



Matthew Portnoy

# Virtualization

## ESSENTIALS

## Understanding Virtualization

*We are in the* midst of a substantial change in the way computing services are provided. As a consumer, you surf the web on your cell phone, get directions from a GPS device, and stream movies and music from the cloud. At the heart of these services is *virtualization*—the ability to abstract a physical server into a virtual machine.

In this chapter, you will explore some of the basic concepts of virtualization, review how the need for virtualization came about, and learn why virtualization is a key building block to the future of computing.

- ▶ **Describing virtualization**
- ▶ **Understanding the importance of virtualization**
- ▶ **Understanding virtualization software operation**

### Describing Virtualization

Over the last fifty years, certain key trends created fundamental changes in how computing services are provided. Mainframe processing drove the sixties and seventies. Personal computers, the digitization of the physical desktop, and client/server technology headlined the eighties and nineties. The Internet, boom and bubble, spanned the last and current centuries and continues today. We are, though, in the midst of another of those model-changing trends: virtualization.

Virtualization is a disruptive technology, shattering the status quo of how physical computers are handled, services are delivered, and budgets are allocated. To understand why virtualization has had such a profound effect on today's computing environment, you need to have a better understanding of what has gone on in the past.

The word *virtual* has undergone a change in recent years. Not the word itself, of course, but its usage has been expanded in conjunction with the expansion of computing, especially with the widespread use of the Internet and smart phones. Online applications have allowed us to shop in virtual

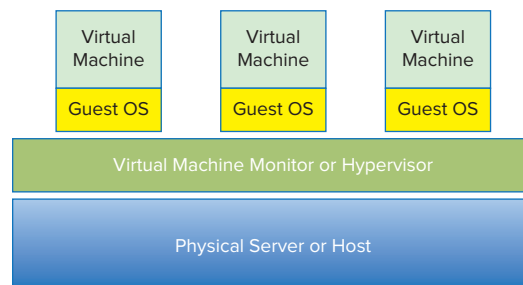
stores, examine potential vacation spots through virtual tours, and even keep our virtual books in virtual libraries. Many people invest considerable time and actual dollars as they explore and adventure through entire worlds that exist only in someone's imagination and on a gaming server.

Virtualization in computing often refers to the abstraction of some physical component into a logical object. By virtualizing an object, you can obtain some greater measure of utility from the resource the object provides. For example, Virtual LANs (local area networks), or VLANs, provide greater network performance and improved manageability by being separated from the physical hardware. Likewise, storage area networks (SANs) provide greater flexibility, improved availability, and more efficient use of storage resources by abstracting the physical devices into logical objects that can be quickly and easily manipulated. Our focus, however, will be on the virtualization of entire computers.

If you are not yet familiar with the idea of computer virtualization, your initial thoughts might be along the lines of virtual reality—the technology that, through the use of sophisticated visual projection and sensory feedback, can give a person the experience of actually being in that created environment. At a fundamental level, this is exactly what computer virtualization is all about: it is how a computer application experiences its created environment.

The first mainstream virtualization was done on IBM mainframes in the 1960s, but Gerald J. Popek and Robert P. Goldberg codified the framework that describes the requirements for a computer system to support virtualization. Their 1974 article “Formal Requirements for Virtualizable Third Generation Architectures” describes the roles and properties of virtual machines and virtual machine monitors that we still use today. The article is available for purchase or rent at <http://dl.acm.org/citation.cfm?doi=361011.361073>. By their definition, a virtual machine (VM) can virtualize all of the hardware resources, including processors, memory, storage, and network connectivity. A virtual machine monitor (VMM), which today is commonly called a *hypervisor*, is the software that provides the environment in which the VMs operate. Figure 1.1 shows a simple illustration of a VMM.

Some examples of virtual reality in popular culture are the file retrieval interface in Michael Crichton's *Disclosure*, *The Matrix*, and *Star Trek: The Next Generation's* holodeck.



**FIGURE 1.1** A basic virtual machine monitor (VMM)

According to Popek and Goldberg, a VMM needs to exhibit three properties in order to correctly satisfy their definition:

**Fidelity** The environment it creates for the VM is essentially identical to the original (hardware) physical machine.

**Isolation or Safety** The VMM must have complete control of the system resources.

**Performance** There should be little or no difference in performance between the VM and a physical equivalent.

Because most VMMs have the first two properties, VMMs that also meet the final criterion are considered *efficient* VMMs. We will go into these properties in much more depth as we examine hypervisors in Chapter 2, “Understanding Hypervisors,” and virtual machines in Chapter 3, “Understanding Virtual Machines.”


Let’s go back to the virtual reality analogy. Why would you want to give a computer program a virtual world to work in, anyway? It turns out that it was very necessary. To help explain that necessity, let’s review a little history. It would be outside the scope of this text to cover all the details about how server-based computing evolved, but for our purposes, we can compress it to a number of key occurrences.

## Microsoft Windows Drives Server Growth

Microsoft Windows was developed during the 1980s primarily as a personal computer operating system. Others existed, CPM and OS/2 for example, but as you know Windows eventually dominated the market and today it is still the primary operating system deployed on PCs. During that same time frame, businesses were depending more and more on computers for their operations. Companies moved from paper pushing to running their accounting, human resources, and many other industry-specific and custom-built applications on mainframes or minicomputers. These computers usually ran vendor-specific operating systems, making it difficult, if not impossible, for companies and IT professionals to easily transfer information among noncompatible systems. This led to the need for *standards*, agreed upon methods for exchanging information, but also the idea that the same, or similar, operating systems and programs should be able to run on many different vendors’ hardware. The first of these was Bell Laboratories’ commercially available UNIX operating systems.

