CHAPTER 2

Preparing a Virtual Machine Host

In Chapter 1, we visited important issues surrounding virtualization technology, including hardware, software, virtualization theory, and manufacturer nomenclature. This chapter will address the planning issues for selecting and preparing a host system to support virtual machines. You'll need to confirm that your host computer meets the VM application's minimum hardware requirements, and you must verify that it has the available resources to support the number of VMs you plan to run simultaneously. You'll look at preparation issues for both Windows and Linux hosts. Several factors determine the readiness of a computer to support a VM application, such as the motherboard, RAM type, CPU speed, hard disk type, and network adapter speed. Using best-practice principles, we'll cover how to quantify host requirements that will result in better stability, scalability, and performance for your guest virtual machines.

Implementing Best Practices

You'd think with RAM, processing power, and storage space being so inexpensive it'd be fairly easy to achieve some type of best-practice hardware implementation in today's modern IT infrastructure. Unfortunately, this has never been the case. Budgets, personal preference, and "we always do it this way" all take a toll on the idea of a best practice. Though some may say a best practice is what works for you, don't buy into that definition. A best practice is always the high road—the road less traveled. A best practice is a radical deviation from "if it ain't broke, don't fix it." In general, best practices describe the activities and procedures that create outstanding results in a given situation and can be efficiently and effectively adapted to another situation. For instance, for us IT people, this means if studies and white papers find that commercial-grade servers have less downtime and problems than white boxes, we should quit building our own servers for the enterprise infrastructure. If studies and reports prove that virtualization truly saves money, decreases administrative overhead, and better supports disaster recovery strategies, we should implement it.

In test and educational environments, using the minimum available resources to learn how to use VM applications, such as Microsoft's Virtual PC and VMware's Workstation, is okay. In these environments, the luxury of time is on your side, and occasionally rebooting is no big deal. In a production environment, though, end users don't have time to wait, and rebooting any major system can mean the loss of real money. Given this, take a moment to look at best-practice hardware requirements for each VM application before deploying VMs in your

infrastructure. Take the time to check your hardware against each manufacturer's HCL, and compare your hardware to the listed best practices in each section. Don't be a victim: spend a little time now checking out your deployment hardware; it will save you late nights and weekends later. Simply loading Microsoft's Virtual Server or VMware's GSX Server on an existing system and "letting it eat" is a recipe for disaster.

Evaluating Host Requirements

When it comes to hosting VMs, bigger is always better. However, stuffing quad processors and a Fibre Channel SAN into a notebook isn't an option either. (We'll give you an in-depth look at VMs and SAN interactivity in Chapter 13, but we'll touch on the basics in this chapter.) Now, let's take a moment to put some perspective on the saying "bigger is always better." Host hardware—whether laptops, workstations, or servers—all perform different functions and will have very real and different needs, so focus your budget on where it will produce the most overall good. You can avoid bottlenecks by investing equally in a host's three major systems:

- **Storage systems**: Hard drives, SANs, CD/DVD-ROMs, RAM, and cache
- **Networking systems**: NICs, switches, and routers
- **Processing systems**: CPU and front-side bus (FSB)

When sizing a computer to host VMs, you can easily pinpoint areas of concern by reflecting on the system's purpose and keeping the three major subsystems in mind. For instance, laptops are great because they can be carried everywhere and utilized in sales presentations or classrooms. For the convenience of shoving a mainframe system into the space of a book, you can make performance and hardware longevity sacrifices. Because the space constraints put on portable computers cause them to overheat, manufacturers use lower-spindle speed hard disks, variable-speed processors, and smaller memory capacities. In this case, you'd want to spend money on adding as much RAM as possible to your system, followed by as much processing power as possible.

Workstations, on the other hand, have the dubious honor of being shin-bangers. The purpose of a workstation has stayed consistent over time, and you know they provide fast, reliable access to centralized network resources. Workstations employ higher-speed disk drives, larger quantities of RAM, faster processors, and good cooling systems. To improve VM workstation hosting, you'll want to add additional disk drives and controllers, followed by more RAM quantities rather than add a second processor. Lastly, add a good, fast NIC. By placing the host OS on one bus and hard drive, and the guest OSs on a second bus and hard drive, you more closely approximate the structure of two computers in one physical box; this creates a much better host.

Servers are different from laptops and workstations in that users will be depending on VMs: accessibility and availability will be your highest priority. You'll want to use Redundant Array of Inexpensive Disks (RAID)–configured hard drives (SAN if possible), several-gigabit NICs in a teamed configuration, multiple processors, and maximum RAM quantities. In case your mind is already racing ahead with *what ifs* (and to feed the supergeek in us), we'll cover more closely the mathematic justification for using multiple controllers and disks in the "Considering Storage Options" section toward the end of this chapter. We'll also discuss hard disk and RAID selection.

Table 2-1 is designed to help you begin prioritizing your VM hosting needs. Keep your eye on what type of VM host is being built, and then make decisions that will adequately support the host in its final environment. Table 2-1 summarizes our recommended priorities, but you can prioritize resources according to your needs.

Table 2-1. *Comparing VM Host System Priorities (1 = High, 6 = Low)*

Resource	Laptop	Workstation	Server
Multibus disks		1	2
RAID/SAN			1
RAM	1	2	5
Processing speed	2	3	6
Multiple processors		5	4
Networking capacity	3	4	3

When prioritizing investment and resource allocation for VM hosts, you want to determine what's going to be the subsystem bottleneck and mitigate its impact on performance by adding bigger, better, and faster technology. Bottlenecks will manifest based on the host's purpose:

- Will it be a portable or fixed system?
- Are resources local or available across networks?
- Is the host serving one person or one thousand?

Later in the "Considering Your Network" and "Considering Storage Options" sections, we'll further discuss the impacts of direct-attached storage, SANs, teaming, and load balancing.

All the best-practice suggestions for sizing host systems will do you absolutely no good without applying a good dose of common sense and experience—reference manufacturer OS system requirements, and check hardware compatibility. This is particularly true with regard to memory quantities and disk space. On any given server, the host and each VM both require sufficient memory for the tasks each will perform. You can calculate your total memory needs by simply calculating the number of VMs to be hosted, adding one for the host, and multiplying the sum by no less than the OS's minimum memory requirement. For example, if you plan to host three Windows 2003 Web servers on one piece of hardware, you need enough RAM to support four servers. But what version of Windows 2003 are you using? Windows 2003 has different minimum and maximum supported memory configurations, as listed in Table 2-2.

Table 2-2. *Windows 2003 Server RAM Requirements*

Memory Requirement	Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition
Minimum	128MB	128MB	128MB	512MB
Recommended	256MB	256MB	256MB	1GB
Maximum	2GB	4GB	32GB/64GB Itanium	64GB/512GB Itanium

Assuming the host is using Windows 2003 Standard, you'll want to install 2GB of RAM (4 servers × 512MB of RAM = 2GB). We'd also like to point out that if you're hosting VMs on Windows 2003 Standard with Virtual Server or GSX Server, you'll be limited to a total of 4GB of RAM for the host and its guest VMs. If you plan on taking full advantage of larger memory pools, you'll want to move up to VMware's ESX Server or Windows 2003 Enterprise.

Determining required disk space is similar to sizing memory but with a few added gotchas. You'll need to include hard disk space for each guest's operating system footprint, for memory swap files, for dynamic disks, and for fixed disk storage. You'll also need to include it for suspending running VMs (equivalent to the RAM allocated to the VM). For a disk-sizing example, let's use the three previously mentioned Web servers configured with fixed disks. Table 2-3 lists the Windows 2003 install footprint for each edition.

Table 2-3. *Windows 2003 Disk Space Setup Requirements*

Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition
1.5GB	1.5GB	1.5GB/2GB Itanium	1.5GB/2GB Itanium

You'd need enough disk space to adequately cover four server OS installations (1.5GB × 4 = 6GB), room for four swap files (1GB × 4 = 4GB), fixed disk data storage for four servers (4 × ?), and room for three guest VMs to be suspended (512MB × 3 = 1.5GB). At the moment, you'd need a minimum of 11.5GB just to get the servers loaded and running. The one variable that may be difficult to contend with is the space for data storage: being that needs vary, you'll need to rely on your experience and common sense for fixed disk space sizing. Assume that the servers in this example will store graphics-intensive sites and need to house documents for downloading (3 × 40GB = 120GB). You'll set aside a total of 8GB for the host and its applications. After calculating fixed storage and adding the install, suspend, and swap requirements, your host will need approximately a minimum 140GB of storage. If you're implementing an IDE solution, mirrored 160GB hard drives will be adequate. A SCSI RAID 5 solution can squeeze by with three 80GB hard drives. If you think any of your VMs may run out of space, you'll want to consider adding hard disk space so you can create additional fixed or dynamic disks in the future.

It's impossible to consider every scenario in which VMs will be installed. Therefore, you'll have to rely on the manufacturer's minimum requirements, on postings from Internet forums, on trusted colleagues, and on your experience. In Chapters 3 and 5, where we cover installing virtualization applications, we'll suggest best-practice minimums. The best-practice minimums offered are merely a guide to point you in the correct direction for sizing a VM host system.

Selecting a Motherboard

The motherboard, the centralized circuit board all other devices and chipsets connect to, interacts with every subsystem of a computer. It's largely responsible for an entire computer's performance, stability, and scalability. Choosing a motherboard to use for your VM host may be less of an issue if you're purchasing a system from a major vendor, such as Dell, Sony, or Hewlett-Packard. However, you'll need to make sure the motherboard supplied with a

proprietary system supports your intended purpose and will scale as needed. Selecting the correct motherboard will go a long way to ensuring that your VM host performs optimally. Additionally, the right motherboard may reduce some of the cost involved by including integrated NICs, hard drive controllers, video adapters, and sound cards.

Motherboards can be complicated and conceivably the most important part of any computer. A plethora of motherboard manufacturers exist, so carefully consider your available options and narrow the field of choices by determining if a particular motherboard supports the correct memory, CPU, I/O device options, and reliability constraints you need. You can buy motherboards for less than $20 and spend as much as several thousand. You don't have to know the intricacies of every chipset and processor made, but you should know enough to make sure your requirements and expectations are satisfied.

