"The perfect way to understand both the technology and benefits of virtualization" Simon Crosby, CTO, XenSource

Virtualization FOR DUMMIES®

Boost server utilization to maximize your IT ROI

A Reference for the Rest of Us!®

FREE eTips at dummies.com®

Bernard Golden, MBA

CEO of Navica, Inc. & expert blogger for CIO.com

Virtualization For Dummies®

Chapter 1: Wrapping Your Head around Virtualization

ISBN: 978-0-470-14831-0

WILEY

Chapter 1

Wrapping Your Head around Virtualization

In This Chapter

- Finding out what virtualization is
- Figuring out what has everyone so excited
- Seeing how virtualization is applied
- Working through the challenges of virtualization

It seems like everywhere you go these days, someone is talking about virtualization. Technical magazines trumpet the technology on their covers. Virtualization sessions are featured prominently at technology conferences. And, predictably enough, technology vendors are describing how *their* product is the latest word in virtualization.

If you have the feeling that everyone else in the world understands virtualization perfectly while you're still trying to understand just what it is and how you might take advantage of it, take heart. Virtualization *is* a new technology. (Actually, it's a pretty well-established technology, but a confluence of conditions happening just now has brought it into new prominence — more on that later in this chapter.) Virtualization *is* a technology being widely applied today with excellent operational and financial results, but it's by no means universally used or understood. That's the purpose of this book: to provide you with an introduction to the subject so that you can understand its promise and perils and create an action plan to decide whether virtualization is right for you, as well as move forward with implementing it should you decide it *is* right for you.

Sadly, not even this book can protect you from the overblown claims of vendors; there is no vaccine strong enough for that disease. This book helps you sort out the hope from the hype and gives you tools to feel confident in making your virtualization decisions.

Virtualization: A Definition

Virtualization refers to a concept in which access to a single underlying piece of hardware, like a server, is coordinated so that multiple guest operating systems can share that single piece of hardware, with no guest operating system being aware that it is actually sharing anything at all. (A *guest operating system* is an operating system that's hosted by the underlying virtualization software layer, which is often, you guessed it, called the *host system.*) A guest operating system appears to the applications running on it as a complete operating system (OS), and the guest OS itself is completely unaware that it's running on top of a layer of virtualization software rather than directly on the physical hardware.

Actually, you've had experience with something like this when you used a computer. When you interact with a particular application, the operating system "virtualizes" access to the underlying hardware so that only the application you're using has access to it — only your program is able to manipulate the files it accesses, write to the screen, and so on. Although this description oversimplifies the reality of how operating systems work, it captures a central reality: The operating system takes care of controlling how applications access the hardware so that each application can do its work without worrying about the state of the hardware. The operating system *encapsulates* the hardware, allowing multiple applications to use it.

In *server* virtualization — the most common type of virtualization — you can think of virtualization as inserting another layer of encapsulation so that multiple operating systems can operate on a single piece of hardware. In this scenario, each operating system believes it has sole control of the underlying hardware, but in reality, the virtualization software controls access to it in such a way that a number of operating systems can work without colliding with one another. The genius of virtualization is that it provides new capability without imposing the need for significant product or process change.

Actually, that last statement is a bit overbroad. A type of virtualization called paravirtualization does require some modification to the software that uses it. However, the resulting excellent performance can make up for the fact that it's a little less convenient to use. Get used to this exception business; the subject of virtualization is riddled with general truths that have specific exceptions. Although you have to take account of those exceptions in your particular project plans, don't let these exceptions deter you from the overarching circumstances. The big picture is what you need to focus on to understand how virtualization can help *you.*

WARNING!

Virtualization is actually a simple concept made complex by all the exceptions that arise in particular circumstances. It can be frustrating to find yourself stymied by what seems to be a niggling detail, but unfortunately, that's the reality of virtualization. If you stop to think about it, the complexity makes sense — you're moving multiple operating systems and applications onto a new piece of software called a *hypervisor,* which in turn talks to underlying hardware. Of course it's complex! But don't worry, if you hang in there, it usually comes out right in the end. Chapters 12 through 14 offer real examples of how to install several flavors of virtualization software and successfully put guest OSes onto the software. Work through the examples, and you'll be an expert in no time!

So, if you take nothing more away from this section than the fact that virtualization enables you to share a hardware resource among a number of other software systems, that's enough for you to understand the next topic — what's making virtualization so important *now.*

Why Virtualization Is Hot, Hot, Hot — The Four Drivers of Virtualization

Despite all the recent buzz about it, virtualization is by no means a new technology. Mainframe computers have offered the ability to host multiple operating systems for over 30 years. In fact, if you begin to discuss it, you might suffer the misfortune of having someone begin to regale you with tales of how *he* did virtualization in the old days.

The truth is that your old gaffer is right. Yeah, virtualization as a technology is nothing new, and yeah, it's been around for many years, but it was confined to "big iron" (that is, mainframes). Four trends have come together in just the past couple of years that have moved virtualization from the dusty mainframe backroom to a front-and-center position in today's computing environment.

REMEMBER

When you take a look at these trends, you can immediately recognize why virtualization is much more than the latest technology fad from an industry that has brought you more fads than the fashion industry.

Trend #1: Hardware is underutilized

I recently had an opportunity to visit the Intel Museum located at the company's headquarters in Santa Clara, California. The museum contains a treasure trove of computing — from a giant design-it-yourself chip game to a set of sterile clothes (called a *bunny suit*) you can put on while viewing a live camera feed from a chip-manufacturing plant. It's well worth a visit. But tucked near the back of the museum, in a rather undistinguished case, is ensconced one of the critical documents of computing. This document, despite its humble presentation, contains an idea that has been key to the development of computing in our time.

I refer to the article by Intel cofounder Gordon Moore in the April 1965 issue of *Electronics Magazine,* in which he first offered his observation about the compounding power of processor computing power, which has come to be known as "Moore's Law."

