Realtime publishers

The Definitive Guide™ To

Virtual Platform Management

sponsored by

ca technologies

Anil Desai

Copyright Statement

Chapter 3: Comparing Virtualization Approaches

The ability to choose from among many technologies is a double-edged sword for IT decision makers. On one hand, having many options makes it more likely that one of them will be the best solution for a given scenario. On the other hand, having many methods of solving a particular problem can make the evaluation process more difficult. The arena of virtualization approaches presents exactly this situation.

As with any new technology, terminology is important. When it comes to discussing *virtualization*, the term is often used in a variety of different ways. IT decision makers and upper management might understand virtualization as a general concept—the ability to concurrently run multiple disparate workloads on the same hardware. Their main focus is on the primary benefits, including reduced costs, increased hardware utilization, and improvements to deployment and provisioning times.

Technical users, however, usually need to focus on the details of the solution, and that is where it becomes important to distinguish between the various ways in which virtualization can be implemented. This chapter will look at several approaches to solving common IT problems using virtualization solutions. The focus will be on the benefits and limitations of each approach as well as recommendations about which method makes the most sense for various scenarios.

Evaluating Virtualization Goals

Before diving into the technical details related to various implementations of virtualization technology, let's first take some time to identify the primary goals of virtual machine technology. The overall idea behind installing operating systems (OSs) and applications within a virtual machine rather than a physical one is to provide the ability to run multiple workloads on the same physical computer at the same time. This leads to increased hardware utilization and provides a number of additional, related benefits.

Figure 3.1 offers an example of how very different workloads might be able to coexist on a single physical server. Although this approach often works in a variety of technical situations (such as where a single Web server might provide access to many different Web sites), it is not as easy to do with complex configurations. Specifically, there are some common characteristics that should be considered when evaluating any virtualization implementation. As this chapter later explores, each approach offers a combination of benefits and drawbacks that can have a significant impact on its success.

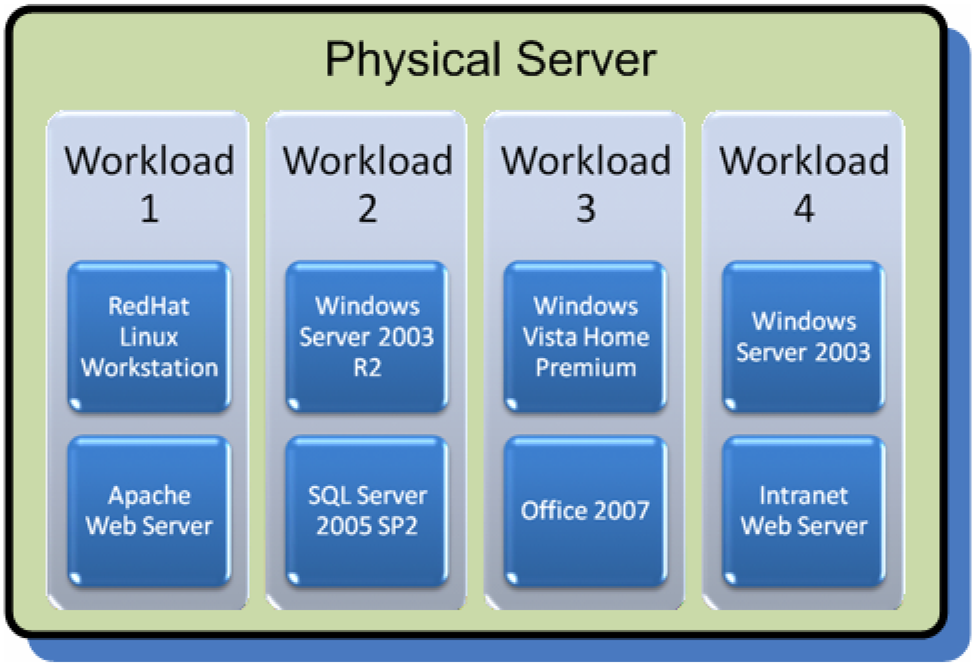

Figure 3.1: Running multiple independent workloads on the same physical computer.

This section will look at some of the ways in which virtualization technology can help enable otherwise incompatible OSs and applications to run concurrently on the same hardware.

Performance and Scalability

When asked about concerns related to virtualization technology, the first issue that many IT managers and systems administrators think of is related to performance. Assuming that the virtual machine approach works properly, response times, throughput, and other details are vital to the success of the implementation. First, let's talk about the bad news: There will always be some amount of performance overhead related to implementing a virtualization layer.

The goal of this solution is to coordinate requests for hardware access among many different independent virtual machines, so the efficiency of the implementation method is important. For example, standard hard disk read-and-write requests from within a virtual machine must traverse several layers before they can be completed. Obviously, this will be slower than directly accessing the hard disk on a physical computer. Each layer could potentially become a bottleneck and could decrease performance.

The task of estimating the overall performance impact of implementing virtualization can be tricky. Unfortunately, there is no single number that will be representative of performance over a variety of workloads. For example, a standard file server might see a 15% decrease in the number of total connections that can be supported, while a low-activity Web server might exhibit a barely-noticeable hit.

This places the burden of testing performance on IT staff. There are several methods by which performance can be measured and evaluated. The ideal testing approach will focus on the experience of the end user. For example, for a Web-based application, the total amount of time it takes to process and complete a transaction can be measured in seconds. IT staff, developers, and end users should provide some method of defining such common transactions along with some guidelines of acceptable times. Automated load- and performance-testing applications and scripts can also be very useful.

In some cases, it can be difficult to perform representative real-world tests that cover all the different uses of an application. For those cases, synthetic workloads can be used to benchmark a system. For example, a database server might have a set of transactions that must be completed in a certain amount of time. As long as the transactions can be run multiple times, this might give a good indication of expected performance in the "real world." Server-level benchmarks can also provide useful feedback. A systems administrator might use a standard memory bandwidth test to compare the performance between a host OS and a guest OS. Figure 3.2 provides an overview of various performance-testing approaches.

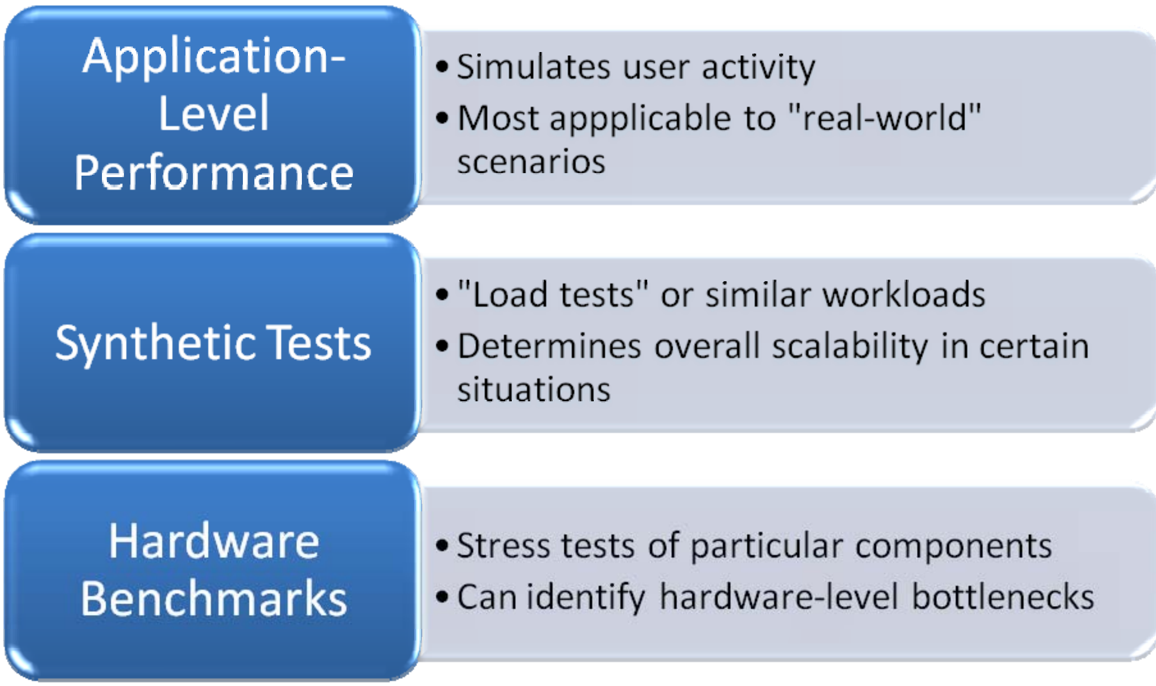

Figure 3.2: Comparing the various methods of testing overall performance.

