Realtime publishers

"Leading the Conversation"

The Shortcut Guide™ To

Selecting the Right Virtualization Solution

sponsored by

Parallels™

Greg Shields

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Chapter 1: The Virtualization Assessment

If you listen to the press, you'd think that virtualization could solve all the world's problems. Virtualization will reduce your total IT overhead. Virtualization will shrink your overall data center footprint. Virtualization will reduce your costs for power and cooling. Virtualization automatically enables entire-network disaster recovery. Virtualization will save you money, make you money, and make you more competitive. Virtualization will even make a decent cup of coffee, turn the lights out when you leave for the day, and automatically mail out birthday cards to your boss and your mom when you forget.

Well, maybe not all those things. But, if you follow the hype surrounding virtualization that has taken over the news cycles in the IT trade magazines, it almost seems like virtualization has every capability to solve all of IT's problems—if you just buy the hype.

In reality, virtualization can change a lot about how you operate and manage your IT environment—if you implement and manage it correctly. With the correct consideration of systems inventory and performance, you can compress tens or dozens of servers onto a single host to help reduce that data center footprint. Truly understanding your physical environment with all its servers and applications can help you see a total reduction in costs for power and cooling. Disaster recovery, though not an automatic recognition, suddenly becomes cost effective for even the smallest networks when coupled with virtualization. And adding in virtualization's capabilities for rapid server deployment, high levels of management automation, and service redundancy and isolation, it can save you money, make you money, and make you more competitive.

But getting virtualization into your environment isn't as easy a flipping a switch. To realize all these potential benefits, it is critically important to truly understand your existing, non-virtualized environment before you ever bring software in the door. That's the goal of this guide. This guide has been written specifically to help you understand the virtualization phenomenon, recognize the types of virtualization that are best fits for your specific network environment, and help you select the *right virtualization solution*.

The goal in these next four chapters is to help you wade through the media hype, recognize some of the common misunderstandings associated with the different classes of virtualization solutions available on the market, and align your knowledge with the realities of what you can implement and how it will benefit you.

Let's take a quick look at each chapter and what we'll be discussing:

- **Chapter 1: The Virtualization Assessment—**In this first, chapter, we'll talk about the different categories of virtualization in the market. If you're most familiar with the virtualization products from companies such as VMware and Microsoft, you may not be aware that their products are only one of many classes of virtualization. We'll talk about each category and the differences among them. We'll then go into detail about the first step in any virtualization implementation—the assessment process.
- **Chapter 2: Virtualization Software Options—**Once you understand the steps necessary in inventorying and otherwise assessing your current systems environment, the next step is to gain an understanding of the virtualization software options available on the market. Some get quite a bit more exposure than others but may not necessarily be the best fit for your particular environment. We'll dig deep into some of the vendors and discuss where they fit and where they may not be the best fit for you.
- **Chapter 3: Best Practices in Implementing Virtualization—**Once you understand the pre-implementation processes that ensure virtualization success, we'll spend some time talking about how best to get it into your environment. This chapter will talk about how virtualization changes both the network and the operating environment. We'll talk about where virtualization augments specific usage scenarios and where best to implement to recognize the greatest return.
- **Chapter 4: Managing Virtualization Environments—**Lastly, we'll talk about managing the new environment's day-to-day operations. We'll discuss the physical-to-virtual machine conversion options available as well as how to tie virtualization into a cost-effective disaster recovery plan to protect your business. We'll conclude with some external tools and environment-specific considerations you'll want to watch out for as you run your environment in steady-state.

What Is Virtualization?

From a purely academic standpoint, *virtualization* is little more than a layer of abstraction between two other layers. What do we mean by "layer of abstraction?" Here, we're talking about modifying the way we think about an item or refer to an item, by not referring to it directly. In the "real world," we deal with layers of abstraction all the time. We do this to hide the physical characteristics of complicated items or concepts.

For example, when I open my cellular phone and down-arrow to the item marked "Eloise," I have created a layer of abstraction between the person "Eloise" and the phone number 303-555-6739. It is more challenging and/or involves a higher cost—in this case, in terms of time—for my cell phone dialing procedure to deal with the 10-digit phone number. Instead, I reduce my procedural complexity by creating a layer on top of the phone number itself. That layer is easier to remember and less cumbersome to deal with operationally.

By adding this layer, I am also adding *transiency* to the underlying number. If at some point in the future the person "Eloise" needs to change her phone number, I need only change my pointer within the virtualization layer—the cell phone's address book—rather than my whole dialing process. This principle of transiency makes me more efficient and reduces the cost associated with change among my phone numbers.

A more IT-focused example involves the network's Domain Name System. DNS is a network service that translates IP addresses into friendly names. By doing its translation, DNS adds a layer of abstraction between the IP address and its associated name. Once added to the database, I no longer need to concern myself with each server's individual IP address. I only need to remember the friendly name and the DNS server will abstract its address for me. If ever a server needs to change IP addresses, I can change that address once in the DNS system and never need to change the name I know it by.