In describing the increasing power of computing power, Moore stated: "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year." Clearly, Moore wasn't in charge of marketing at Intel, but if you translate this into something the average human can understand, he means that each year (actually, most people estimate the timeframe at around 18 months), for a given size processor, twice as many individual components can be squeezed onto a similarly sized piece of silicon. Put another way, every new generation of chip delivers twice as much processing power as the previous generation — at the same price.

This rapid doubling of processing power has had a tremendous impact on daily life, to an extent that's difficult to comprehend for most people. Just over ten years ago, I ran the engineering group of a large enterprise software vendor. Everybody in the group knew that a major part of the hassles involved in getting a new release out the door involved trying to get acceptable performance out of the product on the then-latest generation of hardware. The hardware just wasn't that capable. Today's hardware, based upon the inexorable march of Moore's Law, is around 100,000 times as powerful — in ten short years!

What you need to keep in mind to understand Moore's Law is that the numbers that are continuing to double are themselves getting larger. So, if you take year one as a base, with, say, processing power of 100 million instructions per second (MIPS) available, then in year two, there will be 200; in year three, 400; and so on. Impressive, eh? When you get out to year seven or eight, the increase is from something like 6,400 to 12,800 in one generation. It has grown by 6,400. And the next year, it will grow by 12,800. It's mind boggling, really.

REMEMBER

Moore's Law demonstrates increasing returns — the amount of improvement itself grows over time because there's an exponential increase in capacity for every generation of processor improvement. It's that exponential increase that's responsible for the mind-boggling improvements in computing — and the increasing need for virtualization.

And that brings me to the meaning of this trend. Unlike ten years ago, when folks had to sweat to get software to run on the puny hardware that was available at that time, today the hardware is so powerful that software typically uses only a small portion of the available processing power. And this causes a different type of problem.

Today, many data centers have machines running at only 10 or 15 percent of total processing capacity. In other words, 85 or 90 percent of the machine's power is unused. In a way, Moore's Law is no longer relevant to most companies because they aren't able to take advantage of the increased power available to them. After all, if you're hauling a 50-pound bag of cement, having a truck come out that can carry 20,000 pounds instead of this year's model that can only carry 10,000 pounds is pretty irrelevant for your purposes. However, a lightly loaded machine still takes up room and draws electricity, so the cost of today's underutilized machine is nearly the same as if it was running at full capacity.

It doesn't take a rocket scientist to recognize that this situation is a waste of computing resources. And, guess what? With the steady march of Moore's Law, next year's machine will have twice as much spare capacity as this year's — and so on, for the foreseeable future. Obviously, there ought to be a better way to match computing capacity with load. And that's what virtualization does — by enabling a single piece of hardware to seamlessly support multiple systems. By applying virtualization, organizations can raise their hardware use rates from 10 or 15 percent to 70 or 80 percent, thereby making much more efficient use of corporate capital.

REMEMBER

Moore's Law not only enables virtualization, but effectively makes it mandatory. Otherwise, increasing amounts of computing power will go to waste each year.

So, the first trend that's causing virtualization to be a mainstream concern is the unending growth of computing power brought to you by the friendly folks of the chip industry. By the way, the same trend that's described in chips by Moore's Law can be observed in the data storage and networking arenas as well. They just don't have a fancy name for their exponential growth — so maybe Gordon Moore was a marketing genius after all. However, the rapid improvement in these other technology areas means that virtualization is being explored for them as well — and because I like being complete, I work in coverage of these areas later on in this book.

Trend #2: Data centers run out of space

The business world has undergone an enormous transformation over the past 20 years. In 1985, the vast majority of business processes were paper based. Computerized systems were confined to so-called backroom automation: payroll, accounting, and the like.

That has all changed, thanks to the steady march of Moore's Law. Business process after business process has been captured in software and automated, moving from paper to computers.

The rise of the Internet has exponentially increased this transformation. Companies want to communicate with customers and partners in real time, using the worldwide connectivity of the Internet. Naturally, this has accelerated the move to computerized business processes.

To offer a dramatic example, Boeing's latest airliner, the 787 Dreamliner, is being designed and built in a radically new way. Boeing and each of its suppliers use Computer-Aided Design (CAD) software to design their respective parts of the plane. All communication about the project uses these CAD designs as the basis for discussion. Use of CAD software enables testing to be done in computer models rather than the traditional method of building physical prototypes, thereby speeding completion of the plane by a year or more.

As you might imagine, the Dreamliner project generates enormous amounts of data. Just one piece of the project — a data warehouse containing project plans — runs to 19 terabytes of data.

Boeing's experience is common across all companies and all industries. How big is the explosion of data? In 2003, the world's computer users created and stored 5 exabytes (each exabyte is 1 million terabytes) of new data. A recent study by the Enterprise Strategy Group predicted that governments and corporations will store over 25 exabytes of data by the year 2010. Certainly, the trend of data growth within organizations is accelerating. The growth of data can be easily seen in one key statistic: In 2006, the storage industry shipped as much storage in one month as it did in the entire year of 2000. The research firm IDC estimates that total storage shipped will increase 50 percent per year for the next five years.

The net effect of all this is that huge numbers of servers have been put into use over the past decade, which is causing a real-estate problem for companies: They're running out of space in their data centers. And, by the way, that explosion of data calls for new methods of data storage, which I also address in this book. These methods go by the common moniker of *storage virtualization,* which, as you might predict, encapsulates storage and abstracts it from underlying network storage devices.

Virtualization, by offering the ability to host multiple guest systems on a single physical server, helps organizations to reclaim data center territory, thereby avoiding the expense of building out more data center space. This is an enormous benefit of virtualization because data centers cost in the tens of millions of dollars to construct. You can find out more about this in Chapter 4, where I discuss this trend, which is usually referred to as *consolidation* and is one of the major drivers for organizations to turn to virtualization.

Take a look at your data center to understand any capacity constraints you're operating with. If you're near capacity, you need virtualization — stat!

Trend #3: Energy costs go through the roof

In most companies' strategic thinking, budgeting power costs used to rank somewhere below deciding what brand of soda to keep in the vending machines. Companies could assume that electrical power was cheap and endlessly available.