Regardless of the approach used to evaluate performance, IT staff and end users should find a way to agree upon acceptable levels of performance. Based on these metrics, systems administrators can determine whether a particular virtualization solution is appropriate. The ultimate goal is to maximize scalability—the ability of a solution to support increasing workloads without significantly reducing performance. Ideally, the system will be able to scale in a linear fashion based on the amount of hardware resources that are available.

Reliability and Availability

Entire organizations have become increasingly reliant on their IT infrastructure in order to do business. It's not uncommon for a single server or machine to support dozens or even hundreds of users. Downtime can be costly in terms of productivity. Therefore, when considering the implementation of virtualization solutions, reliability is an important factor. All methods of implementing virtualization involve complex software and hardware interactions. When implemented incorrectly, they can lead to problems that prevent an application or workload from running properly. The end result is downtime or service disruptions.

By their very nature, virtualization solutions present another major risk: many different types of important workloads can be combined onto a single physical computer, so the importance of each physical host machine is increased. It can lead to the IT equivalent of placing many eggs in a single basket. Hardware failures or network connectivity problems that might have resulted in the unavailability of only a single application or service in a non-virtualized environment could affect many systems when running on a single server.

Compatibility, Independence, and Isolation

The basic ideas behind virtualization technology have been available for many years. In the earliest days of computing, centralized architectures were common. For example, mainframe systems usually contained most of their hardware resources within one or a few large machines. These computers were centrally managed and could allow hundreds or even thousands of remote connections. Over time, however, the rise of client-server computing and distributed applications has significantly changed the IT landscape. Now, multi-tier applications with numerous dependencies on other systems are commonplace. Figure 3.3 provides an example of several components that might be required for a standard enterprise application.

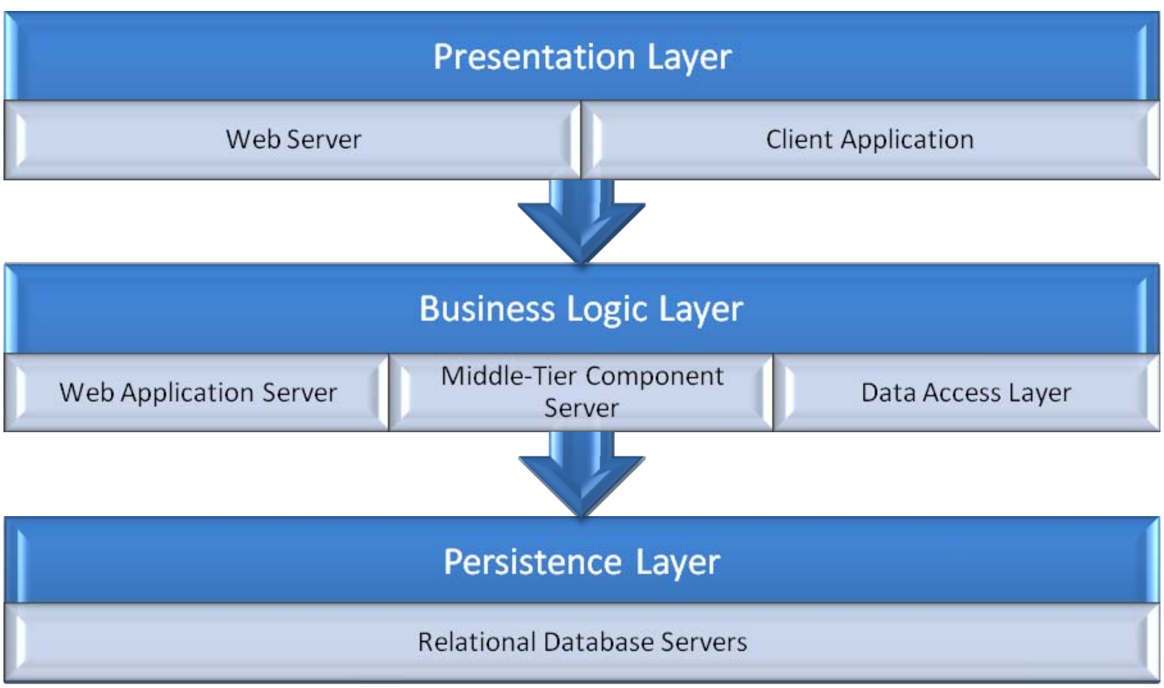

Figure 3.3: Components and dependencies related to a modern multi-tier application.

IT departments have many options when deciding upon application architectures. The different types and versions of client and server OSs alone can present a wide array of potential configurations. In addition, applications have their own requirements such as specific versions of Web server software and OS extensions.

Most OSs, applications, and services were not designed to exist peacefully with differing configurations. Sometimes software types and versions are incompatible (for example, one Web application might not have been tested on Windows 2000 Server while another is not tested on Windows Server 2003). When it comes to the challenge of combining workloads on a single server, all of this must complexity must be taken into account. A primary technical requirement for virtual environments is that of ensuring that each application, service, or workload runs in an isolated environment and independently of other workloads.

When evaluating virtualization solutions, IT departments should keep in mind the types and levels of compatibility that are possible. Support for many types of OSs and workloads are important factors. As we'll see later in this chapter, some virtualization approaches allow for running just about any type of OS and related applications, while others are more limited.

Simplified Administration

For most IT environments, it is common knowledge that costs related to administering and maintaining a technical solution can be far greater than the initial purchase and implementation costs. Virtualization technology is no exception to this rule, as many additional administration factors must be considered. Figure 3.4 presents an example of costs that might be related to the deployment of a new virtualization solution. Typical problems include those that are caused by the simplified deployment process for virtual machines. IT organizations can quickly find themselves managing large numbers of virtual machines that have been deployed throughout the environment. This phenomenon is often referred to as "virtual machine sprawl," and will be covered in greater detail in later chapters.

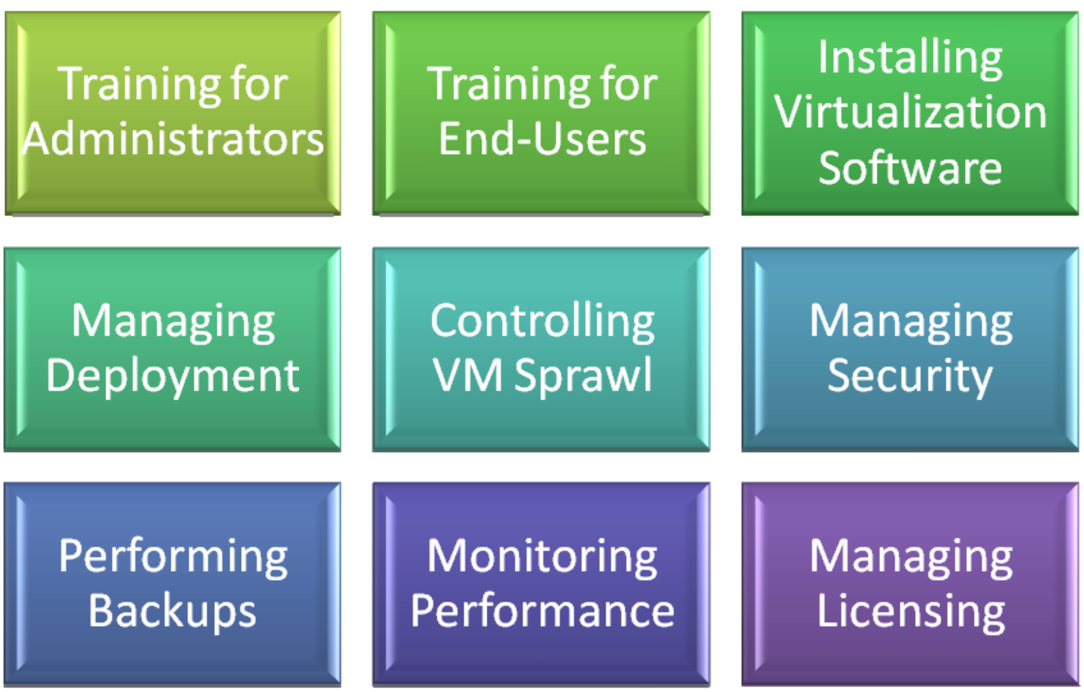

Figure 3.4: Costs related to the implementation of virtualization solutions.

