CHAPTER 19

The Perfect Server

The Network+ Certification exam expects you to know how to

- 3.10 Identify the purpose, benefits, and characteristics of using antivirus software
- 3.11 Identify the purpose and characteristics of fault tolerance: power, link redundancy, storage, services
- 3.12 Identify the purpose and characteristics of disaster recovery: hot and cold spares

To achieve these goals, you must be able to

- Identify methods and hardware used for protecting data
- Describe server-specific hardware used for boosting speed
- Explain methods and hardware used for server reliability

The job of networking demands fundamental hardware differences between a PC that connects to a network and a PC that does not connect to a network. Arguably, the designers of the Personal Computer never considered the PC as a device to participate in a network. You can't blame them. The original PC simply didn't pack the necessary firepower to function in any but the most primitive of networks. The first PCs used tiny (less than 10 megabyte) hard drives—or only floppy drives—and the 4.77-MHz Intel 8088 simply could not handle the many calculations demanded by even the most basic network operating systems. The mainframe-centric world of IBM created the PC to work primarily as an individual computer, a stand-alone system, or to perform as a dumb terminal for mainframe access.

Historical/Conceptual

While networks were not part of the original PC concept, the ongoing improvements in the power and phenomenal flexibly of PCs enabled them to move easily from a world of individual, stand-alone systems into the interactive world of connected, networked machines. Even though any stand-alone PC transforms nicely into a networked machine, the different jobs of a stand-alone versus a networked machine require significantly different hardware in each. What are these requirements? What hardware does a networked PC need that a stand-alone PC can live without? The network functions themselves supply the answers (see Figure 19-1).

A networked PC has four significant functions. First, it must connect to the network. This connection usually runs through a cable of some type, but wireless networks are becoming more common.

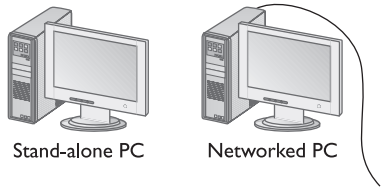


Figure 19-1 Networked PCs need more hardware.

Second, if the PC shares data, the PC needs to protect that shared data by creating more than one copy of the data. The data is usually copied with multiple storage devices—almost always hard drives—that work together to create multiple copies of data.

Third, and again only if the PC shares data, it needs specialized hardware that enables it to share the data as quickly as possible. A sharing PC often uses a number of different hardware technologies to increase the speed with which it shares its resources. A good example of a speed technology is a specialized network card that enables faster data access.

The fourth and last function unique to a network PC is reliability. The shared resources of the network must be available whenever another system accesses them. The networked PC must use special hardware to prevent a sharing system from failing to provide their shared resources. We're not talking about more hard drives here; we've already covered that! Reliability means methods to make sure the PC doesn't stop working due to a failed component. These hardware devices manifest themselves in items such as redundant power supplies or air conditioning units. Together or separately, every network PC has at least one of these four functions (see Figure 19-2).

The process of deciding which functions appear in a network PC is determined by the job of that particular system. The biggest line of demarcation is between systems that share resources (servers) and systems that only access the server's shared resources (workstations). The hardware requirements for a workstation and a server differ fundamentally. The only specialized function of a workstation is connecting to the network via a network interface card (NIC). Workstations do not share resources, so they have little need for reliability, speed, and data protection beyond that already built into any stand-alone PC.

Servers, on the other hand, use all of the functions creating the need for highly specialized systems full of specialized hardware to provide most, if not all, of these four network functions. The incorporation of the specialized hardware in a PC is what makes what we call a *server system*. The incorporation of these extra features makes a server stand out compared to a workstation. Servers are often designed as rack mounts to fit into an equipment rack or are large, floor-mounted units (see Figure 19-3).

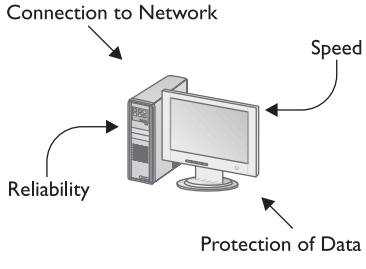


Figure 19-2 The four network functions

Figure 19-3
A typical network server (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

Keep in mind that there is no requirement for a serving system to have the extra hardware. Virtually any PC can act as a serving system—as long as you are willing to put up with lack of reliability, slower response times, and the higher potential for data loss. Equally, in peer-to-peer networks, some, most, or all of the systems act as servers. It's usually logistically impractical and financially imprudent to give every user in a peer-to-peer network a powerful server system (although if you did, you'd be extremely popular!). A good network person considers the network functions of a particular system to determine which ones the system needs. They then balance the needed network functions against cost, time, and support needs to determine what hardware a particular system requires (see Figure 19-4).

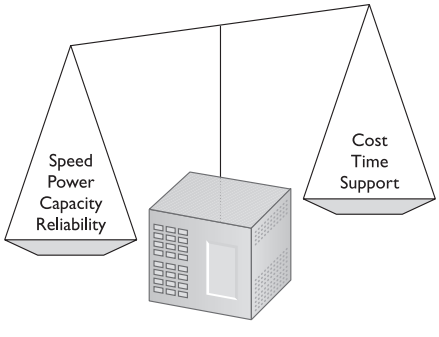


Figure 19-4
Balancing needs versus expense

Test Specific

Server PCs need extra hardware or software to provide data safety, speed, and reliability. In this chapter, we first define conceptually each of these functions and then explore how servers utilize the wide variety of hardware, software, and organization solutions used in today's networks to fulfill the needs of these functions.

Protection of Data—Fault Tolerance

The single most important part of most networks is the *shared data*. The main motivation for networks is the ability for many users to access shared data. This shared data might be as trivial as pre-made forms or as critical as accounts receivable information. The sudden loss of data in their networks would cripple most organizations. Computers can be replaced and new employees hired, but the data is what makes most organizations function. Certainly, any good network must include a solid backup plan, but restoring backups takes time and effort. Unless the data is being continually backed up, the backups will always be a little dated. Backups are a last-resort option. Businesses have failed after the loss of data—even with relatively good backups. The shared data of a network should have better protection than the fallback of laboriously having to restore potentially dated backups! A good network must have a method of protecting data such that if a hard drive fails, a network technician can bring the data instantly, or at least quickly, back online. This requires some sort of instant backup or automatic copy of the data stored on a second drive. The capability of a server to respond to a hardware failure while continuing to operate is called *fault tolerance*.

Okay, so you need to come up with a way to make data redundant on the serving system. How do you do this? Well, first of all, you could install some fancy hard drive controller that reads and writes data to two hard drives simultaneously (see Figure 19-5). This would ensure that the data on each drive was always identical. One drive would be the primary drive, while the other drive, called the *mirror* drive, would not be used unless the primary drive failed. This process of reading and writing data at the same time to two drives is called *drive mirroring*.

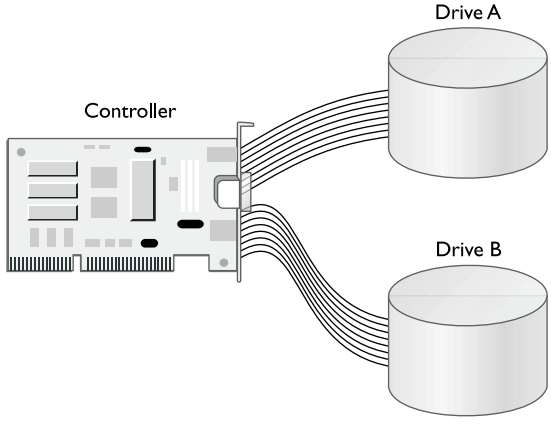


Figure 19-5
Mirrored drives

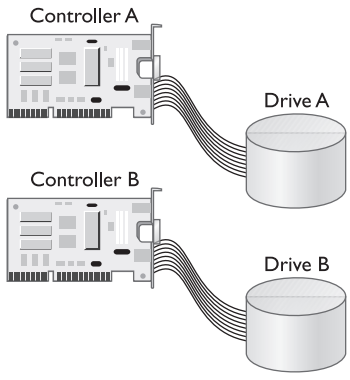


Figure 19-6
Duplexing drives

If you want to make data safe, you can use two separate controllers for each drive. With two drives, each on a separate controller, the system will continue to operate, even if the primary drive's controller stops working. This super-drive mirroring technique is called *drive duplexing* (see Figure 19-6) and is much faster than drive mirroring because one controller does not write each piece of data twice.