> Can you imagine if Google needed to notify all its users every time they went through a change in IP address? It would be functionally impossible. This layer of abstraction, or virtualization, of their IP addressing enhances their operational posture to their customers while enabling transiency of the underlying addressing.

Bringing this definition full-circle, the idea of virtualization in computing systems is to add a layer of abstraction between two layers in that computer system. This "shim" layer allows me to reduce my management reliance on complicated elements, like building new servers or deploying new applications, while also enabling transiency of the underlying virtualized elements:

- *By virtualizing a software application*, I eliminate its direct hooks into its host operating system (OS) allowing me to more easily install, remove, and modify that software installation without affecting the host OS.
- *By virtualizing an entire computer system*, I encapsulate its configuration into a data structure that is more portable, easier to manage, and has more capability for being backed up and restored.

Let's take a look now at some of the categories of computer virtualization currently available on the market and how they differ.

Categories of Virtualization

In this section, we'll talk about four of the current categories of virtualization. Each of these categories is designed to add that layer of abstraction on top of one where complexity exists. Here, we'll talk about virtual machines and OS virtualization, which are two major players gaining attention in the current market. We'll also talk about *paravirtualization*, a lesser-known branch of Hardware Virtualization with special requirements, and Software Virtualization that wraps applications rather than entire machines in a layer of abstraction.

Hardware Virtualization

Hardware Virtualization involves the concept of incorporating virtualization at a layer below even the OS. With Hardware Virtualization, this virtualization layer—often called a *hypervisor*—acts as a sort of proxy between individual virtual systems that sit above it and the physical resources that sit below it.

In the situation of virtual machines, each of the individual virtual machines is completely and wholly segregated from each other. Each virtual machine has no knowledge of its collocation with other virtual machines within the host computer. Each machine knows only that when it makes a request for system resources (CPU, disk, memory, or network), that request arrives from what it believes to be a physical resource. The virtual machine is abstracted at a layer directly on top of the physical hardware, so one of the benefits associated with virtual machines is that it is OS agnostic. The hypervisor cares only for proxying requests between code in the virtual environment and its physical resources.

With Hardware Virtualization, each individual virtual machine includes all the resources it needs to run itself within its virtualization environment. So for ten instances of an OS, ten copies of that OS' files and other configurations are needed. However, because each virtual machine is atomic and the calls to physical resources are done through a hypervisor, it is possible to run machines of disparate OSs on the same host. For example, among others, the following market solutions incorporate a Hardware Virtualization architecture:

- Microsoft Virtual Server
- VMware Server
- VMware ESX

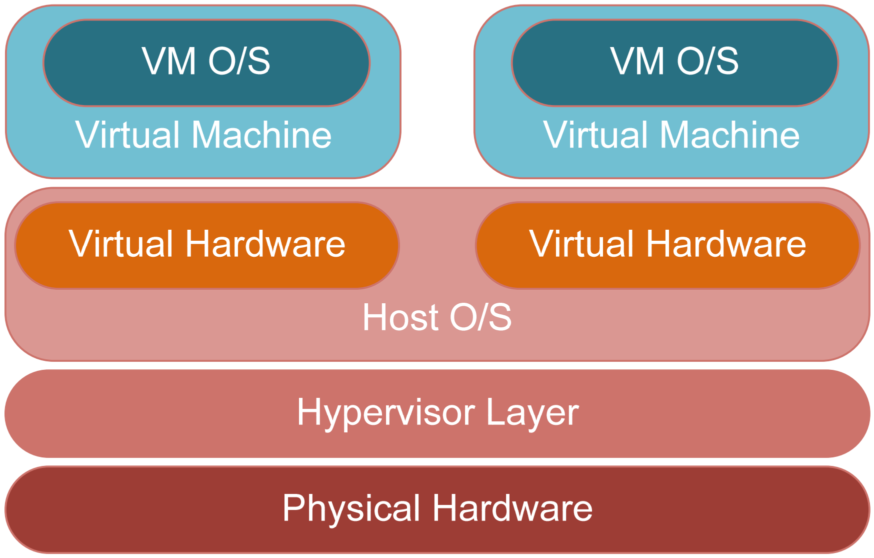

Figure 1.1: With Hardware Virtualization, the internal resources of each virtual machine are replicated. Each virtual machine is atomic and requests physical resources through the hypervisor layer.

OS Virtualization

If we move the layer of abstraction atop the host's OS, we alter dramatically how virtualization within the system works. With OS Virtualization, our virtualization layer gains some dramatic improvements in terms of performance but loses the ability to be agnostic regarding its hosted virtual machines.

How does this work? As with Hardware Virtualization, OS Virtualization includes a host OS and residing virtual machines are completely and wholly segregated from each other. What's different is that host's OS becomes the base OS from which all its hosted virtual machines start their existence. Each hosted virtual machine is almost like a virtual "snapshot" of the software that makes up the host. Like snapshotting technology incorporated into high-end storage systems, each hosted virtual machine's snapshot can then be modified to personalize its configuration. Automation components within the OS Virtualization layer can be incorporated to rapidly deploy additional servers or applications within each virtual machine.