Overview of Virtualization Approaches

With this basic understanding of virtualization goals, it's time to look at different ways in which this technology can be implemented. The first important concept to understand is the layered model of modern IT computing systems. Often, the "base" layer refers to physical hardware and connections themselves. Components such as CPUs, memory modules, hard disks, and network adapters are represented here.

Above this layer sits a set of device drivers—specialized pieces of software that are designed to interact with and manage the hardware itself. Some drivers are implemented in firmware (such as components that are managed through the system BIOS), whereas others run higher in the stack. The OS is the next major component. The OS itself is responsible for providing basic services (such as TCP/IP networking) as well as an environment in which users can interact with applications. Finally, at the top of the stack is the portion that end users are most concerned with—the applications and services that perform useful business functions.

From a conceptual standpoint, virtualization solutions can be implemented at each of these layers (and, in some cases, between the layers). Figure 3.5 provides an overview of various "insertion points" for virtualization technology.

> Later, this chapter will cover specific implementations.

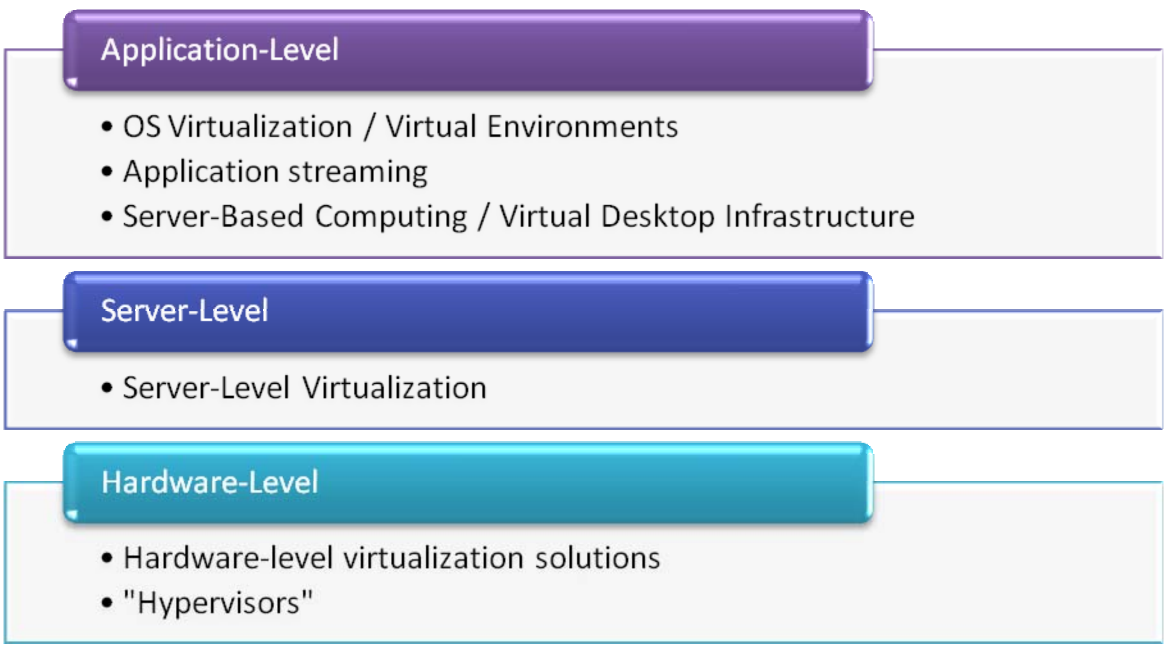

Figure 3.5: An overview of various virtualization approaches.

OS Virtualization

Perhaps the most popular approach to implementing virtualization is that of running a virtualization platform within a host OS. This approach is commonly known as OS virtualization. Figure 3.6 provides an overview of the architecture of this approach. The most important aspect of this figure is that the virtualization layer sits above a primary host OS. It is thereby able to relay and coordinate calls for access to the host hardware through the host OS.

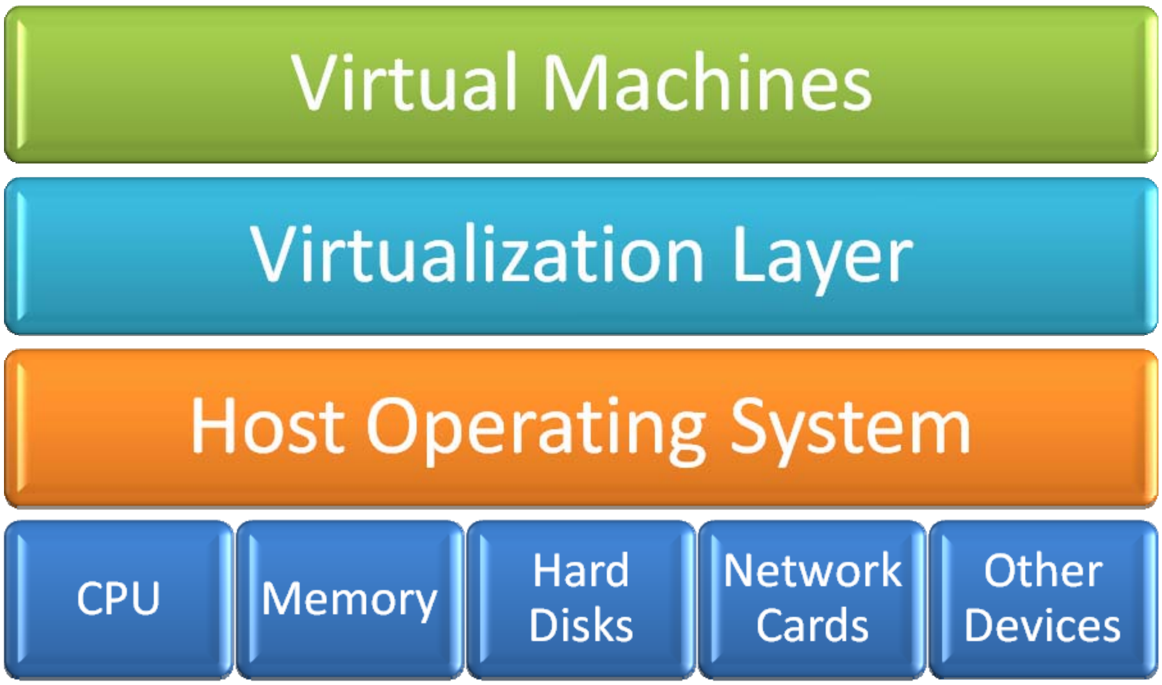

Figure 3.6: An overview of OS virtualization.

The most popular products in the category of OS virtualization include VMware Server, VMware Workstation, Microsoft Virtual Server 2005, and Microsoft Virtual PC. Other solutions are available from Parallels, Inc. and other third-party vendors. All of these products allow for the creation of independent, isolated virtual machine environments. The guest OS within each virtual machine sees the environment as a physical computer. Although details related to the emulated hardware specifications vary between platforms, the general approach is the same. As long as the guest virtual machine environment is supported by particular OSs, applications, and services, any type of workload should work properly within the virtual machine.

Benefits of OS Virtualization

There are many benefits that contribute to the popularity of the OS virtualization approach. First and foremost is the relative ease of getting up and running with this type of solution. Free and low-cost virtualization platforms are available and can be downloaded from their respective vendors. The installation process is usually very straightforward. Client-based virtualization solutions (such as Microsoft Virtual PC 2007 and VMware Workstation) are installed in a way that is similar to other applications. This makes them suitable for end-user use. Server-focused virtualization solutions such as VMware Server and Microsoft Virtual Server 2005 require a little more configuration. Still, the process is usually very quick and easy.

From a technical standpoint, the primary benefit of working with OS virtualization layers is that they can usually take advantage of any hardware that is supported by the host OS. So, for example, if a systems administrator is using a new 10-Gigabit Ethernet network card on the host, the administrator does not have to worry about whether a specific driver is available for the virtualization platform of choice. Although there are some exceptions, in general, as long as a standard driver is available for accessing and managing the hardware on the host, it should be available to guest OSs through hardware emulation.