Good performance levels are critical for ensuring that data processing takes place in a reasonable amount of time. But what's performance without stability and reliability? You'll want to stick with motherboard manufacturers that consistently produce quality merchandise over time and continue to support EOF products: you'll eventually need to update a BIOS or driver to fix a bug or to upgrade when a new OS or system patch is released. Because motherboard drivers are provided in part by the chip manufacturer and the motherboard producer, it's critical you pick a well-known and respected brand to get the most out of available support—Microsoft doesn't build every driver in its operating systems. Because reputable motherboard manufacturers will have already taken the time to thoroughly test their motherboards for stability and reliability over a sizable chunk of time, using respected brands will save you many hours of research and testing time.

Determining hardware compatibility for motherboards can be difficult because of the open nature of computers. Peripheral components that have been in use for longer periods of time in the field are more likely to be supported by reputable manufacturers, such as Adaptec, Intel, QLogic, ATI, and 3Com. If you need a particular device to function with a new motherboard, you'll need to check and verify that the manufacturer has tested and supports the device. Moreover, read the fine print in the warranty. Some manufacturers may void your hardware warranty for not using proprietary or approved hardware.

So, given the preceding rhetoric on buying a quality motherboard, what should you have on your checklist to help narrow the field to create the best possible foundation for your VM host? During the motherboard selection process, you'll want to specifically look at the following:

- CPU speed and quantity
- Controller chipset
- Memory requirements
- Bus types
- Integrated devices
- Board form factor
- Overall quality

We'll cover each of these in the following sections.

CPU Speed and Quantity

The speed and quantity of processors will significantly impact the overall performance of your host. If you're building a production server, you'll definitely want to use multiple processors. Microsoft virtualization applications don't support SMP for guest VMs, but the host can take full advantage of multiple processors (assuming you're using a multiprocessor OS). VMware is similar in that it doesn't support SMP for guest VMs running on Workstation and GSX Server. Multiprocessor support is limited to the host machine's OS. Where VMware differs from Microsoft is that ESX Server supports SMP for guest VMs with SMP licensing. For instance, if your host server has sixteen processors, you can configure four guest VMs, each having four processors.

Purchasing multiprocessor motherboards can be expensive: add the cost of each processor, and you could be in the range of an unrealistic budget. When contemplating using multiple processors, you can keep the lid on runaway costs by avoiding purchasing the fastest processors. There's little difference in performance between 2.8 gigahertz (GHz) and 3.2GHz. But there's a huge difference between 1.8GHz and 2.8GHz. By selecting processors that are two or three steps down from the fastest available, you can keep your system selection within reason. Also, you'll need to keep in mind that multiple processors don't scale linearly. That is, two 3GHz processors aren't twice as fast as a single 3GHz processor. You'll get diminishing returns for each added CPU and each increment in speed. Rein in the desire to go for the fastest possible CPU because there's more to a fast host than clock speed. Additional information you'll want to look for in the motherboard's documentation is compatibility information, such as what processors are supported with the chipset or the requirement of having to implement identical CPUs using the same stepping level.

■ **Caution** Running multiprocessor servers with CPUs that have different stepping levels can and often does create squirrelly computers.

When deciding which CPU to purchase, you'll also need to consider cache memory. Without cache memory, CPU data requests would all have to be sent over the system bus to the memory modules. To avoid this bottleneck and achieve higher performance levels, CPU manufacturers incorporate cache into their processors. Additionally, motherboard manufacturers will incorporate cache into the motherboard. You'll see cache memory listed as Level 1, Level 2, and Level 3. Cache is fast and generally runs at the same frequency of the processor or system bus. Manufacturers are able to produce inexpensive chips by cutting cache levels. Conversely, manufacturers produce high-end chips by increasing cache levels. In general, the more cache made available to the processor, the faster everything will run. You can find Intel processors with cache levels as high as 4MB and as little as 128KB. If you stick with moderate cache levels for production servers, 1–2MB, your guest VMs will be rewarded with significantly better performance. If you want in-depth explanations of processor cache types, you'll want to visit your CPU manufacturer's Web site.

Controller Chipset

The chipset dictates the overall character of the motherboard; it's fundamentally responsible for determining which CPU, memory, and peripheral types are supported. For instance, Intel, Via, and Ali currently produce the majority of chipsets used, and each vendor typically is geared to

support specific processor types, such as Intel or AMD. Communication between the motherboard's components is specifically controlled by the north and south bridge chips. The north bridge chip is generally located near the CPU socket and interconnects the processor, memory, video bus, and south bridge chip. The south bridge chip facilitates communication between the north bridge, peripheral cards, and integrated devices. With so much riding on one chip, the north bridge, it should come as no surprise that the speed at which the FSB functions significantly impacts the host's performance. When it comes to chipsets, you'll want to specifically make sure your CPU selection is supported (including the type and speed) by the motherboard's FSB. You'll also want to look at the motherboard's specifications to see which memory type is supported. Fast processors and RAM require a fast FSB, and a mismatch can create a bottleneck. If you're comparing off-the-shelf systems, you'll want to select a motherboard capable of supporting a processor requiring FSB speeds of 400MHz or greater for production VM hosts. If you're interested in knowing which Intel processors support which bus speeds, the site at `http://processorfinder.intel.com` provides a list of CPUs and required bus speeds.

Memory Requirements

Despite your desires, the memory you choose for your system will be limited to what your motherboard is capable of handling. If you have an idea of the type of memory you want to use and the quantity, you must first make sure the motherboard supports the total memory requirement for your host and guest VMs. Second, you must determine the number of available memory slots, the type of memory supported, and the memory module size supported. Manufacturers are specific about which type of memory can be used in any given motherboard. In large part, the type of memory available for you to choose will be tied to what the CPU and chipset can support. Your motherboard will probably support several memory technology types—double data rate, second generation (DDR2); synchronous dynamic random access memory (SDRAM); and DDR SDRAM. If your system mentions any of the older types of memory—such as single data rate (SDR) SDRAM, fast page mode (FPM), or extended data out (EDO)—choose another motherboard.

When looking at the performance characteristics of RAM, expect to pay more for motherboards using faster types of RAM. When deciding which type of RAM to use, speed isn't as important as quantity in regard to networked VMs. The bottleneck will be the network and not the read/write rate of your RAM. Table 2-4 lists performance data for several memory types.

Table 2-4. *Memory Speeds*

Technology	Speed	Bandwidth
DDR2	PC2-6400	6.4GB/sec
DDR2	PC2-5300	5.3GB/sec
DDR2	PC2-4200	4.2GB/sec
DDR2	PC2-3200	3.2GB/sec
DDR	PC4000	4.0GB/sec
DDR	PC3200	3.2GB/sec
DDR	PC2700	2.7GB/sec
DDR	PC2100	2.1GB/sec
DDR	PC1600	1GB/sec

If you plan on running VMs in RAM only, search for motherboards supporting faster RAM, such as DDR2. Faster RAM will give your VM guests a little extra oomph. Being that your system can be only as fast as the slowest link, you'll probably want to shoot up the middle when it comes to speed and quantity for your host motherboard. That is, there's no point in paying for a motherboard supporting bleeding-edge memory speeds if you're trying to build a server with a single SCSI adapter and hard drive for the host and guest VMs to share. You're better off buying more SCSI controllers, hard drives, and RAM.

When culling the field of motherboard candidates, stick with the boards that come with dual-channel technology. Dual-channel systems allow the bandwidth of two memory modules to be used simultaneously. In cases where dual-channel technology is implemented, you're better off adding memory in pairs to get that added performance boost. For example, if you need 1GB of RAM, get two 512MB modules rather than one large one—you want to make that memory highway as wide as possible.

■ **Caution** Don't get burned by using cheap or generic memory. In the long run, it just isn't worth the headache that junk memory causes. For a few extra bucks up front, you can purchase quality memory products from major memory manufacturers such as Crucial, Viking, and PNY. Quality memory will more than pay for itself over time because it will keep you from troubleshooting the bizarre problems cheap memory causes. Also, you'll find that the many discount vendors peddling heavily discounted memory will offer a "lifetime warranty" with no intent of honoring it.

As a final consideration, you may want to install error correcting code (ECC) RAM in hosts supporting production environments with mission-critical applications. ECC RAM utilizes an extra chip that detects if data is correctly read from or written to the memory module. In many cases, and depending on the type of error, ECC RAM can correct the error. Having the ability to detect and correct errors means a server is less likely to crash, but you'll take a small performance hit by implementing ECC memory.

Bus Types

The availability and quantity of Industry Standard Architecture (ISA), PCI, PCI Extended (PCI-X), and Accelerated Graphics Port (AGP) bus types are extremely important when it comes to hosting VMs. Small motherboards don't offer many expansion slots, and few come with ISA capability. ISA expansion slots are an issue if you need to reuse expensive ISA devices, such as multiport modems. Multiple AGP slots are important if your system requires high-performance video and multiple monitors. Multiple monitors are extremely useful for managing a VM host and guest in full-screen view without the aid of a KVM. A motherboard having only two or three PCI expansion slots limits your ability to correctly equip your servers with sufficient hardware for hosting. For instance, if you need two RAID controllers and three network adapters, you'll need five PCI slots on your motherboard. If five slots aren't available, you'll have to depend on integrated devices or expect less in the performance department. PCI-X is an enhancement over the traditional PCI bus in that the speed of the PCI bus is increased from 133MB/sec to a whopping 1GB/sec. In addition, PCI-X is backward compatible with traditional PCI adapter cards running at the lower speeds. PCI-X was designed to provide the higher performance levels Gigabit Ethernet and Fibre Channel technology demanded. For VM hosts, including PCI-X

technology substantially increases performance and is something you'll need to consider in environments with high network utilization or where SAN interactivity is required.

You'll need to plan for the maximum number of expansion cards your server requires in order to correctly size a motherboard. If you find a motherboard that falls short on available and necessary slots, you'll need to turn to using expansion cards that offer increased capability. For instance, you can use multiport NICs and multihead video adapters in situations where slots are limited. Additionally, you can use integrated AGP video in situations where more PCI expansion slots are a priority. In the end, you need to know how many NICs, SCSI, RAID, and video adapters your host will require.

