very popular with user communities due to long delays for tape mounts. The software was automated but not the hardware.

Today HSM can include optical storage as part of the strategy, and automated tape libraries or jukeboxes are inevitably part of the plan. If your organization has highly critical data that must have very fast restore times, then using disk storage for archives might be cost-justified. Even with automation, the time to restore critical applications from tape can elongate into hours or even days if the data to be restored is scattered among many tapes in a library.

NOTE *HSM is discussed in more detail in Chapter 7, "Hierarchical Storage Management (HSM)."*

Network-Attached Storage

Once disk arrays became common in data centers, the next logical step was to more efficiently use the storage. A single disk array can have multiple terabytes of capacity. If the array is directly attached to a server, only that server can use the disk capacity for applications. Some of the capacity can be shared using LAN shared disks or Network File System (NFS) mounts, but this still isn't an efficient use of disk space. Letting two or more servers access the disk array through direct attachment helps, but there is an even better way to do this.

Network-attached storage (NAS) is an array of disks that has its own little computer in it. The computer establishes network connections just like any other server, but it is a specialized computer. Its primary job is to present logical volume images to the network in order that many servers, both Unix and Windows (and potentially mainframes), can use parts of the NAS storage.

NAS is easy to install (see Figure 2-20), configure, and expand. Managing NAS is very easy. Some storage managers have given the nickname "toaster" to NAS because you plug it in and just let it cook.

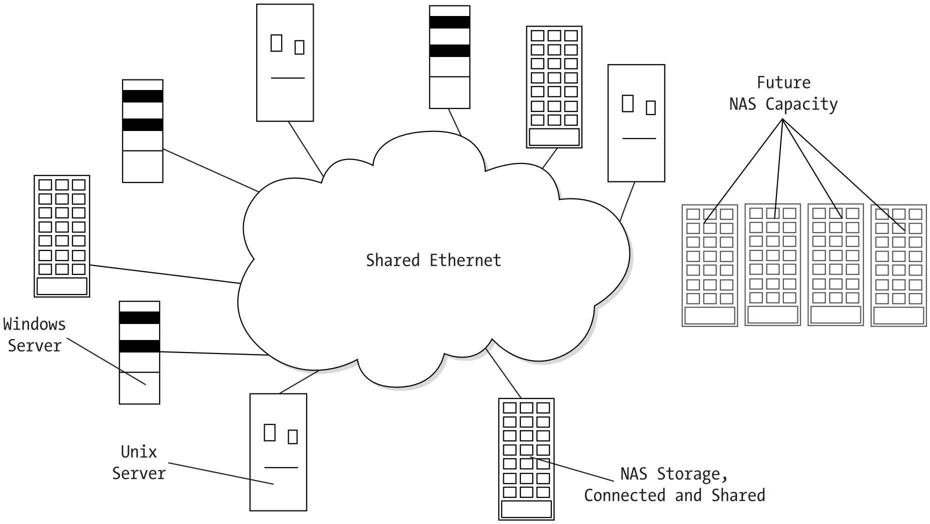

Figure 2-20. NAS installation

Figure 2-20 shows that both servers and NAS are connected to the shared Ethernet IP network. Unix and Windows servers can use storage from any of the network-connected disk arrays. More NAS can be added as more storage capacity is needed.

NAS might be a RAID array or a JBOD (short for just a bunch of disks) array. JBOD doesn't incorporate the fault tolerance of RAID-1, RAID-3, or RAID-5, nor does it stripe data like RAID-0.

The downside of NAS is performance. I/O rates will suffer if the shared IP network it is attached to is overly stressed. You probably don't want to do NAS on anything less than 1 Gbps Ethernet.

NOTE *NAS is covered in more detail in Chapter 4, "Network-Connected Storage."*

Storage Area Network

The storage area network (SAN) was developed to address the network performance issues of NAS, as mentioned previously, by dedicating a network to nothing but storage. Figure 2-21 illustrates the basic idea of SAN. SAN installations generally use Fibre Channel networks; however, IP protocols for SAN have been defined and are being implemented. These new protocols enable using fast Ethernet to build a SAN.

