1

VMware Server: A Brief Introduction

VMware has been on the cutting edge of virtualization since the release of VMware Workstation in 1999. In 2001 VMware released its first of a long line of server-based products. The VMware ESX and GSX Server products have changed the way we look at our server utilization to this day. There are now dozens of virtualization software packages on the market with price points that are all over the map. This increase in competition gave birth to VMware Server, the next iteration of its GSX product line. What makes VMware Server so special? As you will see over the course of this book, VMware Server is a robust and versatile tool for your data center and development process. Oh, did I mention that it was free?

This chapter introduces the basic concepts of hardware virtualization, defines some early terminology, introduces you to VMware Server, and gives you a look at what life was like before virtualization.

What Is Virtualization?

Virtualization itself is a rather broad term that most commonly refers to some sort of abstraction of resources. Throughout this book I will be focusing on one specific aspect of virtualization, software virtualization.

Software Virtualization

Software virtualization is a process in which the actual physical hardware of a machine is decoupled, or abstracted if you will, from the underlying operating system by means of software. This hardware abstraction allows multiple "virtual" environments to function side by side on top of a single physical machine. Through this abstraction process each virtual machine is provided with its own hardware (RAM, CPU, Network Card, and so on) that is independent and isolated from the other virtual machines running along side it.

Figure 1-1 shows how a normal x86 machine presents its hardware to the operating system for your use, and Figure 1-2 provides a visual representation of how software virtualization takes form and how the physical machine hardware is partitioned out into multiple virtual segments for your use.

All the magic happens in the virtualization layer seen in Figure 1-2. This layer decouples your hardware and handles all allocation and utilization of the physical hardware by your virtualized machines.

