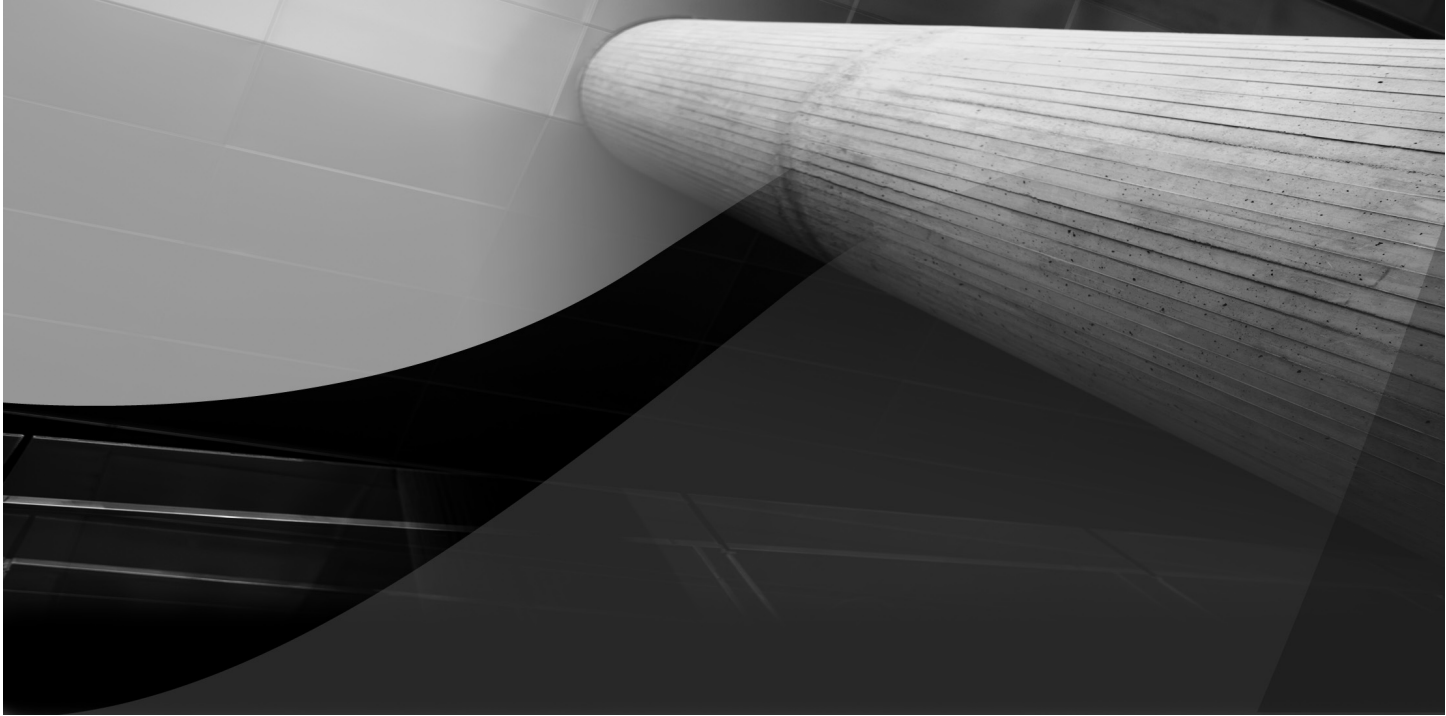


PART
I

Introduction



CHAPTER 1

Introduction to Virtualization



In recent years, virtualization has changed the way we look at computing. Instead of using many different computer systems for different tasks, with virtualization, we can use a single system to host many applications. Not only has virtualization increased in popularity, but it has also sparked new hardware CPU innovation, including Intel VT-x and AMD-V technologies. Over the next few years, virtualization will be a core technology in every data center. Oracle VM, although a relatively new entry in the virtualization market, is based on stable and proven technology.

This chapter provides an overview of virtualization technologies, including the various types of virtualization and its typical uses. The reasons that companies choose to use virtualization and how they use it are also discussed. Subsequent chapters discuss the mechanics of installing, configuring, and managing Oracle VM.

What Is Virtualization?