Where OS virtualization differs from storage snapshotting is that the files in each virtual machine residing on the host are in many cases the same files as those that make up the host. If you change a file on the host, you change the linked file within each virtual machine. With OS Virtualization, the point of commonality, the base OS, becomes a point of central control for all hosted virtual machines. If an OS patch or service pack is needed for that OS, installing that update to the base OS automatically updates each of the residing virtual machines.

> Consider the configuration of any ten servers within your data center. On a file-by-file basis, how different are those server's configurations from each other? If one server is a DNS server and another a Windows domain controller, out of the hundreds of thousands of files that make up that server, it is likely that only a few hundred are actually different between those two systems. As so few differences exist between many servers, the "deltas" between each hosted system in OS Virtualization are likely very little as well.

Moreover, because the physical hardware is not virtualized as with Hardware Virtualization, overall performance is increased because there is no need to emulate a complete set of hardware resources for each virtual machine. The additional overhead required to emulate these resources in the Hardware Virtualization model consumes additional resources. This is not the case in OS Virtualization because no hardware virtualization is required.

This is possible because with OS Virtualization, there is no concept of a proxying hypervisor. Because there is no virtualized hardware that incurs the overhead of translating requests between virtual machines and the host, systems utilizing OS virtualization often show substantially higher performance than those that incorporate Hardware Virtualization. This translates into a much higher consolidation for OS Virtualization environments than in typical Hardware Virtualization environments.

But all this enhanced performance and optimized resource utilization comes at a cost: Each hosted virtual machine must be the exact same OS (and, often, service pack) as the host. With OS Virtualization, you can't, for example, run a Red Hat Enterprise Linux virtual machine on top of a Windows Server 2008 host. As an example, Parallels Virtuozzo Containers is one market solution that incorporates an OS Virtualization architecture.

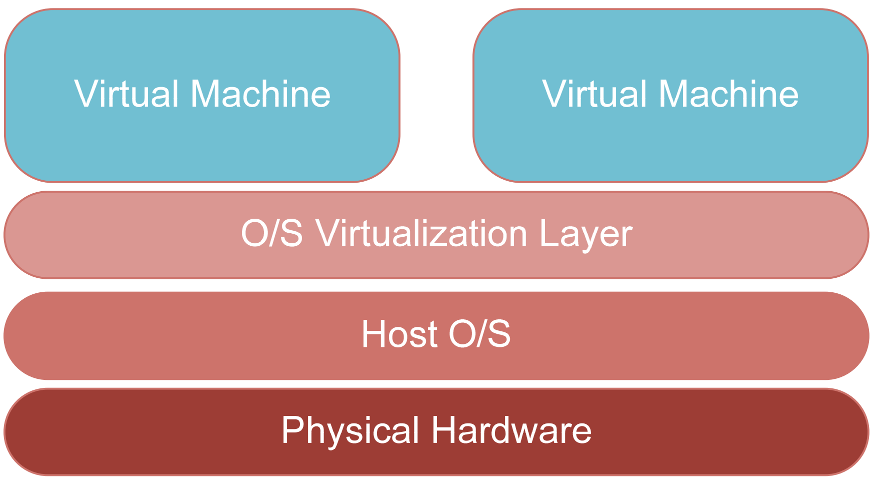

Figure 1.2: With the snapshotting effect of OS Virtualization, the duplication of files and configurations within each virtual machine is eliminated. The lack of a virtual hardware layer enhances performance. But all virtual machines must be similar in OS to the host.

Paravirtualization

Paravirtualization is an oft-misunderstood branch of virtualization software. Paravirtualization works similarly to Virtual Machines in that it enables the hosting of numerous machines on top of an existing host. Where paravirtualization differs is that it does not simulate hardware resources but instead offers a special Application Programming Interface (API) to hosted virtual machines. In order to use that API, the OS must be specifically coded to support it.

Paravirtualization benefits from significant performance improvements over other virtualization solutions, but the special coding requirement limits its usefulness in the marketplace when OS vendors choose not to provide the necessary modifications. As an example, Xen is one market solution that incorporates a paravirtualization architecture.

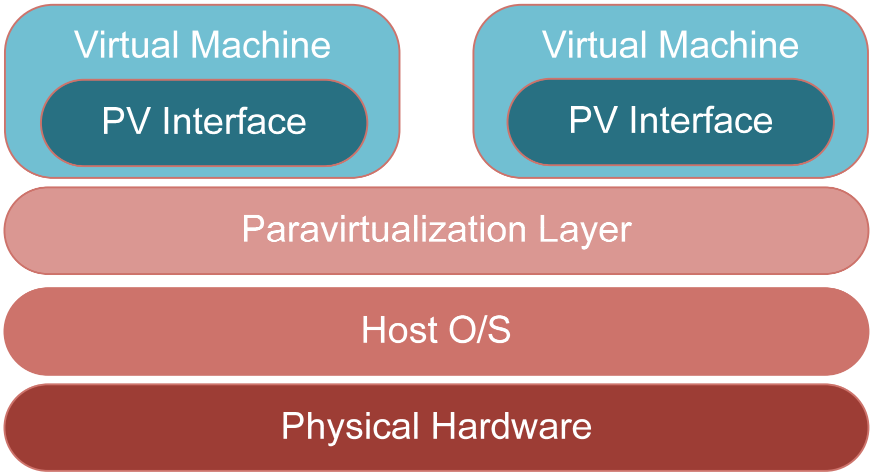

Figure 1.3: Paravirtualization operates much the same as Hardware Virtualization, but specific encoding of the OS is necessary to take advantage of paravirtualization's performance gains.