Integrated Devices

Integrated peripherals can seriously cut the cost of a system. You can safely run a computer without a single expansion card, but you won't generally get the same performance of a system using some or all expansion cards. Typically speaking, integrated video, networking, audio, USB, and hard drive controllers work well for most systems. As important as it is to have integrated devices, it's equally as important to be able to disable them in case of failure or poor performance. You'll find systems that share installed RAM. You'll want to avoid this if possible because the integrated devices will be taking away from what's available for the host and guest VMs. A few megabytes chewed up for video here and a few megabytes for a caching disk controller there can equal one less hosted VM. RAM is cheap, and good-quality manufacturers don't hesitate to provide the appropriate resources for integrated devices. Where integrated devices can really pay off for virtualization is with integrated RAID. Integrated RAID normally doesn't have huge amounts of cache, which therefore makes it perfect for running the host operating system. You can purchase a second RAID controller for guest VMs and scale the add-on cache to meet your data processing needs.

You'll want to specifically look for a minimum of integrated components to support your guest VMs. In particular, a good-quality motherboard will have two Enhanced Integrated Digital Electronics (EIDE) controllers with one being Serial ATA (SATA) capability. SATA controllers and hard drives can give you near-SCSI performance. With VM applications in a workstation configuration, the primary controller and hard drive can be used for the host operation system, and the secondary controller and hard drive can be used for guest VMs and a CD-ROM. On servers, you'll want at least one SCSI interface and one EIDE interface. In a server situation, you can hang a CD-ROM off the EIDE controller, use the SCSI controller for memory swap files, and use SCSI adapter cards for your host's and guests' main execution and storage locations. Integrated Ethernet adapters may not be so important in a tower workstation, but in pizza box–type servers where space is at a premium, integrated NICs are essential. If you're looking at a motherboard with integrated NICs, make sure it has two. You can team the interfaces; if one fails, the host continues to support the networking needs of your hosts. Integrated USB isn't necessarily important, but it's nice to have. If you opt for integrated video or can't find a motherboard without it, make sure the motherboard comes equipped with an AGP. Some manufacturers integrate an AGP video card but don't provide the AGP expansion slot. You'll be forced to use the slower PCI video technology, which may cause a bottleneck on the system bus. Integrated video support on a motherboard isn't uncommon and is generally a good way to save a few dollars. Integrated video isn't usually a problem with servers in that most servers don't require much more than 4–8MB of RAM. If you plan on playing graphics-intensive games on your host machine and using guest VMs for business, integrated video will prove to be useless.

Many new motherboards are capable of booting from USB, and this is a great alternative to booting a computer from a floppy disk or CD-ROM. USB also allows you to use external storage. Backing up your VMs to an inexpensive USB drive is a great way to add a layer of disaster recovery capability. Also, USB-attached storage is an excellent way to move VMs from a test environment to a production environment: it's oftentimes faster than pushing 20–30GB across the network. Lastly, integrated audio is something you may want to consider for basic acoustics. Integrated audio won't give you the advanced features of an adapter card. So, if you're looking to use a VM as a stereo, you can forget about using integrated audio.

Other integrated features you may want to look for in a motherboard are system monitoring and jumperless configurations. System monitoring includes the ability to read BIOS post codes for troubleshooting and the ability to track case and processor temperatures for reliability and performance reasons. Being that VM hosts will be consuming much more of a processor's available cycles than normal, it's important to keep operating temperatures at an optimum level. You can have a server that was once reliable easily become unstable with the addition of three or four VM guests. Being able to track system temperatures is a great way to stay out of the red zone when loading up a server on the road to moderate hardware utilization (60–80 percent). Jumperless settings are more of a convenience than anything else. They prevent you from having to open the system case to make a change on the motherboard. Also, jumperless settings mean that those of us with fat fingers no longer have to worry about dropping little parts in the case. For people who want to tweak performance with overclocking (not recommended in a production environment), jumperless motherboard configurations are a great way to quickly reconfigure the system.

Board Form Factor

If you have a computer case to host your hardware, you want to make sure you get a motherboard to fit it. Nothing is worse than spending a week researching the optimum motherboard, purchasing it, and finding it doesn't fit. Motherboards all require a specific power supply and are designed to fit a particular case. The power connectors differ from older-style AT motherboards to the newer ATX type. Moreover, newer motherboards use *soft-off* functionality, where power switching takes place on the motherboard instead of the power supply. As motherboards begin to support more integrated devices and power-attached peripherals, and as cases take on show-car characteristics, the power supply needs to have a rating sufficient to power the motherboard and everything beyond—hard drives, CD/DVD-ROM drives, fans, cameras, lights, and so on. If you require redundancy or are deploying your VMs in a production environment, you'll want to make sure your motherboard and case support redundant power supplies with high watt ratings (350–500).

Overall Quality

Price isn't necessarily a good indicator of quality, but good-quality products generally cost more. There's a lot of truth in the saying, "You get what you pay for." Nobody goes out of their way to purchase shoddy products, but if you don't research your motherboard purchase, you may think the product is substandard because it doesn't follow through on your expectations. Poor-quality motherboards can cause VMs to perform strangely and spontaneously reboot. You don't want to spend countless hours troubleshooting software problems when it's really a quality issue with hardware. Ferreting out quality requires looking at technical documentation from manufacturer Web sites and seeking advice from trusted professionals, Web forums, and

articles from reputable periodicals. You'll find that trustworthy motherboard manufacturers quickly stand out in your search.

Features that indicate good-quality motherboards are good component layout, good physical construction, a replaceable complementary metal oxide semiconductor (CMOS) battery, and extensive documentation. Nothing should get in the way of adding more RAM or processors to a system. Conversely, motherboard electronic components, such as capacitors, shouldn't get in the way of housed peripherals. Good motherboards have a general "beefy" quality about them. The board will be thick, capacitors will be big, and plenty of real estate will exist between everything to allow for adequate cooling. Lastly, the CMOS battery should be something you can replace from the local electronics store. When the battery goes bad, you'll want an immediate replacement part. A reboot on a bad CMOS battery could mean having to set the configuration for every integrated device manually. After figuring out RAID settings, interrupt requests (IRQs), and boot device priority because of a bad battery, you'll want immediate restitution. Fortunately, guest VMs aren't really affected by CMOS batteries other than that the guest VMs may set their system clocks to that on the physical host. Assuming you don't care about time stamps in log entries, you may be all right. If you host secure Web sites, you may find that your SSL certificates become temporarily invalidated because of an erroneously reported date. In the end, if you reboot your servers with some regularity, you'll want to immediately replace a bad CMOS battery.

Whether you choose to purchase a system or build your own, spend some time researching available options, and make sure the motherboard and vendor can deliver your expectations. Download the motherboard's manual, and make sure it's well-documented; for instance, make sure jumper configurations are clearly marked, processor and RAM capabilities are listed, and warranty information is noted. Being that the motherboard is the most important component in a system, purchasing the best possible motherboard will give you an excellent foundation for VM hosting and further your success with virtualization.

We understand that we're giving you a lot to consider, and at this point you may be completely overwhelmed with all this information. However, all major computer vendors publish data sheets for their systems. Using these data sheets gives you an easy way to compare the hardware features of multiple systems, which in turn will enable you to make a more informed decision. Additionally, reliable vendors will supply presales support to address all your concerns.

Considering Your Network

Building and deploying VM networks isn't really any different from deploying typical physical networks. In fact, the same issues that affect physical networks apply to virtual networks. Knowing how the devices from Chapter 1 function will help you determine what should be included in your VM networking plan. Before utilizing VMs, you'll need to decide on some networking priorities. Specifically, you'll need to decide if your VM implementation requires privacy, performance, fault tolerance, or security. You'll look at these networking requirements in the context of host-only, NAT, and bridged networking.

Public or Private VMs

The first thing you'll have to decide is if your VM network will be public or private. If you're building a private network of VMs, you'll want to use host-only networking. In a private network, you're creating a completely isolated environment in which you're required to supply all

traditional networking services, such as DHCP, Domain Name Server (DNS), Windows Internet Naming Service (WINS), Network Time Protocol (NTP), NetBIOS Name Server (NBNS), and so on; in other words, you'll be building a complete network from the ground up. The advantage to building a private VM network is that it will have zero impact on existing networks, and existing networks will have no impact on it. Host-only networking is a great way to learn about VMs and new operating systems. Host-only networks are also good for engaging in the practice of network forensics: you can analyze the behavior of network applications, ferret out nuances of communication protocols, and safely unleash the ravages of viruses, malware, and scumware for thorough analysis. In private networks, nothing leaves the sandbox.

■ **Note** In the future, if you ever need to connect the private network to production, you can maintain the integrity of your private network by configuring the VM host system to act as a router, for instance, by using Windows RRAS or ICS. This is also useful if you're limited on IP addresses. We'll cover the reconfiguration of host-only networks and NAT in Chapter 4.

If you're building VMs and need to access the outside world, it's still possible to keep a level of privacy and security for your VMs. NAT provides the basic firewall protection you'd expect to get from address translation because network communication can't be initiated from the host (public) network to the private (VM) NATed network.

After building and completing the configuration of your private network, you can use NAT to connect to your host's network. Whatever the host is connected to and has access to, your VMs will be able to access. Oftentimes, the best way to connect VMs to the Internet or local LAN resources is to use NAT. If you're testing new services and want hosts outside your virtual network to have access, you can use port forwarding to direct traffic to the VM hosting the service. Because all VMs share the host's IP address, port forwarding is your only option to open the private network for virtualization applications that support it, such as VMware products. Be advised that using NAT will adversely affect overall networking performance; however, it's a small price to pay for the security benefits. We'll discuss the configuration of port forwarding and NAT in more detail in Chapter 7.

While contemplating if and how you're going to integrate your VMs into your existing infrastructure, you may need to consider the performance impact of each VMware networking model and contrast it with the ease of configuration. The host-only network model provides the best possible performance for guest VMs. This is because network traffic doesn't pass across any communications medium, including the physical network adapters. Virtual hardware is employed to facilitate communications, and, despite the speed rating of the virtual network adapter, VMs in host-only networks pass traffic as fast as they can read and write to the host's memory. Host-networks can be simple to configure because virtualization applications will provide DHCP: you won't need to configure a separate server for this.

NAT networks incur latency during the translation process. This is because NAT is a fairly sophisticated protocol and tracks information entering and leaving the host. The NAT process puts enough overhead on the host system to negatively impact guest VM performance. Additionally, building NAT networks requires a bit more time during the configuration process and can cause some problems with regard to inbound traffic. Generally, inbound traffic is prohibited. In situations where port forwarding can be configured, members of the physical LAN

may still have problems gaining access to resources in the NAT network because not all protocols are supported. NAT networks are best used for private LANs that need access to the outside world while retaining their private state. Members of NAT networks have the speed benefit of host-only networking when communicating with peer members, but NAT networks take a performance hit when outside communication is required.