Another major advantage of OS virtualization is that it provides isolated environments for each virtual machine. Although there are ways to allow network-based access to the host computer or other computers on the network, each virtual machine is independent of the others. This setup allows a single physical computer to concurrently execute requests from a variety of OSs. It also eliminates the potential for interference between different applications and services that are running within virtual machines.

It's important to evaluate the costs and effort related to ongoing management of a virtualization solution. This is another area in which OS virtualization shines: The virtualization layer runs atop a full OS, so it has the ability to take advantage of the host OS's security and administration features (see Figure 3.7).

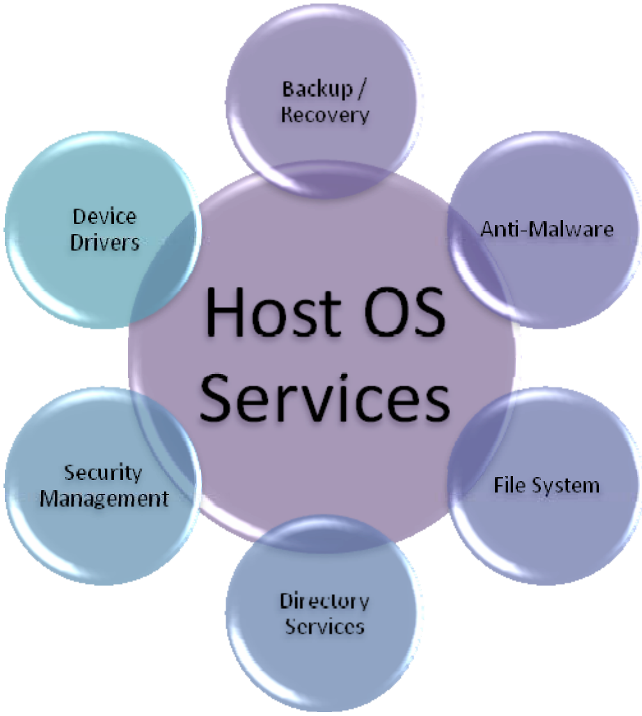

Figure 3.7: Host OS services and features that can be used by virtualization solutions.

For example, systems administrators can leverage Microsoft's Active Directory (AD—or other directory services solutions) to secure and centrally manage access to an installation of Microsoft Virtual Server. Additionally, other features such as backup utilities, malware scanners, and centralized enterprise management software can be installed and executed on the host. Overall, this can make OS virtualization much easier to manage than other approaches.

Drawbacks of OS Virtualization

Although there are many benefits to working with OS virtualization, this approach is not without its drawbacks. Perhaps the most obvious is the need for a host OS. Though the host OS provides many advantages, it also causes issues that can limit scalability and performance. This is due to the fact that the host OS itself must use some amount of CPU, memory, disk, and network resources. If, for example, a host OS requires 512MB of memory to run adequately, this amount should be left as unavailable for use by virtual machines. The overhead added by the host OS might be negligible in some cases (especially if the host system is running other important services), but it can certainly add up when trying to optimize server density.

There is another issue that is related to performance—because hardware-related calls from guest OSs must traverse several layers, the time it takes to perform operations is often increased. Figure 3.8 provides an example of the steps that might be required to complete a disk-read operation. As is the case with all types of virtualization, it's important to test performance before deploying a solution in a production environment.

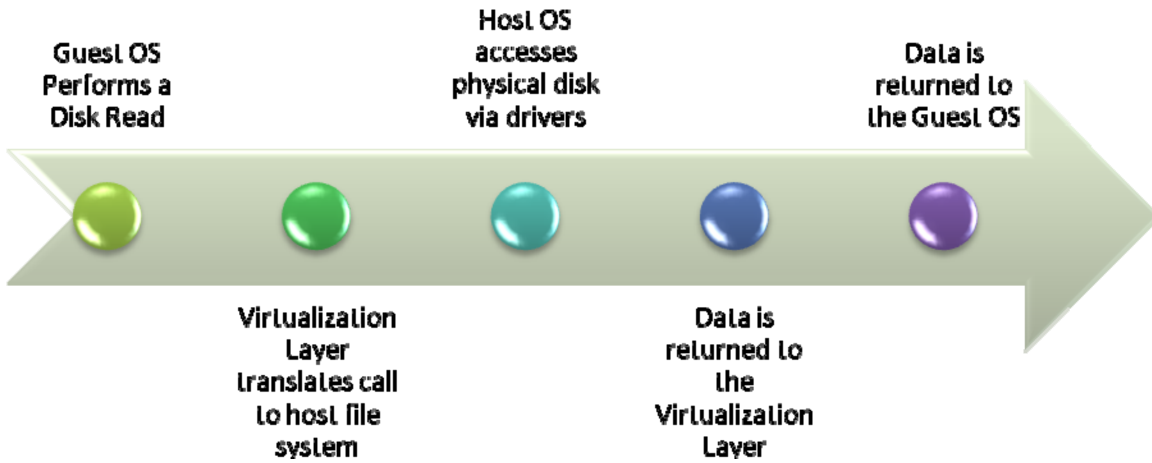

Figure 3.8: Steps required to complete a disk-related call from the guest OS.

Another potential concern for IT departments that are interested in deploying OS virtualization is that of licensing. Generally, an OS license will be required for the host OS as well as for each virtual machine guest OS. This may add to the total cost and must be taken into account when pricing solutions.

Overall, OS virtualization provides many benefits that make it a great "first choice" in the minds of systems administrators and IT decision makers. Based on this information, let's move on to look at some other approaches.

Server-Level Virtualization and Hypervisors

One inescapable "cost of admission" related to working with virtualization is that there will always be some amount of performance overhead. In theory, the simplest and most efficient method for implementing virtualization is to run directly on the host computer's hardware. The idea of server-level virtualization is just that: Rather than requiring a full-fledged host OS, it includes a thinner layer of virtualization management software that handles basic hardware management functions.

The concept is that—because many of the features and functions of the host OS are unnecessary for virtualization—there is no need to waste resources on it. Instead, drivers and system services can be optimized primarily for the purpose of supporting virtual machines. This approach is also referred to as a Hypervisor—a type of system supervisor that sits above the hardware and between the virtual machines that are running on the system. Figure 3.9 provides an architectural overview of this model.

> ✐ The virtualization layer interacts directly with the physical hardware on the host. Compared with OS virtualization, server-level virtualization communications between guest OSs and hardware are more direct.

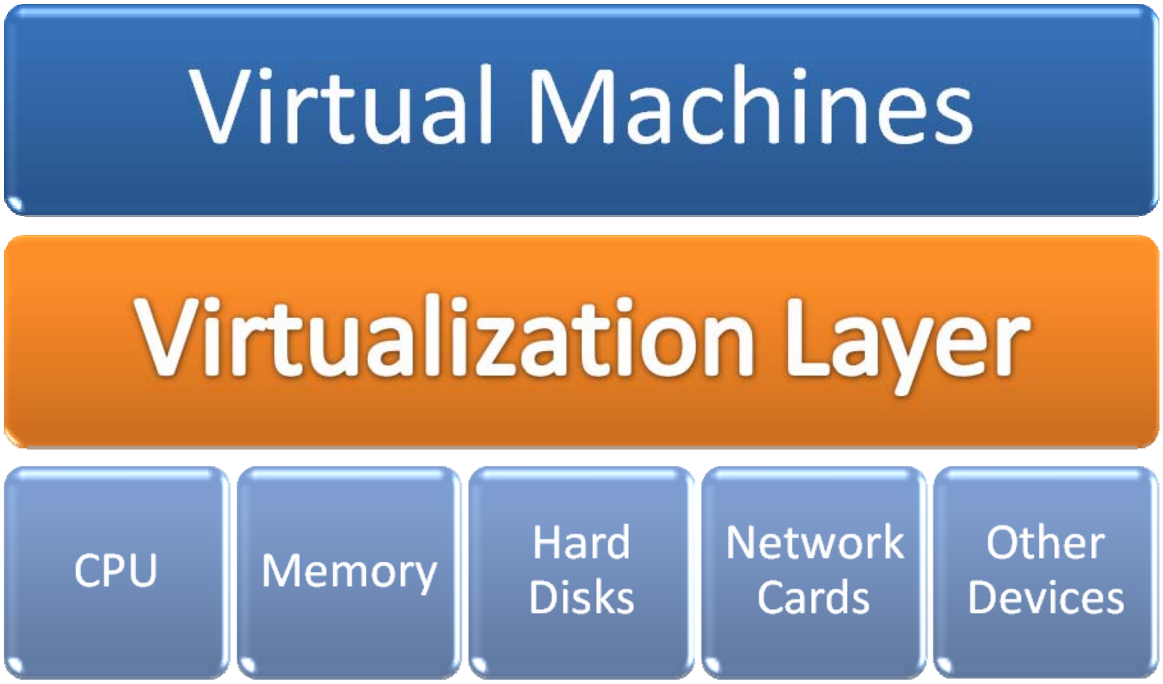

Figure 3.9: An overview of server-level virtualization.