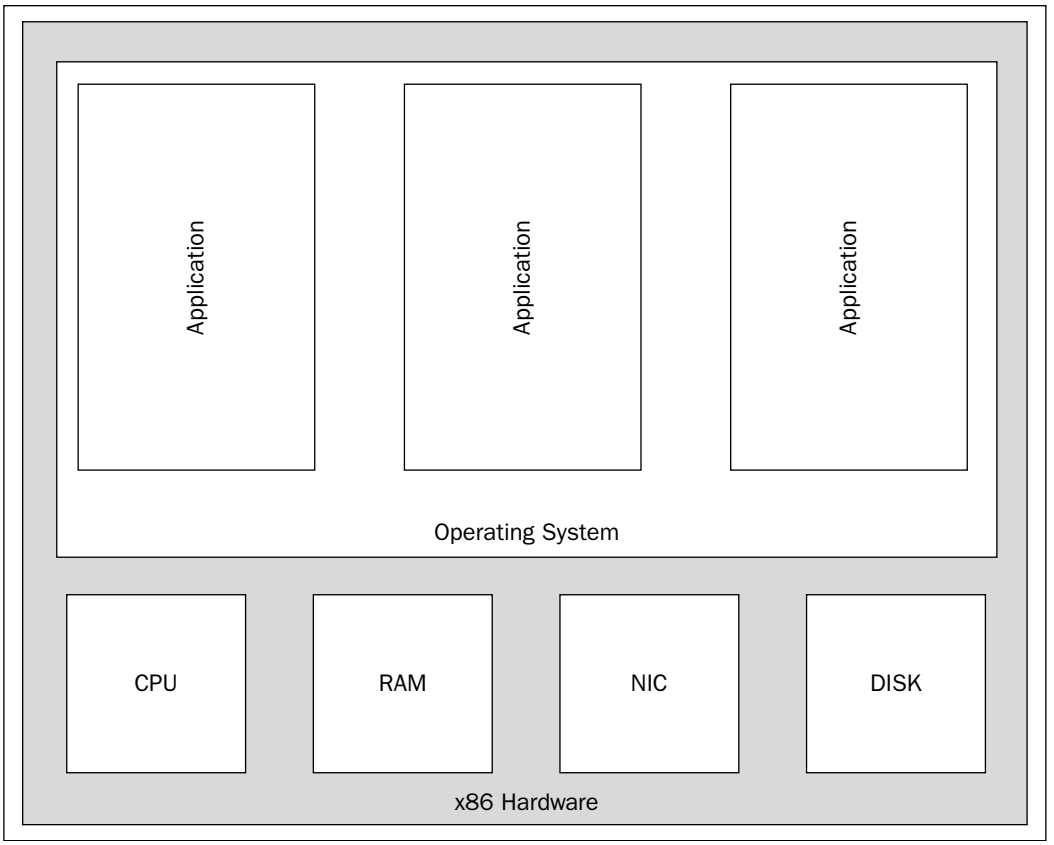

Figure 1-1

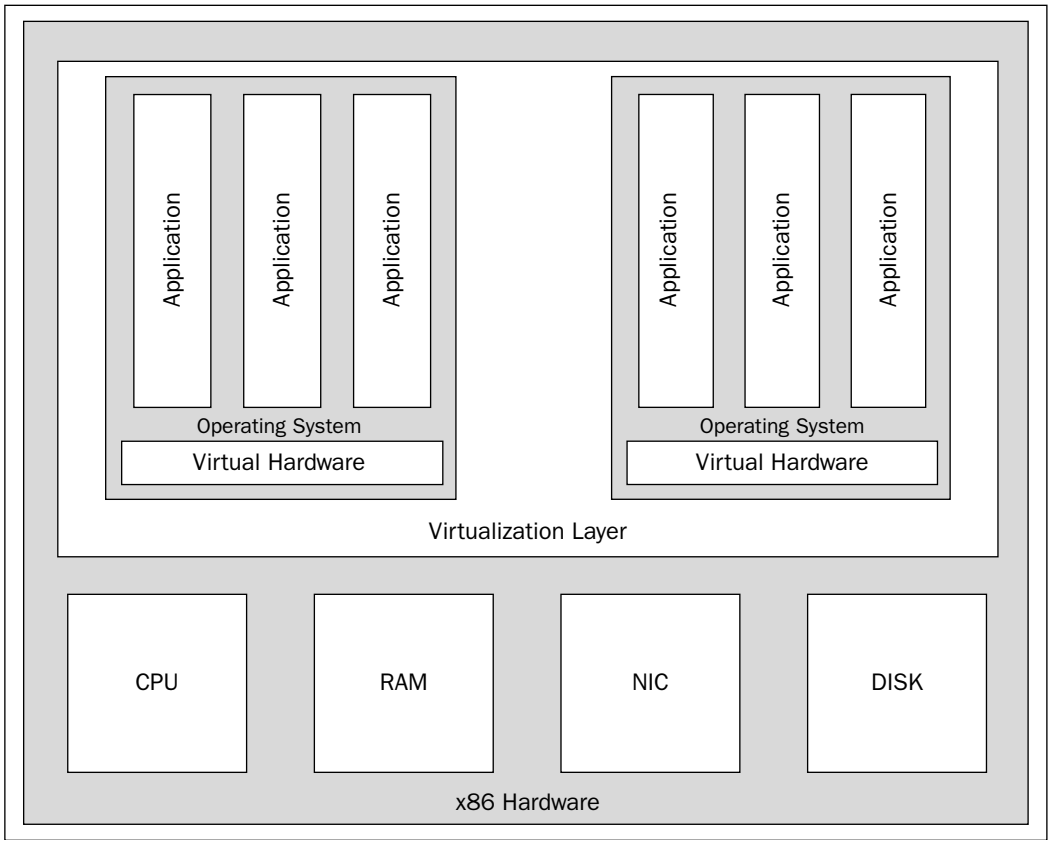

Figure 1-2

Hosted vs. Hypervisor Architecture

You can't really complete the overview of virtualization without a quick discussion of hosted versus hypervisor architecture.

- **Hosted architecture** is where your virtualization software is installed as an application onto the pre-existing host operating system. This means that your virtualization layer relies on your host operating system for device support and physical resource management. VMware Server, VMware Workstation, and Microsoft Virtual PC are good examples of a hosted architecture.
- **Hypervisor architecture** is more of a bare metal approach where there is no pre-existing operating system. The virtualization software is installed on a clean system, and it provides kernel and driver support for the raw physical hardware. VMware ESX server is a good example of virtualization utilizing hypervisor style architecture.

