Chapter 1

The Roots of Virtualization

In This Chapter

- Understanding virtualization
- Examining the virtualization ecosystem
- Looking at x86 server enhancements and multi-core processors

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 in virtualization.

Why Virtualization

Technological evolution both drives, and is driven by, ever-increasing levels of abstraction in hardware and software architectures. High-level programming languages allow programmers to implement software without having to pay too much attention to the operating systems on which it will run. One job of operating systems is to provide the abstractions that free programs from the complex and varied details needed to manage memory and input/output (I/O) devices.

The operating systems that provide this isolation must be totally cognizant of the hardware on which they reside. Details like MAC and IP addresses, SAN LUN assignments, physical memory configurations, processor counts, and system serial numbers become enmeshed within the OS

state at the time of system installation. This *stateful information* locks the OS to the specific hardware on which it was installed and complicates hardware fault recovery, system upgrades, and application consolidation.

Coming Up with the Concept

Hardware and software architects realized that if they could abstract the hardware as seen by the operating system, then they could finesse the software's view of the physical configuration on which it was installed. This approach is kind of like the *black box* model — the software doesn't really need to know what's going on inside the black box as long as computations sent into the box come back out with the correct results. The architects called their approach *virtualization,* and it turns out to be more complex than you might imagine.

Complexity arises because an operating system believes it owns all resources on the hardware on which it runs. More than that, an operating system doesn't like to be fooled. This challenge is made more difficult still, given that the x86 architecture came into existence before notions of virtualization became common currency in mainstream computing.

The Hypervisor

Server virtualization enables you to consolidate many different types of workloads and operating systems onto virtual environments all running on a single hardware platform. Virtual servers or virtual machines are independent operating environments that use virtual resources. One of the most common approaches to virtualization is to use hypervisor technology. *Hypervisors* use a thin layer of code in software to achieve fine-grained, dynamic resource sharing. Today's hypervisors provide a high level of flexibility in how virtual resources are defined and managed and have become a favorite choice for server virtualization.

Two types of hypervisors exist. The two types are discussed in the following sections.

Type 1 hypervisor solutions

You will find that *Type 1 hypervisors* are typically the preferred approach for server consolidation because they can achieve higher virtualization efficiency than Type 2 hypervisors by dealing directly with the hardware. Type 1 hypervisors provide higher performance and efficiency, and use hardware assisted virtualization technology like AMD-V™ technology. Powering ultrathin notebooks to blade servers, all AMD processors shipped are designed to use AMD-V features.

Type 1 hypervisors use a thin layer of code to provide resource sharing within a single hardware platform. In other words, the hypervisor provides a standard emulated hardware environment that the guest OS, sometimes referred to as the *virtual machine (VM),* resides on and interacts with. The VM *encapsulates* the guest operating systems and any applications running on them into a single entity that is isolated from the underlying hardware. It is because of this encapsulation that the VM can be migrated from one physical machine to another without any service interruption.

Not only does this approach support running multiple VMs on the same hardware, it can also support multiple VMs running different types and/or versions of operating systems (for example, completely different operating systems like Windows and Linux can be run simultaneously on the same physical server).

Type 1 hypervisor solutions are often used for server consolidation to achieve higher levels of resource utilization. Software development and quality assurance environments can also benefit greatly from this type of virtualization due to the ability to allow a number of different operating systems to be run simultaneously. This can facilitate parallel development or testing of software in a number of different operating system environments, thus leading to quicker, and perhaps more efficient, testing phases.

Early Type 1 hypervisor solutions experienced some performance degradation due to the overhead of the virtualization software, however, with the introduction of hardware-assisted virtualization that is built into some of today's multi-core processors, and platform improvements, such as expanded memory and I/O capabilities, this performance degradation has been minimized. In fact, there is evidence that some native applications may run as well, if not better, in a virtual environment due to these improvements.