Virtualization is the abstraction of computer hardware resources. This definition is very general; however, the broad range of virtualization products, both hardware and software, makes a more specific definition difficult. There are hardware products that allow for virtualization, software products that create virtual systems, and hardware that assists with software virtualization. All of these products and options perform essentially the same function: they separate the operating system and applications from the underlying hardware.

A number of different types of virtualization are available, including the following:

- Hardware virtualization
- Full software virtualization
- Paravirtualization
- Hardware-assisted software virtualization
- Component or resource virtualization

As this book progresses, you will learn about these types of virtualization and their attributes. You will also obtain the information you need to decide which type of virtualization to use.

Although virtualization allows you to abstract resources away from the hardware layer, you'll still discover some limitations. With today's commercially available technology, virtualization—or at least the most popular types of virtualization—allows you to abstract only like architectures. For example, if you use software virtualization

that runs on x86 or x86_64 architecture, you can run only virtual hosts with either an x86 or x86_64 operating system. In other words, you can't virtualize a SPARC system on an x86 or x86_64 architecture.

At this time, several major virtualization products are on the market:

- VMware was one of the first companies to offer a fully virtualized hardware platform environment, including a range of products with fully virtualized environments. VMware was founded in 1988 and was acquired by EMC in 2003. In addition to hardware virtualization, VMware also offers some paravirtualized drivers.
- Microsoft Hyper-V was recently released. Hyper-V provides both fully virtualized and guest-aware virtualization (if you are running a Windows guest).
- Xen Hypervisor is an open-source standard for virtualization and runs on multiple platforms. The first public release of Xen was in 2003, and the company was acquired by Citrix in 2007. Xen currently supports both Hardware Virtual Machine (HVM) and Paravirtualized Machine (PV). Xen does not offer any paravirtualized drivers.
- Oracle VM is a free, next-generation server virtualization and management solution from Oracle that makes enterprise applications easier to deploy, manage, and support. The Oracle VM hypervisor is an open-source Xen project with Oracle enhancements that make it easier, faster, and more efficient. In addition, Oracle VM is currently the only virtualization product supported for the Oracle Relational Database Management System (RDBMS) and other Oracle products. Oracle VM supports both HVM and PV and provides a set of paravirtualized drivers for both windows and Linux Paravirtualized Hardware Virtual Machines (PVHVM).

In addition, numerous proprietary hardware and software products allow you to virtualize specific vendors' hardware and operating systems. Many products are also available to virtualize networks, storage, and so on; although some of these will be discussed in this book, as appropriate, the focus is on Oracle VM.

Reasons for Virtualizing

There are many reasons for creating systems in a virtualized environment rather than using physical hardware. In general, if your applications allow for virtualization, a virtualized environment lets you allocate sufficient resources to your applications using less power and with fewer cooling and space requirements. If applicable,

in a virtualized environment, many virtualized systems can run on one physical computer, consuming fewer resources. Here are the top reasons for virtualizing:

- Server consolidation
- Server provisioning
- Functional separation
- Performance improvement
- Backup/restore
- Hosting and cloud computing
- Training, testing, quality assurance, and practice

Server Consolidation

Server consolidation is probably the top reason for virtualizing your environment. To consolidate servers, you take multiple computer systems and collapse them into, or consolidate them into, one server. Server consolidation is often, though not necessarily, accomplished using virtualization. From an Oracle database perspective, you can consolidate servers in multiple ways. You can consolidate servers by taking databases from several servers and putting them on a single server, each with their own instance (database). You can take databases from several servers and put them on a single database server within the same Oracle instance, either with or without partitioning. Finally, you can consolidate multiple Oracle databases onto a single server by creating multiple individual virtual machines on a single server and then creating a single Oracle database instance on each virtual machine.

Single Server, Multiple Instances

This option allows you to migrate multiple databases to a single server. You then configure the memory and CPU resources for each instance, thus dividing resources among the various instances. With this option, you consolidate multiple instances onto the same database server and still have some control over how the resources are allocated. The downside, as with the next option, is that some system resources, such as open files and the number of processes, can be exceeded. In addition, a single instance can sometimes utilize a disproportionate amount of system resources.

Single Server, Single Instance

With this option, you run all of the data from all of the instances to be consolidated from the same Oracle instance. This way you can share resources more easily,

but other problems, such as security issues, often arise. In addition, a single instance often utilizes a disproportionate amount of system resources. Performance issues can also occur, as all logical databases must share a single set of initialization parameter values for the Oracle database instance. This option is usually not the one to choose when consolidating Oracle databases.

