Realtime publishers

"Leading the Conversation"

The Shortcut Guide™ To

Selecting the Right Virtualization Solution

sponsored by

Parallels™

Greg Shields

Chapter 2: Virtualization Software Options 21
Rules of Engagement 21
VMware ESX and Virtual Infrastructure 23
What It Does 23
What It Does Not Do 24
Cost Model 25
Where It Excels 26
Where It Lacks 26
Impact on Server Candidacy 27
Microsoft Virtual Server 27
What It Does 28
What It Does Not Do 28
Cost Model 29
Where It Excels 29
Where It Lacks 29
Impact on Server Candidacy 29
Xen/Citrix XenSource 30
What It Does 30
What It Does Not Do 31
Cost Model 32
Where It Excels 32
Where It Lacks 32
Impact on Server Candidacy 32
Parallels Virtuozzo Containers 33
What It Does 34
What It Does Not Do 35
Cost Model 35
Where It Excels 35
Where It Lacks 36
Impact on Server Candidacy 36
Solution Combinations 37
Finding the Right Solution Is Critical to Implementation Success 37

Copyright Statement

Chapter 2: Virtualization Software Options

Wading through the product-centric hype surrounding virtualization is critical to finding the right solution. The idea of virtualization has received so much press that its meaning in the marketplace has morphed from merely a concept towards specific product descriptions. With so many products on the market that advertise virtualization as a feature, it is important to demystify the solutions that are valuable in your environment in comparison with those that will only complicate your operations.

Chapter 1 talked about how virtualization at its core is not so much a specific product focus. From a conceptual level, it describes the addition of a layer of abstraction between two disparate elements, such as the layers within a computer system. Virtualization's layer of abstraction adds transience, allowing elements above the layer to move around in order to gain an operational advantage. It also reduces organizational reliance on elements that are complicated to manage. Specific to our discussion, virtualization's "shim" layer frees us from the limitations of an operating system (OS) that is installed directly onto physical hardware.

The market's confusion about virtualization is centered on where exactly that shim layer should lie within the operational stack. Depending on the focus of today's virtualization solutions, that shim layer may reside directly on top of the physical hardware. It may reside on top of a thin OS, or even a full one. In some cases, it may be an augmentation of the OS itself.

For example, positioning the shim layer high in the stack on top of an existing OS may make software management easy but reduces its overall performance. Positioning it directly on top of the physical hardware enhances raw performance in some ways but at the expense of driver emulation overhead. That shim layer's location plays heavily into its associated product's features and limitations. As we go through this chapter, you'll find that the capabilities and feature sets of each reviewed product are heavily driven by this concept.

Rules of Engagement

This chapter is very different from the other chapters that make up this guide. Focusing heavily on four specific products, we'll take a hard look at the capabilities and feature sets of each in comparison with the others. The intent here is to provide you a punch list discussing each product, what it does, and what it does not do. We will discuss each product's costing model, where it excels, and in what areas it may be lacking. For each, we'll review that product's architecture and the relation of its architecture to the feasibility and performance of the virtual machines that ultimately reside on top. In the end, the candidacy of your existing systems is what will inevitably drive you towards a particular virtualization solution. The solution you choose will depend on the combination of all these factors.

In a guide like this, there is a danger in speaking too specifically about individual product features and costing figures. Products change. Prices change. Thus, the more specificity we provide here results in aging this guide faster than ultimately necessary. Thus, in every case, the intent with this chapter will be to prioritize as much as possible on total product architecture and defocus individual capabilities and point features.

Many people have already developed their opinion about individual virtualization solutions. These opinions are shaped by market forces often as much as product capabilities. As another of our Rules of Engagement, the intent here is not to necessarily politicize individual products one over another.

The intent here is to illustrate that different products are specifically designed to solve specific problems. One product may solve the problem of multi-platform supportability while another may be best at raw performance and systems consolidation. Knowing what you are attempting to solve within your own network will help you understand what product assists in that solution.

Above all, the intent of this chapter is to wade through the marketing hype and help drive you towards the *right virtualization solution*.

We will discuss four products in this chapter. These four were chosen due to their current market share, level of interest within the market today, and contrasting feature sets. They are also representative of the different types of system virtualization architectures available on the market today: *Hardware Virtualization*, *Paravirtualization*, and *OS Virtualization*. These products are capable of driving virtualization in environments both big and small, so enterprise deployments as well as small-scale operations can consider these as valid candidates for their needs.

The four products chosen for this comparison are:

- **VMware ESX and Virtual Infrastructure—**This product is the combination of ESX, which is the virtualization component, and Virtual Infrastructure, which is the management component of VMware's top-end virtualization solution.
- **Microsoft Virtual Server—**Microsoft entered the virtualization arena with its own Virtual Server product that includes a comfortable interface and compelling pricing model. Its ownership by Microsoft natively grandfathers familiarity within certain communities.
- **Xen—**Our only open source candidate as well as our only Paravirtualization candidate, Xen combines high-end performance with Linux-focused management tools. Though multiple distributions of Xen are currently available, our conversation will focus on the Citrix XenSource distribution.
- **Parallels Virtuozzo Containers—**Our OS Virtualization candidate, Virtuozzo Containers' architecture and near-native performance can solve problems not handled well by other platforms. Its architecture works well in scenarios where multiple similar systems require aggregation.

We'll begin our comparison with a look at VMware ESX and Virtual Infrastructure.