Examples of server-level virtualization include solutions from VMware (ESX Server and its associated Virtual Infrastructure tools and technologies), the XEN open source initiative, and third-party hardware-specific options available from various vendors. Microsoft has announced that it will move to a Hypervisor-based model that will be available within 6 months of the release of its next server OS.

Benefits of Server-Level Virtualization

As already mentioned, the primary benefit of server-level virtualization is a reduction in overhead. This should lead to increased performance through an architecture that is designed specifically to support multiple virtual machines. The result is that organizations can run more virtual machines on the same physical server hardware. This increased capacity can translate to significant savings, especially when tasks such as server consolidation are considered.

Security is a primary concern for all IT environments, but it's especially important in the data center. When supporting mission-critical applications and services, it's important to ensure that the entire virtualization stack is as secure as possible. In the server-level virtualization setup, the role of the host OS is limited, so there are generally fewer "moving parts" that need to be configured and managed. As long as an organization has an in-depth understanding of the needs of a server-level virtualization solution, it should be simple to limit access to virtualization tools and features.

Other important items in the list of requirements for a virtualization solution often involve reliability and availability. First- and third-party products are available to help make managing and maintaining virtual machines with minimal downtime a possibility. Again, by simplifying the virtualization solution, it's theoretically easier for an IT department to avoid common causes of downtime.

Drawbacks of Server-Level Virtualization

Although there are numerous advantages of server-level virtualization, there are also some important drawbacks. Perhaps the most important concern for most environments will be that of compatibility. The virtualization layer is designed to directly communicate with host hardware, so it must have the necessary device drivers and support software to do so. Most server-level virtualization solutions will provide a specific hardware compatibility list (HCL) that denotes which types of devices are supported. Solutions such as VMware ESX Server tend to have a fairly comprehensive list of compatible devices and platforms, but they are focused on typical server-side computers. Environments that want to implement proprietary devices or to migrate to newer hardware must first ensure that the virtualization vendor supports the move.

Another important concern is related to the area of administration and manageability. For most current solutions, managing the virtualization layer can be accomplished through the use of a command-line interface or through a Web-based administration console. For example, VMware ESX Server provides both methods, and systems administrators can connect remotely via an OS shell or through a Web browser. However, it is up to the provider of the virtualization solution to offer access to the types of management features that are expected in modern OSs. In some cases, the majority of commonly performed tasks can be carried out easily. However, more advanced features might not be implemented. For example, tight integration with directory services, support for enterprise backup and management tools, and support for anti-malware products might not be readily available.

Finally, cost-related considerations must be taken into account. At the time of this writing, server-level virtualization solutions tend to be significantly more expensive than OS virtualization products (which are currently available for little or no cost). When combined with the costs of guest OS licensing, organizations might have a difficult time demonstrating a financial benefit in non-mission-critical configurations.

Overall, the pros and cons of server-level virtualization make it best-suited for production environments that demand the utmost in reliability, availability, and performance. These environments should be able to accommodate the additional costs and expertise that will be required to keep them running optimally.

Application-Level Virtualization

So far, this chapter has looked at two approaches to creating isolated, independent virtual machine environments within which a guest OS can run. The primary benefit of these approaches is that they allow for compatibility with a broad range of platforms, applications, and services. The primary drawback is that each environment is essentially a separate physical machine that requires a separate OS. When the resource requirements for these layers are factored into the equation, the end result is a significant amount of overhead. In some cases, the need for multiple guest OSs is justified and necessary. Various types of workloads are incompatible, and organizations might have the need to run different OSs and versions on the same hardware.

In other cases, however, the primary requirement might be much simpler. For example, a single computer might need to run multiple versions of the same client application on the same OS at the same time. Without a virtualization solution, this might not be possible. Applications tend to make changes to registry settings and the file system, so there is a chance that they will step on each others' toes. Additionally, some applications were not designed to be aware of multiple users that might be accessing it at the same time. For example, if one user makes a change to various OS settings or program options, the effects should be limited to his or her own environment.

This is where application-level virtualization comes in. Although there are several products and technologies that utilize the application-level approach, the solutions are primarily designed to sit between the host OS and applications. Figure 3.10 provides an overview of this approach.

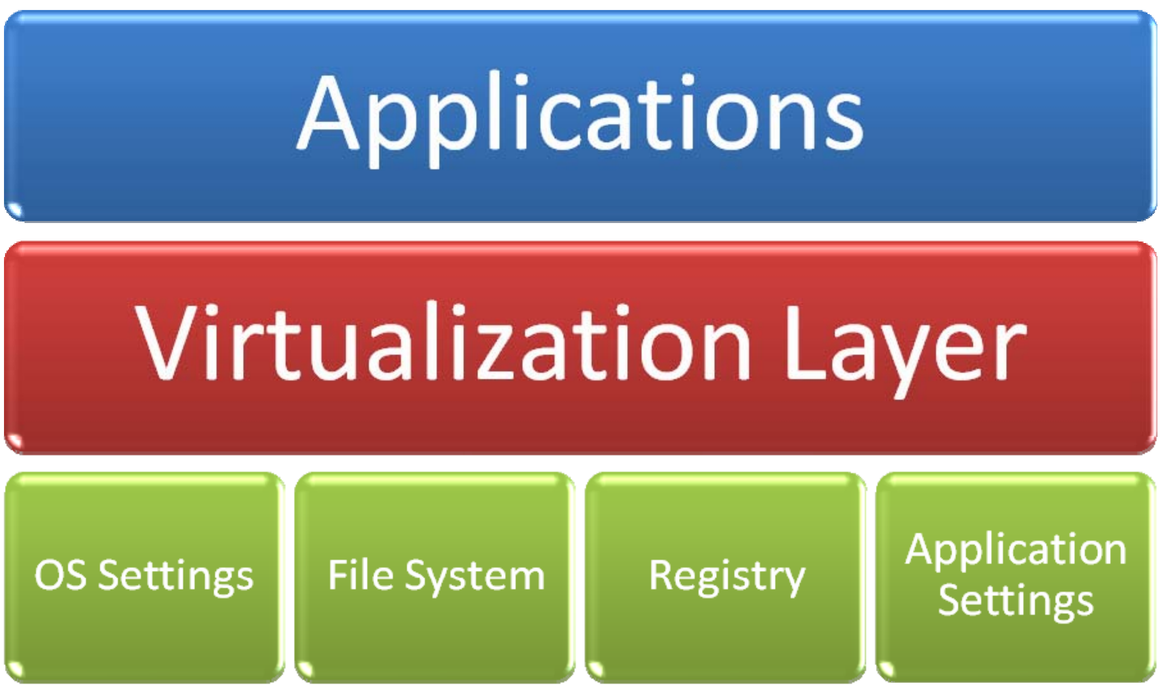

Figure 3.10: An overview of application-level virtualization.

At the base level of this stack are the actual components of the host OS that are regularly modified when applications are installed and used (not shown are the lower layers of the host OS and the actual physical hardware itself). Above this is an application-level virtualization solution that is responsible for coordinating access to these resources. Finally, at the top of the diagram are the applications themselves. The goal is for each application to believe that it is running as expected—with its own access to OS and file system settings—but in a way that allows multiple instances of the application to run at the same time. With this basic understanding of this approach, let's look at some particulars.