Even though drive duplexing is faster than drive mirroring, they both are slower than the classic one drive, one controller setup. The third and most common way to create redundant data is by a method called *disk striping with parity*. *Disk striping* (without parity) spreads the data among multiple (at least two) drives. Disk striping by itself provides no redundancy. If you save a small Microsoft Word file, for example, the file is split into multiple pieces; half of the pieces go on one drive and half on the other (see Figure 19-7).

The one and only advantage of disk striping is speed—it is a fast way to read and write to hard drives. But if either drive fails, *all* data is lost. Disk striping is not something we ever want to do—unless you simply need all the speed you can get and don't care about data.

NOTE A number of popular technical web sites have tested two striped drives against a single drive to see if striping provides any increase in data throughput. In every case, the amount of increase in data throughput with two striped drives was negligible as compared to a single drive. *Disk striping by itself is not a recommended practice.*

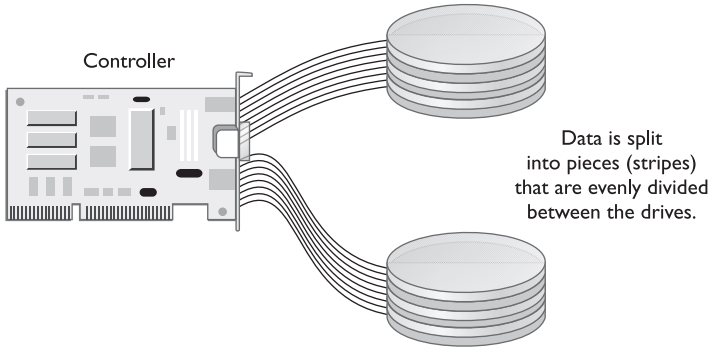


Figure 19-7
Disk striping

Disk striping with parity, in contrast, protects data. Disk striping with parity adds an extra drive, called a *parity drive*, that stores information that can be used to rebuild data should one of the data drives fail. Let's look at that same Microsoft Word document used earlier. The data is still stored on the two data drives, but this time a calculation is done on the data from each equivalent location on the data drives to create parity information on the parity drive. This parity data is created by a simple, but accurate calculation. It's similar to dividing two numbers and storing the result of the division. The calculation is not important; the fact that the parity data can be used to rebuild either drive is, however.

NOTE Modern implementations of disk striping with parity spread the parity information and the data across all three drives, as you'll see in the next section on RAID.

Disk striping with parity must have at least three drives, but it's common to see more than three. Unfortunately, the more drives used, the higher the chance one might fail. Disk striping with parity can only recover data if one drive fails. If two drives fail, you're heading for the backup tapes!

Disk striping with parity combines the best of disk mirroring and plain disk striping. It protects data and is quite fast. In fact, the majority of network servers use a type of disk striping with parity.

RAID

The many different techniques of using multiple drives for data protection and increasing speeds were organized by a couple of sharp guys at Berkeley back in the 1980s. This organization was presented under the name *Random Array of Inexpensive Disks (RAID)* or *Random Array of Independent Disks*. There are seven official levels of RAID, numbered 0 through 6, which are as follows:

- **RAID 0** Disk striping
- **RAID 1** Disk mirroring and disk duplexing
- **RAID 2** Disk striping with multiple parity drives. Unused, ignore it.
- **RAID 3 and RAID 4** Disk striping with parity. The differences between the two are trivial.
- **RAID 5** Disk striping with parity, where parity information is placed on all drives. This method combines data redundancy with a performance boost (or at least no performance hit like you see with RAID 1). RAID 5 is the most common RAID implementation on server machines.
- **RAID 6** RAID 5 with the added capability of asynchronous and cached data transmission. Think of it as a Super RAID 5.

A lot of modern motherboards come with RAID controllers built in and sporting one or two non-traditional RAID modes called RAID 0+1 and RAID 10. Both methods purport to offer RAID 5 data redundancy and performance, but this claim is problematic.

Both modes require four physical drives, rather than the three needed in a RAID 5 array. RAID 0+1 mirrors two sets of striped drives. RAID 10 stripes two sets of mirrored drives. The implementations are pretty much flip sides of the same coin.

The only problems with these nontraditional RAID modes are the quantity of drives required—four hard drives is quite an investment, even at today's prices—and the performance does not equal a good RAID 5 array. On the other hand, if you've got the drives, power for the drives, and a built-in controller, why not? RAID 5 controllers generally cost a lot more than a controller that can do RAID 0+1 or RAID 10, and both of the nontraditional RAID modes are better than RAID 0 or RAID 1 for data redundancy and performance combined.

No network tech worth her salt says things like "We're implementing disk striping with parity." Use the RAID level. Say, "We're implementing RAID 5." It's more accurate and impressive to the folks in Accounting!

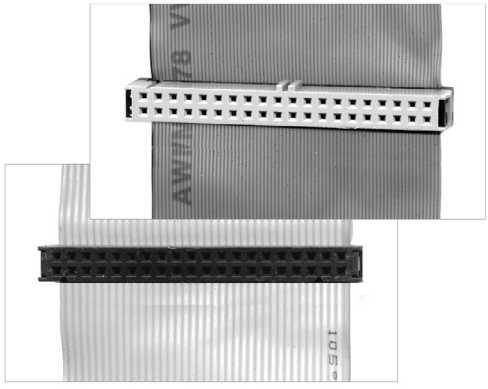
Drive Technologies

Talking about RAID levels is like singing about Einstein's Theory of Brownian Motion. You may sound good, but that doesn't mean you know what you are talking about! Remember that RAID levels are a general framework; they describe methods to provide data redundancy and enhance the speed of data throughput to and from groups of hard drives. They do not say *how* to implement these methods. There are literally thousands of different methods to set up RAID. The method used depends largely on the desired level of RAID, the operating system used, and the thickness of your wallet. Before we delve into these solutions, however, let's do a quick run-through of the three leading hard drive technologies—parallel ATA, serial ATA, and SCSI—to make a few terms more clear.

PATA If you peek into most desktop PCs, you will find hard drives based on the ultra-popular *Parallel Advanced Technology Attachment (PATA)* standard. PATA drives are always internal—inside the PC, which is designed traditionally to use up to four PATA drives. PATA drives can be identified by their unique 40-pin ribbon cable connection (see Figure 19-8). The cables are either 40-wire (for the older drives) or 80-wire (for newer drives). Figure 19-9 shows the two ribbon cables.

Figure 19-8 PATA connection on hard drive

Figure 19-9
PATA cables



NOTE You will often hear PATA drives referred to by their older technology names, such as Integrated Device Electronics (IDE) and Enhanced Integrated Device Electronics (EIDE). Although we old techs can argue about the distinctions, they're pretty irrelevant to the modern network tech. All three terms—PATA, IDE, and EIDE—are used synonymously.

TIP the Network+ exam uses the term "IDE" to describe all PATA drives.

The price, performance, and ease of installation explain the tremendous popularity of PATA drives. PATA can accept any type of storage device, including CD- and DVD-media drives, tape backups, and removable drives. Even with the capability to handle diverse devices, the PC cannot handle more than the maximum of four PATA devices without special additional hardware. My new high-end motherboard, for example, has a total of four PATA connections, two of which are standard connections, and two of which are RAID-capable. Each connection can handle two drives, so I can theoretically put eight PATA drives on this system. Plus, it has two additional hard drive connectors that utilize the current drive technology to which every one is turning, serial ATA.

SATA For all its longevity as the mass storage interface of choice for the PC, parallel ATA has problems. First, the flat ribbon cables impede airflow and can be a pain to insert properly. Second, the cables have a limited length, only 18 inches. Third, *hot swapping* isn't possible with PATA drives—that is, you can't add or remove such a drive with the system running. You have to shut down completely before installing or replacing a drive. Finally, the technology has simply reached the limits of what it can do in terms of throughput.

Serial ATA (SATA) addresses these issues. SATA creates a point-to-point connection between the SATA device—hard drive, CD-ROM, CD-RW, DVD-ROM, DVD-RW, and so

on—and the SATA controller. At a glance, SATA devices look identical to standard PATA devices. Take a closer look at the cable and power connectors, however, and you'll see significant differences (Figure 19-10). Because SATA devices send data serially instead of in parallel, the SATA interface needs far fewer physical wires—seven instead of the eighty wires that is typical of PATA—resulting in much thinner cabling. This might not seem significant, but the benefit is that thinner cabling means better cable control and better airflow through the PC case resulting in better cooling.

Further, the maximum SATA device cable length is more than twice that of an IDE cable—one meter (39.4 inches) instead of 18 inches. Again, this might not seem like a big deal, unless you've struggled to connect a PATA hard drive installed into the top bay of a full-tower case to a controller located all the way at the bottom!