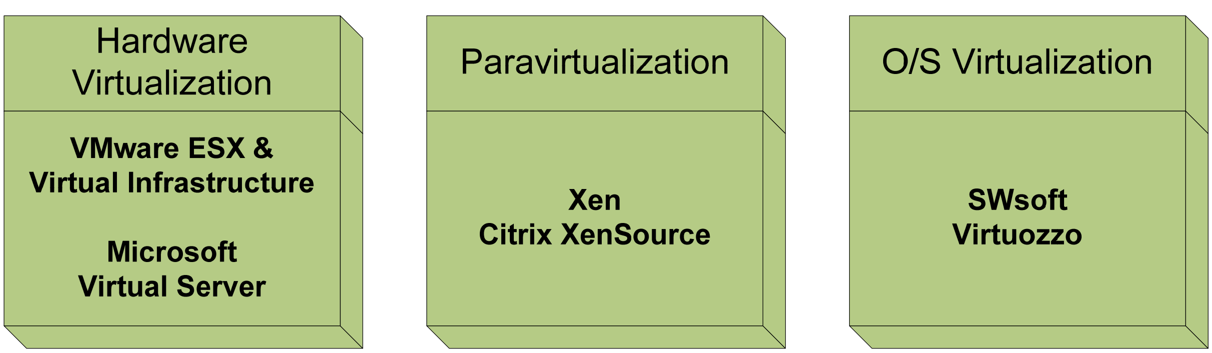

Figure 2.1: The categorization of our virtualization solutions in terms of their architecture.

VMware ESX and Virtual Infrastructure

VMware ESX is VMware's high-end, enterprise-worthy solution for native systems virtualization. ESX encapsulates the virtualization system, the hypervisor code, as well as the OS on which its hypervisor partially relies on for management. Virtual Infrastructure describes the management components that wrap around ESX to enable multi-host management, policy-based configuration, administrator control, and directory-based permission and rights management.

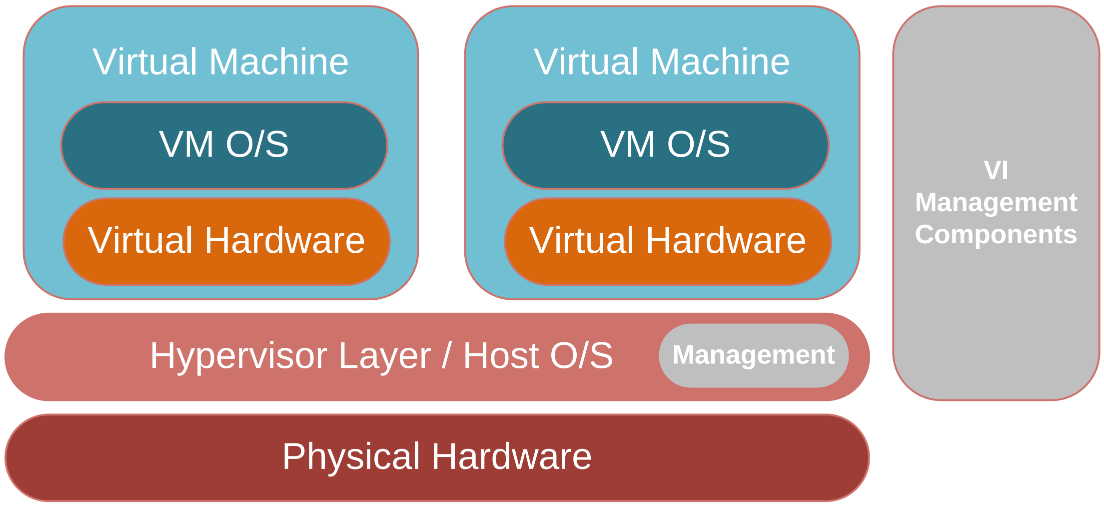

Figure 2.2: VMware ESX hybridizes the hypervisor with an onboard host OS. Atop this layer reside individual virtual machines. The Virtual Infrastructure components manage both the hybrid layer and components of the virtual machines themselves.

What It Does

Taking a look at Figure 2.2, ESX incorporates a type of "hybrid" hypervisor layer that consolidates the hypervisor itself with an onboard host OS running directly on top of the physical hardware. ESX's hypervisor relies partially—though not completely—on the host OS for functionality. On top of this hybrid hypervisor are the individual virtual machines that run productive workloads. ESX's onboard host OS is a Linux derivative, yet is one substantially slimmed down to provide just the functionality required by ESX.

Individual virtual machines that reside within an ESX-based virtualization environment are installed and managed nearly the same as physical machines installed directly to hardware. The ESX virtualization environment supports a set of emulated drivers that enables the host to support a standard virtual machine build with no necessary customizations. This enables new computers to be brought online without requiring customized OS code or specialized training and installation procedures.

ESX's virtualization environment does not provide direct access to physical resources for virtual machines, instead proxying them through a set of emulated hardware drivers. These drivers are installed to all residing virtual machines after installation to obtain best performance and reliability. VMware provides a standardized set of hardware drivers that are represented the same on all ESX platforms. As all drivers are functionally equal across all residing virtual machines, they have the effect of making all virtual machines representative of the exact same physical hardware.

Operating as an optional—though strongly suggested—management component is the Virtual Infrastructure layer shown in Figure 2.2. This external management interface ties into components native to the onboard OS that manage the virtual host and provide access and manageability for residing virtual machines. Onboard management components can be directly used in the absence of the Virtual Infrastructure components. However, use of the Virtual Infrastructure components is required when use of enterprise-level features such as VMotion is desired and when permissioning and rights management must be tied to an external directory such as Microsoft Active Directory (AD).