Companies had both Windows-based PCs and other operating systems in-house, managed and maintained by their IT staffs, but it wasn’t cost effective to train

Between the late 1970s and mid-1980s there were more than 70 different personal computer operating systems.



**Current versions of Microsoft Windows run concurrent applications much more efficiently than their predecessors.**

IT staffs on multiple platforms. With increasing amounts of memory, faster processors, and larger and faster storage subsystems, the hardware that Windows could run on became capable of hosting more powerful applications that had in the past run on minicomputers and mainframes. These applications were being migrated to, or being designed to run on, Windows servers. This worked well for companies because they already had Windows expertise in house and no longer required multiple teams to support their IT infrastructure. This move, however, also led to a number of challenges. Because Windows was originally designed to be a single-user operating system, a single application on a single Windows server ran fine, but often when a second program was introduced, the requirements of each program caused various types of resource contention and even out and out operating system failures. This behavior drove many companies, application designers, developers, IT professionals, and vendors to adopt a “one server, one application” best practice; so for every application that was deployed, one or more servers needed to be acquired, provisioned, and managed.

Another factor that drove the growing server population was corporate politics. The various organizations within a single company did not want any common infrastructure. Human Resource and Payroll departments declared their data was too sensitive to allow the potential of another group using their systems. Marketing, Finance, and Sales all believed the same thing to protect their fiscal information. Research and Development also had dedicated servers to ensure the safety of their corporate intellectual property. Sometimes companies had redundant applications, four or more email systems, maybe from different vendors, due to this individual ownership attitude. By demanding solitary control of their application infrastructure, departments felt that they had control of their data, but this type of control also increased their capital costs.

Aiding the politics was the fact that business demand, competition, Moore’s Law, and improvements in server and storage technologies, all drastically drove down the cost of hardware. This made the entry point for a department to build and manage its own IT infrastructure much more affordable. The processing power and storage that in the past had cost hundreds of thousands of dollars could be had for a fraction of that cost in the form of even more Windows servers.

Business computers initially had specialized rooms in which to operate. These computer rooms were anything from oversized closets to specially constructed areas for housing a company’s technology infrastructure. They typically had raised floors under which the cables and sometimes air conditioning were run. They held the computers, network equipment, and often telecomm equipment. They needed to be outfitted with enough power to service all of that equipment. Because all of those electronics in a contained space generated considerable heat, commensurate cooling through huge air-conditioning handlers was



mandatory as well. Cables to interconnect all of these devices, fire suppression systems in case of emergency, and separate security systems to protect the room itself, all added to the considerable and ever-rising costs of doing business in a modern corporation. As companies depended more and more on technology to drive their business, they added many more servers to support that need. Eventually, this expansion created data centers. A *data center* could be anything from a larger computer room, to an entire floor in a building, to a separate building constructed and dedicated to the health and well-being of a company's computing infrastructure. Entire buildings existed solely to support servers, and then at the end of twentieth century, the Internet blossomed into existence.

"E-business or out of business" was the cry that went up as businesses tried to stake out their territories in this new online world. To keep up with their competition, existing companies deployed even more servers as they web-enabled old applications to be more customer facing and customer serving. Innovative companies, such as Amazon and Google, appeared from nowhere, creating disruptive business models that depended on large farms of servers to rapidly deliver millions of web pages populated with petabytes of information (see Table 1.1). IT infrastructure was mushrooming at an alarming rate, and it was only going to get worse. New consumer-based services were delivered not just through traditional online channels, but newer devices such as mobile phones compounded data centers growth. Between 2000 and 2006, the Environmental Protection Agency (EPA) reported that energy use by United States data centers doubled, and that over the next five years they expected it to double again. Not only that, but servers were consuming about 2 percent of the total electricity produced in the country, and the energy used to cool them consumed about the same amount.

**TABLE 1.1** Byte Sizes

Name	Abbreviation	Size
Byte	B	8-bits (a single character)
Kilobyte	KB	1,024 B
Megabyte	MB	1,024 KB
Gigabyte	GB	1,024 MB
Terabyte	TB	1,024 GB
Petabyte	PB	1,024 TB
Exabyte	EB	1,024 PB

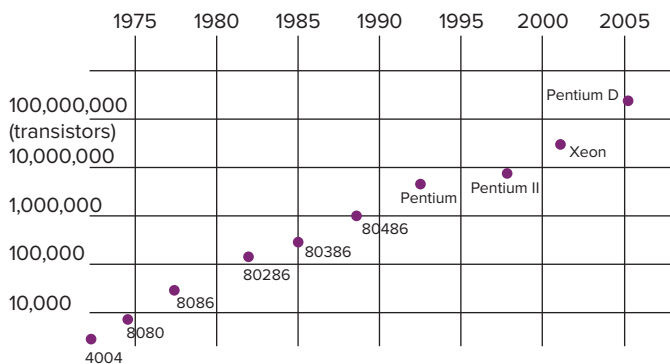
Let's take a closer look at these data centers. Many were reaching their physical limits on many levels. They were running out of actual square footage for the servers they needed to contain, and companies were searching for alternatives. Often the building that housed a data center could not get more electrical power or additional cooling capacity. Building larger or additional data centers was and still is an expensive proposition. In addition to running out of room, the data centers often had grown faster than the people managing them could maintain them. It was common to hear tales of lost servers. (A *lost server* is a server that is running, but no one actually knows which line of business owns it or what it is doing.) These lost servers couldn't be interrupted for fear of inadvertently disrupting some crucial part of the business. In some data centers, cabling was so thick and intertwined that when nonfunctioning cables needed to be replaced, or old cables were no longer needed, it was easier to just leave them where they were, rather than try to unthread them from the mass. Of course, these are the more extreme examples, but most data centers had challenges to some degree in one or more of these areas.