Bridged networking directly connects your guest VMs to the host network adapter(s). Many organizations use this to place their VMs right on the production LAN. Bridging guest VMs has several benefits:

- First, bridging a guest VM to the host is often the fastest way to connect to an existing network.
- Second, the guest will experience better network performance in that no additional routing protocols, such as NAT, are required.

Like physical computers, bridged VMs require unique settings, such as a host name, an IP address, and a MAC address. With bridged networking, you'll be limited to the speed of the physical adapter. For instance, if you have a guest with a virtual gigabit NIC and the physical NIC is rated at 100Mb, transmission will take place at 100Mb. Also, if you have a physical NIC rated at 1Gb and a guest's virtual NIC is rated at 100Mb, you can expect to take a performance hit. It isn't that your physical NIC is going to slow down; it's because of the driver technology utilized by the virtualization application.

Availability and Performance

Good networks can be characterized as fast, predictable, and highly available. Good networks limit user wait times and streamline infrastructure operations. Good networks are created over time by embracing simplicity and using redundancy techniques to achieve maximum uptime. The first milestone to cross on the way to the high-availability finish line requires load balancing and teaming; without these, you'll never be able to achieve high-availability performance goals.

The theory behind network performance tuning and high availability is to build fault tolerance into all systems to eliminate any single point of failure for hardware and software while maintaining fast service. Because building good network performance is inclusive of load balancing and teaming, high-performing networks readily lend themselves to high availability and disaster recovery requirements in that failover and standby systems are already present in the network configuration.

With this lofty network performance/availability definition set, what part do VMs play? VMs play into network availability and performance by supporting what you'd expect of a highly available and good-performing server:

- The virtualization layer neutralizes hardware dependence. If the VM application can be installed on new hardware, your VM server will run.
- Clustering on single-server hardware eliminates single-server failure. If one server panics or blue-screens, the other keeps working.
- The single-file design of VMs supports the portability needs of disaster recovery. Copy and save a server like any other file, and then run it anywhere.

- VMs take advantage of teamed NICs to increase throughput and prevent a common single point of failure. Caveats exist, though! We'll discuss this in further detail in a moment.

We already established server portability and virtualization concepts in Chapter 1. In the following sections, while touching on some commonsense networking tips, we'll cover the specifics of network adapter teaming; we'll save server clustering for Chapter 9. Despite the performance promises of adapter teaming and server clustering, keep in mind that simple is almost always better. As networking professionals, we know that needlessly complicated systems are just needlessly complicated systems. If a problem can be solved, corporate objectives can be upheld, and budgets can be satisfied with two tin cans and a piece of string, then you've built a good network. No excuse exists for not building reasonably fast, predictable, and accessible networks—this is the purpose of VMs, and you can add VMs to your bag of tricks for creating and supporting highly available networks employing internetworking best practices.

Simplicity

If you're looking to build a quick network of VMs capable of interacting with your existing infrastructure, then keep IT simple (KITS). Configure your VMs using the defaults of VM applications that are utilizing bridged networking. Microsoft and VMware have considered many of the common options and scenarios network professionals need to have a VM quickly viable; therefore, you can capitalize on the manufacturers' research and development efforts and reap the rewards. You can make virtual disk adjustments as necessary.

When you've outgrown boxed-product tuning and need to increase the load and redline on your servers, you'll have to spend a bit more time testing NIC teaming and server clustering in the lab to achieve the higher levels of reliability and performance you may be accustomed to getting. For instance, if you're looking to ease the bottleneck on a loaded-up VM server or achieve the possibility of increased uptime, you'll want to pursue adapter teaming and server clustering.

You have much to gain from the initial implementation of simple VM workgroup-type networks, but if you want enterprise-class disaster preparedness, you'll have to employ mesh networks using redundant switching and redundant routing. If you want enterprise-class performance, you'll have to employ network adapter teaming and VM server clustering.

Mesh Networks

Implementing redundant switching and routing really isn't within the scope of this book, but we wanted to mention it because of load balancing and teaming. You can achieve redundant switching and routing by using multiple physical NICs, switches, routers, and multiple internetwork connections utilizing teaming and load balancing. Load balancing and teaming are technologies used to keep mesh networks running in the event of a system failure. For instance, if you're building a highly available Web server, you'll need multiple physical servers with multiple physical connections to multiple switches connecting to multiple physical routers that connect to at least two different carriers using two different connectivity fabrics; you shouldn't use a major carrier and a minor carrier that leases lines from the major carrier. If you lose the major carrier, you also lose the minor carrier, and you can kiss your redundant efforts and mesh network goodbye.

The idea with redundant switching and routing is to ensure that your subnet's gateway is multihomed across truly different links. In the event of a NIC, switch, or router failure, multihoming maintains connectivity to your subnet and everything beyond your subnet using alternate hardware and routes to get the job done.

Teaming and Load Balancing

To gain some of the redundancy benefits and all the performance advantages of teaming and load balancing, you don't have to liquidate the corporate budget to create mesh networks. You can achieve the act of teaming in networked environments by aggregating servers or NICs together. In both cases, the end result is more reliable network communications by removing the physical server or NIC as a single point of failure and gaining the advantage of being able to potentially process more client requests (traffic) by having multiple devices performing similar service tasks. For those not in the know, joining multiple servers into a teamed group is called *clustering*, and joining multiple NICs together is termed *network adapter teaming*.

Load balancing is often confused with teaming; however, *load balancing* is the act of processing client requests in a shared manner, such as in a round-robin fashion. One network adapter services the first client request, and the next adapter services the next; or, one server processes the first request, and the next server processes the second client request. Standby NICs and servers are different from load-balanced ones because in the event of a NIC or server failure, you'll have a brief service outage until the server or NIC comes online. Conversely, service failures on networks using clustering and adapter teaming are transparent to the end user.

A network load balancing cluster combines servers to achieve two ends: scalability and availability. Increased scalability is achieved by distributing client requests across the server cluster, and availability is increased in the event of a server failure in that service requests are redirected to the remaining servers. Server clustering technology is available in both Microsoft and Linux operating systems. We'll discuss clustering at length in Chapter 9 and focus on network adapter scalability and availability here.

Network Adapter Teaming

Network adapter teaming provides servers with a level network hardware fault tolerance and increased network throughput. Teamed adapters are logically aggregated by software drivers. The teaming driver manipulates each adapter's MAC address and logically assigns a new MAC address so the team acts like a single NIC. A teamed NIC is transparent to guest VMs. If one physical network adapter fails, automatic failover seamlessly processes network traffic on the remaining NICs in the team. The software drivers involved with teaming manipulate the MAC address of the physical NICs, and this is generally why OS manufacturers avoid supporting related issues with teaming and even suggest avoiding them in server clusters.

Caution VMware doesn't support network adapter teaming for GSX Server hosts running on Linux, and VMware has yet to officially test it as of this book's publication. Additionally, Microsoft doesn't support the teaming of network adapters and sees it as something beyond the scope of its available OS support services. Both VM application manufacturers lay the responsibility for network adapter teaming at the feet of your NIC vendor.

VMware does offer limited adapter teaming support for GSX Server running on Windows OSs. The Windows-hosted GSX Server supports Broadcom-based network adapters and teaming software in three modes:

- Generic trunking (Fast EtherChannel [FEC]/Gigabit EtherChannel [GEC]/802.3ad-Draft Static)
- Link aggregation (802.3ad)
- Smart load balance and failover

Additionally, VMware supports the Windows-hosted GSX Server utilizing Intel-based network adapters running Intel PROSet version 6.4 (or greater) in five modes:

- Adapter fault tolerance
- Adaptive load balancing
- FEC/802.3ad static link aggregation
- GEC/802.3ad static link aggregation
- IEEE 802.3ad dynamic link aggregation

From the ESX Server install, *adapter bonding*, the functional equivalent of teaming, is available. ESX Server doesn't support teaming for the console interface.

Assuming you run into problems with teaming, and keeping in mind that teaming has traditionally been the domain of hardware manufacturers and not OS manufacturers, you'll want to start your research at the NIC vendor's Web site, and you'll want to consult any accompanying network adapter documentation. In addition, most vendor Web sites have good user communities in which knowledgeable people are eager to help share information and solve problems. Be sure to implement any suggestions in a test environment first.

Don't be discouraged by the lack of OS manufacturer support for adapter teaming; the redundancy and performance benefits are too great not to support team adapters. Teaming software from enterprise-class vendors comes with excellent support, seamlessly installs, and functions smoothly with supported switches. Some switches must be configured before teaming can be properly implemented. You'll need to research the specifics regarding the switch to which you'll connect teamed adapters. The trick to ensuring your adapter team is bound to your guest VMs is making sure the bridge protocol is bound to the teamed NICs and unbound from the physical network adapters. If you set up teaming prior to installing VM application software, you'll have solved nearly all your teaming problems before they ever happen.

If, after teaming NICs in a server, you don't experience a fair level of increased network performance, you may have a server that's suffering from high CPU utilization or just experiencing nonlinear scaling of teaming. For instance, if a server is consistently near 90 percent utilization on a 1GHz processor, availing the server to increased network capacity does nothing for packet processing. If you use the metric that 1Hz is required for every 1bps of data being processed, then in the previous scenario, potential throughput is limited to 1bps—that's a 100Mb NIC running at half duplex. Additionally, you'll find that teaming 100Mb NICs improves performance more than gigabit NICs. The reason for this is that 100Mb NICs are generally overtaxed initially, and installing 5Gb NICs in a team doesn't equal a fivefold increase in performance. With basic network adapter teams, you make available the potential to process more traffic and ensure network connectivity in the event of an adapter failure.

■ **Note** Virtual NICs support different throughput ratings. Currently, Microsoft's Virtual PC and VMware's Workstation can support emulated NICs rated 100Mb. Microsoft's Virtual Server and VMware's GSX Server and ESX Server support emulated gigabit NICs. Whether your VMs are server or workstation class, your VMs will probably never see their rated available bandwidth, but the potential is there.