Application Virtualization

Moving away from the concept of entire-system virtualization is the idea of Application Virtualization. Like the title suggests, Application Virtualization encapsulates the files, registry keys, and other configurations of an individual application into a construct—often a single file—that can be easily installed to computers, removed from computers, and updated as necessary.

Think of this construct as a "bubble" in which the application exists. Within this bubble, its walls allow information only to flow outward. Configurations within the bubble can utilize the OS' resources to run itself, and the user can interact with the application in the bubble to perform useful work. But the OS cannot directly interact with the elements inside the bubble. By encapsulating applications in this manner, the application itself has no hooks into the system, which means that its installation does not impact the configuration of the host system.

Application Virtualization solutions are typically coupled with tools that enable the "streaming" of the application down to the client. This streaming process copies the necessary components of the bubble to the client system in a just-in-time manner as the operator needs those components. Because a software installation is unnecessary, and only the necessary bits are downloaded as needed, users can utilize automatic request systems to check out the software when needed and check it back in when no longer in use. Depending on license restrictions, this can provide a substantial savings on licensing costs for an organization. Among others, the following market solutions incorporate Software Virtualization architecture:

- Microsoft SoftGrid
- Citrix Streaming Server
- Thinstall Virtualization Suite
- Altiris Software Virtualization Solution

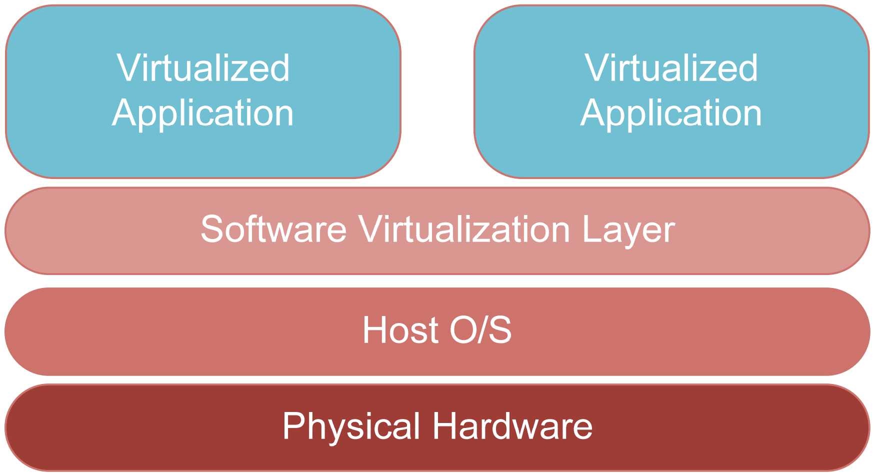

Figure 1.4: Software virtualization, unlike the other types, does not deal with whole systems. Instead, it encapsulates applications into virtualization environments for enhanced distribution and administration.

The concepts of virtualization don't only provide benefit at the systems level. Like with our cell phone number example, virtualization can provide measurable return whenever complicated systems require abstraction. Storage virtualization, network virtualization, and others are all other productized concepts available in the marketplace today. For simplicity, our discussion throughout this guide will limit itself to just those we've discussed above.

Throughout the rest of this chapter, we'll limit our discussion to just the topics of Hardware Virtualization and OS Virtualization.

Why Should You Care About Virtualization?

Moving a network environment to virtualization adds a number of immediately recognizable benefits to an IT organization. The benefits associated with cost reduction in both real numbers and time saved are compelling. These benefits are well known throughout the industry, but many of the actual numbers associated may be difficult to calculate without some understanding of their drivers. In this section, we'll attempt to demystify some of the driving factors to justifying why you should care about virtualization.

For many of these, we'll discuss the concept of *consolidation*. For Hardware Virtualization and OS Virtualization, consolidation describes the number of network services an environment can aggregate onto a single host. So, for example, notwithstanding the software chosen, if your virtualization deployment allows you to take ten separate network systems and aggregate them onto a single host server, you have recognized a consolidation of 10:1.

- Obviously, the higher the consolidation, the fewer the total number of physical hosts you need to run your network services. But that consolidation is not an unlimited number. Due to the physics limitations associated with any physical piece of media, there are physical limits to how far consolidation can go.
- Aspects such as application response time also are a factor in determining the best consolidation for an environment. Later on in this chapter, we'll discuss how you can use physical server performance metrics to gain an understanding about how well those servers will "play together" when they're compressed onto a host system.