## Explaining Moore's Law

So far you have seen how a combination of events—the rise of Windows, corporations increasing reliance on server technology, and the appearance and mushrooming of the Internet and other content-driven channels—all contributed to accelerated growth of the worldwide server population. One 2006 study estimated that the 16 million servers in use in 2000 had grown to almost 30 million by 2005. This trend continues today. Think about all of the many ways you can pull information from the world around you; computers, mobile devices, gaming platforms, and television set tops are only some of the methods, and new ones appear every day. Each of them has a wide and deep infrastructure to support those services, but this is only part of the story. The other piece of the tale has to do with how efficient those computers were becoming.

If you are reading an electronic copy of this text on a traditional computer, or maybe on a smart phone or even a tablet, you probably have already gone through the process of replacing that device at least once. Phone companies typically give their customers the ability to swap out older smart phones every couple of years for newer, more up-to-date models, assuming you opt for another contract extension. A computer that you bought in 2000 has probably been supplanted by one you purchased in the last three to five years, and if it is closer to five years, you are probably thinking about replacing that one as well. This has little to do with obsolescence, although electronic devices today are rarely engineered to outlive their useful lifespan. It has more to do with the

incredible advances that technology constantly makes, packing more and more capability into faster, smaller, and newer packages. For example, digital cameras first captured images at less than one megapixel resolution and now routinely provide more than 12 megapixel resolutions. PCs, and now smart phones, initially offered memory (RAM) measured in kilobytes; today the standard is gigabytes, an increase of two orders of magnitude. Not surprisingly, there is a rule of thumb that governs how fast these increases take place. It is called Moore's Law, and it deals with the rate at which certain technologies improve (see Figure 1.2).



**FIGURE 1.2** Moore's Law: transistor count and processor speed

Gordon Moore, one of the founders of Intel, gets credit for recognizing and describing the phenomenon that bears his name. His original thought was publicized back in 1965, and though it has been refined a few times along the way, is still very true today. Simply stated, Moore's Law says that processing power roughly doubles every 18 months. That means a computer you buy 18 months from now will be twice as powerful as one you buy today. As it turns out, Moore's Law applies not just to *processing power* (the speed and capacity of computer chips) but to many other related technologies as well (such as memory capacity and the megapixel count in digital cameras). You might think that after almost 50 years, we would be hitting some type of technological barrier that would prevent this exponential growth from continuing, but scientists believe that it will hold true for somewhere between 20 years on the low side and centuries on the high. But what does this have to do with straining data centers and ballooning server growth?

Servers are routinely replaced. There are two main models for this process. Companies buy servers and then buy newer models in three to five years when those assets are depreciated. Other corporations lease servers, and when that



lease runs its course, they lease newer servers, also in three to five year intervals. The servers that were initially purchased for use were probably sized to do a certain job; in other words, they were bought, for example, to run a database. The model and size of the server was determined with help from an application vendor who provided a recommended configuration based on the company’s specific need. That need was not the company’s requirement on the day the server was purchased; it was purchased based on the company’s projected need for the future and for emergencies. This extra capacity is also known as *headroom*. To use the server for three to five years, it had to be large enough to handle growth until the end of the server’s life, whether it actually ever used that extra capacity or not. When the server was replaced, it was often replaced with a similarly configured model (with the same number of processors and the same or additional memory) for the next term, but the newer server was not the same.

Let’s take six years as an example span of time and examine the effect of Moore’s Law on the change in a server (see Table 1.2). A company that is on a three-year model has replaced the initial server twice—once at the end of year three and again at the end of year six. According to Moore’s Law, the power of the server has doubled four times, and the server is 16 times more powerful than the original computer! Even if they are on the five-year model, and have only swapped servers once, they now own a machine that is eight times faster than the first server.

**TABLE 1.2** Processor Speed Increases Over Six Years

Year	2005	2006	2007	2008	2009	2010
Processor Speed	1x	2x	4x	4x	8x	16x
3-year plan			purchase			purchase
5-year plan					purchase	

In addition to faster CPUs and faster processing, newer servers usually have more memory, another benefit of Moore’s Law. The bottom line is that the replacement servers are considerably larger and much more powerful than the original server, which was already oversized for the workload it was handling.

The last item you need to understand here is that the server’s actual workload does not typically increase at the same rate as the server’s capabilities. That means that the headroom in the server also increased substantially. Although that performance safety net began somewhere in the 20 to 50 percent range, that

unused capacity after a server refresh or two could be well over 90 percent. Across a data center it was not uncommon to average about 10 to 15 percent utilization, but the distribution was often arranged so that a few servers had very high numbers while the large bulk of servers were actually less than 5 percent utilized. In other words, most CPUs sat around idle for 95 percent of the time, or more!