Multiple VMs, Multiple Instances

This option allows you to configure and manage each database separately, as with the first option. However, because each Oracle instance runs on its own virtual server, this option has a few additional benefits. Because you can move a VM dynamically between underlying VM servers, you have more flexibility of resources. If a system needs more resources, you can move it or another VM sharing that server, thus lessening the load.

This option's downside is that the OS is duplicated, causing more overhead than with the first option. When consolidating an Oracle database, you'll find both this option and the first option are often good choices.

Server Provisioning

One of the primary advantages of using virtual machines is the ability to provision servers quickly and efficiently. A virtual environment allows you to keep templates of various types of servers—such as application servers and database servers—and deploy and put them into production easily in a matter of minutes or hours. The process of provisioning a server typically involves either copying a server or building a new server from a template. Once you have deployed this new server, you run scripts to rename the server, change its IP address, etc., getting it into production quickly.

By prestaging servers and templates, you can quickly, efficiently, and precisely deploy servers. By using a prestaged server, you also avoid mistakes, which makes the provisioning process faster and more exact. You can easily pick what type of server to deploy from your library of templates and prestaged servers. The Oracle VM architecture allows you to deploy the new server into any system in the server farm that you desire, creating a high performance, scalable server in a very short time.

Functional Separation

Various applications don't work well together in an operating system. The reason is primarily due to resource contention in certain subsystems in the OS, such as the memory subsystem, the I/O subsystem, or the OS scheduler. Often application servers do not scale (achieve higher performance) when additional resources such as CPUs or memory are added. In these cases, using several smaller servers (two CPUs, 2GB of RAM) is better than using a larger server with many CPUs and

lots of RAM. This is especially true of Java, which doesn't scale very well with large amounts of memory and multiple CPUs.

Even when application servers don't scale well with large numbers of CPUs and lots of RAM, you can still take advantage of the cost effectiveness of these systems by virtualizing them. Scaling problems typically occur in the application itself; in the OS scheduler, where the system simply can't keep track of too many processes; or in Java. Java applications typically can't take advantage of more than 2–3GB of RAM. By virtualizing, you can assign each system a reasonable amount of resources that the application can handle. Because the process tables and application queues exist in the virtual server and not on the host, the virtual environment will continue to scale. Fortunately, the Oracle database server scales very well with CPU and memory resources.

In addition, you can use functional separation to split management responsibilities, provide additional security, and reduce contention. By virtualizing and splitting up applications by vendor or group, you can allocate a single application to a single server and still not waste hardware resources.

Performance Improvement

As mentioned in the previous section, application servers often perform better as several smaller servers rather than one large server. Therefore, virtualizing a large server into multiple smaller ones can often improve application server performance. By splitting up a large host system into multiple smaller VMs, you can often achieve not only higher performance but also a greater level of scalability.

In addition, the virtual environment provides a feature that you can't find in physical servers: the ability to move the VM to another host. This gives you the flexibility and ability to alter the physical characteristics of the underlying host while the virtual machine is still up and running, giving you the benefits of live load balancing and performance improvements without interrupting service to the VM user.

By moving VMs from one host to another while live, you use resources efficiently and you can reuse them when needed. If a host becomes too busy, you can add another to the server pool and move VMs to it as needed.

Backup/Restore

An often overlooked feature of virtual machines is the ability to back them up quickly and efficiently because the virtual machine looks essentially like several large files to the host OS. Most performance problems that occur during backups are due to the number of small files in a system. When backing up a small file, the backup software takes more time opening and closing the file and finding all of the parts of the file than it actually takes to read the contents and back them up. The amount of data backed up per second drops significantly for small files.

When backing up an operating system, the backup software traverses the entire directory structure and opens, copies, and then closes each file. With large files, the software spends more time copying data. With small files, the software takes more time opening and closing files. Because the virtual disks are actually a large file in the OS, backups are very fast.

Backups from the VM can be problematic if the VM is running because the state of the files is constantly in flux (a file might only be halfway written when the backup occurs). Memory may also be in flux. Considering that most systems use hard drive files for swap, the backup could become corrupted using this approach. Therefore, you must consider a backup of a virtual machine file a “crashed server.” You should back up the virtual server image when the virtual OS is shut down.