Several events over the past few years have changed that mindset dramatically:

- The increasing march of computerization discussed in Trend #2, earlier in this chapter, means that every company is using more power as their computing processes expand.
- The assumption regarding availability of reliable power was challenged during the California power scares of a few years ago. Although later evidence caused some reevaluation about whether there was a true power shortage (can you say "Enron"?), the events caused companies to consider whether they should look for ways to be less power dependent.
- As a result of the power scares and Pacific Gas & Electric's resulting bankruptcy, power costs in California, home to Silicon Valley, have skyrocketed, making power a more significant part of every company's budget. In fact, for many companies, electricity now ranks as one of the top five costs in their operating budgets.

The cost of running computers, coupled with the fact that many of the machines filling up data centers are running at low utilization rates, means that virtualization's ability to reduce the total number of physical servers can significantly reduce the overall cost of energy for companies.

TIP

Data center power is such an issue that energy companies are putting virtualization programs into place to address it. See Chapter 5 to find out about an innovative virtualization rebate program Pacific Gas & Electric has put into place.

Trend #4: System administration costs mount

Computers don't operate on their own. Every server requires care and feeding by system administrators who, as part of the operations group, ensure that the server runs properly. Common system administration tasks include monitoring hardware status; replacing defective hardware components; installing operating system (OS) and application software; installing OS and application patches; monitoring critical server resources such as memory and disk use; and backing up server data to other storage mediums for security and redundancy purposes.

As you might imagine, this job is pretty labor intensive. System administrators don't come cheap. And, unlike programmers, who can be located in less expensive offshore locales, system administrators are usually located with the servers due to their need to access the physical hardware.

The steady increase in server numbers has meant that the job market for system administrators has been good — very good.

As part of an effort to rein in operations cost increases, virtualization offers the opportunity to reduce overall system administration costs by reducing the overall number of machines that need to be taken care of. Although many of the tasks associated with system administration (OS and application patching, doing backups) continue even in a virtualized environment, some of them disappear as physical servers are migrated to virtual instances. Overall, virtualization can reduce system administration requirements by 30 to 50 percent per virtualized server, making virtualization an excellent option to address the increasing cost of operations personnel.

REMEMBER

Virtualization reduces the amount of system administration work necessary for hardware, but it doesn't reduce the amount of system administration required for guest OSes. Therefore, virtualization improves system administration, but doesn't make it vanish.

Four trends mean virtualization is hot

Looking at these four trends, you can see why virtualization is a technology whose time has come. The exponential power growth of computers, the substitution of automated processes for manual work, the increasing cost to power the multitude of computers, and the high personnel cost to manage that multitude all cry out for a less expensive way to run data centers. In fact, a newer, more efficient method of running data centers is critical because, given the four trends, the traditional methods of delivering computing are becoming cost prohibitive. Virtualization is the solution to the problems caused by the four trends I outline here.

Sorting Out the Types of Virtualization

If you've made it this far in this chapter, you (hopefully) have a rough idea of virtualization and why it's an important development. Your next step involves determining what your options are when it comes to virtualization. In other words, what are some common applications of the technology?

Virtualization has a number of common uses, all centered around the concept that virtualization represents an abstraction from physical resources. In fact, enough kinds of virtualization exist to make it a bit confusing to sort out how you might apply it in your organization.

I do what I can to sort out the virtualization mare's nest. If you're okay with gross generalizations, I can tell you that there are three main types of virtualization: client, server, and storage. Within each main type are different approaches or *flavors,* each of which has its benefits and drawbacks. The next few sections give brief descriptions of each of the three types of virtualization, along with examples of common implementations of them.

Client virtualization

Client virtualization refers to virtualization capabilities residing on a *client* (a desktop or laptop PC). Given that much of the earlier discussion of the driving forces behind virtualization focuses on the problems of the data center, you might wonder why virtualization is necessary for client machines at all.

REMEMBER

The primary reason organizations are interested in pursuing client virtualization solutions has to do with the challenges they face in managing large numbers of computers controlled by end users. Although machines located in data centers typically have strict procedures about what software is loaded on them and when they're updated with new software releases, end user machines are a whole different story.

Because loading software is as easy as sticking a disc into the machine's CD drive (or a thumb drive into a USB slot), client machines can have endless amounts of non-IT-approved software installed. Each application can potentially cause problems with the machine's operating system as well as other approved applications. Beyond that, other nefarious software can get onto client machines in endless ways: via e-mail viruses, accidental spyware downloads, and so on. And, the hard truth is that Microsoft Windows, the dominant client operating system, is notorious for attracting attacks in the form of malware applications.

Added to the end user–caused problems are the problems inherent to client machines in general: keeping approved software applications up to date, ensuring the latest operating system patches are installed, and getting recent virus definitions downloaded to the machine's antivirus software.

Mixed together, this stew is a miserable recipe for IT. Anything that makes the management of client machines easier and more secure is of definite interest to IT. Client virtualization offers the potential to accomplish this.

Three main types — or flavors, if you will — of client virtualization exist: application packaging, application streaming, and hardware emulation.

Application packaging

Although the specifics of how application packaging is accomplished vary from one vendor to another, all the methods share a common approach: isolating an application that runs on a client machine from the underlying operating system. By isolating the application from the operating system, the application is unable to modify underlying critical operating system resources, making it much less likely that the OS will end up compromised by malware or viruses.

You can accomplish this application-packaging approach by executing the application on top of a software product that gives each application its own virtual set of system resources — stuff like files and registry entries. Another way to accomplish application packaging is by bundling the application and the virtualization software into a single executable program that is downloaded or installed; when the executable program is run, the application and the virtualization software cooperate and run in an isolated (or *sandboxed*) fashion, thereby separating the application from the underlying operating system.

WARNING!

Application packaging is a great way to isolate programs from one another and reduce virus transmission, but it doesn't solve the problem of end users installing nonpackaged software on client machines.