SATA devices are *hot-swappable,* meaning that they can be plugged into or removed from the PC without having to shut down. This makes SATA a natural fit for RAID technology on operating systems that support it.

The big news, however, is in data throughput. As the name implies, SATA devices transfer data in serial bursts instead of parallel, as PATA devices do. Typically, you don't think of serial devices as being faster than parallel, but in this case, that's exactly the case. A SATA device's single stream of data moves much faster than the multiple streams of data coming from a parallel IDE device—theoretically up to 30 times faster!

SATA devices currently have a rated maximum data burst throughput rate of 150 Mbps. Granted, this isn't much of an immediate gain over current PATA speeds, but the SATA technology specification calls for eventual throughput speeds of up to 600 Mbps! Obviously the potential for greatly improved performance is the biggest draw to SATA.

Installing SATA hard drives is even easier than PATA devices because there's no master, slave, or cable select configuration to mess with. In fact, there are no jumper settings to worry about at all, as SATA only supports a single device per controller channel. Simply connect the power and plug the controller cable in as shown in Figure 19-11—the operating system automatically detects the drive and it's ready to go! The keying on SATA controller and power cables makes it impossible to install either incorrectly.

Figure 19-10 SATA hard drive data (left) and power (right) connections

Figure 19-11 Properly connected SATA cables

SCSI *Small Computer System Interface (SCSI)* accomplishes much the same goals as EIDE—making hard drives and other devices available to the PC. SCSI, however, is not a hard drive technology. Think instead of SCSI as a mininetwork that connects many different types of devices. Virtually any kind of storage device you can imagine comes in a SCSI version, but SCSI hard drives are the most common type of SCSI storage device. SCSI manifests itself in PCs via a card called a *host adapter*. This host adapter then connects to SCSI devices in a daisy-chain (see Figure 19-12). An installed set of SCSI devices is called a *SCSI chain*.

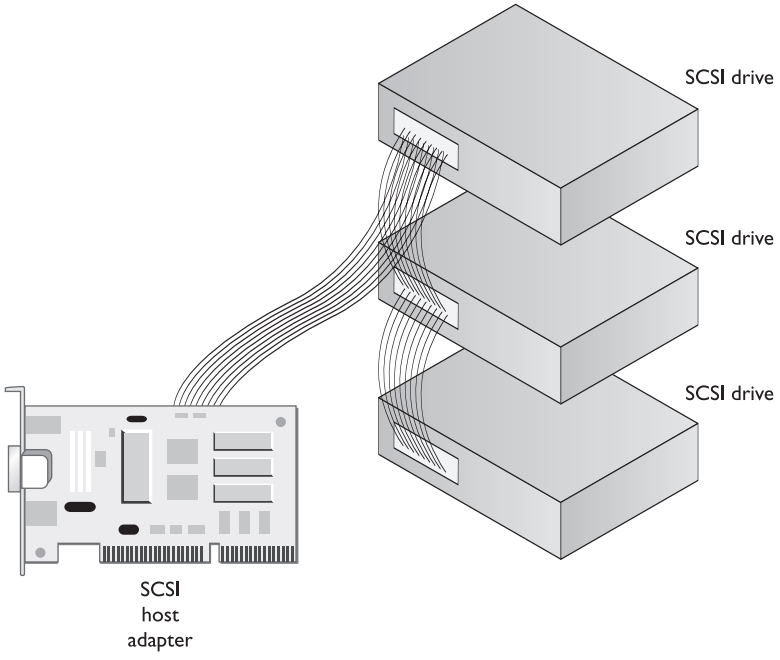


Figure 19-12 A SCSI chain

Each SCSI device on the SCSI chain must have a unique SCSI ID. Older SCSI devices are numbered 0 through 7, with 7 usually reserved for the host adapter itself. More advanced versions of SCSI can support up to 16 devices (including the host adapter).

SCSI devices can be internal or external. Better host adapters come with an internal and an external connector, enabling both types of devices to exist on the same SCSI chain. Figure 19-13 shows a SCSI chain with both internal and external devices. Note that each device gets a unique SCSI ID.

SCSI Connections Fortunately, the Network+ exam isn't interested in your ability to configure SCSI. It does, however, demand you know the many connections unique to SCSI devices. No other class of device has as many connections as SCSI. This is because SCSI has been in existence for a long time and has gone through four distinct standard upgrades, fostering many variations within each standard over the years.

SCSI connections differ for internal and external SCSI devices. There are two types of internal SCSI connections, both of which are inserted into a ribbon cable, just like PATA: the 50-pin narrow connection and the 68-pin wide SCSI. Figure 19-14 shows a typical 50-pin narrow connection with a ribbon cable attached. Figure 19-15 shows a 68-pin connection.

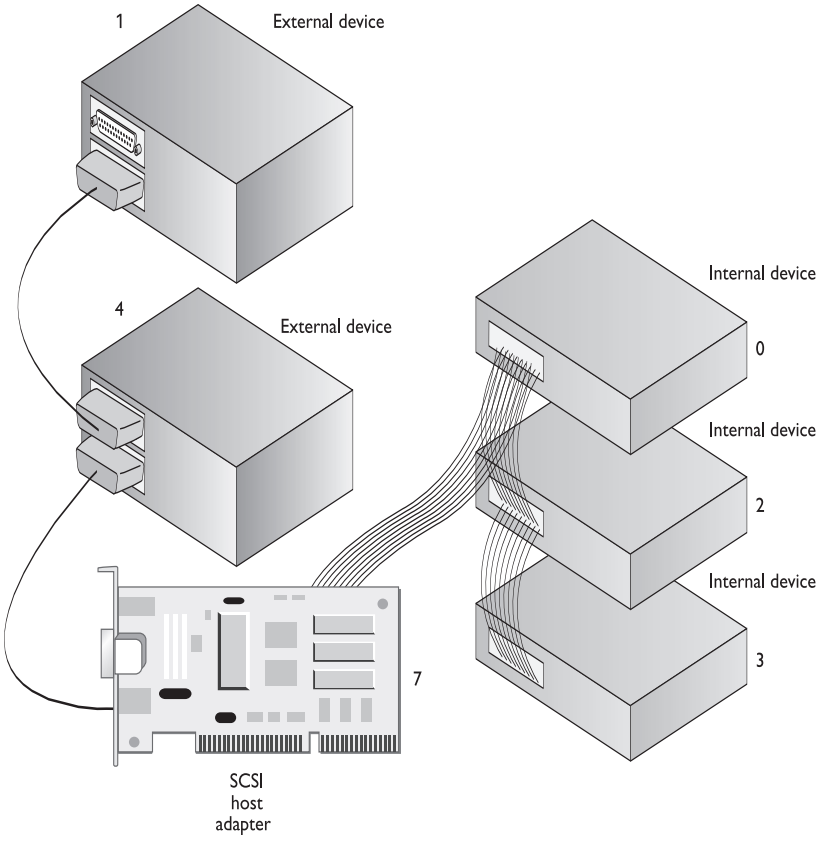


Figure 19-13 A typical SCSI chain with internal and external devices

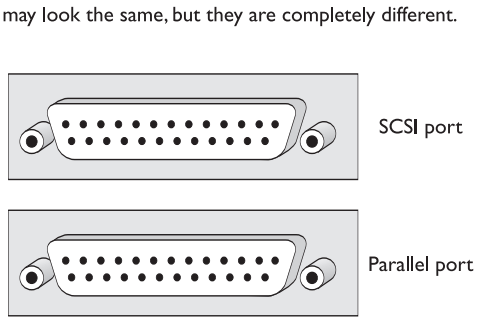
Figure 19-14 The 50-pin narrow SCSI connection

Figure 19-15 The 68-pin wide SCSI connection

The oldest external SCSI connection is a 50-pin Centronics. Although dated, a large number of SCSI devices still use this connector. It looks like a slightly longer version of the printer Centronics (see Figure 19-16).

Figure 19-16 Two 50-pin SCSI Centronics connections

Figure 19-17
Parallel and SCSI connections—both DB-25s

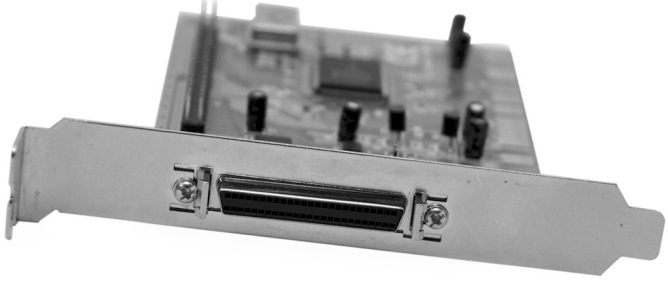


Many host adapters use a female DB-25 connector. Apple has been using female DB-25 connectors for SCSI on its computers for many years, but they are fairly new to PCs. This Apple-style SCSI connector is identical to a PC parallel port (see Figure 19-17), which is unfortunate because they are not electrically compatible. If you plug your printer into the SCSI port, or a SCSI device into the printer, it definitely will not work—and in some cases may damage devices!