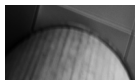
If you’re running backups from the VM itself in order to back up specific items or to back up the database directly using RMAN, then the advantages you achieve by backing up the entire VM image are not realized.

For the same reasons that backing up the entire virtual disk is much faster than backing up each individual file within the virtual disk, restoring is similarly optimized. Restoring one large virtual disk file is much faster than restoring each file individually within the virtual machine.

Hosting and Cloud Computing

One of the most common uses of virtualized machines is for hosting environments. Hosted environments and cloud computing use virtualized machines to provide on-demand computing to their clients. Depending on the underlying hardware, dozens of virtualized systems might be running on the same underlying host server(s). This model allows the hosting vendor to provide virtual systems to its clients on an as-needed basis. In addition, the hosting vendor can easily size the system to the desired number of CPUs and RAM instantly.

Many vendors now offer hosted solutions and cloud computing. *Cloud computing* defines a type of computer resource that is available over the Internet and is made up of resources that you know little about. These resources, usually in the form of a Linux or Windows cloud computer, are provided with a set number of CPUs, a set amount of memory and disk space, and are available to be used anywhere. Typically the end user doesn’t really know anything about the underlying hardware or software. This resource is just a commodity.



NOTE

Two types of cloud computing are available: private and public. Private clouds are hosted using internal networks. Public clouds are available via the Internet to different users or customers.

Cloud computing has become more popular in recent years as many customers have decided to abandon expensive data centers and to acquire just the resources

needed to run their business. Cloud computing is less expensive than traditional computer systems—up to a point. The higher-end cloud computers tend to be a little pricey.

Training, Testing, Quality Assurance, and Practice

Some of the most popular uses of virtualization are for training, testing, quality assurance (QA), and, of course, just plain practice. Virtual environments are absolutely ideal for testing environments, because they are easy to set up, use, and refresh. All of these types of activities benefit from those features.

Training

Training is by far one of the best uses of virtualization. Oracle employs a great deal of virtualization for training purposes. By using virtual machines, you can reset the VMs for the next week's classes by simply removing and replacing them with new copies. You can add, ahead of time, the correct software, class exercises, and configuration and then re-deploy whenever necessary. Oracle has an entire library of training classes, including ones on:

- Oracle 11g
- Oracle 10g
- Oracle RAC
- Oracle on Windows
- Oracle Application Server

By storing several classes, you can easily retask your training facility to handle any class you need. Since many classroom environments are not CPU intensive, you can create several virtual machines on the same underlying host.

Training systems are easy to set up as well as access. By using technology such as Virtual Network Computing (VNC), students can access the systems and run X-Windows programs such as the Oracle Universal Installer (OUI) and the Database Configuration Assistant (DBCA). Later in this book, I will cover VNC in more detail. You can also create a training system assigned to individual students, so they can experience managing an entire OS environment. When the class is finished, or if the student causes irreversible problems with the system, you simply remove and re-create it.

Testing

Testing is an important part of every system. Unfortunately, many companies do not have the resources to purchase test systems, leading to severe problems and risk to

the production environment. I have actually been called in on occasion to assist with recovering from problems that resulted from performing an upgrade or patch without thoroughly testing it on a test server before putting it into production. The reason for not having a test server is always budgetary.

With a virtual environment, you can put a test system into place efficiently and economically. This way, your database administrators (DBAs) can test upgrades, patches, installations, new configurations, and even undertake performance testing. Without a proper test system, the DBA might miss some step or important detail of a patch or upgrade that he or she might not otherwise notice. By testing installations, configurations, and deployments, the DBA will be more comfortable with and better at doing these tasks. Test systems are one of the most important resources you can have. And once you've finished testing, with one click, you can move the test system into production.

By using Oracle VM, you can create multiple test environments and also save these environments at various stages, so you can deploy the next step over and over again, thus testing exactly what you need to test. You also can preconfigure virtual machines with different versions of the OS and various configurations, including Real Application Clusters (RAC).

QA

Having a set of virtual machines for QA has many advantages. You can keep multiple versions of operating systems, databases, and applications and test modifications multiple times using different variations. This way, you can perform multiple tests with very little setup time. You can also refresh and reset the QA system whenever a new code drop is released so you test it efficiently.

Practice

Having a system to practice and test new things on is always good. You can hone your skills and try new things such as backup methods, patching, and upgrades. By practicing and testing new things, you improve your skills, leading to professional advancement and self-confidence.