Figure 1-3 and Figure 1-4 provide a visual representation of the architecture types and their relation to the physical hardware.

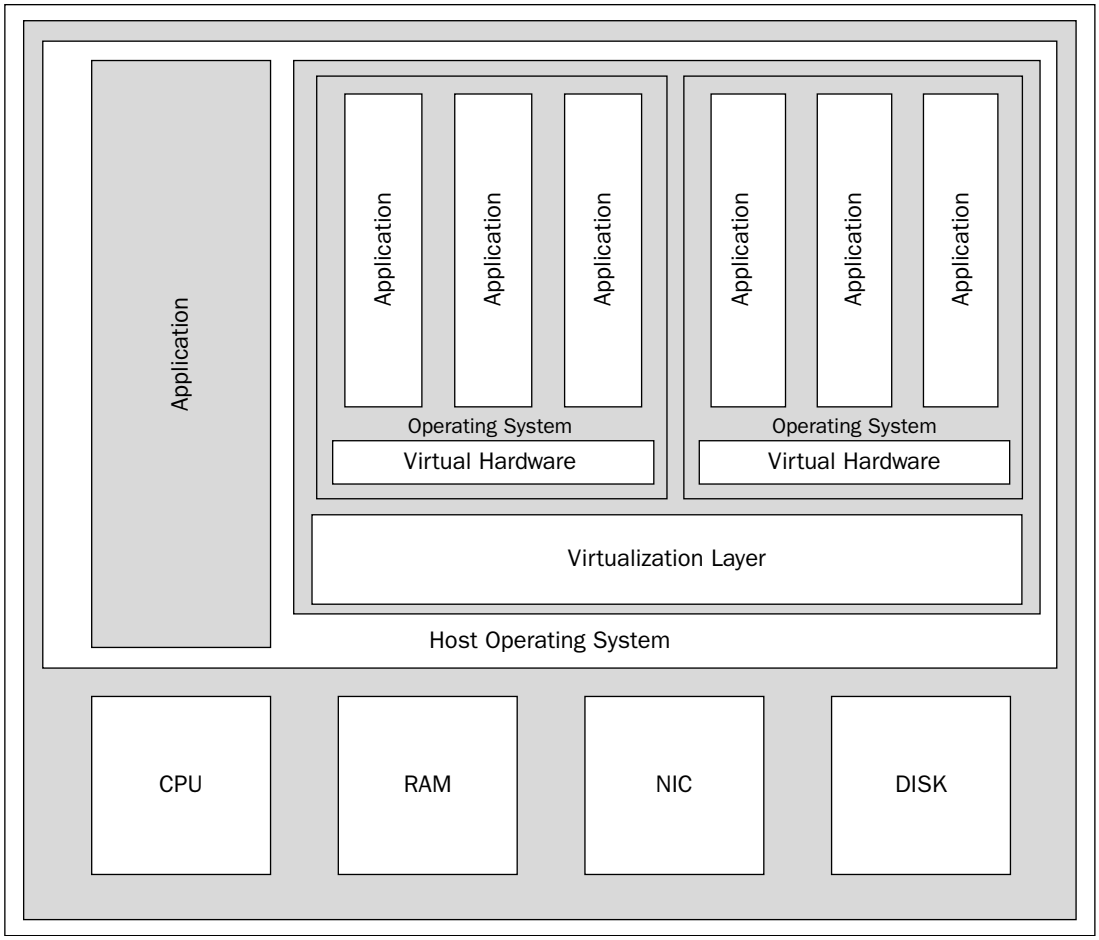

Figure 1-3

Hosted style architecture relies on the underlying operating system for hardware support. This means that theoretically as long as the underlying operating system supports the hardware it should be available to the virtualized machines. This allows for a greater variety of supported hardware. This comes at a price however, the added overhead of the physical machine's operating system. This is where the hypervisor architecture comes in. It doesn't require the physical machine to have a pre-existing operating system to be installed. In practice the virtualization software will install its own kernel and drivers for hardware. This allows for a dramatic decrease in overhead over the hosted architecture but it also seriously reduces the choices in compatible hardware. The reason is that companies, like VMware, have to choose a subset of hardware to write compatible drivers for, whereas drivers are readily available for hardware under Windows XP or Windows Server 2003 because the hardware manufacturers produce the drivers.