What It Does Not Do

The virtualization shim layer within VMware ESX lies directly atop the physical hardware. Thus, ESX's proxying of hardware resource requests to virtual machines occurs on a per-virtual machine basis. Thus, resource sharing between individual virtual machines is not as optimized as in other virtualization architectures.

If one virtual machine requires the use of a processor resource, the hypervisor schedules the use of that resource exclusive to the requesting virtual machine for its configured time period. A similar activity is done with virtual memory requirements on the physical host. Virtualized sharing of resources is done via rapid swapping of ownership.

Where the architecture of ESX fails to fully optimize the sharing of resources is within the stovepiped nature of resource allocation itself. Implementing 10 virtual machines onto a single physical host, each of which requires 2GB of RAM, will require a total of 20GB of physical RAM on the host to support virtual machine needs. The swapping effect enabled by the hypervisor layer on the host reduces this total requirement by a percentage. However, that percentage is variable based on the actual RAM usage of the individual virtual machines. This means that some sharing of physical RAM can be done by individual virtual machines. But if those virtual machines are actively and regularly using RAM, their sharing will incur performance degradation.

Disk requirements for ESX-based virtualization environments can be substantial. Because each virtual machine is in and of itself an atomic entity, its files are not shared by other systems on the host. Thus, if you have 10 virtual machines, all of which require 50 GB of disk space to operate, you will need 500GB of disk space to fully house their required files.

This high disk space requirement is a problem not only with ESX but also a component of essentially all Hardware Virtualization architectures. Some architectures provide a type of “consume on demand” feature for disk file sizes, creating virtual files that have the ability to expand as necessary to fill the total space. Due to performance reasons, ESX does not support this.

As an example of this “consume on demand” capability, with some virtualization architectures, creating a 50GB disk and installing 2GB of actual files to it will consume only 2GB. Conversely, ESX disks are created in full at creation time. A 50GB disk will consume 50GB of real disk space, even if no files are installed to the disk.

From the perspective of intra-virtual machine configuration management, ESX enables little in the way of added functionality. Each individual virtual machine is atomic and logically isolated from others and the physical hardware. Onboard virtual machines do not recognize that they are installed within a virtualization environment and have no recognition that they’re not installed to a typical physical host. Thus, installed applications and the management of these applications have little to do with the host’s installation into an ESX virtualization environment.

You will not implement ESX as a tool to manage applications and installed services. ESX is merely a facility for hosting entire machines. The management of applications and installed services within an ESX-based virtualization environment will be done through other tools.

Cost Model

VMware ESX is licensed on a per-socket basis. For physical servers, this licensing model encourages the purchase of servers with larger numbers of CPU cores per socket. In other words, you pay the same price for a copy of ESX installed onto a dual-processor computer as you do for one installed onto a dual-core, dual-processor computer—but you get double the processing power.

Remember that each individual socket in a system relates to the physical device where a processor chip is placed onto the motherboard. A single socket can contain either a single CPU or, in multi-core processor architectures, it can contain multiple CPUs.

Adding to the ESX pricing is an additional cost for the Virtual Infrastructure components as well as additional advanced functionality we’ll discuss next. Also adding to the ESX pricing are any third-party backup/restore and disaster recovery components you might want to implement to augment the environment.

ESX’s hardware requirements are relatively heavy and usually require hardware on their Hardware Compatibility List (HCL) for the best operation and performance. At a minimum multiple network cards, large hard drives, and multiple processors are required to run an ESX installation, making the hardware more expensive than physical hardware used for other infrastructure purposes.

Where It Excels

Competitive advantage for the ESX product lies along two major axes—performance and resiliency. The core component architecture of ESX is designed in such a way that it is well-performing in comparison with other Hardware Virtualization products currently on the market. Much of this performance stems from the hybrid hypervisor concept, streamlining the host OS down to a very thin layer. Even with the sharing difficulties discussed earlier, this extremely thin layer does an excellent job of scheduling and swapping resources between needy virtual machines.

The "thinness" of the OS means that many functions nominally available on other standard OSs are simply not available on ESX. You will likely not be installing applications directly to the host OS with ESX. It is designed to operate effectively as an appliance OS with little added functionality other than to serve virtual machines.

ESX's other competitive advantage is in its enterprise features that arrive with the highest purchase levels for Virtual Infrastructure. Here, Virtual Infrastructure's management enables all hosts in the virtualization environment to operate functionally as a single unit. A process called VMotion allows for virtual machines to move—without incurring downtime—from host to host based on preconfigured conditions. If one host sees higher-than-acceptable resource usage while another is relatively unused, the virtual machine can migrate without loss of service to the less-used host. Additionally, if a host crashes, residing virtual machines are identified and relocated to the remaining healthy hosts.

> When a host outage occurs, the relocation process involves a short downtime for the virtual machines. However, this downtime is often much less than would be required for an operator to manually bring the machine back online.

Where It Lacks

ESX lacks in two areas: Its pricing model and its aggregation of system resources. The pricing model as explained earlier involves a large upfront cost to buy into the environment. Recurring costs are similarly expensive as high-end server equipment and virtualization software alike require annual maintenance and support.

Exacerbating this problem are the costs to buy into the enterprise-level features discussed earlier. These are only available when purchasing the highest-level SKU for Virtual Infrastructure. The enterprise features require that virtual machines are stored on shared storage devices such as a fibre channel or iSCSI Storage Area Network (SAN). This incurs additional cost if not already present in the environment.