Overview of Virtualization Technologies

Virtualization has been around for quite a while now; however, its mass appeal has only been realized with the extensive improvements that have appeared in the last few years. In virtualization's early days, you had to purchase very expensive hardware; now, due to its commoditization, you can use commodity PC servers and download free virtualization software to get started.

A number of different virtualization technologies are available in the market today, including the following:

- Full software virtualization
- Hardware-assisted software virtualization or Hardware Virtual Machine (HVM)
- Paravirtualization or Paravirtualized Machine (PV)
- Hybrid virtualization technology (PVHVM)
- Component or resource virtualization

Depending on your situation, you might be able to take advantage of one or more of these virtualization types. In this section, I'll explore each virtualization type, along with their pros and cons. Each type has its own attributes, which provide specific benefits. The type of virtualization that you choose depends on your needs. Oracle VM supports hardware-assisted software virtualization, paravirtualization, and the hybrid PVHVM.

Full Software Virtualization

In *full software virtualization*, all the hardware is simulated by a software program. Each device driver and process in the guest OS believes it is running on actual hardware, even though the underlying hardware is really a software program. Software virtualization even fools the OS into thinking that it is running on hardware.

One of the advantages of full software virtualization is that you can run any OS on it. It doesn't matter if the OS in question understands the underlying host hardware or not. Thus, older OSs and specialty OSs can run in this environment. The architecture is very flexible because you don't need a special understanding of the OS or hardware.

The OS hardware subsystem discovers the hardware in the normal fashion. It believes the hardware is really hardware. The hardware types and features that it discovers are usually fairly generic and might not be as full-featured as actual hardware devices, though the system is functional.

Another advantage of full software virtualization is that you don't need to purchase any additional hardware. With hardware-assisted software virtualization, you need to purchase hardware that supports advanced VM technology. Although this technology is included in most systems available today, some older hardware does not have this capability. To use this older hardware as a virtual host, you must use either full software virtualization or paravirtualization.

**NOTE**

Only hardware-assisted software virtualization requires advanced VM features; full software virtualization does not. VMware ESX works on older hardware that does not have any special CPU features. This type of virtualization is also known as emulation.

Unfortunately, full software virtualization adds a lot of overhead. This overhead translates into extra instructions and CPU time on the host, resulting in a slower system and higher CPU usage. With full software virtualization, the CPU instruction calls are trapped by the Virtual Machine Manager (VMM) and then emulated in a software program. Therefore, every hardware instruction that would normally be handled by the hardware itself is now handled by a program.

For example, when the disk device driver makes an I/O call to the “virtual disk,” the software in the VM system intercepts it, then processes it, and finally makes an I/O to the real underlying disk. The number of instructions to perform an I/O is greatly increased. This process is diagrammed and explained in further detail in Chapter 3.

With networking, even more overhead is incurred since a network switch is simulated in the software. Depending on the amount of network activity, the overhead can be quite high. In fact, with severely overloaded host systems, you could possibly see network delays from the virtual switch itself. This is why sizing is so important.

Hardware-Assisted Software Virtualization

Hardware-assisted software virtualization is available with CPU chips with built-in virtualization support. Recently, with the introduction of the Intel VT and AMD-V technology, this virtualization type has become commoditized. This technology was first introduced on the IBM System/370 computer. It is similar to software virtualization, with the exception that some hardware functions are accelerated and assisted by hardware technology. Similar to software virtualization, the hardware instructions are trapped and processed, but this time using hardware in the virtualization components of the CPU chip.

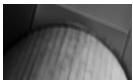
By using hardware-assisted software virtualization, you get the benefits of software virtualization, such as the ability to use any OS without modifying it, and, at the same time, achieve better performance. Because of virtualization’s importance, significant effort is going into providing more support for hardware-assisted software virtualization. Hardware-assisted virtualization also supports any operating system.

Using hardware-assisted software virtualization, Oracle VM lets you to install and run Linux and Solaris x86-based OSs as well as Microsoft Windows. With other

14 Oracle VM Implementation and Administration Guide

virtualization techniques, Oracle VM only allows Linux OSs. This technique also makes migrating from VMware systems to Oracle VM easier, as described in Chapter 10.