One thing to keep in mind with this approach is that it causes additional work as the IT folks prepare the application packages that are needed and then distribute them to client machines. And, of course, this approach does nothing to solve the problem of end users installing other software on the machine that bypasses the application packaging approach altogether. If you're loading a game onto your business laptop, you're hardly likely to go to IT and request that someone create a new application package so that you can run your game securely, are you?

Products that provide application packaging include SVS from Altiris, Thinstall's Virtualization Suite, and Microsoft's SoftGrid.

Application streaming

Application streaming solves the problem of how to keep client machines loaded with up-to-date software in a completely different fashion than application packaging. Because it's so difficult to keep the proper versions of applications installed on client machines, this approach avoids installing them altogether. Instead, it stores the proper versions of applications on servers in the data center, and when an end user wants to use a particular application, it's downloaded on the fly to the end user's machine, whereupon he or she uses it as though it were natively installed on the machine.

This approach to client virtualization can reduce the amount of IT work necessary to keep machines updated. Furthermore, it happens transparently to the end user because the updated application is automatically delivered to the end user, without any physical software installation on the client. It also has the virtue of possibly allowing client machines less capability to be deployed because less disk space is required to permanently store applications on the client hard drive. Furthermore, if this approach is taken to its logical conclusion and the client machine has no hard drive, it is possible that less memory is required because only the official IT applications can be executed on the machine. This result is because the end user can't execute any programs other than the ones available from the central server.

Although at first glance, this approach might seem like a useful form of virtualization, it is really appropriate only in certain circumstances — primarily situations in which end users have constant connectivity to enable application downloads when required. Examples of these situations include call centers and office environments where workers rarely leave the premises to perform work duties. In today's increasingly mobile workforce world, these circumstances apply to a small percentage of the total workforce. Perhaps the best way to think about this form of virtualization is as one that can be very useful in a restricted number of work environments.

This type of virtualization is offered by AppStream's Virtual Image Distribution, Softricity's Softgrid for Desktops, and Citrix's Presentation Server. Softricity has recently been acquired by Microsoft, and its SoftGrid product will soon be available as part of the Windows Server platform. SoftGrid will offer the capability of streaming applications to remote desktops.

REMEMBER

Application streaming is best suited for static work environments where people don't move around much, such as call centers and form-processing centers, although some organizations are exploring using it for remote employees who have consistent network connectivity to ensure that applications can be streamed as necessary.

Hardware emulation

Hardware emulation is a very well-established form of virtualization in which the virtualization software presents a software representation of the underlying hardware that an operating system would typically interact with. (I discuss hardware emulation in more detail in the "Server virtualization" section, later in this chapter.) This is a very common type of virtualization used in data centers as part of a strategy to get higher utilization from the expensive servers that reside in them.

Because of the spread of *commodity hardware* (that's to say, hardware based on Intel's x86 chip architecture; these chips power everything from basic desktop machines to huge servers), the same hardware emulation type of virtualization that can be used in data centers can also be used on client machines. (The term *commodity* refers to the fact that the huge volumes of x86 processors sold make them so ubiquitous and inexpensive that they're almost like any other mass-produced, unspecialized product — almost as common as the canned goods you can get in any grocery store.

In this form of client virtualization, the virtualization software is loaded onto a client machine that has a base operating system already loaded — typically Windows, but client hardware emulation virtualization is also available for systems running Mac and Linux operating systems.

After the hardware emulation software is loaded onto the machine, it's ready to support guest operating systems. Guest OSes are installed via the virtualization software; that is, rather than just sticking a CD into the machine's drive and rebooting it to install the operating system directly onto the hardware, you use the virtualization software's control panel to indicate your desire to install a guest OS (which can be either Windows or Linux). It sets up the container (often called the *virtual machine,* or VM for short) for the guest operating system and then directs you to put the CD in the drive, whereupon the normal installation procedure occurs.

After the installation completes, you control the virtual machine (which is a normal Windows or Linux system) through the virtualization software's control panel. You can start, stop, suspend, and destroy a VM from the control panel.

Interacting with the VM guest OS is just like interacting with it if it were the only OS on the machine. A guest OS displays graphics on the screen, the VM responds to keyboard and mouse commands, and so on. That's why it's called virtualization!

Products offering this type of virtualization are VMware's VMware Server and Microsoft's Virtual Server. On the Macintosh, SWsoft's Parallels product provides hardware emulation virtualization.

Server virtualization

When discussing trends driving virtualization, you'll soon discover that most of the examples that come up are focused on issues of the data center — the server farms that contain vast arrays of machines dedicated to running enterprise applications, databases, and Web sites.

That's not an accident. Most of the action in the virtualization world right now focuses on server virtualization — no surprise, then, if you see me spending most of my time in this book on precisely that topic.

IT organizations are avidly pursuing virtualization to gain more control of their sprawling server farms. Although client virtualization is interesting to them, server virtualization is critical because many IT organizations are running out of room in their data centers. Their inability to add more machines means they can't respond to important business initiatives, meaning they can't deliver necessary resources so that the other parts of the business can implement the company's strategy. Obviously, this inability to provide IT resources is unacceptable, and many, many IT organizations are turning to server virtualization to solve this problem.

Three main types of server virtualization exist:

- **Operating system virtualization:** Often referred to as *containers*
- **Hardware emulation:** Similar to the same type of virtualization described in the client virtualization section, earlier in the chapter

- **Paravirtualization:** A relatively new concept designed to deliver a lighter-weight (in terms of virtualization application size), higher-performance approach to virtualization

Check out the next few sections for an in-depth treatment of each of these three types.

Each type of server virtualization has its pros and cons. It's important to evaluate your likely use of virtualization to understand which virtualization technology is best suited for your needs. See Chapter 7 for a discussion of how to evaluate virtualization use.

Operating system virtualization (containers)

In the preceding "Client virtualization" section, I talk about hardware emulation being a virtualization architecture that installs a piece of software onto a machine. Guest operating systems are subsequently installed using the hardware emulation software. Many times, this approach to virtualization, in which the virtualization software is installed directly onto the machine, is described as a *bare-metal* approach, meaning there is no software between the virtualization software and the underlying hardware.