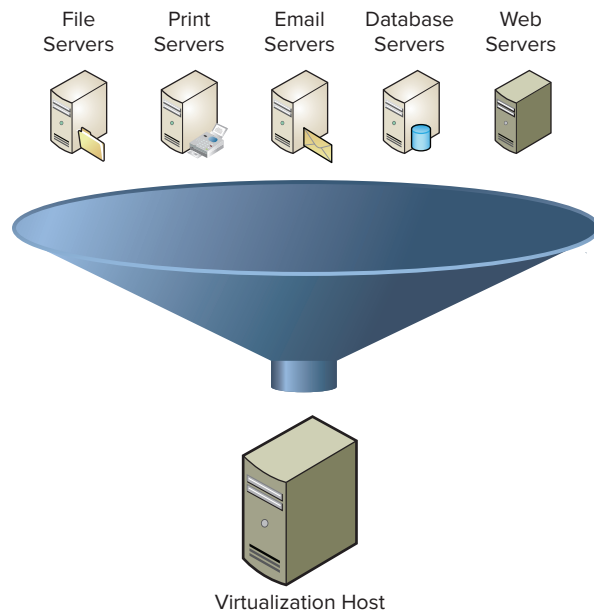
## Understanding the Importance of Virtualization

This is where the two stories come together. There was a wild explosion of data centers overfilled with servers; but as time passed, in a combination of the effect of Moore's Law and the "one server, one application" model, those servers did less and less work. Fortunately, help was on the way in the form of virtualization. The idea and execution of virtualization was not new. It ran on IBM mainframes back in the early 1970s but was updated for modern computer systems. We'll come back to the specifics of virtualization in a moment, but in keeping with Popek and Goldberg's definition, virtualization allows many operating systems to run on the same server hardware at the same time, while keeping each virtual machine functionally isolated from all the others. The first commercially available solution to provide virtualization for x86 computers came from VMware in 2001.

A parallel open-source offering called Xen arrived two years later. These solutions (VMMs, or hypervisors) took the form of a layer of software that lived either between an operating system and the virtual machines (VMs) or was installed directly onto the hardware, or "bare-metal," just like a traditional operating system such as Windows or Linux. In the next chapter, we'll go into much more depth about hypervisors.

What virtualization brought to those overfull data centers and underutilized servers was the ability to condense multiple physical servers into one server that would run many virtual machines, allowing that physical server to run at a much higher rate of utilization. This condensing of servers is called *consolidation*, as illustrated in Figure 1.3. A measure of consolidation is called the *consolidation ratio* and is calculated by counting the number of VMs on a server—for example, a server that has eight VMs running on it has a consolidation ratio of 8:1. Consolidation was a boon to beleaguered data centers and operations managers because it solved a number of crucial problems just when a critical threshold had been reached. Even a modest consolidation ratio of 4:1 could remove three-quarters of the servers in a data center.

◀ The moniker x86 refers to the processor architecture originally based on Intel's 8086 CPU and subsequent chip generations that ended in "86." Other vendors now also produce processors with this architecture.



**FIGURE 1.3** Server consolidation

In larger data centers, where hundreds or even thousands of servers were housed, virtualization provided a way to decommission a large portion of servers. This reduced the overall footprint of a data center, reduced the power and cooling requirements, and removed the necessity to add to or construct additional data centers. By extension, with fewer servers, it reduced a company's hardware maintenance costs and reduced the time system administrators took to perform many other routine tasks.

### CONSOLIDATION DRIVES DOWN COSTS

Many studies show that the total cost of ownership for an individual server is somewhere between 3 and 10 times the cost of the server itself over three years. In other words, if a server costs \$5,000, the cost of maintaining that server is at least another \$5,000 per year. Over three years, that is \$20,000 per server (the initial hardware spend plus three years of maintenance costs). Those ownership costs include software, annual software and hardware maintenance, power, cooling, cables, people costs, and more. So in this example, for every hundred servers the company can consolidate, it can save two million dollars the first year and every year afterward.

Aside from consolidation, a second development took place. As companies began to see the benefits of virtualization, they no longer purchased new hardware when their leases were over, or if they owned the equipment, when their hardware maintenance licenses expired. Instead, they virtualized those server workloads. This is called *containment*. Containment benefited corporations in multiple ways. They no longer had to refresh large amounts of hardware year after year; and all the costs of managing and maintaining those servers—power, cooling, etc.—were removed from their bottom line from that time on. Until the time when virtualization became commercially viable, Moore’s Law worked against the existing application/server/data center model; after it became feasible, it actually helped. The consolidation ratios of the first generation of x86 hypervisors were in the range of 5:1. As time continued to pass, more powerful chips and larger memory enabled much higher consolidation ratios, where a single physical server could host dozens or hundreds of VMs. Instead of removing three out of four servers, virtualization today can comfortably remove nine out of ten; or with sufficiently configured servers, ninety-nine out of a hundred. As a result, most corporate data centers have reclaimed much of the space that they had lost before virtualization.

### **VIRTUAL SERVERS NOW OUTNUMBER PHYSICAL SERVERS**

IDC reported that in 2009, more virtual servers were deployed than physical servers. They predicted that while physical server deployment would remain relatively static over the following five years, virtual machine deployment would double the physical deployments at the end of that span.

## **Examining Today’s Trends**

Consolidation and containment are just two of the many examples of how virtualization enhances traditional server usage that we will cover. They are also the two that most analyses deal with because they are the easiest to quantify from a financial standpoint—remove or significantly diminish the associated hardware cost from your budget, and your bottom line will be directly impacted. We’ll introduce some of those other examples now and examine them more closely later in the book.