Most modern SCSI devices now come with the special, SCSI-only, high-density DB connectors. High-density DB connectors look like regular DBs at first, but have much thinner and more densely packed pins. High-density DB connectors come in 50- and 68-pin versions, the former being the more common of the two (see Figure 19-18).

They All Work! PATA, SATA, and SCSI drives work beautifully for RAID implementations. People who are new to RAID immediately assume that RAID requires some special, expensive stack of SCSI drives. Such is not the case. You certainly can spend the money on fancy RAID boxes, but you do not have to go that route. You can easily implement RAID using nothing but inexpensive PATA drives and cheap, sometimes free software. Furthermore, RAID can use combinations of PATA, SATA, and SCSI (although trying to keep track of combinations of drives is not recommended!). In fact, PATA and SATA RAID arrays have lately become so stable that they rival the security that only SCSI used to promise. The only real distinction nowadays is a difference in access speed and price.

Figure 19-18
The high-density DB-50



Most people prefer SCSI drives for RAID, because they tend to be faster than PATA drives and you can put more drives into a system (7 to 15, rather than the 4 in PATA). The only drawback with SCSI is cost—hard drives are more expensive and you often must purchase a host adapter as well. When speed outweighs cost as a factor in what type of hard drive technology to use in a RAID array, SCSI implementations win out. Finally, if you need serious speed and extra bells and whistles, you can install any number of expensive "stack of SCSI drives" solutions.

RAID Implementations

All RAID implementations break down into either hardware or software methods. *Software RAID* means to use the regular drives on your system, and then to use software, usually the operating system, to create the RAID arrays. The operating system itself is in charge of running the array. Each hard drive in the array is visible to the operating system. If you go into Disk Management in Windows, for example, you'll see every drive in the array. Software RAID is often used when price takes priority over performance and is not popular for real-world servers.

Hardware RAID means to use dedicated RAID controllers to create the RAID arrays. Hardware RAID uses either a CMOS-like configuration or proprietary configuration software to set up the array. Once the array is configured, the RAID controller handles the running of the RAID array. The individual drives in hardware RAID arrays are invisible to the operating system. If you use a hardware RAID array and go into Disk Management in Windows, for example, you'll see the RAID array as a single drive. Hardware is used when you need speed along with data redundancy.

The most famous software implementation of RAID is the built-in RAID software that comes with Windows NT Server/2000 Server/Server 2003. The NT Disk Administrator and 2000/2003 Disk Management can configure drives for RAID 0, 1, or 5, and they work with PATA, SATA, and SCSI drives (see Figure 19-19). Windows 2000 and XP Professional only support RAID 0.

You can use Disk Management in Windows 2000 and XP Professional only to create RAID 0 on Windows 2000 and XP Professional machines. If you start Disk Management on a Windows 2000 or XP Professional machine and attach to (choose to manage) a Windows 2000 Server or Server 2003 system, you can create a RAID 1 or RAID 5 array on that server machine remotely.

The one great downside of RAID stems from the fact that with the exception of RAID 0, every version of RAID sacrifices total storage capacity for safety. Take RAID 1 for example. If you have two 160-GB drives in your system not running as RAID, you'll have a total storage capacity of 320 GB. If you then mirror those two drives, each drive stores an identical copy of the same data, reducing your total storage capacity to only 160 GB.

More advanced RAID versions suffer from the same loss of capacity for the sake of safety. Let's say you have three 100 GB drives, making a total storage capacity of 300 GB. If you make those three drives into a RAID 5 array, one third of the total capacity is used for parity data, reducing the total storage capacity down to 200 GB.

Windows NT/2000/2003 are not the only software RAID games in town. There are a number of third-party software programs available that can be used with other operating systems. There are even third-party software RAID solutions for NT that add a number of extra features above what the Disk Administrator or Disk Management provide.

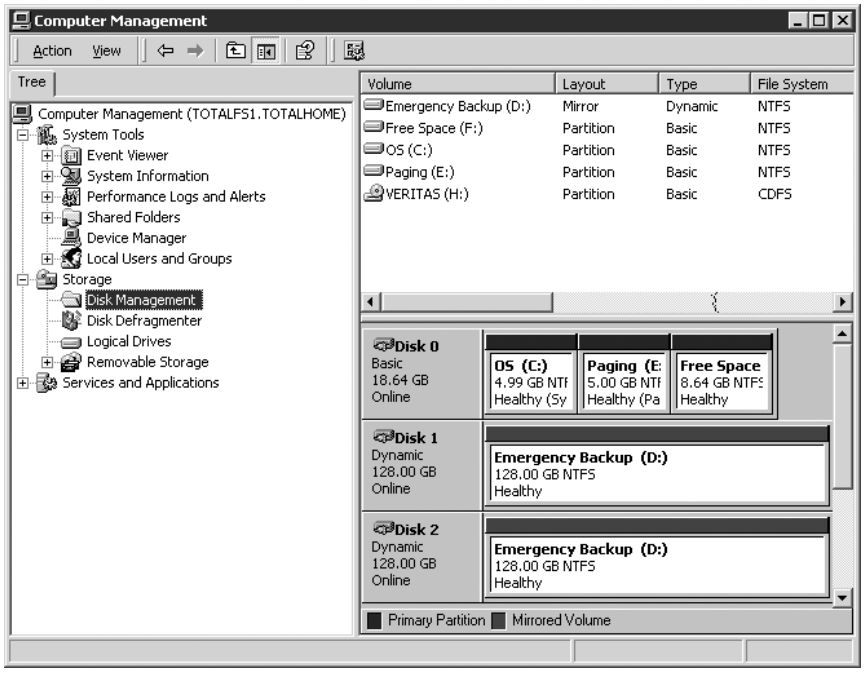


Figure 19-19 Disk Management at work

Most techs and administrators prefer hardware RAID. Software RAID works for small RAID solutions, but tends to run quite slowly and usually requires shutting down the PC to reconfigure and replace drives. When you *really* need to keep going, when you need RAID that doesn't even let the users know there was ever a problem, hardware RAID is the only answer. Because most organizations fit into this latter category, most RAID in the real world is hardware-based. There are a large number of hardware RAID solutions, and almost all these solutions rely on SCSI. SCSI can do one thing that PATA still cannot do—assuming that you have the right type of host adapter, you can yank a bad SCSI drive off of a SCSI chain and replace it with another one without even rebooting the server. This hot-swapping process is common in hardware RAID (see Figure 19-20). SATA, as noted earlier, can do hot swapping very nicely, thank you!

Figure 19-20 Hot swapping a drive

Okay, now that you have an idea of how to RAID, the next big question is "What do you want to RAID?" Granted, RAID 5 is popular but most techs when first exposed to RAID simply assume that they'll drop at least three drives into a server and make one big RAID 5 array. This solution will work, but the demands of the different types of data on a server often require a more refined and complicated approach.

One standard trick often performed with RAID is to separate the operating system itself from the data. The operating system files are neither unique nor do they change often compared to your data. If you lose the operating system you can simply reinstall it, assuming your server can afford to go down the amount of time necessary for reinstalling the operating system. In these cases, you put the operating system files on a non-RAID partition. If you want to get the operating system up more quickly, hold the operating system files on a mirrored partition. Most RAID mirroring solutions require an operating system reboot, but at least you'll be up in a minute or two as compared to the one hour (or more) rebuilding the operating system from scratch.

Another area to consider are swap files and temporary files. These files take up massive amounts of space and are useless if your system crashes. Many server admins place these files on a totally separate, non-RAID drive. There are exceptions to this, but those exceptions are usually operating system– or application-specific. One big exception is the "server that must never go down." In this case, the operating system, complete with the swap and temporary files, usually sits on its own separate RAID 5 or better array.

On most servers, the important data of your business has its own separate RAID 5 array. The low cost of today's RAID 5 solutions makes RAID almost a given on any server holding any data that's important to you or your organization.