VM Networking Configurations

From Chapter 1, you know that Microsoft and VMware can accomplish several types of networking with VMs. Determining the network mode you need to use with your VMs is based on whether interactivity with the outside world is required, such as physical LAN access or Internet connectivity. If the physical LAN needs access to your VMs, you'll want to stick with bridged networking. If you require an isolated network of VMs devoid of external connectivity and invisible to the physical LAN, then host-only networking will be your best choice. If you want a network of VMs that are invisible to the physical LAN and you require external Internet connectivity, you'll want to use NAT networking.

VMware controls the configuration of network types by using virtual network adapters and virtual switches. VMware's VM application software has ten available switches, VMnet0–9, and can use up to nine NICs. The NICs are in turn mapped to a switch. The network configuration of the switch will determine the network access type for the VM. VMware preconfigures three switches, one each for bridged (VMnet0), host-only (VMnet1), and NAT (VMnet8) networking. You can configure the remaining switches, VMnet 2–7 and 9, for host-only or bridged networking. You can create complex custom networks by using multiple network adapters connected to different switches. For instance, by using proxy-type software, a VM can be multihomed on a host-only network and NATed network to create a gateway for an entire virtual LAN.

Bridged networking makes VMs appear as if they're on the same physical network as the host. With bridged networking, you'll impact the production network as if you just set up and configured a new physical machine. For instance, your VM will need an IP address, DNS/NBNS/NetBIOS server configuration, virus protection, and domain configuration settings: if your VM is to be a server, you'll configure it like a typical server, and if it's a workstation, you'll configure it like a typical workstation. It's that easy. The major performance impact

you'll experience is a decrease in network performance from having to share the host's physical network adapter with the host.

Host-only networking is the act of building a private network between guest VMs and the host. These private networks aren't natively visible from the outside world. For instance, you can build a host-only network with multiple VMs, complete with domain controllers, mail servers, and clients, and the traffic will never leave the host computer. Host-only networks are an excellent way to test new software in a secure sandboxed environment. With host-only networking, you can safely experiment with different networking protocols, such as TCP/IP and Internetwork Packet Exchange (IPX)/Sequential Packet Exchange (SPX), and test solutions on isolated prototypes.

NAT can connect a VM to virtually any TCP/IP network resource that's available to the host machine. NAT is extremely useful in situations where a production network DHCP pool is nearly exhausted, Token Ring access is desired, Internet access is required, and security is an issue. NAT performs its magic by translating the addresses of VMs in private host-only networks to that of the host.

Microsoft controls the configuration of virtual networks through the use of virtual network adapters and a preconfigured emulated Ethernet switch driver. You can find the driver in the properties of the physical computer's network adapter settings; it's called the Virtual Machine Network Services driver. If you find yourself troubleshooting network connectivity for Virtual PC guest VMs, you may want to begin fixing the problem by removing the driver and then reinstalling it. The driver file is labeled `VMNetSrv.inf` and is located in the `VMNetSrv` folder of your installation directory.

Virtual PC VMs can support up to four network interfaces, and each VM can be configured for network types that Microsoft refers to as Not Connected, Virtual Networking, External Networking, or Machine-to-Host Networking. Of the four adapters available to you, the first adapter can be set to Not Connected, Local Only, Bridged, or Shared (NAT). The rest of the adapters can be configured as Not Connected, Local Only, and Bridged. Microsoft VM applications allow you to create complicated virtual networks by using the virtual network adapters that include firewalls, routers, and proxy servers.

When you configure a VM as Not Connected, it's a stand-alone computer and is isolated from all machines (including the host). Configuring a VM as Not Connected is a good way to test software or learn new operating systems.

Virtual networking is local-only networking in Microsoft's VM applications, and it's the same as host-only networking for VMware. Local-only networks allow guest VMs to communicate using virtual Ethernet adapters via the host's memory. Local networks don't generate traffic on the physical network; therefore, local networks can't take advantage of the host's network resources.

External networking, or the Adapter on the Physical Computer setting, bridges traffic to the physical adapter, and it's the same as bridged networking for VMware. External networking allows VMs to act as if they were a standard physical computer on your LAN. For example, the physical LAN is responsible for providing a DHCP address and Internet connectivity. External networking is inclusive of shared networking. Shared networking, known as NAT in VMware, provides VMs with a private DHCP address and network address translation services that connect the VM to the host's physical network.

With shared networking, VMs are invisible to the physically attached LAN. Unlike VMware, Microsoft doesn't support port mapping for inbound traffic. Computers that aren't on the virtual network won't be able to access virtual machine services or any of the VM's ports. Additionally, shared networking doesn't directly support host or guest VM intercommunication.

Virtual machine-to-host networking allows VMs to communicate with the host system via the Microsoft Loopback adapter. You'll have to manually configure the Microsoft Loopback adapter to take advantage of virtual machine-to-host networking. You'll configure Microsoft Loopback adapters in Chapter 3. VMware automatically configures a loopback adapter for VM-to-host communication. Assuming that the proper proxy or routing software is installed, as in ICS, you can use host-only networking and loopback networking to connect VMs to the Internet via the host's dial-up adapter. You can also connect to non-Ethernet-type networks, such as Token Ring.

Supporting Generic SCSI

To provide support for physical SCSI devices, such as tape drives, tape libraries, CD/DVD-ROM drives, and scanners, VMware employs a generic SCSI device for Linux and Windows OSs. Generic SCSI allows a VM to directly connect to the physical SCSI device. Assuming the guest OS can supply a driver for the attached physical SCSI device, then VMware VMs can run the SCSI device after installing the appropriate driver. Microsoft virtualization software doesn't currently support generic SCSI devices. The only other thing you need to worry about with generic SCSI is troubleshooting. We'll show how to install some generic SCSI devices in Chapter 3.

Windows Guests

Generic SCSI is intended to be device independent, but it may not work with all devices. Therefore, when using generic SCSI (as with anything new), test your configurations in the lab. To get generic SCSI to properly work with your VM host and guest VMs, as in Windows XP, you may need to download an updated driver from the VMware Web site. Though rare, if, after adding a generic SCSI device to a Windows VM, the guest doesn't display your desired device, you'll have to manually edit the VM's configuration file (*filename*.vmx). Listing 2-1 shows a typical configuration file.

Listing 2-1. *Typical VM Configuration File*

```
config.version = "7"
virtualHW.version = "3"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
memsize = "128"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Windows Server 2003 Standard Edition.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
floppy0.fileName = "A:"
Ethernet0.present = "TRUE"
usb.present = "FALSE"
displayName = "W2K3"
guestOS = "winNetStandard"
priority.grabbed = "normal"
```

```
priority.ungrabbed = "normal"
powerType.powerOff = "default"
powerType.powerOn = "default"
powerType.suspend = "default"
powerType.reset = "default"
ide1:0.startConnected = "FALSE"
Ethernet0.addressType = "generated"
uuid.location = "56 4d f2 13 f1 53 87 be-a7 8b c2 f2 a2 fa 16 de"
uuid.bios = "56 4d f2 13 f1 53 87 be-a7 8b c2 f2 a2 fa 16 de"
ethernet0.generatedAddress = "00:0c:29:fa:16:de"
ethernet0.generatedAddressOffset = "0"
floppy0.startConnected = "FALSE"
Ethernet0.virtualDev = "vmxnet"
tools.syncTime = "FALSE"
undopoints.seqNum = "0"
scsi0:0.mode = "undoable"
scsi0:0.redo = ".\Windows Server 2003 Standard Edition.vmdk.REDO_a03108"
undopoint.restoreFromCheckpoint = "FALSE"
undopoint.checkpointedOnline = "TRUE"floppy0.startConnected = "FALSE"
tools.syncTime = "FALSE"
```

We hope you won't have to manually edit VMware configuration files to fix issues related to generic SCSI. In the event that you find yourself poking around in a text file similar to Listing 2-1, you can count on four reasons for the SCSI device not properly installing for Windows guests:

- The device isn't physically configured correctly (is it plugged in and turned on?).
- The driver isn't installed on the host.
- The host driver prevents the SCSI device from being detected by the guest.
- The guest requires a device driver that doesn't exist for the host system.

After eliminating any of these reasons for generic SCSI failure, you'll need to use a text editor, such as Notepad or Vi (Vi can be used in Windows and is even available on a Windows Resource Kit), and modify the VM's configuration file to solve the problem. To start, you'll need to locate the VM's configuration file: it uses a `.vmx` extension, and it should be edited only while the VM in question is powered down.

VMware suggests that only experts edit VM configuration files, but you can safely edit the file if you make a copy before making changes. If something goes awry during the reconfiguration process, you can simply write over your changes with the previously copied file and start again. When troubleshooting generic SCSI for Windows, you'll want to focus on one of three things:

- Is this the installation of a new SCSI adapter?
- Is this the installation of a new SCSI device?
- Is this a failure of the Add Hardware Wizard?

In the first instance, if you're troubleshooting the installation of a new SCSI adapter (meaning this isn't a preexisting SCSI adapter), make sure the downed VM's configuration file contains this:

```
scsiZ:Y.present = "true"
scsiZ:Y.deviceType = "scsi-passthru"
scsiZ:Y.fileName = "scsiX:Y"
```

In the second instance, if you're troubleshooting the installation of a new SCSI device on an existing VM-recognized SCSI adapter, make sure the downed VM's configuration file contains this:

```
scsiZ:Y.deviceType = "scsi-passthru"
scsiZ:Y.fileName = "scsiX:Y"
```

In the third instance of troubleshooting, you may have to address failures related to the Windows Add Hardware Wizard. The wizard often doesn't properly recognize newly installed or preexisting SCSI adapters and devices. If this is the case, make sure the downed VM's configuration file contains this:

```
scsiZ:Y.fileName = "scsiX:Y"
```

In all three troubleshooting scenarios, X, Y, and Z will be defined according to the following:

- X is the device's SCSI bus on the host system.
- Y is the device's target ID in the virtual machine and on the host. It must be identical to function correctly.
- Z is the device's SCSI bus in the virtual machine.

When determining which numbers to use for X, Y, and Z, keep in mind that SCSI buses are assigned numbers after available IDE buses, and the device target ID is normally assigned via switches or jumpers on the device.