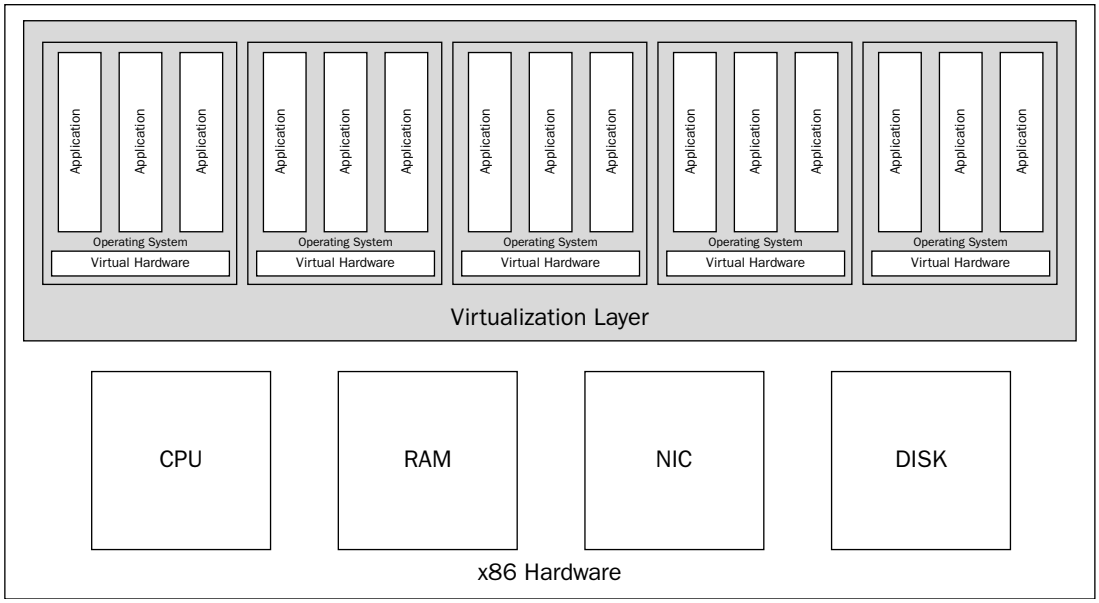

Figure 1-4

In conclusion, the following tables outline the advantages and disadvantages of each type of architecture.

Hosted Architecture	
Advantages	**Disadvantages**
Ease of deployment	Increased cost due to licensing of host software
Greater hardware support	Physical resources such as memory are shared between the host and virtual machines
Decrease cost for virtualization software	Less server utilization possible due to the additional overhead of the underlying host operating system

Hypervisor Architecture	
Advantages	**Disadvantages**
Greater resources for the virtual machines	Decreased hardware support
Decreased cost for licensing because no host operating system is required	Increased cost for virtualization software. Hypervisor software like VMware ESX server is considerably more expensive.
Increased utilization of server hardware	

Terminology, Host vs. Guest

If you remember the definition of hosted architecture you'll undoubtedly see where the term host comes from. The *host machine* is the physical machine itself, the machine where you will be installing VMware Server. The *host operating system* is the actual operating system installed on the host machine. Many times you will simply see them both referred to as just the host.

In the discussion of virtualization earlier in the chapter you learned that there are segments, or partitions, created in order to isolate and house your virtual machines. Once a new virtual machine is created, it is essentially a clean slate. Just like with a new PC you must install an operating system onto the newly created virtual machine. The *guest* refers to the operating system installed on the virtual machine. As with host, many times you'll see guest refer to both the virtual machine and its operating system.