Operating system virtualization — Type 2 hypervisor

Type 2 hypervisors run on a host operating system that provides virtualization services such as I/O device support and memory management. These services give an application the illusion that it is (or they are, if there are multiple applications) running on a machine dedicated to its use. 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 interacts with its virtual OS as though it has sole control of the resources of the virtual OS. However, it can't see the applications or the OS resources located in another virtual OS.

Virtualization solutions that use a Type 2 hypervisor are also referred to as operating system (OS) virtualization, and in some environments are called *containers*.

This approach to virtualization is 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 container virtualization to allow a hosted Web site to "believe" it has complete control of a machine, while in fact each hosted Web site shares the machine with many other Web sites.

There are some limitations to operating system virtualization, though. First and foremost, this approach typically limits operating system choice. Containerization usually means that the containers must offer the same operating system as the host

OS and even be consistent in terms of version number and patch level. As you may imagine, this 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. Consequently, operating system virtualization is best suited for homogenous configurations.

Virtualization Makes Hardware More Important

Even though virtualization is a software technology, it has the effect of making hardware more important. Removing lots of servers and migrating their operating systems to virtual machines makes the remaining servers that support all those virtual machines vital to running your business.

For example, in an organization that has a "one application, one server" environment, if a single server goes down, this inconveniences a single user population. However, virtualization is very different. If each server supports multiple virtual machines, then a server that goes down would inconvenience multiple user populations. Thus, the hardware can take on greater importance in a virtual environment.

The greater importance placed upon hardware by virtualization technologies is one reason why hardware manufacturers like AMD have introduced virtualization-ready technologies like AMD-V technology. AMD-V technology can be found in all AMD multi-core processors, and other hardware manufacturers have introduced similar enhancements to support virtualization implementations.

The Virtualization Revolution in Hardware

Because virtualization inevitably leads to more apps running on a single hardware instance, hardware becomes more important. And hardware has responded. Well, actually, hardware *manufacturers* have responded, but you get the point.

The virtualization-enabling hardware developments break into two categories:

- **Virtualization extensions to processors and associated low-level hardware like network interface cards (NICs).** These extensions allow virtualization to integrate more closely with the underlying hardware in order to improve performance and security. For more on these hardware changes, see Chapter 3.
- **Form factor changes to servers to support virtualization workloads more effectively.** Originally, standard 1U servers (sometimes called *pizza boxes*) were used as virtualization platforms. These servers typically had one processor, up to 16GB of memory, and a limited number of NICs — a design that makes sense in a one application, one server world. In the new virtualization-oriented world, though, every one of these capabilities turned into a bottleneck: To run large virtualization workloads, these servers had too little processing power, not enough memory, and nowhere near enough network connections. So server manufacturers have transformed their server designs to better support virtualization with these design extensions:
    - **Multi-core processors.** Today's multi-core processors help to improve utilization and increase VM density by allowing many virtual machines to share a single physical server. Additionally, today's multi-core processors can be more energy efficient than single core processors, depending on their usage and configuration, potentially providing further energy cost savings.
    - **Multi-socket servers.** Today, single-socket servers are passé for virtualization. Servers with two or four multi-core processors, sometimes referred to as *2P* or *4P* systems, are the norm.
    - **Big memory servers.** Of all the bottlenecks in virtualization, memory is typically the first encountered. Solution? More memory. Manufacturers are now releasing servers with memory capacities of up to half a terabyte of memory, and, as with all things hardware, you can expect to see memory capacity climbing in the future.

- **Blade servers.** In yet another scheme to pack more processing into less room, server manufacturers have pushed all this hardware goodness into blade servers — servers that contain *blades* (actually, an entire server contained on a small plug-in board) are slotted into a multi-U chassis. Instead of 4 units in a rack holding four computers, a blade server can contain 16 or even 32 blade computers in the same space. Bottom line, more server for a given amount of rack space, and more virtualization for a given amount of rack space.