Data Center Footprint

As your consolidation increases, it's obvious that the number of systems necessary to service the needs of your network environment will decrease. Let's take a look at some example numbers that can help illustrate the change to the look and feel of your data center after implementing virtualization.

- The average server rack can hold 42U of space. A U is 1.5" of vertical space.
- Subtracting 4U at the top and bottom of each rack for power distribution devices and networking gear and averaging 2U for each individual server, a fully loaded server rack can handle upwards of 17 servers per rack.
- If we assume a data center with five racks of equipment, four of these racks housing servers, there is the potential to house 68 individual servers within the available storage space. The additional rack's space will typically house storage, networking equipment, battery backup, tape backup, and other equipment.
- When virtualization is added to the environment, if we assume the environment can support a consolidation of 15:1 and include an N+1 configuration for virtualization host systems, it is not unreasonable to expect that we could compress 60 of the 68 systems into four host systems plus one additional system for redundancy. We leave eight systems non-virtualized due to extenuating circumstances we'll talk about later in this chapter.

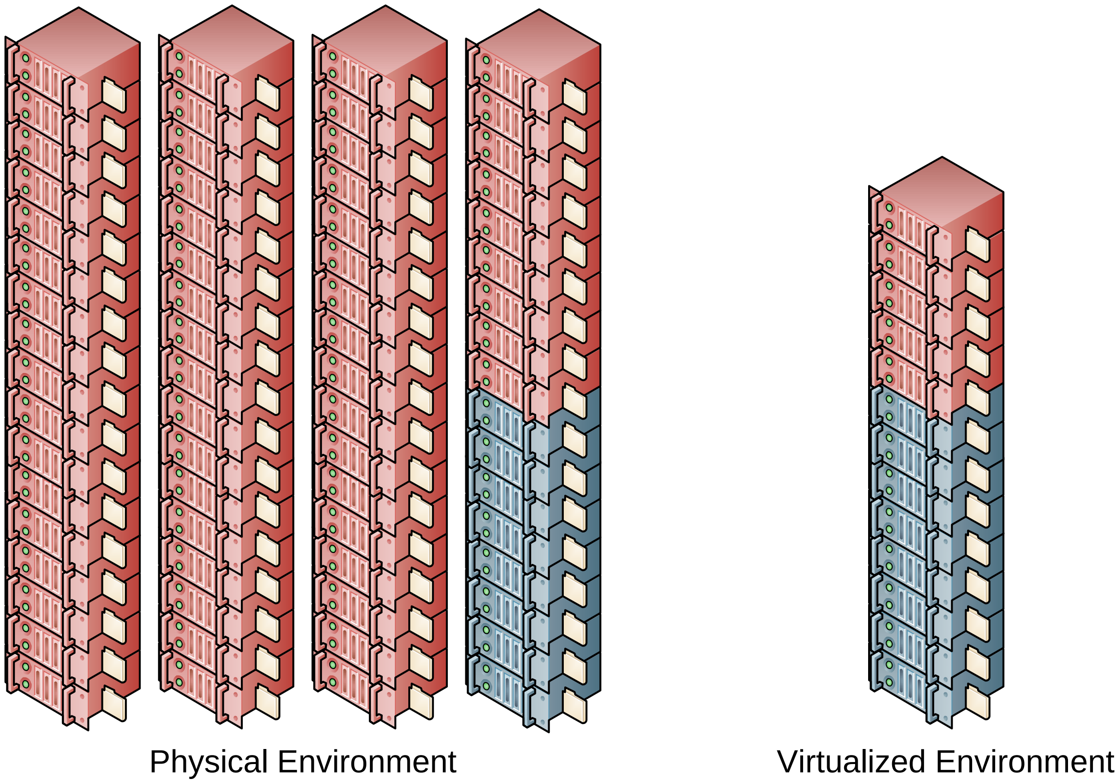

Figure 1.5: Conservative estimates on virtualization can support a 15:1 consolidation on servers. This has the effect of potentially reducing four server racks in a data center down to one.

In this conservatively-built scenario, we have the potential of compressing our total data center footprint from four racks of servers to only one. We've shown a visual representation of this in Figure 1.5. Our rack of networking, storage, and backup equipment remains unchanged, but 60 of our 68 servers have now been compressed onto five virtualization hosts. If your business leverages a collocation facility for your data center that bills based on square footage, this has the potential for a 60% reduction in recurring cost for leasing the space.