Linux Guests

Like Windows operating systems, VMware supports generic SCSI for Linux VMs. Generic SCSI is nothing more than pass-through access to physical SCSI devices for which the host loads drivers. Assuming a driver has been successfully loaded, guest VMs can use any SCSI device that the host can. The generic SCSI driver is responsible for mapping attached SCSI devices in the `/dev` directory. To take advantage of generic SCSI, it requires driver version 1.1.36 (`sg.o`) and must be used with kernel 2.2.14 or higher. Each device will have an entry in the directory beginning with `sg` (SCSI generic) and ending with a letter. The first generic SCSI device would be `/dev/sga`, the second would be `/dev/sgb`, and so on. The order of the entries is dictated by what's specified in the `/proc/scsi/scsi` file, beginning with the lowest ID and adapter and ending with the highest ID and adapter. You should never use `/dev/st0` or `/dev/scd0`. You can view the contents of the `/proc/scsi/scsi` directory by entering `cat /proc/scsi/scsi` at the command-line interface (CLI).

Juggling host and guest connectivity to disk drives (`sd`), DVD/CD-ROM drives (`scd`), and tape drives (`st`) can be tricky. If you permit simultaneous connectivity for the host and guest systems, your Linux system may become unstable and data corruption/loss can occur. If you don't have two SCSI disk controllers, one for the guest and one for the host, this is a good time to rethink your configuration strategy.

Simultaneous access becomes an issue because Linux, while installing the generic SCSI driver, additionally recognizes SCSI devices, such as the previous devices, as `sg` entries in the `/dev` directory. This creates a dual listing for SCSI devices. The first entry was created during the install of the host's Linux OS. Though VMware does an excellent job of arbitrating access to a single resource, it isn't always successful in making sure the dual device listings aren't simultaneously accessed under the `/dev` and `/dev/sg` listings. Therefore, don't use the same device in a host and guest concurrently.

After adding a generic SCSI device under Linux, you'll need to check the permissions on the device. For the device to be of any use to guest VMs, read and write permissions must be assigned to the device. If standard users, other than the superuser, will need access to the device, groups should be created for access control. We'll show how to install and set permissions on generic SCSI devices in Chapter 3.

Considering Storage Options

Determining which hard disk type and configuration to use for hosts and guests is based on the purpose of the final VM system. Presentation, educational, test, and production environments all have different reliability and availability requirements and widely differing budget constraints. Correctly identifying the purpose of a VM host will help you budget available resources so you can prioritize host needs, as shown in Figure 2-1.

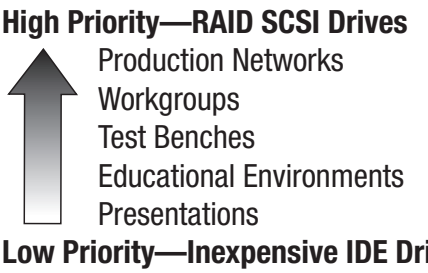

Figure 2-1. *Prioritizing host storage*

In the following sections, we'll cover physical and virtual SCSI/IDE disks and their relative performance characteristics. We'll also briefly discuss the commonsense application of each.

Physical Hard Drive Specifications

Physical hard drives have several specifications with which you need to be familiar. Being that virtual hard drives are mapped to the physical drive, the performance of the virtual hard drive is similar or equivalent to the physical drive. That is, you can't have a better virtual hard drive than you do a physical one. The hard drive you choose impacts the host and guest OSs equally. With SCSI and IDE hard drives, you'll want to concern yourself with four specifications: spindle speed, cache buffer size, access time, and data transfer rate.

Spindle Speed

Spindle speed is the rotation speed of the hard drive's platters. Speeds generally step from 4,200 revolutions per minute (RPM), 5,400RPM, and 7,200RPM for IDE drives and from 10,000–15,000RPM for SCSI drives. The higher the speed, the better the performance you'll experience. Conversely, more speed means more money and heat; therefore, you'll need to increase your budget and ventilation. You generally will find low spindle speed (4,200–4,500RPM) small form factor hard drives in notebooks, and these drives are sufficient only for testing or demonstrating a handful of concurrently running VMs. Avoid using similar spindle speeds in production VM host workstation and server systems, or you'll take a serious performance hit.

Cache

The buffer cache on a hard drive is similar to that found on a motherboard. The memory is used to speed up data retrieval. Most hard drives come with a 2MB, 8MB, or 16MB cache. If you're building a workstation for testing, then you can save some money by going with smaller buffer sizes. For servers, video processing, and extensive data manipulation using spreadsheets or large documents, you'll want to go with larger buffers to get that extra performance boost. Moreover, if you're using RAID controllers, you'll want increased cache levels for file servers. Cache creates performance differences significant enough for VM host applications and guest VMs that investing in larger buffers is merited because it decreases overall system response time, which allows you to more seamlessly run multiple VMs.

Access Time

The rate at which a hard drive can locate a particular file is referred to as *access time*. This factor becomes extremely important for file server–type VMs. For instance, if a VM will be frequently manipulating thousands or tens of thousands of files, being able to find each one in a reasonable time becomes problematic on drives with high access times. High access times will cause VMs to appear to hang while the hard drive searches for files. By sticking with the lowest possible access times, you reduce latency. As you stack more VMs onto a single host, this access time becomes more critical.

Transfer Rate

Transfer rate generally refers to one of two types, internal and external. *Internal* transfer rate is the rate a hard disk physically reads data from the surface of the platter and sends it to the drive cache. *External* transfer rate is the speed data can be sent from the cache to the system's interface. Trying to compare the transfer rates among different manufacturers may be difficult; they all typically use different methods to calculate drive specifications. You will, however, find that the transfer rates of parallel ATA IDE drives are less than that of SATA IDE drives, and both are less than that of SCSI transfer rates. When it comes to service life, SCSI will win in the warranty and longevity department.

You'll get better sustained transfer rates from SCSI drives. If you want to keep the data flowing on a busy VM, such as an e-commerce Web server, and sustain higher levels of performance, transfer rates are critical. Transfer rates are less critical if a system operates in bursts, such as a print server. Because external transfer rates depend on the internal rates being able to keep the cache full, internal transfer rates are a better indicator of performance when selecting drives for your VM host machines. You can download data sheets from hard drive manufacturer Web sites to compare transfer rates for hard drives.

We mentioned earlier that having multiple disk controllers and disks provide for better VM hosting, so let's feed the supergeek in us and look more closely at the mathematic justification for using multiple controllers and disks in your VM host. You'll need to pull hard drive data sheets to compare throughput averages of disks to get a better feel for overall performance as compared to theoretical maximums. For this discussion, we'll make some sweeping generalities about average data throughput that many hard drives support. To that end, what do the numbers look like between a VM host utilizing two Ultra 320 SCSI drives across two channels compared to a VM host using two Serial ATA 150 drives across two channels? Serial ATA drives have a maximum throughput of 150MB/sec and can sustain an average transfer rate between 35MB/sec and 60MB/sec. Your total throughput for a two-channel system would provide host and guest VMs with a maximum of 300MB/sec, and in constant data delivery mode, VMs would see about 70–120MB/sec total. Looking at Ultra 320 SCSI drives using one channel, maximum throughput could reach 320MB/sec and would average about 50–80MB/sec. Using a two-channel Ultra 320 SCSI controller, theoretical maximum throughput would be 640MB/sec and would be 100–160MB/sec for constant data delivery. To get equivalent speeds from SATA devices, you'd need a five-channel adapter and five hard drives: this would put you in the price range of a SCSI solution. The upshot with SATA is that storage is inexpensive. In fact, SATA storage is so inexpensive that it's not uncommon for disk-to-disk backup systems to be composed entirely of SATA drives. If you combined the space of five 350GB hard drives, you'd have 1.7TB. To get the equivalent from 143GB SCSI drives, you'd have to purchase 13 hard drives! When you look at SCSI and SATA for large storage solutions, you can't question the cost benefit of using SATA. When it comes to performance, SCSI blows the doors off SATA technology. Determining if you should go with SCSI or SATA can be difficult, though. You can use Table 2-5 as a guide to help determine if your host implementation situation should use SATA or SCSI technology.

■ **Note** VMware's ESX Server requires SCSI-attached storage for running guest VMs. However, you can store guests on IDE devices.

Table 2-5. *Hard Disk Implementation Situations*

System	Usage	Storage Disk Type
Desktops	Office suites/games	All SATA
Workstations	Engineering/CAD	Mostly SATA/some SCSI
Entry-level servers	Web/e-mail/automation	All SATA
Mid-level servers	Web/e-mail/automation	Partly SATA/partly SCSI
High-end servers	Web/e-mail/automation/databases	All SCSI
Enterprise servers	Disk-to-disk backup	Mostly SATA/some SCSI

We also want to point out that the performance increase SCSI drives experience over IDE drives isn't all attributed to drive design. The increase in performance is generated by the structure of the SCSI bus. The SCSI controller bus can manage hard drives without having to interrupt the processor for support, and the controller can use all drives attached to the bus simultaneously. IDE drives are limited to sharing the IDE bus. For instance, if you connected two IDE drives to the same bus, the drives would have to take turns communicating over the IDE bus. IDE drives are ideal for single-user computers, low-end servers, and inexpensive storage. If you're building a VM host that will be pounded in a multitasking environment, the extra expense of SCSI drives is well worth the performance boost your VMs will experience.

RAID

To create a RAID group, you can use one of two approaches: RAID implemented in software or RAID implemented in hardware. As with anything, each RAID type has positive and negative points. When considering RAID for your VM solution, you're looking to increase capacity, gain a performance edge, or add fault tolerance.

Hardware RAID vs. Software RAID

If you want the best performance and security from a RAID solution, you'll want to implement a hardware solution. You obviously will not get more performance from a software implementation, as it makes the VM host do more work. The independent architecture of hardware RAID, meaning that the operating system functions independently of the management features of the RAID controller, will deliver better performance and security for VMs. Software RAID is implemented within the host's operating system. Many operating systems come with a software RAID capability as a standard feature, and using it will save you the cost of a RAID controller. Unfortunately, software RAID adds overhead to the host by loading up the CPU and consuming memory. The more time the CPU spends performing RAID tasks, the less time it has to service guest VMs. The extra overhead may be a moot point in low utilization environments where you gain the redundancy advantage RAID has to offer.

■ **Note** Let's take a quick reality check with software RAID implementations. If you've purchased server-class virtualization applications to use VMs in a production environment, money exists within your budget for a hardware RAID controller. If you don't have a hardware RAID controller for your host and insist on using software RAID, do everyone a favor and return the VM software. Buy a hardware RAID controller first.