Secondly, as stated previously, ESX is not necessarily the best tool for sharing of system resources. The fully atomic nature of ESX-based virtual machines entails a massive horizontal scaling of physical resources necessary to simultaneously host all necessary virtual machines.

Impact on Server Candidacy

ESX provides easy-to-use tools for migrating physical machines into the virtualization environment. These tools are mature and function well, though they can take an extended period of time to complete a migration activity. That being said, two types of machines must be carefully thought out prior to consideration as virtualization candidates due to constraints of the ESX architecture:

- High resource use—Machines that make high use of resources, especially disk and processor resources, can significantly reduce the consolidation ratio within ESX environments. Pay careful attention to performance use on candidate systems to validate their use of these types of resources.
- Attached peripherals—Physical machines that require attached peripherals may not be candidates for ESX as some types of attached peripherals are not supported by ESX.

Microsoft Virtual Server

Our second example of a product that leverages the Hardware Virtualization architecture is Microsoft Virtual Server. This product enjoys many of the same benefits to virtual servers associated with a standalone ESX environment but does not include many of the enterprise-level benefits associated with a high-end Virtual Infrastructure deployment.

A differentiator here is the price point for Microsoft Virtual Server when compared with ESX. Microsoft Virtual Server is a free download from Microsoft that can be added to any licensed installation of Microsoft Windows Server.

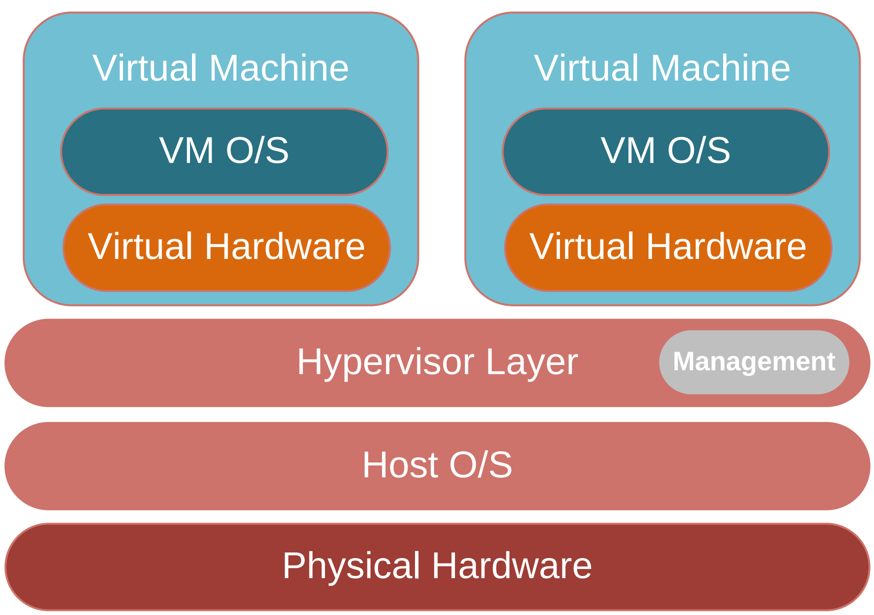

Figure 2.3: Microsoft Virtual Server arrives as an application installed onto an existing full OS. Its management is in line with its hypervisor application.

What It Does

Microsoft Virtual Server operates much the same as ESX, providing hosting for virtual machines residing on top of its installed hypervisor. Machines with Virtual Server are similarly atomic, operating ignorant of the virtualization environment in which they reside. Virtual Server provides many of the same features available with an individual instance of ESX.

Virtual machine disk files with Virtual Server are encapsulated into single disk files. Unlike ESX, where these disk files are monolithic and must consume their full available space at the time of creation, Virtual Server disk files can be configured to occupy physical disk space only as they are used. For environments in which space is particularly at a premium, Virtual Server disk files can be compressed using NTFS compression for additional disk conservation. This is handy when multiple virtual machines or virtual machine backups are kept online for redundancy, restoration, or disaster recovery purposes. It also serves to further reduce the total cost of ownership with Virtual Server.

Virtual Server can host many of the same OSs possible with ESX. Linux-based virtual machines as well as Microsoft-based virtual machines can co-locate on the same host for simultaneous operations.

One component of Virtual Server that is compelling to many organizations is its relationship with Microsoft. As a Microsoft product, the interface for Virtual Server is comfortable for existing Microsoft administrators. Thus, training and support issues are lessened. In comparison, other virtualization environments, such as ESX and Xen, have a heavy reliance on Linux-based OSs. For these OSs, troubleshooting can involve high-end, specialized skills not present in many IT environments. Thus, troubleshooting problems with virtualization environments in production can be more challenging than with Virtual Server.

Moreover, Virtual Server's tie to Microsoft also enhances its tie to Microsoft AD. Virtualization environments based on Virtual Server are automatically managed through existing AD components, which enhances centralized management and reduces the cost of operations.

What It Does Not Do

Virtual Server does not provide some of the enterprise-level redundancy, resiliency, and availability features associated with other virtualization solutions. Virtual machines in Virtual Server instances can be relocated from host to host without incurring downtime, but that movement is limited to between two host nodes. This is due to Virtual Server's reliance on Microsoft Cluster Services in a 2-node configuration for redundancy. Because of this limitation, cluster hosts must be configured such that the loss of an individual host and subsequent migration of machines will not overload the remaining host. Relocation of virtual machines as a function of resource load balancing are similarly impacted as load balancing can only occur between two possible hosts in a single environment.