Heating and Cooling Costs

Continuing on with our example, let's add into our equation the costs to power and cool these systems. Here, we'll use the same 68 servers in our data center footprint example:

- The current United States average retail price for electricity as of March 2007 is 9.35 cents per kilowatt-hour (Source: http://www.ppinys.org/reports/jtf/electricprices.html—as of August 19th, 2007).
- A moderately configured server-class computer running at steady-state under nominal load consumes about 295 watts of electricity (Source: http://h30099.www3.hp.com/configurator/calc/Power%20Calculator%20Catalog.xls; server system rated is an HP Proliant DL380 G4 dual-processor 3.0 GHz with 2G of RAM, three 72G hard drives, and averaging 10% utilization).
- Combining these two numbers, the cost to power one of these servers for 1 year is about $242. For our 68 systems in our full physical environment, the cost to power these servers per year is $16,456.
- For us to be able to compress 60 of these systems into five virtualization systems, it is reasonable to assume that higher power systems are required and those higher-power systems will also operate with greater utilization. A high-end server-class computer running at steady-state under a high load consumes about 507 watts of electricity (Source: http://h30099.www3.hp.com/configurator/calc/Power%20Calculator%20Catalog.xls; server system rated is an HP Proliant DL380 G5 dual-processor 3.0Ghz with 16G of RAM, six 72G hard drives, two PCI-E cards, and averaging 80% utilization).
- Combining these two numbers, the cost to power one of these servers for 1 year is about $415. For five of these systems in our virtualized environment, plus the remaining eight that were not virtualized, the cost to power these servers per year is $4,012.

In this continuation of our scenario, we can see that reducing the total number of servers in our environment from 68 to 13 reduces our total cost for power by more than 75%, or about $12,000 annually.

This number is purely for the cost to power the servers. Adding to this is the cost associated with keeping those servers cool as well. Typically, due to the physics associated with tons of cooling needed to remove a quantity of BTUs of heat, the cost to cool a data center is roughly equivalent with the cost to power it. This has the effect of doubling our total numbers.

> Some electricity providers, such as Pacific Gas & Electric, even provide credits to businesses that incorporate virtualization into their data centers. This is particularly telling with data center costs consuming upwards of 48% of the IT budgets of some companies (Source: http://www.blade.org/tech_energy_efficiency.cfm). Adding to this are the environmental benefits and enhanced customer goodwill associated with the move towards "green" operating practices.

Computer Refresh Costs

The average life cycle for a computer server is between 3 and 5 years. Businesses typically refresh servers as they come due for their extended warranty. This is due to the non-linear increase in maintenance fees that occur with servers as they age beyond their initial warranty. That being said, buying new servers is usually a portion of every IT organization's budget, just to keep up with the demand for aging servers and new services.

No matter where the layer of abstraction occurs, Hardware Virtualization and OS Virtualization alike add the ability to inexpensively deploy additional servers often without requiring an additional hardware purchase. Virtualization environments like the one in our earlier example are built with additional resources in mind for future expansion. So for many cases, the process of building a new server for the network environment is little more than copying and pasting a template file and purchasing an additional OS license.

Moving away from actual numbers and into estimates, if we assume that a moderately powerful server-class computer costs $5,000 to purchase and a high-end server costs $20,000 to purchase due to the extra processors, RAM, hard drives, and internal bus speeds, the break-even point for virtualization consolidation approximates 4:1. Thus, once you've consolidated four low-use servers onto a single server, you've broke even on cost.

Adding to this calculation for Microsoft Windows environments are a change to the licensing model for certain OS editions that further encourages virtualization (Source: http://www.microsoft.com/licensing/highlights/virtualization.mspx). This change means that for purchases of Microsoft's mid-tier server OS edition—Server 2003 Enterprise Edition—a customer that implements virtualization gains the privileges of running four virtual OS instances for every licensed physical instance. This incurs an 80% reduction in costs for Microsoft licensing. For their top-tier Datacenter Edition licensing, the benefits are even more liberal with the ability to run unlimited virtual instances on top of a physical instance. This translates to a linear reduction in costs as consolidation increases.

Management Costs

The costs associated with management of all types virtualization environments are similarly reduced, notwithstanding their architecture. This is due to the consolidation of virtual servers into a single environment with remote management-capable entry points and automation features built-in to the virtualization software. Let's take a look at some of the more common benefits gained by systems administrators upon a movement to virtualization within their environments of responsibility:

- Rapid Deployment—The process of building a new server is a long, tedious process that can take hours or days, depending on the number of tasks that must be accomplished—and this is just to bring that server to the point where its hosted applications can be installed. Similar to how the assembly line brought massive economies of scale to the automaking industry, virtualization's ability to "templatize" servers reduces the time for server deployment from hours to minutes.
- Commonality—As a business' network matures and several refresh cycles end the life of some servers while keeping others around, many networks find themselves in the situation of needing to support numerous different classes of server equipment. An HP DL360 Generation 1 server has quite a different set of drivers than does an HP DL365 Generation 2. Because of the differences in hardware between server classes, it is operationally impossible to relocate OS instances. With virtualization, each of the virtual servers on that host leverage the same driver sets as the host itself.
- Temporal Nonlinearity—Native to nearly all virtualization solutions is the ability to "snapshot" that system's existence at any point in time. Many virtualization solutions also support multiple, iterative snapshotting, which enables the property of temporal nonlinearity to the server environment (or, more plainly, "you can go backwards in time"). If the incorporation of a set of bad patch code, an improper configuration, or malware into the server environment renders that server inoperable or unstable, it is a trivial process to revert the server to a state some period in the past. This ability to turn back the clock enhances the stability profile of servers and reduces the risk of change operations to the production environment.
- Loosely-Coupled Simultaneity—Taking our "snapshotting" capabilities one step further is the ability to offload those snapshots to a remote location for emergency purposes. Software exists for virtualization systems that enable the automated snapshotting, offloading, and transfer of this data to off-site locations. By doing so, it is possible to have copies of the same server simultaneously in two different places. The term "loosely-coupled" is used here based on the medium used for the transfer. That medium can be network-based, which moves the information in near-real time. Or physical media can be used. Physical media tends to store fewer and less up-to-date snapshots, but within a more stable media.