RAID provides data redundancy. Implementing RAID requires that you decide the level of RAID you want to use and whether you want to go the hardware or software route. For the exam, make sure you can quote the different levels of RAID—and know your hard-drive connections. You'll fly through those questions without any difficulty!

NAS

If there's one thing no network ever seems to get enough of, it's space for file storage. For many years, the way we increased file storage space was to add more and larger capacity hard drives to our servers. This works well and is still a way to increase file storage space on many networks. But as networks grow, the burden of increased file *handling* begins to take its toll on the servers. This problem is exacerbated by the fact that most servers are already doing a lot of other jobs, such as name resolution, authentication, and e-mail serving—all of them critical jobs that we need servers to do to make our networks run. Over the years, I've seen a trend to spread these many jobs out to different servers. In my network, for example, I have one system that handles DNS, another that takes care of DHCP and WINS, and a third that handles authentication. However, all of these systems are also tasked with providing file sharing. What if we had a server that did absolutely nothing but file sharing?

This is one of many situations where *network attached storage (NAS)* is handy. NAS is a prebuilt system, usually running Linux with Samba and/or NFS, which you snap into your network to provide quick and easy storage with little or no setup involved. An NAS

is a server, but it doesn't come with all of the extras programs you'll find on most server systems. Instead, it's optimized to share folders or tape backups. A true NAS doesn't have a monitor, keyboard, or mouse. Configuration is handled through programs run from another system or a web interface. An NAS is usually much cheaper and much faster than a traditional server with the same storage capacity. Figure 19-21 shows a common brand of NAS, a Snap Server from the company Snap Appliance.

Most NAS devices have DHCP enabled and will run right out of the box. Even though they can run with no or little configuration, all NAS devices come with the capability to create security groups, user names, and passwords. It's common to keep a NAS in a Windows environment on its own domain—early NAS systems had to be on their own domain—but most will now join an existing domain or even an Active Directory.

The important issue to remember here is that an NAS is a standalone system running an operating system, usually Linux. It has a regular NIC and runs TCP/IP. The NAS server runs using either NFS or Samba to enable other systems to access its shared folders. This is important because NAS is often confused with something far more complex called a SAN.

SAN

A *storage area network (SAN)* system is a group of computers connected to an array of hard drives using an advanced serial technology such as SCSI fibre channel, a high-speed interface that functions similarly to SCSI interfaces. Designed for multiple-drive systems that can afford to have little or no downtime, Fibre Channel enables hot swapping of drives, RAID, and cable distances of up to 30 meters (!) between a device and the Host Bus Adapter (HBA). These capabilities (and many more), combined with data throughput speeds of up to 100 Mbps, put Fibre Channel in a class by itself.

All of the systems in the SAN may or may not have their own internal hard drives. In a SAN, each system connects to a Fibre Channel switch via a special NIC called a *host bus adapter (HBA)*. Figure 19-22 shows a Fibre Channel HBA. Note that the Fibre Channel HBA is virtually identical to a fiber-optic NIC.

The power of the SAN is in the disk array. One of the great aspects about Fibre Channel SCSI is that there is no practical limit to the number of drives in a single array. It's common to see a single Fibre Channel array with over one hundred drives. Figure 19-23 shows just such an array. This flexibility enables users of SANs to look at a single array as

Figure 19-21
Snap Server 2200

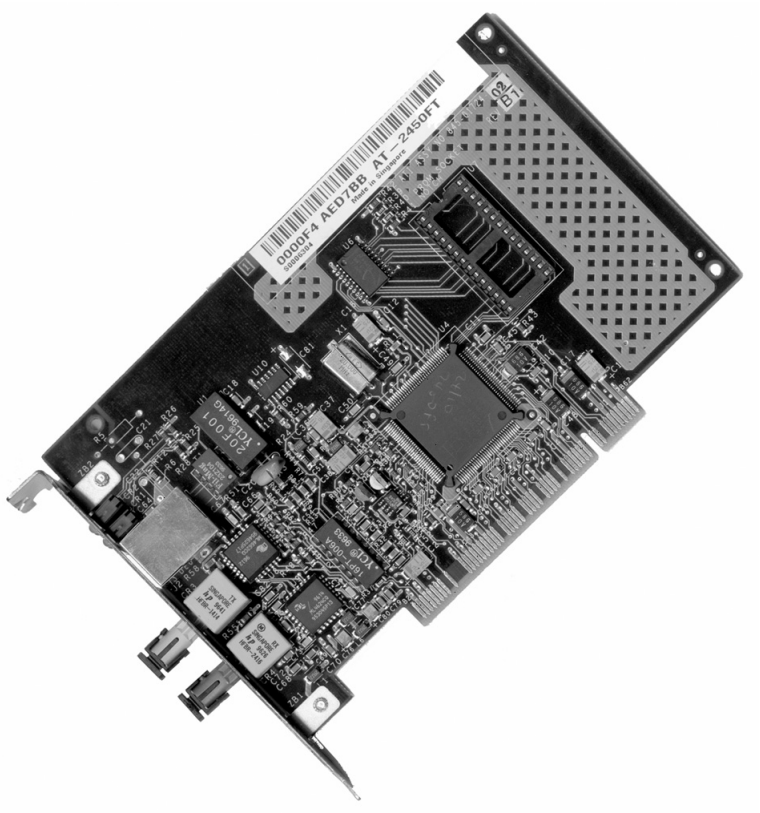


Figure 19-22 Fibre Channel HBA

one huge "glob" of hard drive that they can then take chunks out of and partition and format in any way they want. These partitions can be RAID or *Just a Bunch of Disks (JBOD)*. (No, I am not making this up! That is a real term!) Users can then attach or detach drives from their systems using the standard disk manipulation tools, such as Disk Management in Windows XP.

SANs are fast and can handle vast amounts of data, but they are also incredibly expensive. Odds are good that you could go your entire tech life and never see a SAN.

Tape Backup

Various RAID solutions provide data redundancy to a certain degree, but to secure your server data fully, nothing beats a tape backup. If the RAID solution works properly, that tape backup can happily collect dust on an offsite shelf somewhere. In the event of a catastrophe such as a hardware crash or a flood in the server room, only that tape can save the day.

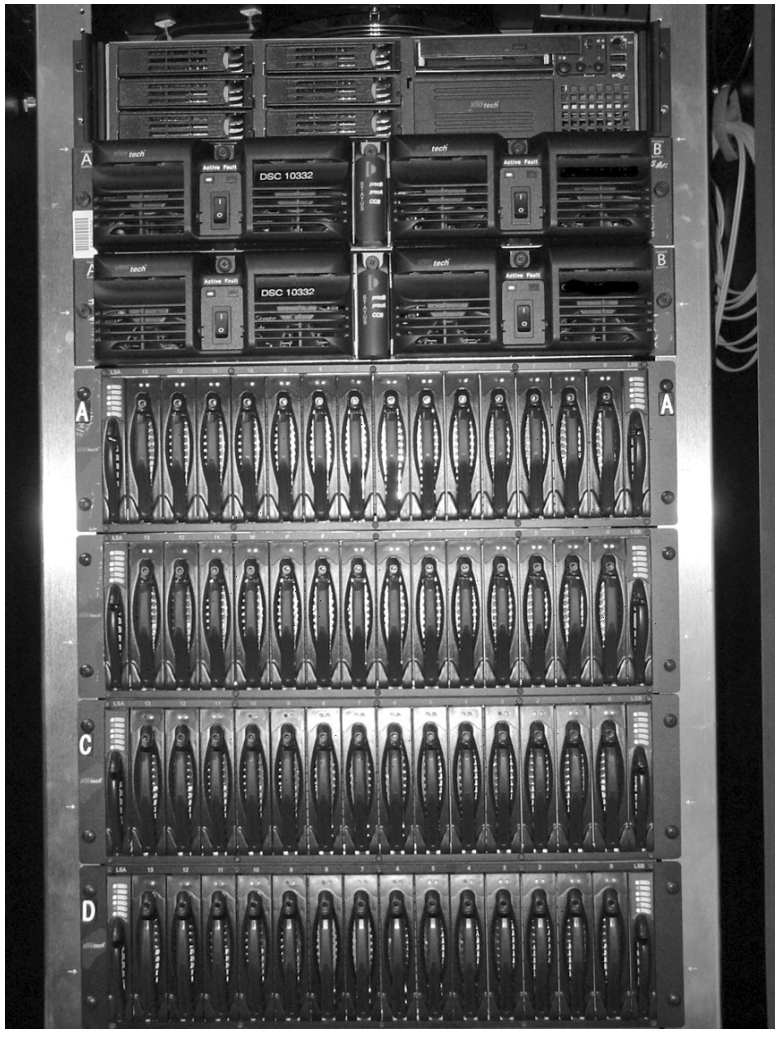


Figure 19-23
Fibre Channel SAN array

Magnetic tape is the oldest of all methods for storing data with computers. Who has not seen an episode of the old TV shows like "Time Tunnel" or "Voyage to the Bottom of the Sea" and watched the old reel-to-reel tapes spinning merrily in the background? The reel-to-reels are gone, replaced by hard drives; tapes are now relegated to the world of backup. Nothing can beat magnetic tape's capability to store phenomenal amounts of data cheaply and safely.