As mentioned earlier, both Intel and AMD are committed to supporting hardware-assisted software virtualization. They both introduced virtualization technology around the mid-2005–2006 period, which is not that long ago, and their support has improved the functionality and performance of virtualization. Intel and AMD do not yet fully support paravirtualization; however, hardware-assisted software virtualization components are changing at a very fast pace, with new features and functionality being introduced continually.



NOTE

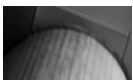
Hardware-assisted virtualization is really the long-term virtualization solution.

Intel

Intel supports virtualization via its VT-x technology. The Intel VT-x technology is now part of many Intel chipsets, including the Pentium, Xeon, and Core processors family. The VT-x extensions support an Input/Output Memory Management Unit (IOMMU) that allows virtualized systems to access I/O devices directly. Ethernet and graphics devices can now have their DMA and interrupts directly mapped via the hardware. In the latest versions of the Intel VT technology, extended page tables have been added to allow direct translation from guest virtual addresses to physical addresses.

AMD

AMD supports virtualization via the AMD-V technology. The AMD-V technology includes a rapid virtualization indexing technology to accelerate virtualization. This technology is designed to assist with the virtual-to-physical translation of pages in a virtualized environment. Because this operation is one of the most common, by optimizing this function, performance is greatly enhanced. AMD virtualization products are available on both the Opteron and Athlon processor families.



NOTE

The virtual machine that uses the hardware-assisted software virtualization model has become known as the Hardware Virtual Machine, or HVM. This terminology will be used throughout the rest of the book and refers to the fully software-virtualized model with hardware assist.

Paravirtualization

In *paravirtualization*, the guest OS is aware of and interfaces with the underlying host OS. A paravirtualized kernel in the guest understands the underlying host technology and takes advantage of that fact. Because the host OS is not faking the guest 100 percent, the amount of resources needed for virtualization is greatly reduced. In addition, paravirtualized device drivers for the guest can also interface with the host system, reducing overhead. The idea behind paravirtualization is to reduce both the complexity and overhead involved in virtualization. By paravirtualizing both the host and guest operating system, very expensive functions are offloaded from the guest to the host OS.

The guest essentially calls special system calls that then allow these functions to run within the host OS. When using a system such as Oracle VM, the host operating system acts in much the same way as a guest operating system. The hardware device drivers interface with a layer known as the hypervisor. The hypervisor, which is also known as the *Virtual Machine Manager (VMM)*, was mentioned earlier in this chapter.

Hybrid Virtualization Technology (PVHVM)

Since Oracle Enterprise Linux (OEL5), Oracle has provided the ability to create a HVM that uses a few specific paravirtualized device drivers for network and I/O. This hybrid virtualization technology provides the benefits of a paravirtualized virtual machine with the additional hardware accelerations available within the HVM. This technology is still new but might be the future of virtualization. This type of virtual machine is known as a Paravirtualized Hardware Virtual Machine or PVHVM. This virtual machine type has the benefits of both the HVM system and the PV system.

Component or Resource Virtualization

Component or resource virtualization has come to mean many different things, depending on the context. Probably the most popular form of component or resource virtualization is in the area of storage. Network resources can also be virtualized but are not as common.

Storage virtualization is simply the abstracting of logical storage from physical storage. This process has been going on for many years and is very popular. Storage virtualization has tremendous advantages as well as a few drawbacks.

Benefits of Storage Virtualization

Storage virtualization has many benefits. It is easy to use and easy to expand. Storage virtualization lets you add drives to a virtual disk while the system is live and running. Management of the storage system can be done external to the database and system administrators. Storage virtualization also supports large numbers of disk drives and virtual disks.

Virtual disk systems have been around for a long time—especially if you consider Redundant Array of Inexpensive Disks (RAID) to be a virtual storage system. RAID meets the definition of the virtual storage system as an abstraction of logical storage from physical storage. RAID systems provide both performance (from multiple disk drives) and redundancy to protect the system from the loss of a component. As you can see, storage virtualization actually provides many benefits.

Drawbacks of Storage Virtualization

The drawback of storage virtualization is you lose control of I/O resources. The I/O subsystem is one of the most important subsystems for database performance. If the I/O subsystem is overloaded, it can lead to high latencies, which, in turn, lead to slow query performance. The I/O subsystem's performance depends on several factors, including the number of disk drives (or spindles) and how busy those drives are.