Operating system virtualization, by contrast, is installed on top of an existing operating system. It doesn't enable installation of virtual machines, each of which is isolated from any other virtualization machine. Rather, operating system virtualization runs on top of an existing host operating system and provides a set of libraries that applications interact with, giving each application the illusion that it is running on a machine dedicated to its use.

If this seems a bit confusing, take a look at Figure 1-1, which illustrates the concept. Here you can see a server running a host operating system. *That* operating system is running software that provides operating system virtualization, and a number of virtual OSes are running within the operating system virtualization software. Each of the virtual OSes has one or more applications running within it. The key thing to understand is that, from the application's execution perspective, it sees and interacts only with those applications running within its virtual OS, and it interacts with its virtual OS as though it has sole control of the resources of the virtual OS. Crucially, it can't see the applications or the OS resources located in another virtual OS. It's as though multiple operating systems are running on top of the real host OS. You can see why this approach to virtualization is often referred to as *containers:* Each set of applications is contained within its assigned virtual OS and cannot interact with other virtual OSes or the applications running in those virtual OSes.

You might wonder what use this approach to virtualization is. It can be extremely useful if you want to offer a similar set of operating system functionalities to a number of different user populations while using only a single machine. This is an ideal approach for Web-hosting companies: They use operating system virtualization to allow a hosted Web site to believe it has complete control of a machine, but in fact each hosted Web site shares the machine with many other Web sites, each of which is provided its own container. Every container has its own file system that looks like a complete operating system, but in fact the file system is mapped to the underlying host OS file system in such a way that it isolates each Web site's file system from the others. Another common use for operating system virtualization is when different organizations within a company each need to be provided with their own server, each of which is identical to the others.

The benefit of operating system virtualization is its efficiency. Rather than running a number of complete guest OSes so that applications can each have access to dedicated operating resources, operating system virtualization uses a set of libraries to provide operating system functionality and file-mapping services; consequently, much less software is required to enable the applications to be isolated. Therefore, operating system virtualization is quite efficient, enabling high performance for the overall system. Put another way, operating system virtualization imposes little overhead for the virtualization capability achieved, thereby ensuring most of the machine's resources are available to the applications running in the containers. This has the effect of allowing more containers to run on a given piece of hardware than would be possible with the more heavyweight hardware emulation virtualization approach.

Also, rather than having to license separate instances of guest OSes, operating system virtualization requires only one copy of the underlying OS while providing OS services via the virtualization software. The reduction in licensing costs for the guest OSes can be quite significant.

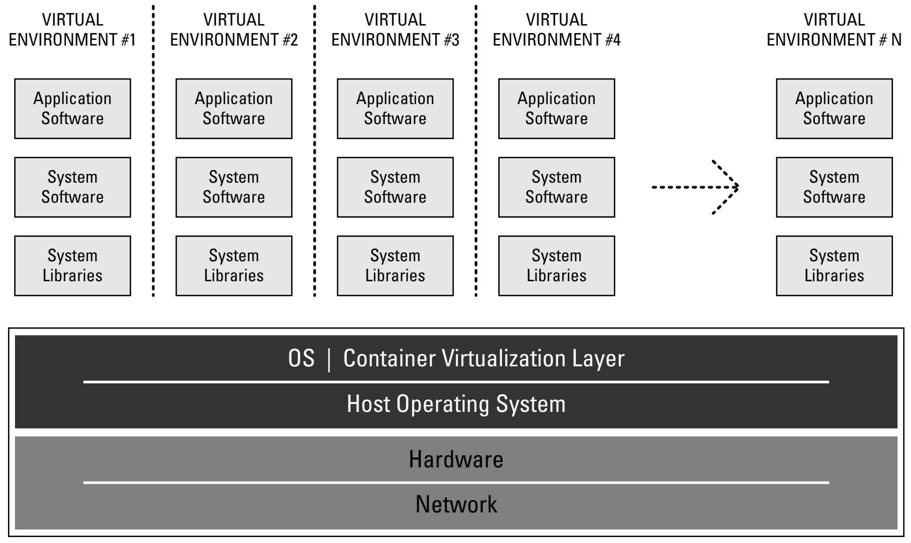

Figure1-1: Operating system virtualization.

Operating system virtualization has some limitations, though. First and foremost, this approach typically limits operating system choice. Containerization usually means that the containers offer the same operating system as the host OS. In other words, if the host OS is Linux, only Linux containers are available.

For many application profiles, having to stick with the same OS isn't really a problem. If you're supporting multiple Web sites, it's usually okay — or even ideal — that every guest OS is the same. However, for certain application profiles, it's not enough to just stick with the same general OS; the containers also have to be consistent with the host OS in terms of version number and even patch level. This need for an exact match can cause problems if you want to run different applications in the containers because applications are often certified for only a certain OS version and patch level; if you have different applications that require different certified OS and patch levels, you have a problem. Consequently, operating system virtualization is best suited for homogeneous configurations — for those arrangements, going the operating system virtualization route is an excellent choice.

Operating system virtualization is well suited when an organization requires large numbers of homogeneous operating environments. It's not such a good choice when different environments are required due to application requirements or operating system differences in version or patch level.

Companies offering operating system virtualization include Sun (as part of the Solaris operating system) and SWsoft, which offers the commercial product Virtuozzo and sponsors the open source operating system virtualization project called OpenVZ.

Hardware emulation

In hardware emulation (see Figure 1-2), the virtualization software (usually referred to as a *hypervisor*) presents an emulated hardware environment that guest operating systems operate upon. This emulated hardware environment is typically referred to as a *virtual machine monitor* (VMM). The guest OSes are installed in a *virtual machine* (VM) that interacts with the VMM rather than to the physical hardware upon which the virtualization software runs. In other words, rather than an operating system being installed on a physical machine, it's installed on a virtual machine that emulates the hardware that the OS would usually interact with. The VMM coordinates access between the guest VMs and the actual underlying hardware.