Similar to other issues associated with ESX, the architecture of Virtual Server stands up a virtualization environment strictly for the hosting of entire computers. As with ESX, this architecture means that resource sharing is not as optimized as is enjoyed by other architectures. As the virtual machines are atomic in relation to the host—even with a full OS driving the host—those virtual machines do not share resources with the host like we'll see later on with OS virtualization.

Cost Model

Microsoft Virtual Server is a freely downloadable tool that can be installed onto any Microsoft Server OS. It has no annual maintenance costs. Support is enabled through the standard organizational support agreement in place for Microsoft Windows Server licenses.

Where It Excels

Considering its pricing model and installation onto an existing full OS, Virtual Server excels in environments that require neither the highest levels of virtual machine performance nor the highest resiliency and availability requirements for virtual machines. Virtual Server provides a solid virtualization environment that can aggregate multiple OSs and segregated workloads onto a single physical machine or, alternatively, a 2-node cluster.

In environments in which Linux experience is lacking or Microsoft homogeneity is a priority for contractual, support, or training needs, Virtual Server provides an interface that integrates with existing Microsoft management and support components likely already in production within the organization. This ease of integration means that the initial cost of entry for a Virtual Server-based virtualization environment is low compared with that of other tools.

Where It Lacks

Virtual Server suffers from performance limitations primarily due to its architecture. First, some background: As noted in Figure 2.3, the hypervisor layer arrives as an installed application atop an existing full OS. Positioning the shim layer at this level in the processing stack enables it to operate in conjunction with other applications collocated on that computer. A server that runs Virtual Server can also operate in other roles and provide other services. For organizations in which physical hardware is at a premium, this capability allows for the consolidation of virtual machine operating functionality with other necessary infrastructure-type services.

Though this capability sounds like a benefit, it comes with an associated performance loss. Virtualization hosts incur greater-than-normal processing than other typical services. This happens because they are effectively processing the needs of additional OSs on top of their own. When additional requirements are placed onto the host OS for other services and functionality, these resources are taken away from the residing virtual machines. Additionally, as the OS is a full OS rather than a streamlined one, the overall resources available to give to virtual machines are fewer.

> Comparing Virtual Server with other tools that reside on top of a full OS, such as Virtuozzo Containers, Virtual Server lacks the resource-sharing capabilities attributed to Virtuozzo Containers' OS Virtualization. Thus, consolidation ratios with Virtual Server will be less than with Virtuozzo Containers.

Impact on Server Candidacy

The reduced capability of Virtual Server to support simultaneous virtual machines due to resource requirements will impact the number and type of machines that can be hosted within a Virtual Server instance. This reduction in the consolidation ratio will have the effect of increasing the number of virtualization hosts required to support an equal number of virtual machines.

Xen/Citrix XenSource

Xen is a moniker used to describe any of a number of "forked" distributions of an open source virtualization tool originally designed as a university research project. One of the most known and widely used distributions of Xen is Citrix XenSource. Citrix XenSource is a commercialization of the open source code that offers numerous additional GUI-based components. These components are intended to ease the otherwise command-line heavy installation, management, and usage of the software.

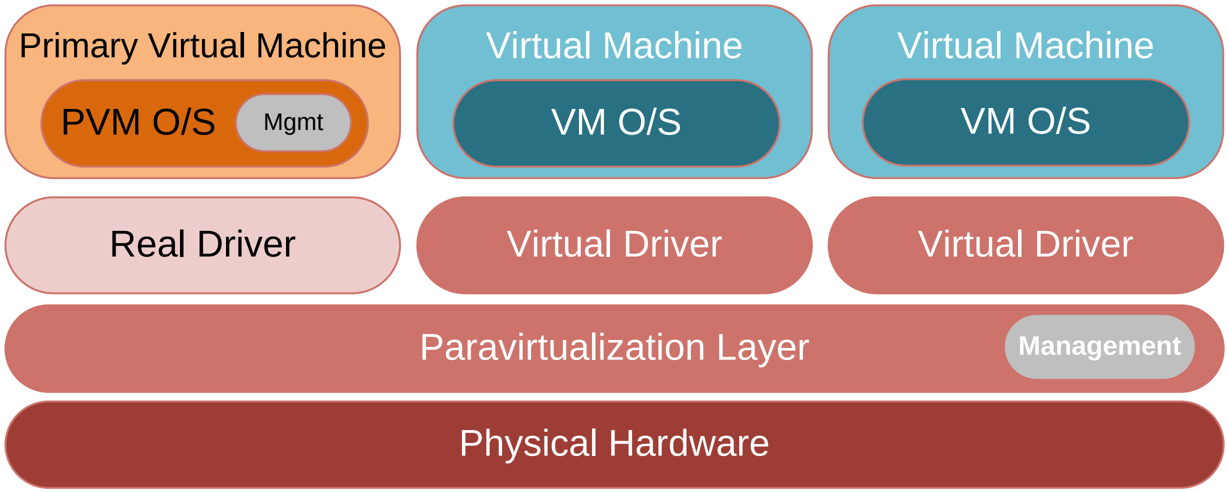

Figure 2.4: Xen's paravirtualization layer sits directly on top of the physical hardware. This layer enables a primary virtual machine for driver control and systems management in parallel with all residing virtual machines.