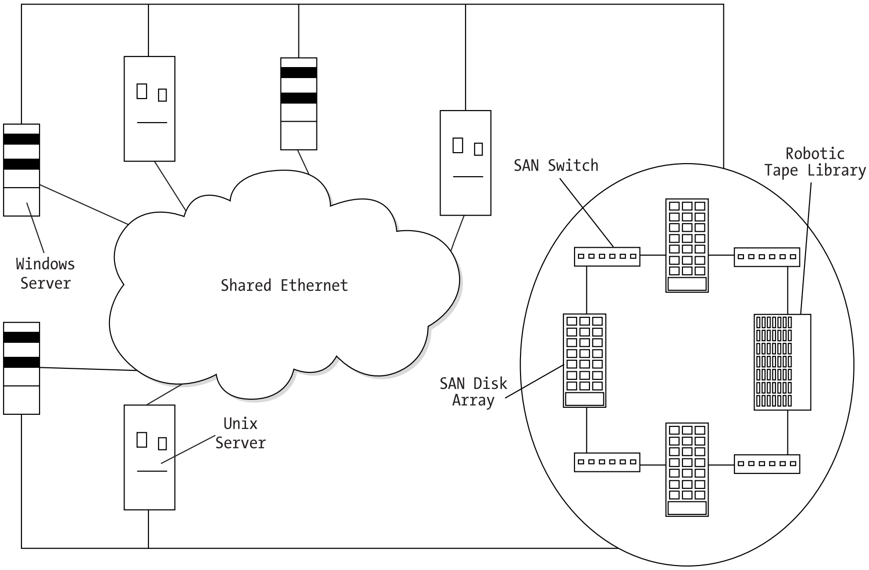

Figure 2-21. SAN installation

Fibre Channel, by its own definition, has an upward speed limit of 4 Gbps. Ethernet has no upward limit, and 10 Gbps equipment is entering the market now.

SAN hasn't been as popular as NAS due to its complexities. Setting up a Fibre Channel network is a major undertaking, as opposed to just plugging a disk array into the existing network. Additionally, managing a SAN requires a fair amount of Fibre Channel monitoring and troubleshooting expertise.

SAN does have some compelling advantages over NAS, namely the ability to mix many kinds of storage from several vendors; the ability to provide very strong centralized management; and the potential to improve storage performance.

NOTE *SAN and Fibre Channel are covered in greater detail in Chapter 4, "Network-Connected Storage."*

Storage Management Software

Along with the changes in storage hardware, storage management software has been changing rapidly too. What was once a loose collection of utilities, scripts, and home-grown programs has developed into full-featured suites of highly sophisticated software packages.

The major storage management software vendors include Computer Associates, EMC, HP, Legato, StorageTek, Tivoli, and VERITAS. Some vendors, EMC and Hitachi for example, have concentrated on their own line of storage hardware. Others like Legato cover other vendors' hardware.

Minimally, storage management software should perform and automate backups, provide nondisruptive backup of databases, streamline routine storage management tasks, and report on storage performance and capacity. If the operating system's logical disk volume manager isn't up to par, this may also be a desirable part of the suite. Deluxe features include SAN management and reporting, performance enhancements, and extended troubleshooting support.

For complex storage environments, the storage manager may find that several different vendor software packages are in use. For example, Legato NetWorker may be in use for Compaq StorageWorks equipment, EMC Navisphere may be installed to support CLARiiON NAS, and VERITAS Backup Exec may be the overall backup software.

NOTE *The features of storage management software from the previously mentioned vendors are listed in Appendix D, "Storage Management Software Vendors."*

Several storage management software vendors try to cover many storage hardware platforms and operating systems at once, and this has led to the development of another protocol standard, Network Data Management Protocol (NDMP). NDMP is an open standard that was started in 1996 to simplify backup software development. Prior to NDMP, software developers had to code for each hardware platform and operating system, including different hardware versions and operating system releases.

The value of NDMP to storage software vendors is obvious—it saves a lot of programming effort. The value to the storage administrator is that potentially one suite of storage management software can be used to control and monitor a complex storage environment consisting of a multitude of different vendor hardware products. In order for NDMP to work, the hardware has to support the protocol, and so NDMP is gradually entering the storage arena. The protocol promises to make your life as a storage manager a lot easier.

More information on NDMP is available at `http://www.ndmp.org`.

Acronyms Used in This Chapter

The acronym blizzard gets thicker the further we move into the details of storage management. Although this is unavoidable while discussing technologies used in storage, we provide the following list as a reference to help in your navigation of the storm.

ACRONYM	FULL NAME	EXPLANATION
ACM	Association for Computing Machinery	An organization for computer engineers.
AIT	advanced intelligent tape	A type of tape storage that uses helical scan technology.
CD-ROM	compact disk read-only memory	A write once, read many optical disk.
CD-RW	compact disk rewritable	An erasable and rewritable optical disk.
CPU	central processing unit	The brains of a computer.
DAS	direct attached storage	Storage that directly attaches to a computer's bus.
DASD	direct access storage device	A storage device that allows the read/write head to access data directly rather than reading it all up to the desired part. This follows the way sequential access works.