I cover how to create virtual machines in Chapter 4.

I could spend the next several pages going over some of the other terminology used in virtualization and VMware Server; however, I will leave that for the chapters ahead. Nevertheless, I did want to stress the importance of understanding this difference between the host and guest. Though it is not a difficult concept, it is a very common one over the next several chapters.

So What Is VMware Server?

VMware Server is a multi-platform (Windows Server and Linux) application that provides host-based software virtualization. This gives you the ability to create and operate multiple x86-based virtual machines simultaneously on the same physical machine. Of course all of that is great, but what does it really do for you? Here are just a few of the advantages to running your server or development environment within virtual machines:

- Rapid deployment/decreased setup and tear down of new machines
- Multiple platforms available on a single physical machine
- Ability to move virtual machines from one server to another with no loss of information
- Increased utilization of your hardware/decreased costs

The next few sections go over each of these advantages in more detail.

Rapid Deployment

Especially in a development environment you'll find the ability to set up and tear down a set of virtual machines quickly is worth its weight in gold. This rapid setup and tear down can be accomplished by creating a library of base images. These images will contain a static install of, for example, Windows XP. Having a library of base images will allow you to quickly unpack and deploy virtual machines in a matter of minutes.

Chapter 4 covers creating and organizing a base image library.

As an example, say you have an obscure application that works right only on Windows 98. A customer calls with a serious problem to troubleshoot. In the past you would have to either dual boot a workstation into Windows 98 or have an old junk PC with Windows 98 on it for testing. Of course they aren't just lying around when you need them to be; you have to go find what door they are propping open, and what should be a 15-minute support call turns into a whole afternoon.

Now, say you receive the same phone call but this time you have a handy virtual machine already set up with Windows 98 installed. You simply connect to the VMware Server console and fire up your Windows 98 machine. You have a testing environment set up and ready within minutes, usually with the customer still on the phone. Once you finish the call you simply shut it down and forget it.

Multiple Platforms Available on a Single Physical Machine

If you're like me you spend most of your day on a Windows PC doing Windows development. The problem is that I like to tinker with new technology. I always want to try out that new Linux distro, but I never have a spare machine just lying around. This is where one of my favorite benefits of virtualization comes into the picture. If you remember earlier I talked about how the virtualization layer abstracts out the hardware from its implementation. This allows you to not just run Windows-based virtual machines, but you can run Linux, BSD, and so on. Almost anything that will run on an x86, you can run side by side with anything else in VMware Server, regardless of the host operating system. I just love that part. I would have to say the most common use I've found for this, from a developer's view, is setting up an Apache environment for web development. PHP and Perl just work so much better on an Apache installation in Linux.

The Ability to Move Virtual Machines from One Server to Another

As you will learn in Chapter 3, virtual machines in VMware Server are really nothing more than files on your hard disk. This coupled with the independent nature of the image with respect to the host hardware makes it extremely portable. In other words, there is nothing that keeps you from moving that Windows XP base image from your development VMware Server over to your VMware Workstation installation or over to a VMware ESX Server.

> *There are some minor incompatibilities between VMware Server, ESX, and Workstation images. The product you use to create the image will prompt you if a choice is made that will render the image incompatible with other versions.*

As an example, suppose you have two highly critical servers running in virtual machines on one server. Now, and we've all had this happen, a fan goes out on that server. You can't buy the fan anywhere but direct, it always seems to be that way. You can rig the machine to come up but you don't have room in your data center for a machine cracked open with a box fan propped above it. All you have to do is get that server up long enough to copy those virtual machines over to another VMware Server and switch them on. Problem solved and your users are never the wiser. Suppose this server was simply a firewall or instant messaging server. Backup images of machines like that can be kept on another VMware Server machine for easy, quick fall over.

The flexibility all the VMware products offer in this regard is outstanding. If care is taken to maintain compatibility when the virtual machines are created, you can move, mix, and match your virtual machines in any way you see fit. You can even use virtual machine images from other virtualization products like Microsoft Virtual PC and Microsoft Virtual Server. VMware Server can read and convert the images into a format compatible for use on your VMware products.