As virtualization takes hold in an organization, its progress takes a very predictable course. The initial beachhead is in infrastructure services and in older servers, two areas where server management and cost issues are typically most

acute. Infrastructure servers deliver an organization's technology plumbing in the form of print services, file servers, and domain services. These servers are critical to the day-to-day business but often run on less reliable, less expensive hardware than the tier-one applications that drive the business. Older servers are also a concern. Data centers frequently host applications that do not run on newer operating systems—for example, a seven-year-old Windows NT system running a custom-built analytics system continues to run on its original hardware, which may be obsolete and no longer reliable or even serviceable. A company can also have applications that it no longer knows how to manage (don't laugh, it happens)—the vendor is no longer in business or their internal expert is no longer with the company, but the application runs, so they just hope it will continue to do so. Virtualization, as you will see, makes these applications much more available, scalable, and manageable than they ever were on physical servers, and for less cost as well.

Once the infrastructure services are virtualized and an organization starts to reap some of the fiscal benefits of their new strategy, an active program is put in place to move to the next level. As servers come off their leases, those workloads are migrated to the growing infrastructure. Companies usually adopt virtualization-first policies, which state that as new projects come in house, any server requirements will be satisfied by the virtual resources, rather than by paying for new physical resources. Actual hardware will be purchased only if it can be proven that the need cannot be satisfied with the virtual environment. Right behind the infrastructure services are the test and development servers. For every production application that a corporation runs, there are somewhere between 2 and 10 times as many servers in the data center that support that application. Tier-one applications require many environments for new update testing, quality and assurance tests, user acceptance testing, problem resolution environments, performance tuning, and more. Moving these systems to the virtual infrastructure, aside from again saving costs through consolidation, gives developers and application owners greater flexibility in how they can manage their processes. Preconfigured templates allow them to rapidly deploy new servers in minutes rather than in the weeks it would have taken prior to the change.

At this point, an organization's infrastructure is somewhere between 50 and 75 percent virtualized, at least on the x86 platforms that run their Windows and Linux servers. They have built up expertise and confidence in the virtualization technologies and are still looking to take even further advantage of virtualization. From here companies go in a number of different directions, often simultaneously.

Larger applications often require larger hardware and specialized operating systems to run that hardware. Databases, for example, run on a variety of UNIX

systems, each with a vendor-specific version. Sun servers run Solaris, HP servers run HP/UX, and IBM servers run AIX. Companies invest large sums of money for this proprietary hardware, and just as much time and effort in training their people to work with these open but also proprietary operating systems. But again, Moore's Law is working in their favor. In the past, an x86 platform would not be powerful or reliable enough to run this mission-critical type of workload; today that is no longer true. There are almost no workloads today that cannot be run in a virtual environment due to performance limitations. Linux, which is an open source flavor of UNIX, can run the same application software as the vendor-specific hardware and software combinations. Although we'll focus mostly on Microsoft Windows, Linux can also be easily virtualized, and that is leading many companies to migrate these critical workloads to a more flexible, less expensive, and often more available environment.

As we touched on earlier, virtual servers are encapsulated systems, essentially just a set of files that can be copied and moved like any other files. As Internet computing has evolved, availability has become crucial, whether it is maintaining 24/7 operations through enhanced software and features, or *disaster recovery* capabilities (restoring operations after an interruption). Virtualization enables availability in a number of ways. Virtual machines can be moved from one physical host to another without interruption. Instead of scheduling application downtime for a physical host to do maintenance, the workload can be moved to another host, the physical work done on the server, and the workload returned, all without interrupting the users. With Linux and newer versions of Microsoft Windows, you can add additional resources, processors, and memory to a virtual machine without having to reboot the operating system. This ability allows an administrator to resolve resource shortages without impacting application uptime. By replicating the files that comprise a server to a secondary site, in the event of an environmental disaster, such as a hurricane or flood, the entire data center can be restored in a matter of hours or even minutes, instead of the days or weeks it would have taken previously. These are just a few examples of the increased availability virtualization provides.

Finally, the remaining physical servers are addressed. These are the ones that run the *tier-one applications*, strategic business applications that give each company its competitive advantage. They take the form of email services such as Microsoft Exchange or Lotus Notes, database servers such as Microsoft SQL Server, Oracle, or MySQL, enterprise business applications such as SAP, business intelligence and analytics systems such as SAS, hospital healthcare applications, financial services applications, custom-built JAVA applications, and on and on. Because the health and well-being of these applications directly affect a company's profitability, administrator and application owners are hesitant to make



changes to a time-proven environment or methodology, even if it has flaws. But after working with virtualized servers in test, development, and QA environments, they are comfortable enough to virtualize these remaining workloads.

Moving to an entirely virtualized platform provides enterprises a much greater degree of availability, agility, flexibility, and manageability than they could have in a solely physical environment. You will find out more about many of the capabilities of virtual machines and what a virtual environment can provide throughout this text, but one large benefit of virtualization is that it provides the foundation for the next phase of data center evolution: cloud computing.

## Virtualization and Cloud Computing

Five years ago, if you said the words “cloud computing,” very few people would have had any idea what you were talking about. Today it would be difficult to find someone who is engaged in the worldwide business or consumer markets who has not heard the term *cloud computing*. Much like the rush to the Internet during the mid-to-late 1990s and early 2000s, many of today’s companies are working on cloud enablement for their offerings. Mirroring their actions during the dot-com boom, consumer services are also making the move to the cloud. Apple for example, recently offered the iCloud where you can store your music, pictures, books, and other digital possessions and then access them from anywhere. Other companies, such as Microsoft, Amazon, and Google are offering similar cloud-based services. Rather than define the cloud, which would be outside the scope of this text, let’s look at what the cloud is providing: a simple method for accessing and utilizing resources.