What It Does

Unlike the two tools described earlier, Xen and Citrix XenSource (hereafter referred to collectively as XenSource) are tools that enable *paravirtualization* of systems. Paravirtualization differs from Hardware Virtualization in that residing OSs must incorporate a set of low-level modifications to operate within the environment. These modifications, in combination with the virtualization architecture, provide a substantial performance enhancement for certain workloads.

Modern versions of XenSource can also integrate with hardware-assist modifications coded directly into processor chip instruction sets. These combine to eliminate the OS modification requirement. *Hardware-assisted virtualization extensions* enable XenSource to operate with unmodified OSs installed directly off the vendor's CD. As many OS vendors are unlikely to modify their code to operate within a paravirtualization environment, XenSource's hardware-assisted virtualization capability has enabled it to operate with new sets of OSs where it was previously unable. The most prevalent of these is Microsoft Windows.

The effective functionality of a virtual machine within a XenSource-based virtualization environment is similar to those within both ESX and Virtual Server-based virtualization environments. Virtual machines operate as atomic units, segregated from each other. Virtual machines with XenSource are managed similarly as well. XenSource provides a management interface for connecting to the console of residing virtual machines. That console eases common virtualization management functions such as virtual machine creation, migration, and removal.

XenSource—and Xen in general—originated within the Linux community, originally designed as a tool for virtualizing Linux workloads, so its relationship with various Linux distributions is longer than with Microsoft Windows. Hardware-assisted virtualization is a relatively new capability, only possible with processor manufacturers' recent release of onboard virtualization extensions. Thus, its history with non-Linux OSs such as Microsoft Windows can be considered nascent.

XenSource is distributed on a number of levels that provide incrementally more functionality. Higher levels of XenSource can provide many of the high-end enterprise functions enjoyed by higher levels of ESX. These hot migration, load-balancing, and high-availability functions position XenSource as a tool for enterprises who require high levels of availability and resource load balancing.

What It Does Not Do

Similar to both of our previous options, XenSource is also a virtualization tool for entire machines. With XenSource, virtual machines consume resources in much the same way as with our other two solutions. One difference here is involved with the driver sets assigned to those virtual machines. XenSource does not emulate hardware device drivers like ESX and Virtual Server, instead leveraging paravirtualized device drivers. These paravirtualized device drivers are substantially higher performing under certain workload types than with the emulated device drivers used in the other two products.

The unique difference that occurs with paravirtualized device drivers is that virtual machines are aware they are operating within a virtualization environment. Though this tends to have little impact on the effective operation of those machines, it is an important difference compared with the use of emulated device drivers.

> XenSource supports the use of emulated device drivers for the installation of a Microsoft Windows system. This is required to support direct-from-media installations. Once the installation is complete, the emulated device drivers must be replaced with paravirtualized drivers for production operation.

Unlike the other two solutions discussed thus far, XenSource does not support the backing up of entire virtual machines from the host layer. Whereas ESX and Virtual Server can create single-file backups of individual virtual machines for portability and improved restoration, XenSource recommends that all virtual machine backups occur within the virtual machine using traditional backup clients installed into the virtual machine.

Lastly, with the noted exception of the commercialized XenSource distribution, many Xen distributions are administered completely via the Linux command line. XenSource provides additional functionality through the use of its installed management consoles.

Cost Model

Xen, in any of its distributions, is an open source application. Thus, any of the distributions of Xen can be freely downloaded and used in production. However, be aware that different distributions may have different capabilities or feature sets.

Conversely, the XenSource commercialization involves a for-cost purchase. Among other features, XenSource's additional cost buys a set of GUI management consoles that ease the management overhead of administering virtual machines. XenSource is distributed with a number of levels, the highest of which includes the enterprise availability and load-balancing features discussed earlier.

Where It Excels

Xen and XenSource excel in terms of raw performance. Taking a look at Figure 2.4, the XenSource paravirtualization layer is installed directly onto physical hardware. This aligns with ESX's architecture and eliminates the extra layers and associated performance loss with Virtual Server.

Due to its history, XenSource excels in Linux-heavy environments. If administrators are versed in Linux command-line management and do not require the enterprise-level features in the highest levels of XenSource, its added costs may be unnecessary. This would have the tendency to reduce the initial cost as well as the total cost of ownership for a Xen-based virtualization environment.

Where It Lacks

The complexity of the various Xen distributions as well as the supportability of Xen/XenSource installations may be an inhibitor to some environments. There are a number of Xen distributions, many of which are "forks" off the original code. As open source software, these distributions arrive with little-to-no paid support offerings for production environments with high-availability requirements. The XenSource distribution does provide formal paid support options, which make it the most likely to be used for high-availability production environments.

As to management, even with the use of XenSource management consoles, the host OS code runs on a variant of Linux. For environments that do not have the necessary Linux experience in-house, the troubleshooting and administration of host machines can be more complex than with other virtualization solutions.

Impact on Server Candidacy

As XenSource's use of paravirtualized device drivers entails higher performance with some types of workloads than with Hardware Virtualization, this will have the effect of increasing the consolidation ratio within virtualization environments. Many of the same impacts discussed in the previous two products also apply within Xen environments.

Parallels Virtuozzo Containers

A virtualization product much different than the others we've discussed throughout this chapter is Parallels Virtuozzo Containers. Considered an OS Virtualization tool rather than one for Hardware Virtualization, Virtuozzo Containers virtualizes OS instances and changes within those instances rather than an encapsulation of an entire segregated virtual machine.