In a virtualized storage environment, the storage presented to the server is abstracted from the physical storage. Logical Unit Numbers (LUNs), or virtual disks, are made up of multiple disk drives, and multiple LUNs can span the same disks. Unfortunately, this means you might not know exactly where your storage is coming from and how many other systems are sharing the same drives with that storage, which can cause unpredictable performance.

Because I/O performance is so important to the Oracle database, Oracle Automatic Storage Management (ASM) disks should be made up of LUNs with dedicated drives (i.e., no other LUNs sharing the disk drives). In some virtualized storage environments, this is not possible, however. In fact, some storage systems automatically stripe all LUNs across all drives in the system.

Depending on the I/O subsystem's flexibility, different configuration options are available that might let you have more control over storage. In a theoretical environment, the abstraction of logical to physical shouldn't matter. Unfortunately, in the real world, the speed of the disk drives are finite and thus must be allocated properly.

Miscellaneous

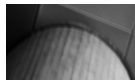
Other forms of virtualization exist that haven't been mentioned in this chapter. These virtualization methods typically involve expensive proprietary hardware. Since this book is focused on Oracle VM, which doesn't use any proprietary hardware, those technologies have been excluded from this chapter. This book will, however, discuss technologies that are similar to and share some of the same ideas as Oracle VM, primarily VMware ESX Server and Xen. As you will learn in the next chapter, Oracle VM has its roots in Xen technology.

The Hypervisor

The hypervisor is what makes virtualization possible. The hypervisor is the component that translates the virtual machines into the underlying hardware. There are two types of hypervisors: The type 1 hypervisor runs directly on the host hardware; the type 2 or hosted hypervisor runs in software. In addition, some proprietary hardware has a hypervisor built into the hardware. The hypervisors that we are concerned with are the type 1 and type 2 hypervisors.

Type 1 Hypervisor

The *type 1* (or *embedded*) hypervisor is a layer that runs directly on the host hardware, interfacing with the CPU, memory, and devices. Oracle VM and VMware ESX Server both use the type 1 hypervisor. The hypervisor treats the host OS in much the same way as a guest OS. The host OS is referred to as Domain 0 or dom0 and guests are referred to as Domain U or domU, as shown in Figure 1-1. Here you can see that all virtual machines must go through the hypervisor to get to the hardware. The dom0 domain is a virtual system just like the domU virtual machines (but has more capabilities as discussed later). Currently, the type 1 hypervisor is considered the most efficient and is the most recommended hypervisor.



NOTE

Even though Oracle VM and VMware both use a type 1 hypervisor, these hypervisors are significantly different. VMware handles device drivers directly in the hypervisor; Xen handles them in dom0 or a driver domain.

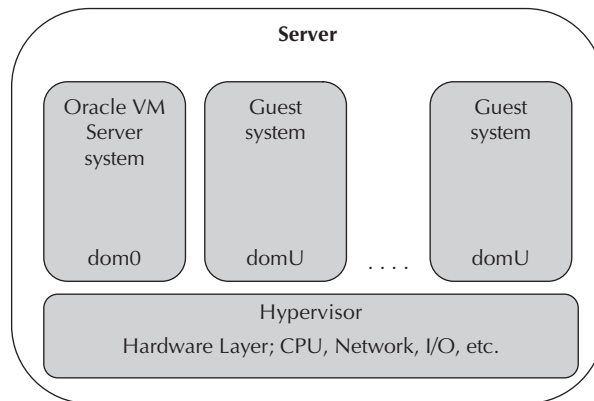


FIGURE 1-1. *The type 1 hypervisor represented graphically*

As you can see from Figure 1-1, the dom0 domain does not differ much from other domains in the virtual environment, except that you access it differently and it is always enabled by default. In addition, dom0 has unlimited rights to hardware, whereas domU only has access through a layer of indirection and only to what dom0 grants it. Because the type 1 hypervisor is essentially part of the OS, it must be installed on the hardware itself and support the devices installed on the system.

Type 2 Hypervisor

The *type 2* or *hosted* hypervisor runs as a program and is used for software virtualization. Because the type 2 hypervisor runs as a program, it has neither the same priority as the type 1 hypervisor nor the ability to access the hardware directly. Its main advantage is you can install it on a variety of host systems without modification. The type 2 hypervisor works with both full software virtualization and hardware-assisted software virtualization. VMware Server is an example of a type 2 hypervisor.