You implement hardware RAID using a special controller card or motherboard chipset. This is more efficient than software RAID. The controller has its own processing environment to take care of RAID tasks. Hardware RAID is managed independently of the host, and all data related to the creation and management of the RAID array is stored in the controller. Software RAID stores RAID information within the drives it's protecting. In the event of operating system failure, which solution do you think is more likely to boot? In the event of disk failure, which solution is more likely to preserve the state of your data? If you choose to use a software implementation of RAID, it should be used only for redundancy purposes and when hardware RAID is cost prohibitive. For better all-around performance and security, hardware RAID is your best choice.

In the not-too-distant past, RAID controllers were available only for SCSI hard drives. With the increased throughput of SATA technology and the need for inexpensive RAID controllers, manufacturers are making hardware RAID available for SATA technology. In situations where hardware RAID is needed and the budget is looking thin, SATA RAID is an excellent choice. These new breeds of RAID controllers come equipped with several disk interfaces, some as high as 16 ports! Like their SCSI counterparts, SATA controllers can have hotswap capability, allowing for the replacement of failed disks on the fly. It isn't uncommon for new entry-level or mid-level servers to come equipped with an SATA RAID option.

RAID Types

With RAID implementation selection out of the way, let's look at the available RAID types and how they impact the performance and redundancy of your VM solution. RAID types commonly found in production environments are RAID 0, RAID 1, and RAID 5, and each offers specific benefits and drawbacks.

RAID 0 isn't really a RAID type because redundancy isn't available (no parity). Despite the lack of redundancy, RAID 0 offers the best possible performance increase for VMs. RAID data is striped across two or more disks. *Striping* is the process of splitting data into blocks and distributing it across the drives in the array. Distributing the I/O load across the disks improves overall VM performance. RAID 0 is great to use where a single hard disk is normally used and the loss of data isn't critical. Generally, you'll find RAID 0 implemented for a computer-aided drawing (CAD) or video-editing workstation.

RAID 1, or *disk mirroring*, maintains an identical copy of all data on different disks in the array. At a minimum, you must use two disks. If one disk goes bad, service availability of the host and guest VMs will be maintained. RAID 1 provides for faster disk reads because two data locations are working to service one request. Conversely, RAID 1 negatively impacts disk write performance because data must be written to two locations. RAID 1 is good for protecting servers that don't require lots of disk space or where extensive file writing will not be an issue, such as for a server hosting a read-only database. In terms of using RAID 1 for a host, it isn't a

bad choice for redundancy when drives are limited. You'll see a noticeable decrease in performance for guest VMs that use virtual disk files.

RAID 5, also known as *block-level striping with distributed parity*, stripes data and parity information across a minimum of three hard drives. RAID 5 has similar performance characteristics as RAID 0 save for the overhead of writing parity information to each drive in the array. The performance impact of having to write the parity information isn't as significant as having to perform disk mirroring. Fault tolerance is derived from data blocks and its parity information being stored on separate drives in the array. If one drive should fail, the array would continue to function because the data parity is stored on a different disk. When a drive fails in a RAID 5 array, it functions in a degraded state until the failed drive is replaced. RAID 5 is common in production environments and offers the best of all worlds—excellent capacity potential, good performance, and better fault tolerance for your host and VMs.

Host Disk Sizing

Sizing system storage is extremely important for both virtual and physical machines. Because you'll be collapsing the infrastructure onto one box, your storage needs in a single host computer will be a multiple of the servers you want to run on it. If you have four servers with 100GB storage capacity, for example, you'll need a host with a minimum of 400GB of space and additional room for the host operating system. If you're using dynamic disks, you'll find that VMs will quickly consume your attached hard drive space. When choosing permanent storage for your hosts and guests, you need to decide if the final solution is for portable demonstrations and academic situations, for a test bench, or for the production environment.

Presentation and educational environments are generally restricted to small budgets and lower-end hardware. VMs typically will be loaded onto inexpensive large-capacity IDE devices and notebooks with limited quantities of RAM. In these cases, rather than spending your budget on SCSI drives, investing in more RAM is the key to getting three or four reasonably responsive dynamic disk VMs running simultaneously in host mode–only networks. For instance, if you need to turn up four VMs, you'll need 256MB for the host and 128MB for each guest, totaling 768MB. Running Windows 2000 and 2003 on 128MB is painful. In practice, you'll find you slide by fairly nicely by running Windows VMs with 192MB of RAM. In the previous situation, that now requires you to have 1GB of RAM.

Being that a test bench should more closely approximate an existing network environment, and being that VMs are often built on the test bench and then copied into production, you'll want to use SCSI drives in this case. Moreover, SCSI drives are a prerequisite if you plan on testing ESX Server. To get the most out of the two to three servers you may have available for testing, you'll want to have two SCSI controllers in your server (one for the host and one for guests) and a moderate amount of RAM: both SCSI controllers and ample quantities of RAM (256–512MB per VM) are required to adequately support a test domain of six to ten VMs and any enterprise applications, such as Oracle, Lotus Notes, or Exchange.

■ **Note** You'll also want to have similar, if not identical, NICs in your existing production environment to avoid undue network complications. You'll also want to have enough physical NICs to test any teaming or load balancing implementations that may be in production.

Production environments require high availability, good reliability, and good response times. It's necessary to have multiple RAID controllers per host (or SAN access), high-end SCSI drives, plenty of NICs, and tons of RAM. The more closely you can approximate the contents of multiple servers in one box, the better the VM host you'll have. The key to successfully deploying VMs is to consistently utilize 60–80 percent of the total physical server resources. The extra 30–40 percent is a buffer to offset utilization spikes. The more a server approaches 100 percent utilization, the more it appears to be frozen to an end user. Fast RAID controllers and SCSI disks with large buffers can easily feed a hungry network. RAID drives are more important than ever during the event of system failure because it isn't one server going down—it's a host and all its VMs! Determining the amount of drives you need is as easy as figuring the total number of fixed-disk VMs plus room for the host. You may find yourself "specing out" a physical server with a terabyte of storage or investing in a SAN.

Guest Disk Sizing

IDE device controllers are limited to four devices total, two each for the primary and secondary channels. You can use any CD-ROM and hard drive combination, but you can't exceed the limit of four devices: this is true for physical and virtual machines. All the VMs in this chapter are limited to 128GB virtual IDE devices, whereas physical systems can use IDE drives larger than 350GB. If you have a guest VM that requires more than 384GB of storage, using virtual IDE hard drives isn't a viable option. However, you can run virtual SCSI drives from a physical IDE drive to meet your sizing demands. The virtual SCSI disk performs better than the virtual IDE disk because of the difference in driver technology. Using virtual SCSI disks reduces the impact of virtualization overhead and will more closely approximate the physical performance characteristics of the hard drive.

Virtual SCSI disks are available for VMs in Microsoft Virtual Server and all VMware virtualization applications. Microsoft allows you to configure up to four virtual SCSI adapters with a maximum of seven virtual SCSI disks for a total of twenty-eight virtual SCSI disks. Each virtual SCSI disk can be as large as 2TB. Virtual Server emulates an Adaptec AIC-7870 SCSI controller. For VMware, you can configure up to seven virtual SCSI disks across three virtual SCSI controllers at 256GB per disk. VMs can directly connect to physical SCSI disks or use virtual disks. In either case, you can't exceed a total of 21 virtual drives. When configuring your VM, you can choose between a BusLogic or LSI controller card. The LSI card provides significantly better performance for your VMs. The trick to utilizing the LSI card for your VMs is to have the driver available on a floppy disk while installing your OS. You can download the LSI driver from the VMware Web site. When given the choice, opt for SCSI virtual disks.

The performance difference between mapping a VM to a physical disk and a virtual disk file is an important component to consider in your system design. In the first scenario, the VM treats the disk as a typical server would: it's raw storage, and the VM interacts directly with the physical disks. The VM reads and writes to the disk like a typical server with its own file management system. The only overhead really experienced by the VM guest is that of the virtualization layer.

Virtual disks, in essence, emulate physical hard drives and are stored as files on the host's physical hard drive. The guest VM will mount the file and use it like a hard drive. The guest OS will use its own file management system to read and write to the file. All this reading and writing takes place within the file management system of the host creating the extra overhead. If you opt to use dynamically expanding virtual disk files, this creates additional overhead because the file must be resized before data can be written to it. Dynamically expanding virtual disks also tend to fragment the host's hard drive, which further negatively impacts performance for the host and guest VMs alike. The major advantage virtual disk files have over mapped drives is portability. Moving or backing up a VM is like copying a regular file. In Chapters 4 and 6, we'll further discuss virtual disk performance.

Storage Area Networks

A SAN generally consists of a shared pool of resources that can include physical hard disks, tape libraries, tape drives, or optical disks that are residing on a special high-speed subnetwork (different from the existing Ethernet) and utilizing a special storage communication protocol, which is generally either Fibre Channel Protocol (FCP) or Internet SCSI (iSCSI). Together, it's all known as the *storage fabric*. The storage fabric uses a SAN-specific communication protocol tuned for high-bandwidth data transfers and low latency.

Most SAN interconnectivity is accomplished by using a collection of Fibre Channel switches, bridges, routers, multiplexers, extenders, directors, gateways, and storage devices. SANs afford connected servers the ability to interact with all available resources located within the SAN as if the devices were directly attached. In a reductive sense, you can think of a SAN as a network composed of interconnected storage components. We'll discuss SAN technology, as well as its configuration options and relationships to VMs, in much further detail in Chapter 13.

Servers generally gain access to the SAN via fiber-optic cabling; however, connectivity is also available over copper. What SANs offer VMs is the same thing offered to physical servers: increased performance levels, better reliability, and higher availability. These benefits are achieved because the SAN is responsible for data storage and management, effectively increasing the host's capacity for application processing. Extra processing capability results in lower latency for locally running VMs and faster access times for the end user. Disaster recovery capabilities are enhanced in that the time to restore a VM is significantly decreased and the physical data storage can be in a remote location.

Microsoft VMs and SANs

Microsoft VM virtual hard disks can be stored on a SAN. VMs don't need anything in particular to use a SAN because the host views the SAN as a local volume—the physical computer treats the SAN as a typical storage device, assigning it a local drive letter. The host simply places all files associated with VMs on the SAN via the assigned drive letter. If you're wondering if you can use SAN as a physical disk, you can't. Virtual Server doesn't provide emulation of SAN host bus adapter (HBA) cards. Microsoft will treat a SAN as a locally attached device.