The primary difference in Virtuozzo Containers-based virtual machines is that the files that make up each residing virtual machine are components of the host. From a conceptual level, a virtual machine in Virtuozzo Containers is little more than the summation of any differences between that virtual machine and the host itself.

> A very important point worth noting here is that updating a file on a residing virtual machine will not impact the configuration of the host. However, updating a file on the host can impact the configuration of each residing virtual machine. This phenomenon bestows several advantages possible only with OS Virtualization environments. We'll discuss those advantages in a minute.

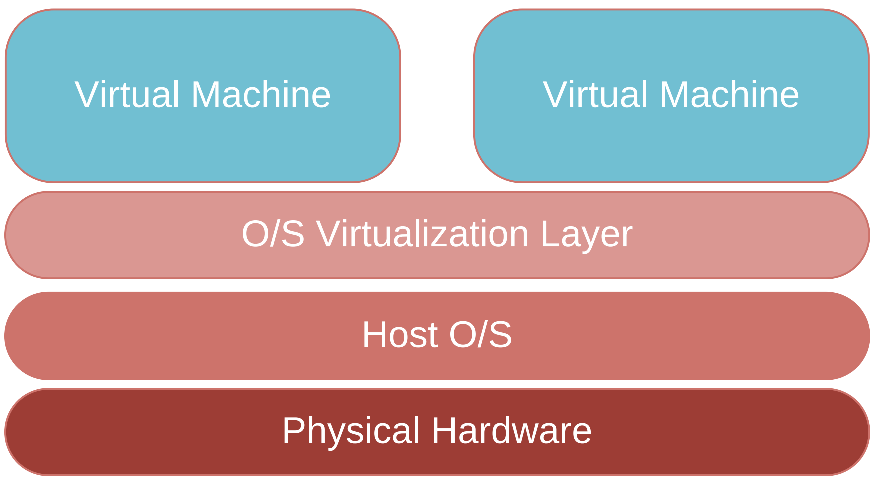

Figure 2.5: Virtuozzo Container' OS virtualization layer resides on top of an existing OS. It creates virtual "containers" where residing virtual machines and their individual configuration changes are stored.

What It Does

The OS Virtualization architecture enjoys benefits not possible through any of our other architectures. As all virtual machines begin their operational lifetime as a “snapshot” of the host, their disk consumption is considerably less than with other architectures. The process to create a new virtual machine (also called a *Container*) creates effectively an empty file whereby differences between the virtual machine’s configuration and the host are stored. These partitions or *containers* are logically isolated from each other. They operate as individual machines that are functionally segregated from each other.

As Virtuozzo Containers virtualizes machines at the OS level rather than the level of the entire machine, no emulation or paravirtualization of driver sets is required to drive the residing virtual machine. Both emulated and paravirtualized drivers add some level of performance drag to virtual machines due to the “proxying” of physical resources to virtual machines, so Virtuozzo Containers-based virtualization environments enjoy substantial performance benefits beyond other architectures with all types of workloads.

Management of virtual machines and individual virtual machine configurations are likewise improved through this architecture. As residing virtual machines are the “deltas” between the host and the individual virtual machine configuration, a change to the host’s configuration can immediately manifest itself into all residing virtual machines at once. This proves itself valuable during regular patching and other massive machine reconfiguration operations where many computers must be updated within a short timeframe. By updating the host machine, all residing virtual machines immediately enjoy the benefits of the update.

Continuing with the benefits associated with this architecture, virtual machine configuration updates are encapsulated into automatic installation files. If configuration updates need to occur to a subset of all virtual machines, these automatic configuration files can be distributed through the virtualization management interface to residing virtual machines as necessary. The encapsulation of changes enables their later reuse.

Finally, Virtuozzo Containers enjoys a greater level of dynamic resource allocation than with other architectures. Memory, processor, and disk resources for residing virtual machines can be reconfigured on the fly to add or subtract resources as necessary. This process occurs without requiring downtime on the part of the OS.

> ✐ As an example, if it is determined that a virtual machine needs to increase its RAM by an additional 1GB to support its current activities, that additional memory can be added on-the-fly with no reconfiguration or reboot done on the virtual machine itself. The virtual machine simply and immediately recognizes the change and begins utilizing the extra resources.

What It Does Not Do

The biggest limiting factor with OS Virtualization is a component of its strength. As all residing virtual machines are the summation of the differences between their configuration and that of the host, all machines that make up the environment must be the same OS and often the same major revision (that is, Service Pack for Microsoft Windows). For heterogeneous environments, this can incur an added cost as machines at different OSs and levels must be virtualized onto their own similar host.

Also different than the other architectures we've discussed in this chapter are the nature of the virtual machine files themselves. As each virtual machine requires its host to operate, individual virtual machines do not enjoy the same atomic properties as with other architectures.

Cost Model

Virtuozzo Containers is priced per CPU, all capabilities are included in the same price including a full-featured backup package, live migration, and physical to virtual tools. The management tools are sold separately as all management functionality can be driven from the command-line interface, but the value of the management tools in time savings and ease of use makes it an easy addition to justify. The high density possible with Virtuozzo Containers also enables cost savings in other areas such using the Microsoft Data Center license very thoroughly. Microsoft Datacenter is a set price for as many VEs as you can load on a single server, so the more you put on the lower your OS/cost per Container becomes.