Every properly designed network uses a tape backup, so every network tech must learn to use them. The type of tape backup implemented varies from network to network, as do the methods for backing up data. This section covers the types of tape backup; refer to Chapter 20, "Zen and the Art of Network Support," for the methods.

There are a dizzying number of tape backup options, each with different advantages and disadvantages. They basically break down into three major groups: QIC, DAT, and DLT. All of the groups similarly use *cartridge tapes*—square tapes like fat audio cassettes—but the physical cartridge size, capacity, recording method, tape length, and speed vary enormously.

All tape backup solutions can back up data in compressed format. How much any data might compress varies on the type of data getting compressed. Tape manufacturers will advertise their capacities based on an assumption of 50 percent compression. When you see a tape that will store 30 GB, for example, that probably means it will store 15 GB of uncompressed data. Most manufacturers now advertise their capacities in both uncompressed and compressed values. A recent tape I purchased advertised itself as 10/20 GB—10 GB uncompressed and 20 compressed. Be advised, however, that the compressed value is just a guess! Without knowing the data that's being compressed, there's no way to know the compressed value!

QIC

Quarter-inch tape (QIC) is an old standard and rarely used in any but the smallest of networks. *QIC* was one of the first standards used for PC backups, but it has gone through many evolutions in an attempt to keep up with the demand for increased capacities over the years. The earliest versions of QIC could store about 40 megabytes—fine for the days when tiny hard drives were the rule, but unacceptable today. There have been a number of increases in QIC capacities, as high as two gigabytes, but QIC has fallen out as a desired tape standard. Imation Corporation created an improved QIC format called *Travan* that is quite popular, again on smaller networks, with capacities of up to 8 gigabytes. Under the Travan banner, QIC lives on as a tape backup option. Older QIC/Travan drives used a floppy connection, but EIDE or SCSI connections are more common today.

DAT

Digital audio tape (DAT) was the first tape system to use a totally digital recording method. *DAT* was originally designed to record digital audio and video, but it has easily moved into the tape-backup world. DAT tapes have much higher storage capacities than QIC/Travan tapes—up to 24 gigabytes—and are popular for medium-sized networks. DAT drives use a SCSI connection.

DLT

Digital linear tape (DLT) is quickly becoming the tape backup standard of choice. It's a relatively new standard that has massive data capacity (up to 70 gigabytes), is fast, incredibly reliable, and quite expensive compared to earlier technologies. When the data is critical, however, the price of the tape backup is considered insignificant. DLT drives use a SCSI connection.

Data Redundancy Is the Key

Data redundancy provides networks with one of the most important things they need—security. Improper preparation for the day a server hard drive dies leads to many quickly prepared résumés for the suddenly out-of-work network technician. When the data is

important enough (and when *isn't* it?), providing data redundancy via RAID solutions is required for the properly designed network.

Speed

A system providing a resource to a network has a tough job. It needs to be able to handle thousands, millions, even billions of transactions over the network to provide that shared resource to other systems. All of this work can bring a standard desktop PC to its knees. Anyone who has taken a regular desktop PC, shared a folder or a printer, and watched their PC act as though it just shifted into first gear can attest to the fact that sharing resources is a drain on a PC. Systems that share resources, and especially dedicated servers, require more powerful, faster hardware to be able to respond to the needs of the network.

There are a number of methods for making a serving system faster. Making a good server isn't just a matter of buying faster or multiple CPUs. You can't just dump in tons of the fastest RAM. Fast CPUs and RAM are important, but there are two other critical areas that tend to be ignored—a good server needs fast NICs and fast drives.

Fast NICs

The first place to look when you think of making a server faster is the NIC. Placing the same NIC in your server that you place in your workstations is like putting a garden hose on a fire hydrant—it just isn't designed to handle the job. There are a number of methods for making the NIC better suited to the task. You can increase the megabits (the data throughput), make the NIC smarter and pickier, and make it do more than one thing at a time. A lot of this was covered in detail in Chapter 6, "Modern Ethernet," so let's simply do the high points here.

Increase the Megabits

Most networks are a mix of 10BaseT, 100BaseT, and 1000BaseT. With autodetecting NICs and switches, different speed devices can communicate; sometimes a little bit of NIC organization or rearranging can work wonders to speed up a network—especially when it comes to your server. The trick is to have the server part of the network run at a faster speed than the rest of the network. If you have a 10BaseT network, you can purchase a switch that has a couple of 100 megabit ports. Put a 100BaseT NIC in the server and connect it to one of the 100BaseT connectors on the switch. The server runs at 100 Mbps while the workstations run at 10 Mbps (see Figure 19-24). This optimizes the server speed and, because the server does most of the work in the network, optimizes your network as well.

Smarter NICs

Many NICs still need the CPU to handle most of the network job, but several companies make powerful NICs with onboard processors that take most of the work away from the CPU. Every NIC manufacturer has a different method to provide this support and those methods are way outside the scope of this book. From a network person's standpoint,

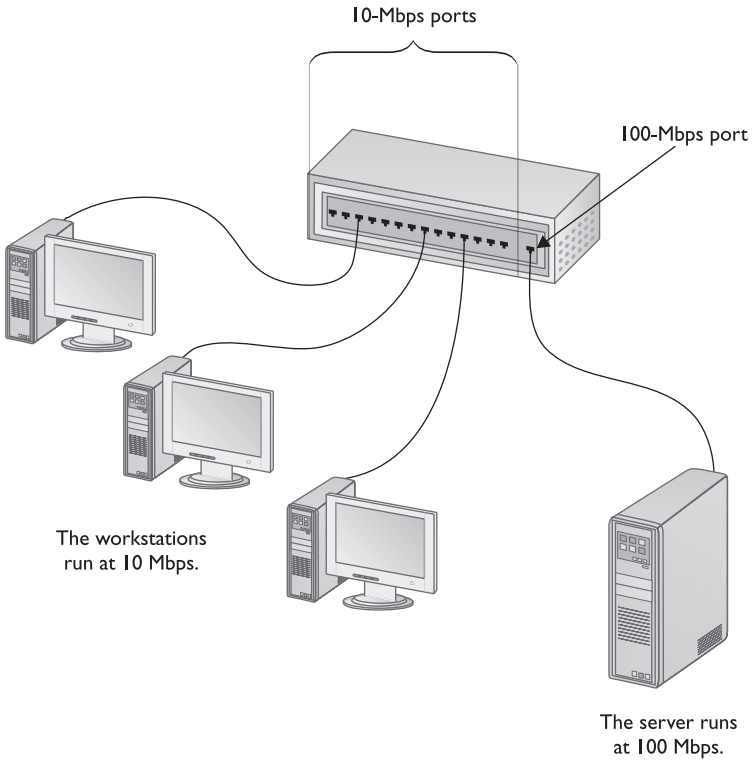


Figure 19-24 The server runs at 100 Mbps; workstations run at 10 Mbps.

just buy a special server NIC, plug that sucker in, and enjoy the benefits of faster response times.

Full-Duplex NICs

Most network technologies consist of send and receive wires, and most NICs can handle only sending or receiving at a given moment. Full-duplex NICs can both send and receive data at the same time, which practically doubles the speed of the network card. Make sure your server NICs are full-duplex, but be warned that you may need to upgrade the server's hub to take advantage of full-duplex!

Making the NIC better is one of the easiest upgrades to a server as it usually means simply yanking out an inferior NIC and replacing it with something better. At worst, you may have to replace a hub or a switch. Make your NIC better and you'll see the results.

Make the Drives Faster

The other big way to increase a server's speed is to make the process of getting the data to and from the shared drives faster. There are two big options here. First is to get fast drives. Using run-of-the-mill PATA drives in a busy serving system is not smart. Try using high-performance SCSI drives on a fast controller. It makes a big difference. Second, use RAID 5. Because you probably need it for data protection anyway, you'll also enjoy the speed.