Virtualization is the engine that will drive cloud computing by turning the data center—what used to be a hands-on, people-intensive process—into a self-managing, highly scalable, highly available, pool of easily consumable resources. Before virtualization, system administrators spent 70 percent or more of their time on routine functions and reacting to problems, which left little time for innovation or growth. Virtualization and, by extension, cloud computing provide greater automation opportunities that reduce administrative costs and increase a company’s ability to dynamically deploy solutions. By being able to abstract the physical layer away from the actual hardware, cloud computing creates the concept of a virtual data center, a construct that contains everything a physical data center would. This virtual data center, deployed in the cloud, offers resources on an as-needed basis, much like a power company provides electricity. In short, these new models of computing will dramatically simplify the delivery of new applications and allow companies to accelerate their deployments without sacrificing scalability, resiliency, or availability.

## Understanding Virtualization Software Operation

Although we've spent the bulk of our time discussing server virtualization and it will be our focus throughout the remainder of the text, there are other methods and areas of virtualization. Personal computers are changing into tablets and thin clients, but the applications that run on PCs still need to be offered to users. One way to achieve this is desktop virtualization. Those applications can also be virtualized, packaged up, and delivered to users. Virtualization is even being pushed down to the other mobile devices such as smart phones.

### Virtualizing Servers

The model for server virtualization, as you saw earlier, is comprised of physical hardware augmented by two key software solutions. The hypervisor abstracts the physical layer and presents this abstraction for virtualized servers or virtual machines to use. A hypervisor is installed directly onto a server, without any operating system between it and the physical devices. Virtual machines are then *instantiated*, or booted. From the virtual machine's view, it can see and work with a number of hardware resources. The hypervisor becomes the interface between the hardware devices on the physical server and the virtual devices of the virtual machines. The hypervisor presents only some subset of the physical resources to each individual virtual machine and handles the actual I/O from VM to physical device and back again. Hypervisors do more than just provide a platform for running VMs; they enable enhanced availability features and create new and better ways for provisioning and management as well.

While hypervisors are the foundations of virtual environments, virtual machines are the engines that power the applications. Virtual machines contain everything that their physical counterparts do (operating systems, applications, network connections, access to storage, and other necessary resources) but packaged in a set of data files. This packaging makes virtual machines much more flexible and manageable through the use of the traditional file properties in a new way. Virtual machines can be cloned, upgraded, and even moved from place to place, without ever having to disrupt the user applications. We will focus exclusively on hypervisors in Chapter 2 and look closer at virtual machines in Chapter 3.

### Virtualizing Desktops


Just as virtualization has changed the model of how traditional server computing is being managed today, virtualization has moved into the desktop

computing model as well. Desktop computing for companies is expensive and inefficient on many fronts. It requires staffs of people to handle software update rollouts and patching processes, not to mention hardware support and help desk staffing. Virtual desktops run on servers in the datacenter; these servers are much more powerful and reliable hardware than traditional PCs. The applications that users connect to are also in the data center running on servers right next door, if you will, so all of the network traffic that previously had to go back and forth to the data center, no longer needs to, which greatly reduces network traffic and extends network resources.

Virtual desktops are accessed through thin clients, or other devices, many of which are more reliable and less expensive than PCs. Thin clients have life spans of 7 to 10 years so can be refreshed less frequently. They also only use between 5 and 10 percent of the electricity of a PC. In large companies, those costs add up quickly. If a thin client does break, a user can replace it himself, instead of relying on a specialized hardware engineer to replace it. The virtual desktop where all of the data is kept has not been affected by the hardware failure. In fact, the data no longer leaves the data center, so the risk that a lost or stolen device will cause security issues is also reduced.

That data is now managed and backed up by a professional, instead of an unsophisticated or indifferent user. Creating desktop images as virtual machines brings some of the cost savings of server virtualization but really shines on the desktop management side. A desktop administrator can create and manage fewer images that are shared among hundreds of people. Patches can be applied to these images and are guaranteed to reach a user, whereas that is not always the case with a physical desktop. In the event that a rolled-out patch or other software changes breaks an application, an administrator can direct users back to the original image, and a simple logout and login will return them to a functional desktop.

One of the biggest differences comes in the area of security. Today PCs routinely have antivirus software applications that help protect their data from malware and more. Virtualization allows new methods of protection. Rather than just loading the malware software on individual virtual desktops, there are now virtual appliances, specifically designed virtual machines that reside in each host and protect all of the virtual desktops that run there. This new model reduces the overall I/O and processor usage by downloading new definitions once instead of individually by guest. This is an area of rapid change and growth at the moment, and it looks to continue that way as new user devices become more common.



**Two popular solutions for desktop virtualization are Citrix's XenDesktop and VMware's View. There are also many others that provide desktops using various combinations of hardware and software.**

## Virtualizing Applications

Computer programs, or applications, can also be virtualized. Like both server and desktop virtualization, there are a number of different solutions for this problem. There are two main reasons for application virtualization; the first is ease of deployment. Think about the number of programs you have on your PC. Some companies must manage hundreds or even thousands of different applications. Every time a new version of each of those applications is available, the company, if it decides to upgrade to that newer version, has to push out a copy to all of its PCs. For one or a small number of computers, this may be a relatively trivial task. But how would you do this to a hundred PCs? Or a thousand? Or ten thousand? Corporate IT staffs have tools that help manage and automate this task to happen repeatedly and reliably.