Increased Utilization of Your Hardware

Figuring the Return on Investment gives me a headache; however, it doesn't take too much to see that a $3000 server that averages a CPU load of 25 percent has a lower return than a $3000 server with an average load of 75 percent. The simple fact is your servers are probably underutilized. Take any server currently in your datacenter and monitor it for several hours. It may shock you to learn that an average server is rarely utilized over 10 to 20 percent. That leaves a bit to be desired in terms of your investment. VMware Server can keep that server doing what it was bought for, pushing bits. By consolidating your servers into a single host machine capable of multiple virtual machines, you increase the return on your investment. I haven't even figured in the decrease in power consumption, air conditioning, hardware failure, and other issues that plague a datacenter made up of several individual servers where each performs just a handful of tasks.

A perfect example that I've seen and heard from many people is that they have a piece of software that cannot be installed alongside a different piece of software; however, both are needed in the complete system. The product documentation probably tells you that you need two servers. Why? You just need virtualization. Create two virtual machines. Install the operating system and software needed on each and away you go. You save yourself the cost of an additional server and most likely still do not max out the hardware on your existing server.

Life without VMware Server

Looking at this in terms of your development process, what is life like without VMware Server? I may not be able to answer that question from your perspective, but it is certainly my goal to show you what your development process can be like with VMware Server as a part of it.

In terms of physical machines the common developer has three requirements, depending on the project of course:

- **Developer machine** — Usually the machine on your desk.
- **Servers** — Where you develop web-based code (Web Services, ASP.NET, and so on).
- **Test bed** — The bank of machines you use to test your code.

In a world where virtualization did not exist, the preceding requirements could result in a large number of physical machines. The test bed alone, if you were doing any serious stress or regression testing, could be comprised of several machines itself. Of course there can be many other types of machines added to the preceding list, but for the purposes here it covers the basic requirements.

Now, as you probably already know, I'm going to tell you the machines required for the preceding scenario would fit perfectly into virtual machines.

First, having your developers write code from within a development virtual machine has numerous advantages. In Chapter 4 I cover the advantages and disadvantages of placing your developers solely within a virtual machine. One of the biggest advantages, for the sake of discussion here, is a static environment. You can create a static development image complete with all the tools, IDE, and so on needed to function within your environment and then distribute that out to your development team. Now all of your developers are writing code on the same platform with the exact same settings and install. No more worrying about your developers installing the latest game or spyware on their work PC. Everything is static and isolated within your development image.

> *There are numerous advantages and disadvantages to developing within a virtual machine as well as a few instances where this approach might not produce the results you desire. Further information is provided in Chapter 4.*

Development servers are a perfect candidate for virtualization. Typically your development servers are not production servers, and most companies don't have the budget or extra hardware lying around to house a server just for testing code. There's almost no better use for a virtual machine.

The final common requirement is the test bed. There were two ways to handle multiplatform testing in the past. Either you had a room full of physical computers with various operating systems installed on them or you had one machine that you just spent hours in front of. More often than not in the latter situation testing would take a back seat due to the huge time requirement of testing on one machine. This is prime time stuff for VMware Server. Not only can you accomplish multiplatform testing for your code with multiple virtual machines, but I show you later in this book how you can automate the process so you can push builds out, automatically start virtual machines, and even retrieve status back from your installers and applications while testing. Imagine your nightly build starting up the necessary virtual machines, copying over the bits, and e-mailing your QA department that it's ready to be tested. That's some pretty cool stuff.

Summary

Over the past few sections you've learned about the following:

- ❑ The basics of virtualization, the types of architecture utilized, and terminology
- ❑ What VMware Server is and the advantages of using virtual machines
- ❑ The difficulties of life without VMware Server, or virtualization in general

Chapter 2 goes over the installation steps required to get VMware server up and running on your Windows or Linux-based server. In the following chapters you will take that installation and go through some ways in which you can integrate virtual machines into your existing development processes and hopefully increase your productivity.