It's Not Just Hardware

The demands of networking require servers to have better hardware than your run-of-the-mill, stand-alone PC. Improving CPUs, adding RAM, using powerful NICs, and running fast hard drives all work together to make your serving PC more powerful. But hardware is not the only answer. Good maintenance, such as defragging and setting up good disk caches, also plays an important role. Many times, slow resource access is due to poor network design and is not the fault of the serving system. Be careful about throwing hardware at the slow access issues; it can often be a big waste of money!

Reliability

The last network function, primarily for serving systems, is reliability. The shared resource must be there when the user needs it. *Reliability* is achieved by providing a secure environment for the server and by adding redundant hardware to compensate for failed components. There is a nasty tendency to mistake reliability for data protection. Don't confuse the two. All the pretty RAID systems aren't going to do you any good if somebody steals the server. Tape backups are useless if the power supply dies. Clearly, other technologies are needed to keep the serving system reliable. There is no logical order to explaining these technologies and safeguards, so we will cover them in no particular order.

NOTE The disaster recovery folks like to use terms like *hot spares* and *cold spares* when discussing any type of redundant equipment, but these terms work especially well for PCs. A hot spare is any redundant device that will instantly take over if the primary device fails. A cold spare is a redundant device that is onsite and ready to go, but is normally turned off.

Good Power

All of the components in the PC run on DC current electrical power. Without clean, steady, DC power, the components stop working. There are a number of steps that electrical power must take between the power company and those components. At any given moment, if one of those steps fails to do its part, the PC no longer works. You can take several actions to safeguard your hardware to make sure this doesn't happen, starting with the power company.

Electrical power in the United States is a wonderful commodity. Electrical service is pretty reliable, and the electricity is generally of high quality. Most folks in the United States can count on a good electrical service 98 percent of the time. It's that other 2 percent that will get you! Electrical power sometimes stops (power outages) and sometimes goes bad (electrical spikes and sags). Additionally, techs (and nontechs alike) can screw up perfectly good electricity on their own by overloading circuits with too much equipment. You can protect the servers from problems of power outages, electrical spikes, and overloaded circuits with several important technologies—dedicated circuits, surge suppressors, UPSes, and backup power.

Dedicated Circuits

A *dedicated circuit* is a circuit that runs from the breaker box to only certain outlets. In most homes and offices, a circuit might have many jobs. The circuit that runs your PC might also run the office water cooler and the big laser printer. Using too many devices on one circuit causes the power to sag, which might cause your computer to do nothing, lock up, or spontaneously reboot. It all depends on how lucky you are at that moment! Dedicated circuits keep this from happening. In most cases, dedicated circuits have outlets with bright orange faceplates to let you know that they are dedicated. This will (theoretically) prevent some uninformed person from plugging a photocopier into the circuit.

Surge Suppressors

It almost sounds silly to talk about suppressors these days, doesn't it? Does anyone really need to be convinced that all PCs, both network and stand-alone, need surge suppressors? An electrical surge—a sudden increase in the voltage on a circuit—can (and will) destroy an unprotected computer. Translation: every computer should plug into a surge suppressor!

UPS

An uninterruptible power supply (UPS) is standard equipment for servers. Any good UPS will also provide excellent surge suppression as well as support for power sags. Most only offer a few minutes of power, but it's enough to enable the server to shut down cleanly. All servers will have a UPS.

Backup Power

When you want serious reliability, get a backup power supply. Many server systems come with two power supplies. If either power supply fails, you can replace it without even turning off the system. But if the power from the power company goes out, you'll need a true backup system. There are a number of small battery-based backup systems that will provide a few hours of protection. If you want something that will last for a few days, however, you will need a gasoline/diesel backup system.

The Computer Virus

Ah, would that the only problem you faced was with faulty power. But alas, this is not the case. There are a large number of computer viruses and malicious code just waiting to infect your network. So what do you do when you think your computer has caught a code? In this chapter, you will find out.

The words "I think your machine has a virus" can send shudders down the back of even the most competent technician. The thought of megabytes of critical data being wiped away by the work of some evil programmer is at best annoying—and at worst a serious financial disaster.

So, where do viruses come from? Just like many human viruses, they live in host bodies—in this case, computers. Your computer can only catch one if it interacts with other computers, or with programs or data from an infected computer. Problem is, these days

almost everyone's computer (aside from folks like the CIA) is connected to the Internet, and thereby to many, many other computers. Also, many viruses are spread through the sharing of programs or information on floppy disks or CD-ROMs.

How do you know if you've caught a virus? You feel sluggish, start sneezing and coughing, want to sleep—or in this case, the computer equivalents of those symptoms might be as follows: your computer seems unusually sluggish, generates strange error messages or other odd emissions, or possibly even locks up and refuses to function entirely. All these are classic symptoms, but you cannot assume your computer is virus-free just because it seems fine. Some viruses do their work in secret, as we shall discuss shortly.

The secret to avoiding viruses is to understand how they work. A *virus* is a program that has two functions: (1) *proliferate* (make more copies of itself) and (2) *activate* (at some signal, count, date, and so on, do something—usually something bad like delete the boot sector). A virus does not have to do damage to be a virus. Some of the first viruses written were harmless and downright amusing. Without going into too much of the nitty-gritty, there are only five typical types of viruses—boot sector, executable, macro, worm, and Trojan—plus a sixth type that is a combination of any two other viruses—bimodal/bipartite.

Boot Sector

A *boot sector virus* changes the code in the master boot record (MBR) of the hard drive. Once the machine is booted, the viruses reside in memory, attempting to infect the MBRs of other drives by spreading themselves to removable media, connecting to network machines, and creating whatever havoc they are designed to do by the programmer.

Executable

An *executable virus* resides in executable files. These viruses are literally extensions of executables and are unable to exist by themselves. Once the infected executable file is run, the virus loads into memory, adding copies of itself to other EXEs that are subsequently run, and again doing whatever evil that the virus was designed to do.

Macro

A *macro virus* is a specially written application macro. Although they are not truly programs, they perform the same functions as regular viruses. These viruses will autostart when the particular application is run and will then attempt to make more copies of themselves—some will even try to find other copies of the same application across a network to propagate.

Trojan

Trojans are true, freestanding programs that do something other than what the person who runs the program thinks they will do. An example of a *Trojan virus* would be a program that a person thinks is a game but that is a CMOS eraser. Some Trojans are quite sophisticated. It might be a game that works perfectly well, but when the user quits the game, it causes some type of damage.

Bimodal/Bipartite

A *bimodal or bipartite virus* uses both boot-sector and executable functions.

Worm

A *worm* is a network-aware virus that spreads through applications such as e-mail and web browsers. *E-mail worms* are currently the greatest single virus threat. These worms propagate by reading e-mail address books and sending copies of themselves to everybody. Most will mask their origin by using a false e-mail address as the sender.

Antivirus Programs

The only way to protect your PC permanently from getting a virus is to disconnect from the Internet and never permit any potentially infected software to touch your precious computer. Because neither scenario is likely these days, you need to use a specialized antivirus program to help stave off the inevitable virus assaults.

An antivirus program protects your PC in two ways. It can be both sword and shield, working in an active seek-and-destroy mode and in a passive sentry mode. When ordered to seek and destroy, the program will scan the computer's boot sector and files for viruses, and if it finds any, present you with the available options for removing or disabling them. Antivirus programs can also operate as virus shields that passively monitor your computer's activity, checking for viruses only when certain events occur, such as a program executing or a file being downloaded.

Antivirus programs use different techniques to combat different types of viruses. They detect boot-sector viruses simply by comparing the drive's boot sector to a standard boot sector. This works because most boot sectors are basically the same. Some antivirus programs make a backup copy of the boot sector. If they detect a virus, the programs will use that backup copy to replace the infected boot sector. Executable viruses are a little more difficult to find because they can be on any file in the drive. To detect executable viruses, the antivirus program uses a library of signatures. A *signature* is a code pattern of a known virus. The antivirus program compares an executable file to its library of signatures. There have been instances where a perfectly clean program coincidentally held a virus signature. Usually the antivirus program's creator will provide a patch to prevent further alarms. Antivirus programs detect macro viruses through the presence of virus signatures or of certain macro commands that indicate a known macro virus. Now that we understand the types of viruses and how antivirus programs try to protect against them, let's review a few terms that are often used when describing certain traits of viruses.