The second reason has to do with how different applications interact with each other. Have you ever loaded or updated an application that broke some functionality that had been working just fine? It is difficult to know how an upgrade to one solution may affect other applications. Even simple upgrades such as Adobe Acrobat Reader or Mozilla Firefox can become problematic. Some types of application virtualization can mitigate or even prevent this issue by encapsulating the entire program and process. Many application virtualization strategies and solutions are currently available. This is a rapidly evolving area with new use cases appearing regularly, especially in conjunction with mobile devices such as smart phones and tablets.



Some popular application virtualization solutions are Microsoft's App-V, Citrix's Application Streaming, and VMware's ThinApp. Each solution approaches the problem differently but is effective.

### THE ESSENTIALS AND BEYOND

Server virtualization is a disruptive technology that allows many logical computers to run on a single physical server. Extreme server population growth driven by application deployment practices, the spread of Microsoft Windows, and Moore's Law have placed physical resource and financial constraints on most of the world's corporations. Virtualization is not a new concept, but was redeveloped and helped relieve those stresses on data centers through server consolidation and containment. Many of the characteristics that server virtualization provides, such as increased availability and scalability, are providing the foundation for corporations as they move to cloud computing.

### ADDITIONAL EXERCISES

- Using Moore's Law, calculate how much faster processors are today than they were in year 2000. Calculate how much faster processors will be 10 years from now.

*(Continues)*

**THE ESSENTIALS AND BEYOND** *(Continued)*

- ▶ Using the Internet, discover how many different types of server virtualization are publicly available. How many separate architectures are represented in what you found?
- ▶ At what minimum amount of servers does it make sense to virtualize a data center? Will the cost savings and soft cost savings (such as increased manageability and availability) outweigh the initial cost of virtualization, cost of education, and effort to effect the change?

# CONTENTS

*Introduction*

*xv*

<b>CHAPTER 1</b>	<b>Understanding Virtualization</b>	<b>1</b>
	Describing Virtualization . . . . .	1
	Microsoft Windows Drives Server Growth . . . . .	3
	Explaining Moore's Law . . . . .	6
	Understanding the Importance of Virtualization . . . . .	9
	Examining Today's Trends . . . . .	11
	Virtualization and Cloud Computing . . . . .	14
	Understanding Virtualization Software Operation . . . . .	15
	Virtualizing Servers . . . . .	15
	Virtualizing Desktops . . . . .	15
	Virtualizing Applications . . . . .	17
<b>CHAPTER 2</b>	<b>Understanding Hypervisors</b>	<b>19</b>
	Describing a Hypervisor . . . . .	19
	Exploring the History of Hypervisors . . . . .	20
	Understanding Type 1 Hypervisors . . . . .	21
	Understanding Type 2 Hypervisors . . . . .	23
	Understanding the Role of a Hypervisor . . . . .	24
	Holodecks and Traffic Cops . . . . .	24
	Resource Allocation . . . . .	25
	Comparing Today's Hypervisors . . . . .	27
	VMware ESX . . . . .	27
	Citrix Xen . . . . .	29
	Microsoft Hyper-V . . . . .	31
	Other Solutions . . . . .	32
<b>CHAPTER 3</b>	<b>Understanding Virtual Machines</b>	<b>35</b>
	Describing a Virtual Machine . . . . .	35
	Examining CPU in a Virtual Machine . . . . .	38
	Examining Memory in a Virtual Machine . . . . .	39



	Examining Network Resources in a Virtual Machine . . . . .	39
	Examining Storage in a Virtual Machine . . . . .	41
	Understanding How a Virtual Machine Works . . . . .	42
	Working with Virtual Machines . . . . .	43
	Understanding Virtual Machine Clones . . . . .	44
	Understanding Templates . . . . .	45
	Understanding Snapshots . . . . .	47
	Understanding OVF . . . . .	48
<b>CHAPTER 4</b>	<b>Creating a Virtual Machine</b>	<b>51</b>
	Performing P2V Conversions . . . . .	51
	Investigating the Physical-to-Virtual Process . . . . .	52
	Hot and Cold Cloning. . . . .	53
	Loading Your Environment . . . . .	54
	Exploring VMware Player . . . . .	60
	Building a New Virtual Machine . . . . .	63
	Thinking About VM Configuration . . . . .	64
	Creating a First VM . . . . .	65
<b>CHAPTER 5</b>	<b>Installing Windows on a Virtual Machine</b>	<b>71</b>
	Loading Windows into a Virtual Machine . . . . .	71
	Installing Windows 7 . . . . .	72
	Installing VMware Tools . . . . .	83
	Understanding Configuration Options . . . . .	89
	Optimizing a New Virtual Machine . . . . .	95
<b>CHAPTER 6</b>	<b>Installing Linux on a Virtual Machine</b>	<b>97</b>
	Loading Linux into a Virtual Machine. . . . .	97
	Installing Linux into a Virtual Machine . . . . .	98
	Installing VMware Tools . . . . .	113
	Understanding Configuration Options . . . . .	117
	Optimizing a New Linux Virtual Machine. . . . .	122
<b>CHAPTER 7</b>	<b>Managing CPUs for a Virtual Machine</b>	<b>125</b>
	Understanding CPU Virtualization. . . . .	125
	Configuring VM CPU Options . . . . .	129