Isolated Virtual Environments

The approach described in the previous section involves the creation of compartments or areas within a standard OS within which applications can run. Although it doesn't allow multiple OSs to run concurrently, it does allow for applications to have their own access to resources that are typically not shared.

SWSoft's Virtuozzo product is designed to created isolated application environments that can be run on the same computer at the same time. It runs above a compatible host OS and handles the logical partitioning of the system into independent instances. Each instance can host applications and services, and users can be given access to one or more partitions. Additional features provide for the ability to change the allocation of resources dynamically and tools for monitoring each environment as it is running. This approach can be used to meet many of the same goals as server-level and OS virtualization solutions without the added overhead of running many different OSs.

Application Streaming

One of the primary headaches for IT departments is that of installing, distributing, and configuring new applications on client computers. The process can be tedious and error-prone, even when it is largely automated. Managing software updates and changes can be time-consuming and might require end-user involvement. Add in challenges related to security and distributed management, and it can quickly become a major source of cost and frustration. The end result is a less-than-ideal experience for both users and systems administrators.

The overall approach with application streaming is to provide users with the applications that they need in an on-demand way. Figure 3.11 provides a high-level overview. Rather than installing several (or sometimes dozens) of applications on a system just because a user might have an occasional use for them, applications can be quickly deployed and configured on a target system from a central location when they are actually called. To speed up the deployment time, portions of the application code are made available as they are requested. Best of all, when the applications are no longer needed, they can be automatically removed.

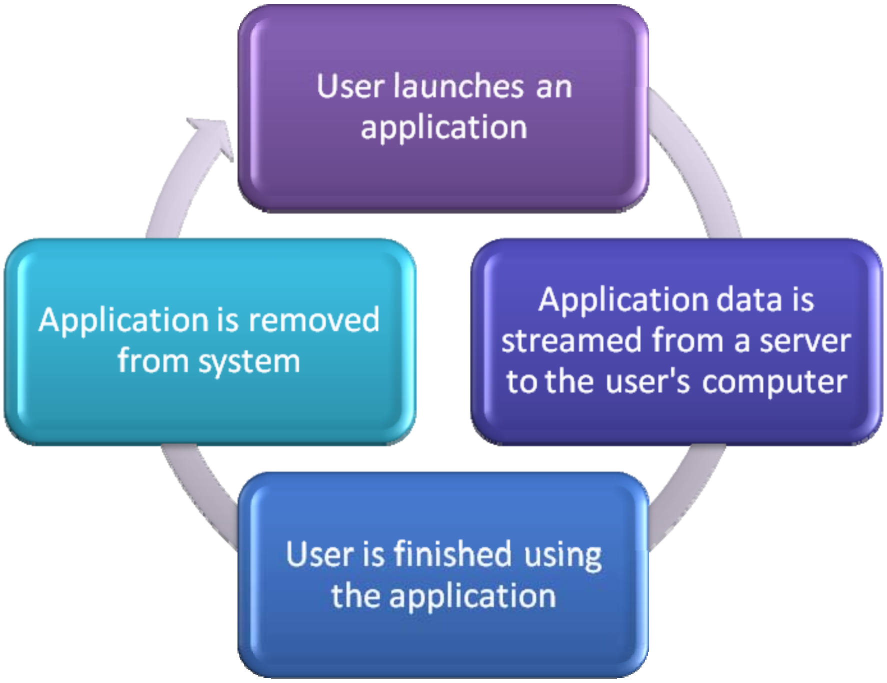

Figure 3.11: Steps in the application-streaming process.

Microsoft's SoftGrid Application Virtualization technology (formerly known as Softricity) is an example of this approach. It provides application streaming functionality along with management infrastructure for keeping track of applications. There are several application requirements necessary to support this approach, but tools are available for making standard applications streaming-compatible.

Server-Based Computing

Although the technology has been referred to by a variety of names over the years, the concept of server-based computing has remained largely consistent. The primary approach, as Figure 3.12 shows, is that client computers access all their resources using a very simple layer of software. The client software is designed primarily to relay keyboard, video, and mouse input/output commands over a network connection. The architecture provides for centralized computing—the actual execution of application code and services is being performed on the central server, and the results are sent as video commands to the end-user's computer.

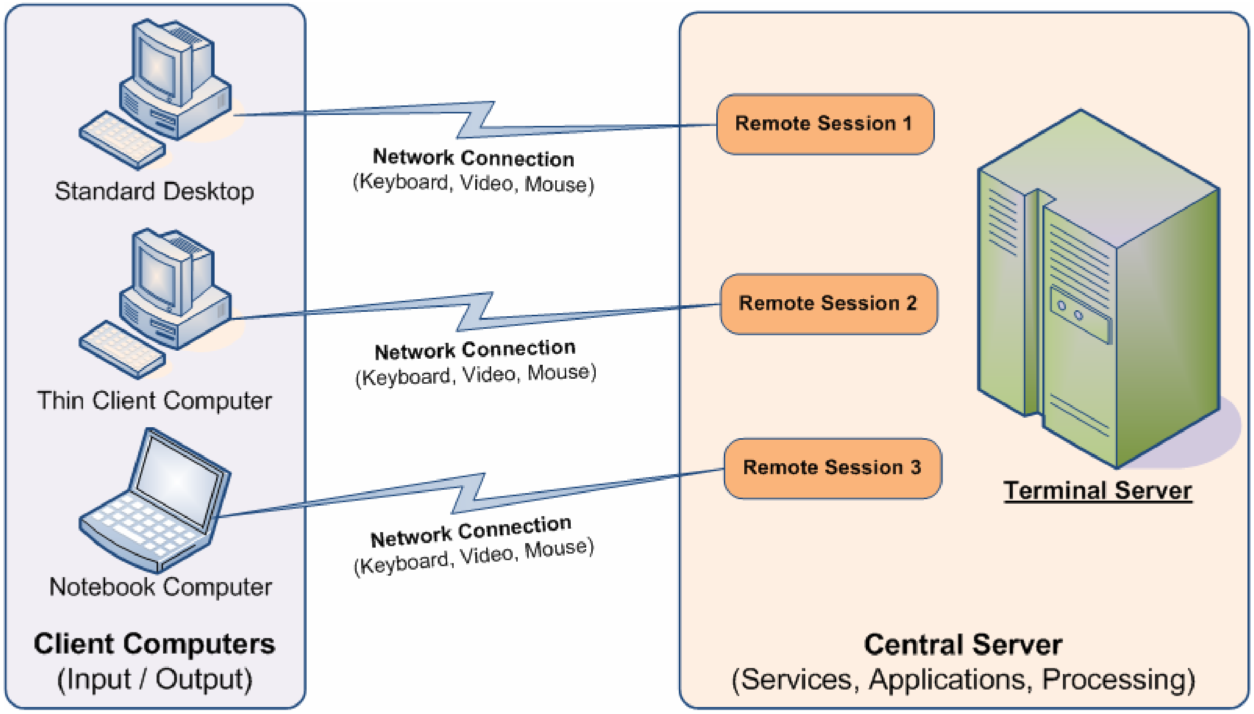

Figure 3.12: The architecture of server-based computing solutions.

Two examples are Microsoft Terminal Services (available on the Windows Server platform) and Citrix Presentation Server.

Virtual Desktop Infrastructure

Time and effort related to managing desktop computers can account for a large portion of overall IT costs. Many organizations have the need to deploy, support, and manage hundreds to thousands of computers. In addition to the standard maintenance and administration tasks, consideration must be made for systems that are located in various regions of the world. Some remote locations might lack the resources and expertise to properly manage these systems. When details such as those related to maintaining security and consistent configuring are added, the process can become very challenging.

The concept of Virtual Desktop Infrastructure (VDI) is based on the idea of configuring end users' computers to connect to virtual machines over a network instead of having their applications and services installed locally. Although this approach is similar to that of the other methods discussed, it's a specific application of virtualization technology. Rather than focusing on consolidating servers or other types of workloads, the idea is that the entire desktop computing infrastructure can be largely replaced. There are numerous obvious advantages, including centralized administration and the ability to reduce hardware requirements on desktop and workstation computers.