Because the guest OS and the VM are both stored in files that together form a complete system image, the image can be migrated from one hypervisor to another even though the hypervisors reside on different physical machines. This virtual machine portability offers great flexibility for system deployment and forms the basis for most of the advanced applications of virtualization, which are described in Chapter 3.

Virtualization enables operating systems that have been designed and delivered to run directly on hardware to run on an intervening layer of software known as a hypervisor. In order for a guest operating system to interact with the hypervisor, those parts of it that make hardware calls must be modified to make calls to the hypervisor instead. The hypervisor takes those guest OS calls and passes them along to the underlying physical hardware. In short, the guest OS must be modified in order to take advantage of virtualization.

Hardware emulation is a powerful virtualization technology because it performs the guest OS modifications at runtime through a technique known as Binary Translation; essentially, when the guest OS is installed, the virtualization software cleverly rearranges the internal guest OS software so that the calls that originally attempted to access physical hardware resources are now directed to resources within the VMM. Consequently, it is quite easy to use hardware emulation, because you install unmodified guest OSes, and the hardware emulation transparently takes care of modifying the various bits so that the entire virtualization assemblage works properly.

TECHNICAL STUFF

Hardware emulation takes advantage of the well-known technical architecture of the x86 chip to create a software emulation of the chip. Because the commodity x86 chip is so widely used, there's a large enough installed base to make hardware emulation virtualization economically viable.

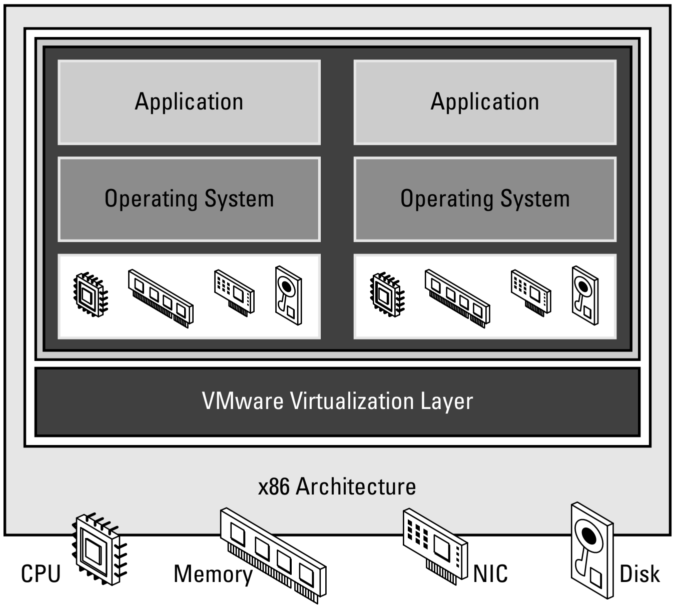

Figure 1-2: Hardware emulation virtualization.

Because the hardware emulation hypervisor presents a consistent interface to guest OSes and takes responsibility for figuring out how to interact with the underlying hardware, this form of virtualization is quite flexible because it allows a guest OS installed and run on one hypervisor to be moved and run on a second hypervisor.

This approach to virtualization means that, rather than running in containers as happens with operating system virtualization (see the previous section), applications run in a truly isolated guest OS with one or more guest OSes running, one per VMM. The VMMs all reside on the virtualization hypervisor, no matter how many happen to be running at any given time. Not only does this approach support multiple OSes, it can support dissimilar OSes, differing in minor ways (such as version and patch level) or in major ways (for example, completely different OSes like Windows and Linux can simultaneously be run in hardware emulation virtualization software).

Common applications for hardware emulation are software development and quality assurance because they allow a number of different OSes to be run simultaneously, thereby facilitating parallel development or testing of software in a number of different operating system environments. Hardware emulation is also used in server consolidation, where a number of operating system and/or application environments are moved from separate physical servers to a single physical server running virtualization software.

If all this makes it sound as though hardware emulation is just what the IT doctor ordered, you need to know that the technology has a couple of drawbacks:

- **Performance penalty:** One drawback is that the virtualization software hurts performance, which is to say that applications often run somewhat slower on virtualized systems than they would if they were running on unvirtualized systems. This makes sense if you think about it. After all, you've piled things on pretty thick, what with the application you're working with, the operating system it runs on, and the VMM/hypervisor layer you've inserted below the operating system. Getting through that VMM/hypervisor layer is bound to take some machine cycles and thereby reduce performance somewhat. In practice, the benefits of virtualization usually outweigh performance concerns.
- **Device driver availability:** Another drawback to hardware emulation is that the virtualization software presents a standardized hardware interface to the guest operating system. The hypervisor translates requests for hardware resources by the guest OS into calls to the actual physical resources on the machine. This means that the hypervisor must contain the interfaces to the resources of the machine; these resources are referred to as *device drivers.* If you've ever installed new hardware in a PC, you know that you often have to install a device driver into the

operating system so that the new hardware and the operating system can communicate. If there is hardware present on the machine that the hypervisor does not have drivers for, the guest OSes cannot access the capabilities of the underlying hardware. This can be a problem if an organization wants to, for example, connect a new type of storage to the hardware; if the hypervisor does not contain a driver for that new storage device, it cannot be used.

- **Device driver inflexibility:** If you install new hardware for a typical operating system and it does not contain a driver for it, the OS will ask you to load a new driver. This dynamic driver capability makes it possible to update the OS to reflect hardware additions. Hardware emulation hypevisors, by contrast, do not have the ability to dynamically load device drivers; hardware support is limited to those products that the hypervisor could support at time of shipment. This can cause problems, especially for organizations that want to take advantage of new hardware developments. For example, organizations that want to move to the latest storage options can find that they're blocked from doing so because their virtualization software doesn't yet support a particular storage hardware device. Depending upon your organization's computing infrastructure, this might or might not be a problem, but it's definitely something to check out before embarking upon your virtualization journey.