	Tuning Practices for VM CPUs . . . . .	130
	Choosing Multiple vCPUs vs. a Single vCPU . . . . .	131
	Hyper-Threading . . . . .	132
	Working with Intel and AMD Servers . . . . .	134
<b>CHAPTER 8</b>	<b>Managing Memory for a Virtual Machine</b>	<b>137</b>
	Understanding Memory Virtualization . . . . .	137
	Configuring VM Memory Options . . . . .	140
	Tuning Practices for VM Memory. . . . .	142
	Calculating Memory Overhead . . . . .	143
	Memory Optimizations . . . . .	144
<b>CHAPTER 9</b>	<b>Managing Storage for a Virtual Machine</b>	<b>151</b>
	Understanding Storage Virtualization . . . . .	151
	Configuring VM Storage Options. . . . .	156
	Tuning Practices for VM Storage . . . . .	162
<b>CHAPTER 10</b>	<b>Managing Networking for a Virtual Machine</b>	<b>171</b>
	Understanding Network Virtualization . . . . .	171
	Configuring VM Network Options . . . . .	181
	Tuning Practices for Virtual Networks . . . . .	187
<b>CHAPTER 11</b>	<b>Copying a Virtual Machine</b>	<b>191</b>
	Cloning a Virtual Machine . . . . .	191
	Working with Templates . . . . .	197
	Saving a Virtual Machine State . . . . .	201
	Creating a Snapshot. . . . .	204
	Merging Snapshots. . . . .	208
<b>CHAPTER 12</b>	<b>Managing Additional Devices in Virtual Machines</b>	<b>211</b>
	Using Virtual Machine Tools . . . . .	212
	Understanding Virtual Devices. . . . .	213
	Configuring a CD/DVD Drive . . . . .	214
	Configuring a Floppy Disk Drive . . . . .	215
	Configuring a Sound Card . . . . .	218
	Configuring USB Devices . . . . .	219
	Configuring Graphic Displays . . . . .	221
	Configuring Other Devices. . . . .	222

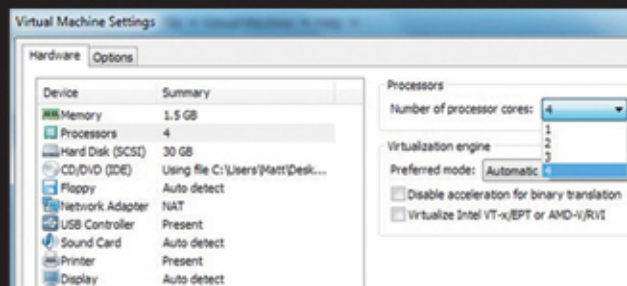
<b>CHAPTER 13</b>	<b>Understanding Availability</b>	<b>227</b>
	Increasing Availability .....	227
	Protecting a Virtual Machine .....	230
	Protecting Multiple Virtual Machines .....	234
	Protecting Datacenters .....	238
<b>CHAPTER 14</b>	<b>Understanding Applications in a Virtual Machine</b>	<b>243</b>
	Examining Virtual Infrastructure Performance Capabilities .....	243
	Deploying Applications in a Virtual Environment .....	248
	Understanding Virtual Appliances and vApps .....	256
<b>APPENDIX</b>	<b>Answers to Additional Exercises</b>	<b>261</b>
	<i>Glossary</i> .....	<i>271</i>
	<i>Index</i> .....	<i>277</i>

# Learn Virtualization Quickly and Easily

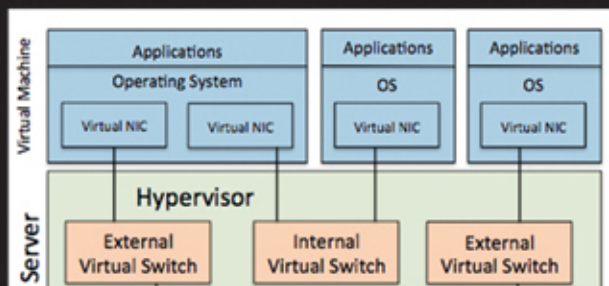
Virtualization is crucial to providing computing services economically on a wide scale, and the topic is essential learning for IT professionals. This full-color, hands-on guide thoroughly covers virtualization fundamentals. You'll discover how today's virtual machines work, explore hypervisors, learn virtual networking concepts, see how to plan for high availability, and much more. It's the perfect virtualization primer, whether you're an aspiring IT professional or just brushing up on your skills.

## Learn these virtualization essentials—and more:

- The importance of virtualization in today's data center environments
- Hypervisors—Type 1, Type 2, their roles, and what's on the market
- How to build a virtual machine from scratch
- Managing CPU, memory, disk storage, and networking
- Ways to map virtual networking to a physical machine
- Configuring devices such as DVDs, USBs, and sound cards
- What you need to do to ensure high availability
- Working with applications in a virtual environment



Change virtual machine CPU quantity



Virtual network configurations

This striking *Essentials* book maximizes skill acquisition and knowledge retention with:

- Chapter-opening learning objectives
- Step-by-step tutorials
- Four-color screenshots and illustrations
- Essentials and Beyond—summaries and additional suggested exercises
- Chapter review questions

## About the Author

Matthew Portnoy is a Senior System Engineer at VMware and has more than 30 years of work experience in the IT industry. He is an Adjunct Professor at Wake Tech Community College, teaching CIS classes since 2007. Matt has presented at many trade shows and industry events, including VMware User Group events and VMworld, which is the largest virtualization conference.

[www.sybex.com](http://www.sybex.com)

[www.sybex.com/go/virtualizationessentials](http://www.sybex.com/go/virtualizationessentials)

Sybex®  
An Imprint of  
**WILEY**

COMPUTERS/  
Operating Systems/  
Virtualization

ISBN 978-1-118-17671-9



9 781118 176719