Impact to Disaster Recovery Metrics

Because virtualization adds this ability to think of an entire machine as a single file, the process to back up and restore that file is greatly enhanced. With many backup systems, one of the major components in time-to-restore is the time needed for the restoration medium to find each file within the tape catalog and restore that file. Virtualization files are few in number and very large, so very little tape drive seek time is required, which allows the drive to operate with much higher efficiency.

Additionally, as virtualization enables the ability to do near-real time offloading of snapshot information to alternative sites, the potential to rapidly power on the remote copies exists if sufficient processing power is available and properly pre-staged at the alternative site.

There are two metrics that are commonly used to describe the desired state after a disaster event occurs. Those metrics are Recovery Time Objective (RTO) and Recovery Point Objective (RPO). The RTO of a disaster relates to the acceptable maximum amount of time that a service can be down related to the disaster. The RPO of a disaster relates to the maximum amount of data loss that is acceptable as part of a disaster event. These metrics are often used in IT Service Level Agreements (SLAs) to identify specific goals for a system's availability after a disaster event.

The major impacts to RTO and RPO will be based on the number of snapshots offloaded to the alternative site (affecting the RPO) and the ability of that site to come online and begin servicing requests (affecting the RTO).

📖 We'll talk more about disaster recovery and these terms in Chapter 4.

Virtualization Assessments Are Critical to Success

Now that we've spent considerable time talking about the types of virtualization and the benefits associated with the move to virtualization, we need to talk about that first critical component of any virtualization project—the pre-implementation assessment. According to a survey announced by Computer Associates in 2007 and referenced at http://www.virtualization.info/2007/03/44-of-companies-unable-to-declare-their.html, "44% of respondents who said they had deployed server virtualization technology were unable to declare their deployment a success."

The article continues, "For organizations claiming success with virtualization, the most important factor was being able to measure performance of the virtualized environment. Other key success factors cited in the study include diligent inventorying of server assets and load distribution, and thorough investigation of available technology solutions."

The assessment process is a much-needed component of the virtualization process. This occurs for a number of reasons: Not all servers are created equal, and not all are used equally. Although Microsoft Windows server utilization averages about 5%, some servers such as database and email servers can go far above that. Additionally, some virtualization solutions cannot support some types of attached peripherals. Understanding the inventory—hardware, software, and performance-associated—of the servers in your environment will go far in ensuring you find the right virtualization solution for your environment.

The Components of a Virtualization Assessment

With any major shift in the structure of the network, such as what occurs with the implementation of virtualization, it is critical to fully understand the "as-is" environment before any changes are made. This understanding must occur along a number of different axes: Understanding the hardware and software inventory, the network topology and its capacity to support the change, performance collection and analysis, and determining the virtualization candidacy of the current physical systems are all components.

As virtualization has a tendency to "put all the eggs in a few baskets," we must ensure that our desired end-state has few if any single points of failure. We must also ensure that the systems we plan to collocate onto our host systems play well with each other and will not contend for resources. This process of getting to know our systems from a performance standpoint feeds into our decisions about virtualization candidacy. In the end, we will use all this data not only to determine the best fit software for our environment but also how to best engineer that software so that it increases our overall system reliability.

Hardware Inventory and Network Topology

Our first step is to develop a complete understanding of the physical nature of the systems we want to virtualize. What is the hardware makeup of these systems? What attached peripherals do they require? How is the network configured? Do we need to deal with multiple networks or a single one? Some components of a good hardware and network topology inventory we need to understand are:

- Processor class and speed—We will later analyze the performance of the system in relation to its current processor. As the new virtualization hosts that will be brought online to support this system will likely have greater horsepower than the current systems, we need to know the current capabilities of our systems.
- Memory—For any virtualization system, RAM is the least "shareable" of the resources within the system. Only one piece of data can exist in a particular piece of memory at a time, and changing that piece of data involves a resource-intensive swapping activity. Thus, we must have an understanding of the quantity of RAM currently installed in the system to use in determining how much RAM we should spec for our host system.
- Attached peripherals—Are there any serial, parallel, SCSI, or USB devices that are currently attached to this system? Are they required? Can those devices be virtualized through a software-based solution? Can those devices be moved to another network location? As many attached peripherals do not function correctly in the virtualization environment, we need to have an understanding of those systems that are immediately non-candidates due to required attached peripherals.
- Network cards and topology—What network cards are currently in this system? Is this system used as a bridge between networks? Are those cards teamed? Is that teaming considered failover or load-balancing teaming? Is this server a member of a DMZ or our production network? Do special firewall rules relate to this server and why? Another of the major architectural decision we'll need to make as part of our virtualization system implementation is the number of network cards to purchase and the networks to which those cards will attach.