REMEMBER

Before committing to hardware emulation, be sure to survey your hardware environment to evaluate whether your potential hardware emulation virtualization product can fully support your needs.

Companies offering hardware emulation virtualization software include VMware (in two versions, VMware Server and ESX Server) and Microsoft, which offers a product called Virtual Server. VMware is the undisputed leader in the virtualization marketplace, and I discuss it extensively throughout this book. Microsoft's Virtual Server has much less presence in the marketplace and will be superseded by the virtualization functionality contained in Microsoft Server 2008, which is referred to as Microsoft Server virtualization. It turns out that Microsoft Server virtualization takes more of a paravirtualization approach to virtualization, which I get to discuss in the next section.

I show you how to install and use VMware Server, one of the VMware products, in Chapter 12.

Paravirtualization

Paravirtualization is the name for another approach to server virtualization. In this approach, rather than emulate a complete hardware environment, the virtualization software is a thin layer that *multiplexes* (that is, coordinates) access by guest operating systems to the underlying physical machine resources; in other words, paravirtualization doesn't create an entire virtual machine to host the guest OS, but rather enables the guest OS to interact directly with the hypervisor (see Figure 1-3). (By the way, the "para" in paravirtualization doesn't really mean anything. It's actually just a smart-alec term used to sneer at hardware emulation virtualization.)

This approach has two advantages: First, it imposes less performance overhead because it uses a very small amount of code. Hardware emulation (see the previous section) inserts an entire hardware emulation layer between the guest operating system and the physical hardware. By contrast, paravirtualization's thin software layer acts more like a traffic cop, allowing one guest OS access to the physical resources of the hardware while stopping all other guest OSes from accessing the same resources at the same time.

The second advantage of the paravirtualization approach when compared to hardware emulation is that paravirtualization doesn't limit you to the device drivers contained in the virtualization software. In fact, paravirtualization doesn't include any device drivers at all. Instead, it uses the device drivers contained in one of the guest operating systems — the one that's referred to as the *privileged guest* (Microsoft uses the term root partition to refer to the same thing). Without going into too much detail about this architecture here, suffice it to say that this is an advantage because it enables organizations to take advantage of all the capabilities of the hardware in the server, rather than being limited to hardware for which drivers are available in the virtualization software, as with hardware emulation virtualization.

TECHNICAL STUFF

Paravirtualization uses an operating system capability called *shared memory* to achieve high performance. Shared memory is memory that can be accessed by two different programs. Paravirtualization uses shared memory to send data back and forth between guest OSes and the hypervisor, thereby achieving high performance levels.

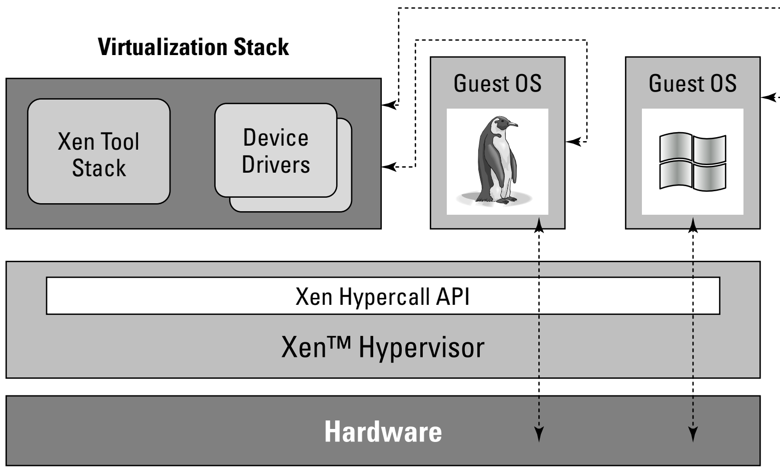

Figure 1-3: Paravirtualization at work.

Okay, so you clearly have a winner with the paravirtualization approach, right? Not so fast, cowpoke. There's one significant drawback to this approach to virtualization: Because it's so lightweight and yet still has to multiplex access to the underlying hardware, paravirtualization needs some help with the heavy lifting. In other words, it requires that the guest operating systems be modified prior to being run to interact with the paravirtualization interfaces. This can be accomplished only by having access to the source code of the guest OS. This access is possible for open source operating systems such as Linux or any of the BSD (Berkeley Software Distribution, a Unix-like open source operating system) flavors, but — crucially — it isn't possible for any Microsoft product. As you might imagine, the inability to virtualize Microsoft OSes is a significant drawback for many organizations. The good news is that the latest generation of chips from Intel and AMD provide functionality (called Intel VT and AMD-V, respectively) that enables unmodified operating systems, particularly Microsoft Windows, to be hosted by a paravirtualized hypervisor. Consequently, this drawback to paravirtualization will diminish as servers with these new chips take their place in production infrastructures.

The new chips from Intel and AMD are usually called *virtualization-enabled,* indicating that they have additional capability that allows some functionality that formerly was provided by the hypervisor to be moved to the chip. Because silicon (the chip) is ordinarily vastly faster than software, virtualization-enabled chips offer better virtualization performance.

The best-known product with the paravirtualization approach is a relatively new open source offering called Xen, which is sponsored by a commercial company called XenSource. Xen is included in the recent Linux distributions from Red Hat and Novell, as well as being available for many community Linux distributions like Debian and Ubuntu. XenSource itself sells Xen-based products as well. The forthcoming Microsoft Server virtualization is based on a paravirtualization approach to virtualization, and is scheduled to be available shortly after Microsoft Server 2008 is released.

Storage virtualization

The amount of data that organizations are creating and storing is exploding. Due to the increasing shift of business processes to Web-based digital applications, every company is being inundated with data.

This explosion of data is causing the following problems for many such companies:

- From a sheer storage capacity, many applications generate more data than can be stored physically on a single server.

- Many applications — particularly Internet-based ones — have multiple machines that need to access the same data. Having all the data sitting on one physical machine creates a bottleneck, not to mention the potential risk of a situation where many virtual machines might be made inoperable if a single physical machine containing all the application's data crashes.
- The explosion of machines mentioned earlier in the chapter causes backup problems; in other words, trying to create safe copies of data is a Herculean task when hundreds or even thousands of machines need data backup.