(continued)

ACRONYM	FULL NAME	EXPLANATION
DLT	digital linear tape	A type of tape storage that uses linear track technology.
DS	Digital Signal	A telephony communications standard.
DVD	digital versatile disk	A high-density optical disk.
DWDM	Dense Wavelength Division Multiplexing	A high-speed fiber optic communication technology that splits the laser light into multiple wavelengths.
EB	exabyte	10E18 bytes.
ECC	error-correction code	A method of correcting data read errors on the fly.
GB	gigabyte	10E9 bytes.
HSM	Hierarchical Storage Management	A method of saving disk space by putting seldom-used data onto cheaper media, usually tape.
IC	integrated circuit	Modern computer chip technology.
I/O	input/output	The action of moving data that resides on auxiliary storage, such as disk drives and tapes, into a computer's main memory (read) and out of it, after processing, back to auxiliary storage (write).
ISDN	Integrated Services Digital Network	A set of standards for digital communications over telephony networks.
LAN	local area network	A network characterized by no use of telephony links.
LTO	Linear Tape-Open technology	A new way of writing to tape, characterized by high density and fast transfer rate.
MB	megabytes	10E6 bytes.

(continued)

ACRONYM	FULL NAME	EXPLANATION
MO	magneto-optical disk	Where data is written with magnetism and read with a laser.
MRAM	magnetic random access memory	A new, nonvolatile memory technology that does not depend on electricity to keep the memory.
MTBF	mean time between failure	A measure of how long some piece of computing equipment might last.
NAS	network-attached storage	A disk array that has built-in computing power for internal management of the array and to provide interfaces to the shared IP network.
NDMP	Network Data Management Protocol	A protocol that standardizes software and hardware interfaces for storage management software development.
NFS	Network File System	A set of Unix daemons that allows a computer to access files on other computers.
NVRAM	nonvolatile random access memory	Memory that is used for various reasons, including the keeping of cached writes during a power failure.
NVS	nonvolatile storage	Memory that is used in the same way as NVRAM.
OC	Optical Carrier	A communications standard for optic fiber networks.
PB	petabyte	10E15 bytes.
RAID	redundant array of independent disks	A set of methods for providing performance enhancements and fault tolerance to an array of disk drives.
RAB	RAID Advisory Board	An organization dedicated to s upporting RAID.

(continued)

ACRONYM	FULL NAME	EXPLANATION
RAM	random access memory	Memory that can be accessed using direct addressing rather than reading from beginning to end.
RAMAC	Random Access Method of Accounting and Control	The first direct access disk drive, announced by IBM in 1956.
rpm	revolutions per minute	A measure of something that spins, such as the disks in a disk drive.
SAN	storage area network	A method of providing high-speed, high-access storage to many computing systems at once through a dedicated network, or by special protocols over a shared IP network.
SIGMOD	Special Interest Group on Management of Data	A suborganization of the ACM.
SLED	single large expensive disk	A type of disk drive that was once used with mainframe computers but is now obsolete.
SSD	solid-state disk	A set of DRAM chips and a controller that makes the memory look like some form of disk drive; characterized by very fast I/O speed and data volatility.
STM	Synchronous Transport Module	A communication standard for optic fiber networks.
TB	terabytes	10E12 bytes.
VSS	virtual storage subsystem	An early attempt to make mainframe I/O more efficient by using a small mainframe between the host mainframe and the SLED drives.
WORM	write once	Read many, same as CD-ROM.
XOR	exclusive OR	A type of gating technology used at the IC chip level.

Vendor and Organization Web Sites

You may have noticed tight associations among corporations and nonprofit organizations in the storage field. This is a reflection of the desire for open systems, standards, and cooperative market sharing in the storage industry. This is also an indication of how important storage is to the organizations that purchase storage equipment, and the value of the market itself. The following is a list of vendors and organizations referenced in this chapter.

VENDOR OR ORGANIZATION	WEB SITE
Association for Computing Machinery	http://www.acm.org
Compaq (now part of HP)	http://www.compaq.com
Computer Associates	http://www.cai.com
EMC	http://www.emc.com
Hitachi	http://www.hitachi.com
HP	http://www.hp.com
IBM	http://www.ibm.com
Legato	http://www.legato.com
Seagate	http://www.seagate.com
StorageTek	http://www.storagetek.com
Tivoli	http://www.tivoli.com
University of California, Berkeley	http://www.berkeley.edu
University of Manchester	http://www.man.ac.uk
VERITAS	http://www.veritas.com

Summary

Storage managers tend to work with diverse conglomerates of hardware and software. This has always been a challenge throughout the years, but the challenge is becoming even greater with new architectures, standards, and technologies coming to market. We have attempted to give you some perspective of the complexities that accompany this diversity in today's storage arena in this chapter. Our hope is that, once the technologies have been defined and explained, you will discover that storage isn't so difficult to grasp. After all, it is simply a matter of data preservation, protection, and delivery to the CPU.

Disk and tape storage have been covered in enough detail that you should be able to make intelligent judgments on the available technologies. The next chapter covers DAS, which goes into more detail regarding the practical use of nonnetworked disk and tape.

Chapter 4, “Network-Connected Storage,” consists of detailed discussions of networked storage (NAS and SAN), plus a closer look at remote mirroring.