VMware VMs and SANs

Like Microsoft, VMware treats a SAN as locally attached storage for GSX Server. Unlike Microsoft, VMware allows for the mounting of a SAN LUN for ESX Server (using a SAN as a physical disk). A LUN is a slice of the total SAN representing a section of hard drives that's in turn labeled to identify the slice as a single SCSI hard disk. Once the LUN is created, the host or ESX Server guest can address the disks in that particular SAN slice. ESX Server supports up to a maximum of 128 LUNs and can emulate QLogic or Emulex HBA cards.

ESX provides support for mulitpathing to help achieve a more highly available network by maintaining connections between the SAN device and server by creating multiple links with multiple HBAs, Fibre Channel switches, storage controllers, and Fibre Channel cables. Multipathing doesn't require specific drivers for failover support.

When installing ESX Server, you'll need to detach the server from the SAN. Detaching prevents accidentally formatting any arrays and prevents the SAN from being listed as the primary boot device for ESX Server. Moreover, VMware suggests you allow one ESX Server access to the SAN while configuring and formatting SAN and VMFS volumes. In a SAN solution, you want to install ESX Server on the local physically attached storage and then run all VMs from the SAN via an HBA card. You'll get near physical server performance from your VMs by allowing them sole use of the HBA cards. This also gives you a major backup advantage. Multiple VMs can directly back up to a tape library in a SAN simultaneously. Additionally, don't place ESX Server's core dump file on the SAN, as it can make the system unstable.

LUN Management

On initialization or when a Fibre Channel driver is loaded, ESX Server scans the SAN for LUNs. You can manually rescan for added or deleted LUNs by issuing `vmkfstools -s` at the CLI. If you're using QLogic HBAs, you'll have to flush the adapter's cache for entries in `/proc/scsi/qla2200` or `/proc/scsi/qla2300`. At the CLI, for example, enter the following:

```
echo "scsi-qlascan" > /proc/scsi/qla2300/0
vmkload_mod -u qla2300
vmkload_mod /usr/lib/vmware/vmkmod/qla2300_604.o vmhba
```

If, after creating your LUNs and them not appearing after a scan, you'll need to change the DiskMaxLun field from Advanced Settings in the Management Interface. The number should be equal to the total number of LUNs plus one. In Figure 2-2, the default value is set to 8. Notice that the name of the server isn't using the default installation name; your system should have FQDN listed.

By default, ESX will see 0–7; you achieve this by typing **8** in the Value field. Rescan to detect the new LUNs. You can verify your result at the CLI by running `ls /proc/vmware/scsi/<fiber channel adapter>`.

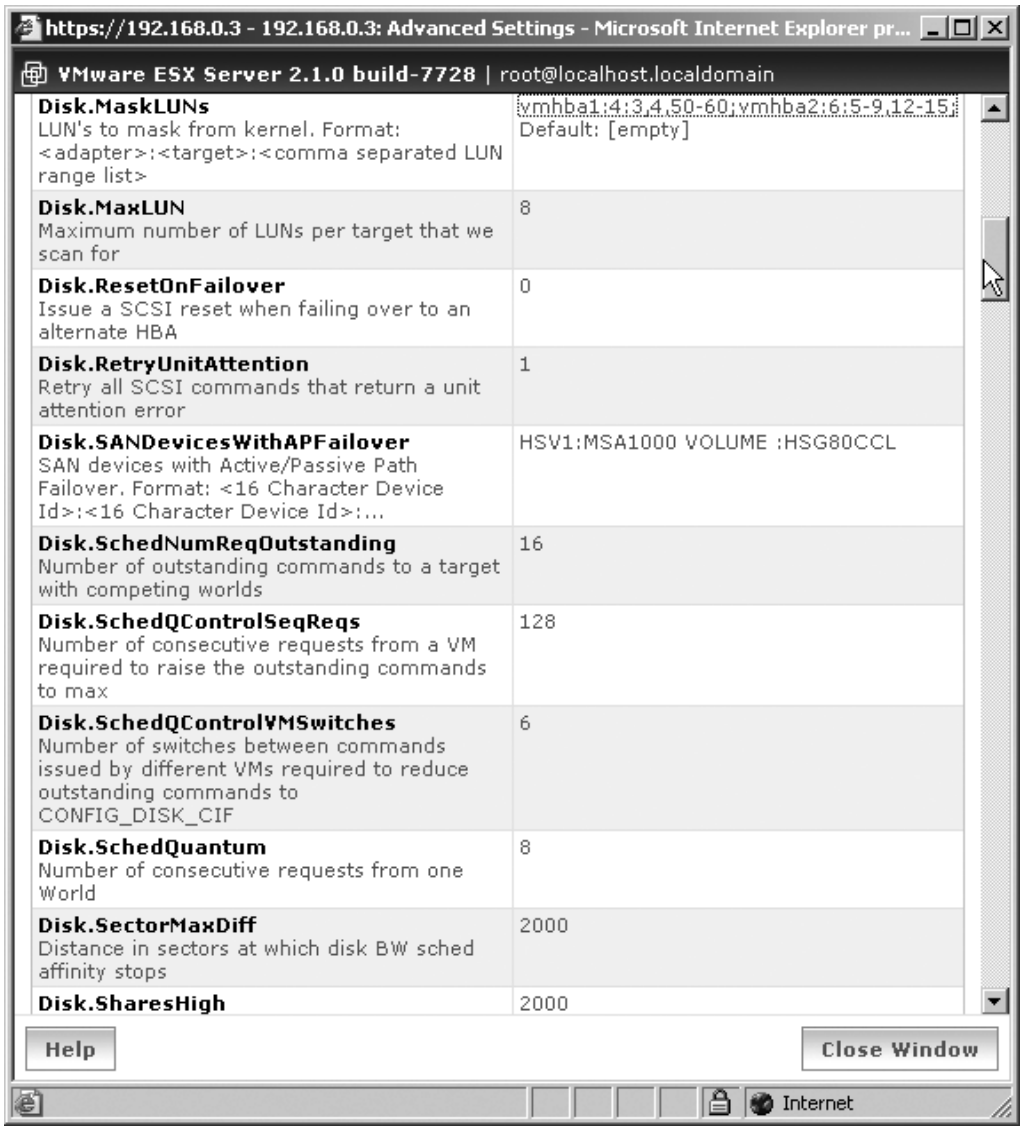

Figure 2-2. *DiskMaxLun and DiskMaskLUNs field settings*

Before assigning SCSI or SAN storage to a virtual machine, you'll need to know the controller and target ID used by the service console and the controller and target ID used by the VMkernel. Though physically dismantling your system is an option for information regarding the service console, you can determine controller and target IDs from the CLI. The boot log file, `/var/log/messages`, and the SCSI file of the proc pseudo file system, `/proc/scsi/scsi`, both contain valuable information. You can view the contents of each by entering the following at the CLI:

```
cat /var/log/messages | less
cat /proc/scsi/scsi | less
```

For data regarding the controllers assigned to the VMkernel, you'll want to look to the `/proc/vmware/scsi` directory. This information will be available only if the VMkernel starts. For every controller assigned to the VMkernel, a corresponding directory should exist in `/proc/vmware/scsi`. The subdirectories of each controller will have a listing of devices, target IDs, and LUNs. For instance, if you had the `vmhba0` subdirectory with a `2:0` file, you'd want to perform the following commands at the CLI to view the file contents:

```
cat /proc/vmware/scsi/hba0/2:0 | less
```

After collecting your data, you can use it to configure your VMs in the Management Interface.

LUN Masking

Masking is also available if you don't want the VMkernel scanning or even accessing particular LUNs. Masking is generally performed for security reasons to prevent operating systems from accessing LUNs. You can accomplish masking by setting the DiskMaskLUNs field on the system's Options tab under Advanced Settings to this:

```
<adapter>:<target>:<comma separated LUN range>;
```

For example, if you wanted to mask LUNs 3, 4, and 50–60 on vmhba 1, target 4, and LUNs 5–9, 15, and 12–15 on vmhba 2, target 6, you'll need to set the DiskMaskLUNs option to `vmhba1:4:3,4,50-60;vmhba2:6:5-9,12-15`, as in Figure 2-2. You're prohibited from masking LUN 0. When using QLogic HBAs, you'll need to select the correct driver version as well (IBM, HP, or EMC).

Bus Sharing

Unlike masking, ESX provides for bus sharing. The bus sharing feature is useful for high-availability environments, such as clustering, where you want two VMs to access the same virtual disk. By default, ESX Server prevents this from happening: if you need to change the settings, you can use the Management Interface to set bus sharing to one of three options:

- **None**: VMs don't share disks.
- **Virtual**: VMs on the same host can share disks.
- **Physical**: VMS on different hosts can share disks.

You'll want to choose Virtual as the sharing type for building a cluster in a box, and you'll need to choose Physical if you want to hedge against hardware failure. Using physical bus sharing requires the virtual disk to be mutually accessible by each VM.

Persistent Binding

Persistent binding is also available for ESX Server VMs and affords the target IDs to be retained during reboots. Persistent bindings, the assignment of target IDs to specific Fibre Channel devices, are necessary if you're using the SAN for VM direct access: persistent bindings treat the SAN as a physical disk by directly mapping the SAN LUN as locally attached storage. As with all ESX Server administration, you can configure persistent binding at the console or through the Management Interface. As a word of caution, persistent binding can create major problems in FC-AL SAN topologies!

Summary

In this chapter, we reviewed no-nonsense techniques for budgeting, building, and installing Windows or Linux VM host solutions. We pointed out some network and system bottlenecks along the way, and we showed how to avoid them in bridged or NATed networks by installing and configuring multiple network and storage adapters in a teamed configuration. In addition, you know that when you use teamed network adapters, you must configure the connecting switch for Etherchannel and the appropriate teaming protocol. If you don't properly configure your teamed NICs, you may wind up with flapping links or loops that adversely impact the network, negating all performance gains.

Additionally, we discussed the importance and impact of network and storage considerations for host and guest VMs, including locally attached storage and SANs.

Now that you're aware of the techniques for selecting and preparing hardware for host systems, you're ready to move onto the next few chapters where we'll focus on hosting and installing VMs. We'll cover the techniques to help students, technology administrators, and sales professionals get the most out of virtualization by installing guest operating systems on both server and workstation VM applications.