Where It Excels

Virtuozzo Containers' virtualization architecture specifically enables the support of very high density environments. Due to the similarities between residing virtual machines and the void of requirement for device driver emulation or paravirtualization, virtual machines that operate within a Virtuozzo Containers-based virtualization environment can operate with very high performance in comparison with other architectures.

As the disk footprint of individual virtual machines is significantly less than with other architectures, Virtuozzo Containers is an excellent tool for use in hosted desktop and same application servers. For environments that require the support of large numbers of similarly configured systems—Virtuozzo Containers' architecture enables greater density with significantly less horizontal scaling than with Hardware Virtualization or Paravirtualization. There is no replication of core OS files within the Virtuozzo Containers virtualization environment, making the total cost of ownership much lower when considering disk impact.

For non-hosted environments that incorporate a large level of homogeneity in the environment, Virtuozzo Containers can also be a boon to administrative management. As all virtual machines are components of the host, regular patching and update operations—both costly and non-value added activities—can be reduced. Patches and updates need only be applied at the level of the host for all residing virtual machines to realize the change.

Virtuozzo Containers includes *hot migration* (live migration) support for Linux workloads. Different than the hot migration capabilities in ESX and Xen, a migration utilizing Virtuozzo Containers does not require a shared storage device for the hosting of virtual machine files. With other products, only state information is moved from host to host during a hot migration, changing the ownership but leaving virtual machine files on the shared storage. Virtuozzo Containers transfers the complete machine from source to target during a live migration.

In environments in which applications can be categorized and packaged, Virtuozzo Containers' management interface provides rapid deployment of configuration changes via *templates* to residing virtual machines. Moreover, configuration update packages can be applied en masse to multiple templates as well as running virtual machines as necessary. With Virtuozzo Containers the management of the Containers can include management of individual machine configurations as well—a feature not a part of other product sets. As templates are merely links within the Container to OS and application data, they are highly flexible. It is possible to create "sample" templates that aggregate this along with necessary resource assignments.

Virtual machine backups are integrated into the Virtuozzo Containers management console as well, incorporating enterprise-worthy features not present natively in other tools. These backups can be created across platforms and restoration is done through the management interface.

Lastly, for high availability environments, Microsoft Cluster Services in multiple-node configurations can be implemented within the environment to fail over virtual machines to available resources when outages occur. The same benefits discussed previously with Virtual Server and Microsoft Cluster Services are available when paired with Virtuozzo Containers.

Where It Lacks

As stated earlier, Virtuozzo Containers' most apparent limitation is related to the nature of OS Virtualization itself. As residing virtual machines must originate with the host's configuration, environment homogeneity is critical to obtaining the greatest return with Virtuozzo Containers.

Virtuozzo Containers is also limited in its abilities to perform certain types of hot migrations between hosts. Although Microsoft Cluster Services are the listed solution for high availability within Virtuozzo Containers environments, cluster-based high availability with Virtuozzo Containers suffers from some of the same limitations as for Virtual Server. One difference between the supportability of clustering between Virtual Server and Virtuozzo Containers is that clusters that house Virtuozzo Containers workloads can operate as multi-node clusters, whereas Virtual Server clusters currently are limited to only two nodes.

Impact on Server Candidacy

With a specific nod to its onboard native driver sets, Virtuozzo Containers' architecture has the tendency to significantly improve the overall performance of residing virtual machines. Suited to high I/O as well as high-density applications makes it cheap enough to do things like file and print servers. This will tend to increase the consolidation ratio as compared with other architectures. Consolidation with Virtuozzo Containers is impacted further when residing virtual machines are relatively similar to each other.

Solution Combinations

Lastly, there is value in discussing the use of virtualization solutions like those discussed here in combination with other IT infrastructure needs. Within the industry, there is currently a large amount of subjective information regarding which types of workloads are candidates for virtualization. Opinions such as "SQL Servers don't virtualize well" or "Terminal Services servers aren't good virtualization candidates" are prevalent when researching the use of virtualization as an IT solution. However, many of those opinions are based on specific product sets.

The issue with workloads is two-fold. First, when considering candidacy for incorporation into a virtualization environment, it is important to recognize the solution chosen. As was discussed throughout this chapter, some virtualization solutions incorporate greater virtual machine performance than others. This has the tendency to enable a greater scope of candidacy for possible systems. Due to the vocabulary prevalent in the industry, the productization of the term "virtualization" may not accurately consider all virtualization solutions chosen. One of the goals of this chapter is to show where that may be the case.

The second issue is related to the measurement of the combination itself. The consideration that certain products "don't virtualize well" can be considered an overgeneralization when related to the needs for specific performance testing. Few, if any, products simply don't work with virtualization. An engineering activity measuring the effective performance of concerning applications is necessary to validate application candidacy.

> Or, in simpler terms, if you hear, for example "Terminal Services doesn't virtualize well," that statement may be based on its inclusion in one virtualization architecture and not another. You'll need to do your own testing to see if it works for you.

Finding the Right Solution Is Critical to Implementation Success

Taking the hype out of the available virtualization options on the market will help you identify which options work best for your particular environment. Some architectures perform better than others under certain workloads and in certain environments. This chapter has attempted to show both the good and the bad of four possible options currently on the market. At the same time, there are other products also available not reviewed here. The analysis in this chapter also gives you the framework to analyze other products in relation to your needs.

In our next chapter, we'll retreat away from our specific product focus and return back to individual architectures and discuss how best to implement them. We'll talk about a set of best practices in implementing virtualization that will ensure the greatest chance of success and best return on your virtualization investment.