Recently Oracle has released a new product, Oracle VM VirtualBox, which is a type 2 hypervisor-based product.

Hardware Virtualization vs. Paravirtualization

Both hardware-assisted software virtualization and paravirtualization have benefits and drawbacks. With the recent changes in hardware, the software and the technology are changing at a rapid pace.

Currently there are three major types of virtualization: full software, hardware-assisted software, and paravirtualization. Full software virtualization and hardware-assisted software virtualization are very similar as no additional coding or configuration is required (except perhaps to enable virtualization in the BIOS). The main comparison, therefore, really falls between hardware-assisted software virtualization and paravirtualization.

Virtualization Changes Are Happening Rapidly

This section provides guidelines rather than recommendations. At the time of writing this chapter, things are probably much different than they will be when you are reading it. The technology is changing very quickly, and these recommendations might have changed. Do your own research and make your own decisions based on the existing technology.

Benefits of Hardware-Assisted Software Virtualization

The main benefit of software virtualization is its flexibility. You can run any OS on the guest that is compatible with the underlying hardware. That is, if your underlying hardware is Intel or AMD, then the guest must run on Intel or AMD hardware. A number of combinations of 32-bit and 64-bit OSs can run in this environment.

With a 64-bit host, you can run 32-bit or 64-bit guests. Either type of guest is supported, and you can combine both 32-bit and 64-bit guests. When running a 32-bit host, a 64-bit is only available with hardware assistance. If the host system is a 32-bit server without VM technology (hardware assist), the host only supports 32-bit guests. This is described in more detail in Chapter 3.

Drawbacks of Hardware-Assisted Software Virtualization

The main drawback of hardware-assisted software virtualization is I/O and network performance. Because hardware calls are being trapped and processed in software, there is a great deal of overhead. The use of hardware assistance accelerates these operations if they are in the supported system calls (mainly memory), but the technology is still very young and evolving at a rapid pace.

Benefits of Paravirtualization

The primary benefit of paravirtualization is performance. Because the guest OS understands that it is actually running in a VM environment, it can bypass some software interpretations of hardware calls and go directly to the underlying host or hypervisor. Both the guest and host must be running the VM (xen) kernel so they understand and can communicate with the underlying hypervisor.

Drawbacks of Paravirtualization

The main drawback of paravirtualization is the restriction on the supported OS. Because the host and guest must both be running a VM kernel, at this time, paravirtual drivers support only Linux, Solaris x86, and Windows.

Another drawback of paravirtualization is the lack of hardware support. Currently, paravirtualization does not take advantage of the new VM hooks that hardware vendors are adding to their CPU chips. In the future, this is likely to change. When paravirtualization is accelerated with hardware, performance will increase dramatically.

Because the VM kernel must be used, converting a VMware or other image to a paravirtualization virtual machine is difficult. This shortcoming can be resolved,

Hybrid Virtualization

As mentioned earlier, there is a new hybrid type of virtual machine known as the PVHVM. This type of virtual machine uses a combination of a HVM kernel that can take advantage of hardware acceleration and paravirtual I/O and network device drivers. The PVHVM is still very new and has not been used extensively. In the near future the use of PVHVM systems will become more common and more information will become available. Information on the latest in PVHVM technology can be found by visiting my website and blog (see the Introduction of this book for that information).

however, and is covered in Chapter 10, but you will find it is a little more difficult than converting a VMware image to a hardware-assisted software virtualization environment.

If you are running a Linux guest, paravirtualization is the recommended method of virtualization because of its performance benefits. Upcoming chapters provide information on creating both hardware-assisted software virtualization and paravirtualization guests.

Summary

This chapter has introduced you to virtualization and its different types. The information provided is intended to be fairly generic, since Oracle VM specifics will be provided throughout the book. Both software and paravirtualization were described, as well as the benefits and drawbacks of both technologies. Because hardware support for virtualization is changing so rapidly, some of the information in this chapter may be obsolete by the time you read this book; however, the general principles and methods will remain the same.

This chapter compared hardware-assisted software virtualization and paravirtualization. This comparison is based on the technology as it exists today. Soon hardware extensions for virtualization will improve as well as hardware support for paravirtualization. Virtualization is now at the forefront of CPU technology and is leading the new technological revolution.