The VDI approach is not without significant limitations, however. The first issue is a strong dependence on the network. If a network connection is unavailable (or unreliable), VDI solutions will not work. In addition, there will be certain types of applications and workloads that are not very "virtualization-friendly." Resource-intensive applications or those that require specific types of hardware (such as third-party peripherals or high-end graphics acceleration) are not good candidates for this approach. Despite these drawbacks, organizations might find it to be a compelling technology for some portion of their user base.

Evaluating Application Virtualization

Overall, application-level virtualization solutions offer very useful advantages. They tend to be more scalable than virtual machine-based approaches, and this can lead to running hundreds of instances of an application on a single physical server. They still provide many important features such as centralized management and easy deployment.

The main potential drawbacks with application-level virtualization are that of compatibility and hardware support. All applications will actually be running on a single physical system, so it may be impossible to accommodate some types of differences. For example, if one application requires Windows XP SP1 and another requires SP2 or later, they will likely be unable to coexist on the same system. Overall, though, application-level virtualization can greatly reduce costs and improve scalability for many common business applications.

Physical Virtualization and Clustering

Despite all the many benefits that are associated with using virtualization technology, it's not always the best solution for every type of workload. In some cases, hardware utilization might already be too high, and other priorities might take precedence. Although the term virtualization commonly refers to the ability to run multiple workloads on a single physical computer, the same concepts can be applied to the opposite—running a single workload spread across multiple physical computers. This approach is often referred to as clustering or load balancing. The technical details involve using multiple physical computers that appear to function as a single logical one. Figure 3.13 shows how multiple servers can work together but still allow users the simplicity of accessing the cluster using a single name.

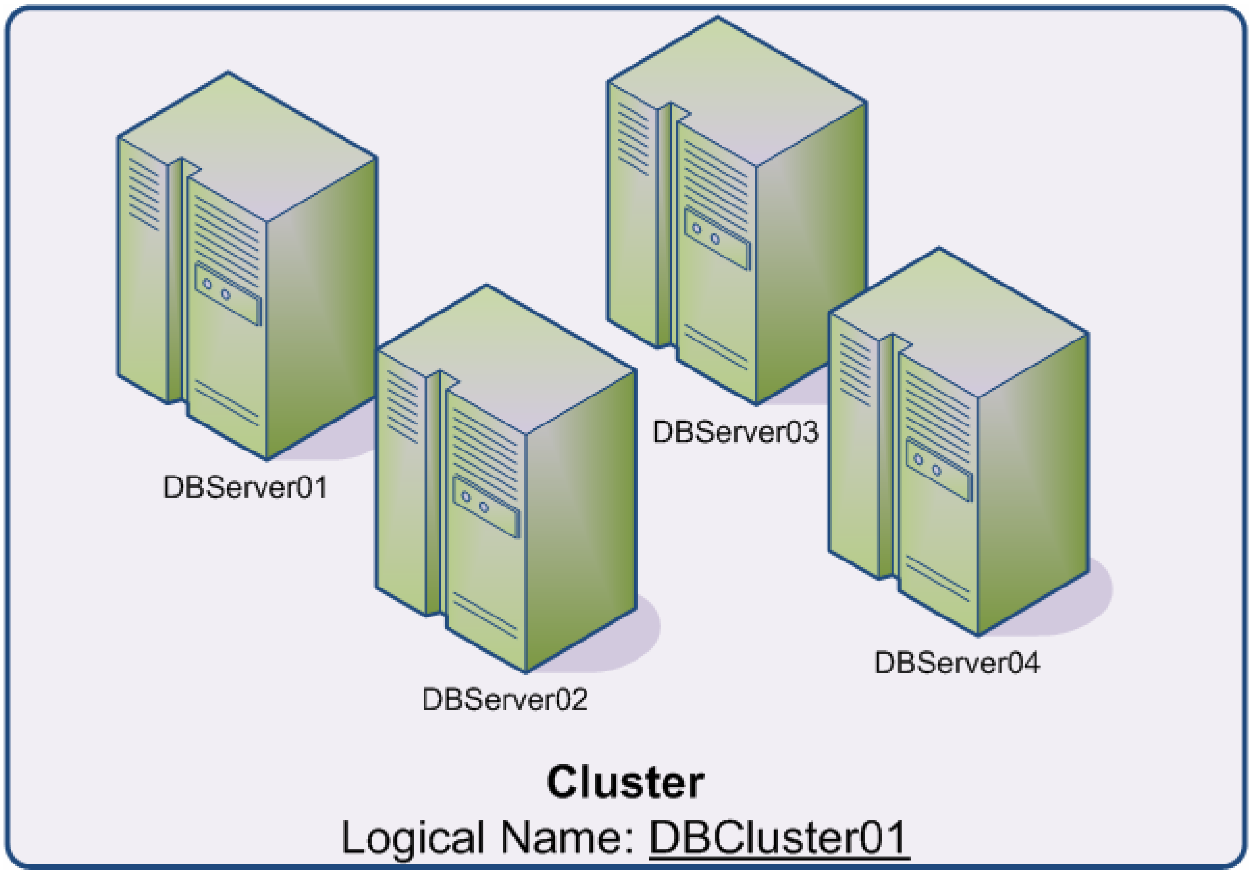

Figure 3.13: A logical and physical overview of clustering.

Clustering and virtualization technology can be implemented at many levels and by using a variety of technology types. The goal is generally to increase scalability, performance, and reliability. This section will look at several examples.

Scaling Out vs. Scaling Up

Although it's probably stating the obvious, even the most powerful physical computers have their limits. In some cases, the limitations are technical. For example, a particular system might support as many as eight physical CPUs. In other cases, scalability might be limited by business issues such as budget constraints. Regardless of the reasons, scalability can be a difficult problem to address. There are two main approaches to this challenge.

The first is known as *scaling up;* this term refers to improving the hardware configuration of a specific machine or group of machines. For example, if it is found that slow disk I/O is causing a bottleneck in serving up Web pages, increasing the amount of memory (to reduce paging) or investing in a faster disk subsystem might increase overall throughput and concurrency. Although this approach can certainly help performance, there are limitations. For example, all types of server hardware have limitations related to their physical hardware configurations. A cap on the number of physical disk drives, types and number of CPUs, and amount of memory can prevent scaling up. Additionally, external factors—such as the types of activities performed on a single server—might cause bottlenecks. If a single database server is used for both reporting and online transaction processing (OLTP) workloads, the two types of activity can reduce overall performance for all users.

The second approach to increasing performance is known as *scaling out*. In this method, additional computers are added to distribute overall workload. The primary requirement is that the type of application or service that is running must be compatible with this architecture. In the case of a Web site with static content, it's easy to implement. More complex applications must take into account caching, locking of records before update, and other details related to managing contention for resources and information. Assuming that a particular workload supports it, however, the scale-out approach can provide nearly limitless scalability. Figure 3.14 provides a comparison of the two approaches.

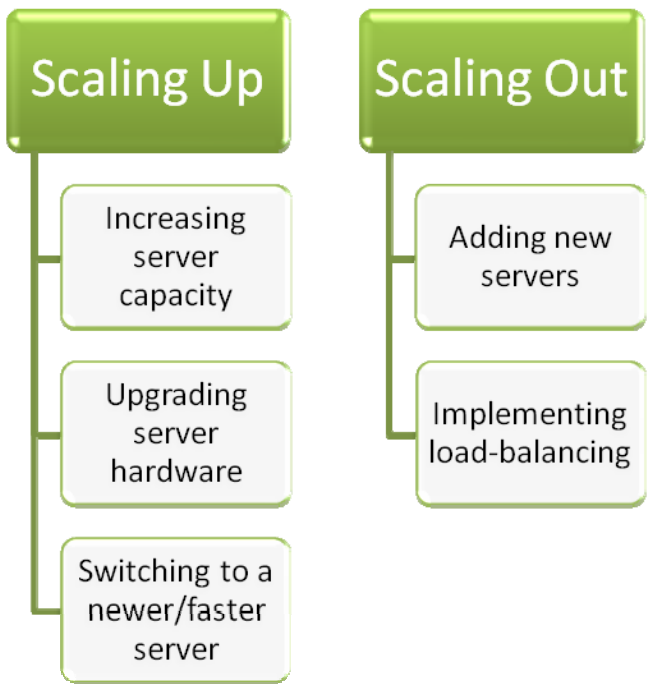

Figure 3.14: Comparing scale-up and scale-out approaches.