Polymorphics/Polymorphs

A *polymorph virus* attempts to change its signature to prevent detection by antivirus programs, usually by continually scrambling a bit of useless code. Fortunately, the scrambling code itself can be identified and used as the signature—once the antivirus makers become aware of the virus. One technique sometimes used to combat unknown polymorphs is to have the antivirus program create a checksum on every file in the drive. A *checksum* in this context is a number generated by the software based on the contents of

the file rather than the name, date, or size of that file. The algorithms for creating these checksums vary among different antivirus programs (they are also usually kept secret to help prevent virus makers from coming up with ways to beat them). Every time a program is run, the antivirus program calculates a new checksum and compares it with the earlier calculation. If the checksums are different, it is a sure sign of a virus.

Stealth

The term "stealth" is more of a concept than an actual virus function. Most *stealth virus* programs are boot sector viruses that use various methods to hide from antivirus software. One popular stealth virus will hook on to a little-known but often-used software interrupt, running only when that interrupt runs. Others make copies of innocent-looking files.

Virus Prevention Tips

The secret to preventing damage from a virus attack is to keep from getting one in the first place. As discussed earlier, all good antivirus programs include a virus shield that will automatically scan floppies, downloads, and so on (see Figure 19-25).

Use your antivirus shield. It is also a good idea to scan a PC daily for possible virus attacks. All antivirus programs include terminate-and-stay resident programs (TSRs) that will run every time the PC is booted. Last but not least, know where software has come from before you load it. While the chance of commercial, shrink-wrapped software having a virus is virtually nil (there have been a couple of well-publicized exceptions), that illegal copy of Unreal Tournament you borrowed from a local hacker should definitely be inspected with care.

Get into the habit of keeping around an antivirus floppy disk—a bootable, write-protected floppy with a copy of an antivirus program. If you suspect a virus, use the diskette, even if your antivirus program claims to have eliminated it. Turn off the PC and reboot it from the antivirus diskette. Run your antivirus program's most comprehensive virus scan. Then check all removable media that were exposed to the system, and any other machine that may have received data from it, or that is networked to the cleaned machine. A virus can often go for months before anyone knows of its presence.

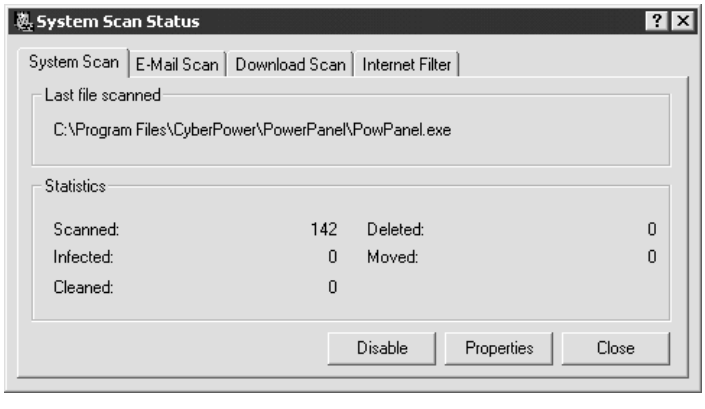


Figure 19-25 A virus shield in action

Environment

Keep the server room locked at all times. Get a card lock or a combination lock doorknob and make sure that only the right people have access. Keep the humidity low, but not too low—around 40 percent is about right for most electronics. Keep the room a little on the cool side—right around 68 degrees is just about perfect, although most PCs can handle about 80 to 85 degrees before overheating becomes a problem. Check with the system's manufacturer for their recommendations.

Redundant Components

Many components inside the system can be made redundant. It is common to find servers with redundant power supplies where a power supply can be removed without even shutting down the PC. You can buy NICs that work together in the same PC, covering for one or the other if one dies—there are even NICs that can be replaced without rebooting the PC! Placing hard drives on separate controllers—like the drive duplexing discussed earlier in this chapter—provides excellent redundancy.

Last, there are methods for making the entire server redundant. For instance, there are a number of methods where two or more servers can be mirrored, providing the ultimate in reliability (assuming the cost is bearable)!

How Much Reliability Do You Need?

Reliability is like any security system—expensive, boring, a pain to administer, and you never have enough when you need it. Measure the cost of being down vs. the cost of reliability to make your decision. You might be surprised to find that it's a lot cheaper to be safe than sorry.

Putting Them All Together

Now that you've got a grasp on the many hardware and software features that make up a server, what's the right server for your needs? As much as I'd love to give you a checklist of every possible issue and the right type of hardware or software to use to deal with that issue, the complexities and continuing new features of networking make such a checklist impossible to create. But that doesn't leave you to nothing but a guess. Here are a few issues to consider and some guidelines to help you get the server you need to your network.

Function

What is this server going to do? Understanding the function of a server is the first step toward defining the hardware and software it needs. Is this server only going to support a small in-house web server or is this going to be a file server supporting a massive database? Equally important to the function is the number of systems accessing that server. How many users will connect to that server at a time? What type of data will they be requesting? By understanding the server's function, you can get the following questions answered.

Fault Tolerance

Servers with complete fault tolerance—the "server that can never go down"—are expensive. Certainly, there are organizations that need this level of fault tolerance but that doesn't mean every server needs dual power supplies, hot-swappable RAID 5, and re-

dundant NICs. Look at the data and visualize the effect on your organization if the server were to go down. Ask others in the organization and make your judgment.

On the same token, don't go cheap on the basics. There's no excuse for a server not to use a good UPS with plenty of standby power.

CPU/RAM

Should you get a server with a single AMD Athlon XP or go for the one with the dual Itaniums? Should you go with 1 GB of RAM or 16 GB? Choosing the CPU and RAM depends on the serving applications of your server and the number of users accessing that system. Check the web sites of whoever makes the server applications—they all provide guidance on how much RAM their applications need.

Speaking of applications, dual CPUs are useless unless you have applications that take advantage of them. Many server applications do take advantage of multiple processors but not all do so. Again, check with the application maker before spending the big cash on multiprocessor systems!

Scalability

Sure, the server may work well now, but what happens as needs expand? Can you add more RAM? Can you add drives? Don't limit yourself to the single server—could you add another server to take away some of the workload as needs increase?

Nothing's Perfect

There is no such thing as the *perfect* server. Certainly every network PC needs to connect to the network, but data protection, speed, and reliability are functions that vary tremendously depending on network size, types of data and applications, the existing network cabling system, demands of growth, and of course, your pocketbook. The Network+ exam does not assume you can build the perfect network PC, but it does expect you to have a feel for the options you have. When it comes time to build or buy that system, you can act as an advocate for your network, to ensure that you get as close to that perfect network PC as possible.

Chapter Review

Questions

1. Of the following, which is the most important to consider when determining the amount of RAM to install in a server?

   A. The NIC

   B. The network protocol

   C. The applications

   D. The Service Pack

2. A computer virus can be categorized as which of the following? (Select all that apply.)
   A. Always destructive
   B. Self-replicating
   C. Self-activating
   D. Self-destructive
3. Which of the following is an improvement of SATA over PATA?
   A. Three drives per cable
   B. Maximum of 16 drives on a system
   C. No more master/slave jumpers
   D. Drives spin more slowly
4. Which is the most common RAID implementation on servers?
   A. RAID 0
   B. RAID 1
   C. RAID 3
   D. RAID 5
5. How many drives attach to a single PATA cable?
   A. 1
   B. 2
   C. 4
   D. Unlimited
6. If Janet mirrors two 100-GB drives, what will be her effective total capacity?
   A. 100 GB
   B. 166 GB
   C. 200 GB
   D. Depends on the RAID level
7. Which term describes the practice of using multiple controllers when mirroring?
   A. Multiplexing
   B. Distributing
   C. Duplexing
   D. Omniplexing
8. Which of the following terms defines a storage technology?
   A. PCI
   B. AGP

C. MATA

D. SATA

9. Disk mirroring is under which level of RAID?

A. RAID 0

B. RAID 1

C. RAID 2

D. RAID 3

10. Which of the following connectors are used with SCSI? (Select all that apply.)

A. 50-pin Centronics

B. 36-pin Centronics

C. Female DB-15

D. Female DB-25

Answers

1. **C.** The applications running on the server are the most important consideration.
2. **B, C.** For a program to be considered a virus, it must be self-replicating and self-activating.
3. **C.** SATA eliminates the need for master/slave jumpers.
4. **D.** RAID 5 is the most common RAID implementation on servers.
5. **B.** PATA allows for a maximum of two drives per cable.
6. **A.** Each drive stores the same data, making an effective storage capacity of 100 GB.
7. **C.** Duplexing is the mirroring method in which each mirrored drive uses its own controller.
8. **D.** Of the answers listed, only SATA is a storage technology.
9. **B.** Disk mirroring is under RAID 1.
10. **A, D.** Both 50-pin Centronics and female DB-25 connectors are used with SCSI.