For these reasons, data has moved into virtualization as well. Moving data from many physical servers to a central location enables backups to be performed more efficiently. Furthermore, the central data repository can be configured with multiple physical storage devices to ensure that no hardware crash will ever make the organization's data unavailable. (For more information on data virtualization, see Chapter 11.) The three basic approaches to data storage are

- Direct-Attached Storage
- Network-Attached Storage
- Storage Area Network

I cover each of these options in the following sections.

Direct-Attached Storage

Direct-Attached Storage (DAS) is the traditional mode of data storage: hard drives attached to whatever physical server is running the application. DAS is easy to use but can be hard to manage. I talk in later chapters about why virtualization tends to cause organizations to reconsider sticking with a DAS architecture and think about moving to more sophisticated methods of data storage.

Network-Attached Storage

Network-Attached Storage (NAS) is a machine that sits on your network and offers storage to other machines. If you've ever used a remote drive in Windows, you've experienced NAS. However, NAS can be much more sophisticated than remote Windows drives; you can find specialized hardware appliances that can be filled with many hard drives and thereby provide multiple terabytes of storage.

You might think of NAS as the first step toward storage virtualization. NAS provides a single source of data, facilitating data backup. By collecting your data in one place, it also avoids the problem of multiple servers needing to access data located on another server.

NAS uses standard data communication protocols to send data back and forth between a server and the NAS device. This makes things simple but puts data traffic that's limited to the server in DAS usage onto the corporate network, which often causes a ripple effect as additional network capacity and hardware are required to support the NAS data traffic. See, I told you virtualization is a journey!

Although NAS helps with some of the problems associated with DAS, it carries its own set of issues. The network traffic caused by moving data back and forth from the NAS to individual servers can be considerable, stressing network capacity. The traffic can also interfere with other work carried on the network. And, of course, moving all data to a single NAS appliance just moves your risk because all your servers now depend on the NAS being available; if it goes down, everything goes down. For that reason, production environments that require absolute data availability typically use the Storage Area Network approach to data storage.

One advantage of NAS is that it's very cost effective. Some NAS appliances out on the market cost less than $1,000, which, when measured against the cost of labor to back up numerous machines, is a small drop in the budget bucket.

Storage Area Network

What if you have rapidly growing data storage needs and also need the assurance that your data is effectively backed up and that, more important, you're protected from data outages due to hardware failure? In that case, a Storage Area Network (SAN) is for you. Naturally, this being the technology industry, they had to choose an acronym that's easy to confuse with a similar solution — NAS, described in the preceding bullet.

SANs use highly specialized hardware and software to transform mere disk drives into a data storage solution that transfers data on its own high-performance network. SANs provide the ability to add additional storage capacity as data storage requirements grow; also, SANs can be configured to use multiple, redundant pieces of storage hardware so that data is always available, even if one or more pieces of storage hardware fail.

Within a SAN architecture, servers don't bother with transferring data across the standard corporate network — the standard Network Interface Card (NIC) to Ethernet cable to corporate network route. Instead, they usually use their own SAN interface device called a Host Bus Adapter (HBA) to connect to their SAN.

SANs use a specialized network protocol — either something called Fibre Channel or the alternative choice, iSCSI — for SAN network communication. At the other end of the SAN network is a hardware device that holds drives. This SAN device makes it easy to shift physical data storage, increase it, back it up, and even ship copies of the data to other locations to ensure business continuity in case of a natural disaster.

Companies move to SAN storage when they recognize that corporate data is a key resource that must be available 24/7 and needs to be conveniently managed. As you might expect, the price tag for that capability is large — very large. SANs can cost upwards of several million dollars, but when you consider the value of data, that level of investment is warranted for many companies.

Fibre Channel SANs use specialized data communication protocols to send data back and forth between a server and the data storage device on a separate network. Because the protocols are different, SANs require specialized hardware to be present on the server. By contrast, iSCSI-based SANs are able to use the standard corporate network for their data traffic, although this can raise the same kind of network congestion issues posed by NAS storage.

Creating the Virtualized Enterprise

If you're convinced that virtualization deserves a closer look and you want to know how to implement it in your organization, this book is for you. The purpose of the entire book is to give you information about how to evaluate virtualization, figure out which type of virtualization (and virtualization product) is right for you, and spell out the different ways you can migrate to (and manage) a virtualized environment. However, I'll give you the one-page preview right here. Here goes.

Most of the energy about virtualization in the industry focuses on server virtualization, and that's what this book primarily focuses on.

A typical first step in virtualization is a so-called server-consolidation project. This is a fancy term for moving the operating system and applications from a number of physical servers to a single server running virtualization software so that you can then run all the operating system/application configurations on a single physical server. (All of the pre-existing physical servers have been *consolidated* onto a single box — get it?)

Server consolidation provides a great payoff. Many organizations then decide to go further and create a pool of virtualized servers to enable load balancing and failover (that is, the ability to bring a new application up if a previous one crashed). Of course, after you move beyond server consolidation, you pretty much have to begin virtualizing your storage.

Although moving to virtualization might seem exciting (and, if not exciting, it can certainly be financially rewarding), it's not entirely painless. There's new software to obtain, people who have to be trained to use it, and project work that needs to be done to migrate all your existing physical servers to the new virtualized environment. All these things are necessary; failing to do any one of them raises the probability that you won't have a good outcome for your virtualization project. Fortunately, I address these topics in this book and provide tools for you to better find virtualization happiness.

In this book, I introduce the concept of the *virtualization life cycle,* the overall process of successfully introducing virtualization to your organization. Many people feel that all they need to do to be successful is buy some virtualization software and install it. That might get them started on the virtualization life cycle, but doesn't complete it. You'll be more successful if you address all the steps in the life cycle.

Okay, the whirlwind tour of virtualization has concluded. Time to move on to the real thing.