Server Clustering

A common goal for IT departments is to ensure the simplest and most efficient experience for end users. Often, the underlying complexity of details such as TCP/IP addresses and physical server names are abstracted so that users need know only a few pieces of information to access the systems they need. By implementing server clustering, the same can be done for groups of physical computers. When a user attempts to connect to the cluster, the user will use the name of the cluster rather than the name of any individual computer. The request itself, however, might be routed to any one of the systems in the cluster.

There are numerous benefits to this configuration. The first is related to performance: through the use of intelligent load-balancing solutions, requests and transactions can be routed for maximum efficiency. In some cases, a round-robin scheme might be used. In others, a load-balancing solution can automatically examine the current load on each node and route requests to the least-busy servers. Like virtualization, this can greatly help in the area of increasing hardware utilization. Other benefits include increased availability. Generally, one or several nodes within the server cluster can be taken offline without affecting uptime. Although it's possible that overall performance will be reduced, users will still be able to access the services they need. Finally, for security reasons, network-level implementations such as a firewall or packet filtering can ensure that no node is accessed directly.

The primary drawbacks related to physical server clustering are cost and administrative complexity. Depending on the specific implementation, setting up a new cluster requires expertise. Some types of applications, such as relational database servers, must be cluster-aware in order to work properly. Hardware and software licenses for clustering solutions can also be expensive. Clearly, these products are not a solution for every type of workload in a data center. However, when uptime and performance are critical requirements, clustering can be a great asset.

Network-Layer Clustering

When accessing devices over a network, end users are rarely concerned with the underlying physical implementation of the servers. Often, they're aware only of the content. Systems administrators, however, might have the need to support hundreds or thousands of concurrent user session on a single Web site. Especially for publicly accessible sites, maintaining performance and availability is an important concern. Even brief outages can be very noticeable. As with server clustering, devices can be clustered at the network layer. This is often done through the use of features in devices such as routers, firewalls, and switches.

Specific network-layer solutions include network-level clusters, load balancing, and various network addressing schemes (such as round-robin DNS). Regardless of the overall approach, the goal is to distribute load across multiple devices. In the case of a Web server farm, a front-end router or other device can be used to dynamically allocate traffic to particular Web server nodes. Additionally, features such as content caching can help reduce the load on popular servers.

As with server clustering, adding network-layer clustering requires a capital investment and understanding of the technology. Overall, however, the benefits provided by these techniques can quickly pay for themselves in added performance and reliability.

Selecting a Virtualization Approach

So far, we've looked at many ways in which virtualization technology can be implemented. We also looked at specific benefits and drawbacks to each approach, along with some suggestions for where each might be a good fit within an overall IT architecture. This, of course, raises questions related to which approach is best. The definitive answer is "it depends." The bottom line is that each virtualization approach has strengths that make it ideal for some situations and weaknesses that prevent it from working in others. Based on the technical details of each approach, let's take a look at ways in which IT departments can evaluate workloads to determine the best approach for a given situation.

Server Consolidation

One of the most compelling reasons to include virtualization within the types of technologies supported by an IT department is that of reducing the number of physical servers that are supported while allowing for an increased workload. The best way to achieve this goal is to combine many types of applications and services on a single physical machine to increase overall hardware utilization. For example, three physical servers, each of which is approximately 25% utilized, can be combined onto a single computer. Assuming that virtualization is the right solution, the challenge then becomes determining which virtualization approach is best.

The evaluation should begin by taking an inventory of the types of OSs, applications, and services that need to be consolidated. Based on this information, IT staff can attempt to group similar applications. In some cases, it might be found that many of the different types of workloads are "compatible." For example, if several Web sites are running on Apache servers, these workloads can likely be combined without using virtualization technology at all. Similarly, two databases that are running on Microsoft SQL Server 2005 might be consolidated onto a single database server without using virtualization.

It's more common, however, to see disparate systems that may require custom environments. For those cases, either OS or server-level virtualization is likely the best candidate.

Software Development/Testing

Software developers often have the need to set up complex environments that allow for creating and testing applications. The types of applications can vary greatly. On one end of the spectrum is a single executable that might be designed to work entirely on a single desktop computer. In this case, the code often must be tested in a variety of configurations to ensure compatibility. For example, QA staff might need to ensure compatibility with Windows 2000 Professional, Windows XP, and Windows Vista. Additionally, various combinations of service packs and security updates might increase the number of systems required.

On the other end of the spectrum are multi-tier, distributed applications and services. In these scenarios, developers and testers must set up environments of multiple virtual machines. Components such as Web servers, business logic tiers, and relational database servers must all work together for the product to be successful. These configurations and requirements tend to lend themselves to OS virtualization. Although server-level virtualization is also a possibility, developers and testers will usually prioritize ease of installation and deployment over reliability and availability.

End-User Productivity Applications

Many line-of-business applications and application suites such as Microsoft Office are among the most-used portions of an IT infrastructure. From a technical standpoint, these applications are designed to work on a fairly wide array of client OSs. That makes the idea for running multiple OSs to support them somewhat excessive. In many cases, however, they have not been designed to support multiple instances of the application running concurrently and independently. This type of workload is ideal for application-level virtualization, unless there is a specific technical requirement that prevents it.

Workload Solutions Summary

To summarize the various scenarios examined so far, Table 3.1 shows a list of common workloads along with some general recommendations.

Workload	Requirements/Characteristics	Virtualization Recommendation
Data Center Server Consolidation	• Performance is a key factor • Scalability is important • Server applications are typically complex and have many requirements	• Server-Level • OS Virtualization
Software Development and Testing	• Manageability is a key requirement • Users must be able to change hardware settings and install OS updates • The ability to roll back system state will be helpful	• Server-Level
Sharing End-User Productivity Applications	• Scalability is important • Applications are less complex	• Application-level virtualization
High-Performance Web Server Farm	• Support for many low-end servers • Fault-tolerance/high availability	• Server clustering • Network-layer clustering

Table 3.1: Comparing virtualization approaches for various workload types.

Managing Heterogeneous Environments

The beginning of this chapter looked at an important goal for any type of IT solution: manageability. An key issue that is raised when supporting many types of virtualization approaches is that of systems administration. In addition to keeping tracking of physical servers, IT departments are tasked with tracking a variety of virtualized solutions. When managed manually, this task can require large teams of experts to ensure that systems are running properly.

Fortunately, through the use of virtualization-aware enterprise management tools, this challenge can be overcome. The key feature to consider is one that allows for all these types of assets—physical and virtual machines—to be managed using a similar model. Indeed, many cases might come up in which physical machines need to be converted to virtual ones (or vice versa). And, the basic goal of all these implementations—to provide services and applications to end users—is the same.

From the standpoint of virtualization, the details and complexity of the actual technical implementation should be abstracted by the management tools. IT decision makers should look for solutions that are able to provide a single, consistent administration model that will allow their departments to manage both physical and virtual assets easily and seamlessly regardless of platform or vendor.

> This topic will be revisited in more depth in later chapters.

Summary

This chapter began by examining the important goals of virtualization technology. It’s crucial that solutions are able to meet requirements related to performance, scalability, reliability, availability, compatibility, and administration. The chapter then presented details about several approaches to meeting these goals and a list of some of the pros and cons related to each. The first method was OS virtualization—the implementation of a virtualization software layer within a host OS. Next was server-level virtualization—a method that minimizes the host OS layer and puts the virtualization solution in charge of hardware.

Another approach that doesn’t directly involve creating virtual machines is a collection of technologies known as application-level virtualization. By providing isolated OS environments and access to applications over a network, these methods can provide for significant scalability. Finally, we explored the topic of physical virtualization. This involves using techniques such as load balancing and automatic failover to implement logical clusters of machines.

Based on all of this information, the chapter then provided examples of factors to keep in mind when determining the best technical solution for a particular type of workload. Finally, it covered points to keep in mind when looking for solutions to manage a heterogeneous virtualization environment (a topic that future chapters will expand on).

The overall message to keep in mind is that there are numerous ways in which virtualization goals can be met. Although OS virtualization is arguably the most popular concept, other approaches may be more appropriate in particular situations. IT decision makers should keep in mind that they have a variety of ways in which to utilize virtualization, and the real challenge is determining which approach is the best for each type of workload.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.