Software Inventory

Once we understand the physical nature of our systems, we then need to understand what they are currently doing. For each system we intend to virtualize, we should have an understanding of its reason for operation and its OS and application composition. This information will be used when pairing systems with each other in the final result. Some components of a good hardware inventory we need to understand are:

- OS—We need to know the OS and patch level for each system. We also need to know how the OS was installed. Was it an OEM installation or did we install the OS ourselves? Is this an OS that is supported by our selected virtualization solution?
- Installed applications—With the exception of core infrastructure services, nearly all servers have some third-party non-OS software that runs the service it provides. Our inventory needs to know those major applications as well as how those applications interrelate with other applications on the network.
- Applications supportability—Some applications may or may not be supported by their vendor within the virtualization environment. For the most part, those applications that are not supported typically will function within the environment. For those that function but are not supported, is the business willing to take the risk of hosting them in an unsupported configuration?

> It is an unfortunate fact that many software applications are still unsupported by their vendors within virtualization environments. You'll find that those applications, with rare exceptions, function with no issues. However, convincing the application vendor to support the configuration is challenging.

License Inventory

In addition to software, a virtualization assessment activity is also an excellent time to undergo a license inventory. Doing so will ensure that we have the correct number and types of licenses for the environment we are currently running. Some components of a good license inventory we need to understand are:

- License count—Do we have the correct number of OS and application licenses for the number of instances we currently have? Do those licenses match with the correct edition of the application or OS? Do we need to add more licenses or upgrade our existing license set? It is also a good idea to understand our license count from the perspective of license compliance, which can be expensive to the business if we are found to be out of compliance.
- License growth—Virtualization environments, due to the ease of building new systems, tend to see an initial hypergrowth in the number of new servers brought online. Often, this is a good thing as IT gets their first opportunity to practice good service isolation (for example, getting file services off Windows domain controllers). Does the business have the budget and the desire to ensure that enough licenses are available for the needed growth? Or, are there too many licenses? With the change to the licensing rules for Microsoft licenses, is there the potential for a license harvest for OS licenses? Can those licenses be redistributed to other uses or even returned back to their vendor for a credit?
- Version age—With the impending massive change to the operating environment, is now a good time to upgrade existing software versions to the latest and greatest? Or is the risk of too many changes too great? As the organization goes through a virtualization activity, they may determine that the changeover is an excellent time to upgrade critical systems.

Performance Collection

Easily the most important of the activities needed during the assessment period is a solid understanding of the core performance of the candidate systems. Interestingly enough, as Moore’s Law has hit the IT industry again and again over the past few decades, the needs of server software have rarely kept up with the advancements in hardware technology. This dissonance between hardware capability and software needs has resulted in what can only be called a surplus of hardware resources within most business networks.

This resource surplus is the main reason why virtualization works. For systems running at 5% utilization, the other 95% is non-productive flipping of 1s and 0s to occupy the system during idle periods. Virtualization consolidates multiple systems onto single hosts and increases the effective utility of those systems from 5% to something much higher. By doing so, the result is more efficient use of the servers on-hand and the electricity used to power those servers.

However, the nature of computer systems is dynamic. Some servers can idly flip 1s and 0s for 23½ hours of each day, only to spike to 100% utilization for their needed 30 minutes. Others never leave 80% utilization during steady-state due to the needs of their users. Even others (and, arguably, the majority of servers) spend most of their days rarely consuming more than 5 to 10% of their available hardware resources.

The performance collection component of the virtualization assessment phase entails little more than enabling performance counters on candidate systems. No matter what the system, the performance metrics should be enabled for a period sufficient enough to get a sizeable data set. Also, if your business has a standard business cycle—per month, per week, per quarter—where natural spikes occur in processing requirements as part of that business cycle, the performance collecting process should occur through at least one full business cycle. This ensures that performance collecting gathers both the low points and the high points in resource needs throughout that cycle.

Additionally, each business typically has performance "dead zones." For the typical office environment, that may be between 5:00 P.M. and 8:00 A.M. when no employees are in the office. Other businesses may operate around the clock. Averaging effects associated with performance collection over "dead zone" periods has a tendency to reduce the total recorded performance. Be aware to configure performance metric solutions not to collect during those periods.

Performance Analysis

Your next thought is likely, "I recognize that I should be monitoring for performance, but I'm just not sure what to look for." It's a valid statement. On a typical Microsoft Windows server, there are hundreds of potential counters that can provide data on performance metrics. Finding just the correct ones is the hard part. Let's talk about a few of the critical performance counters that you'll want to watch for when considering a server for virtualization candidacy:

- System\Threads—System Threads are an indicator of the number of things that the processor needs to do over a period of time. Each currently running process has a number of processing threads that need attention for them to perform their job. A high number of system threads indicate both a high number of simultaneously running processes as well as a high number of processes that require the attention of the processor. A very high count for System Threads can indicate that a particular system is not a candidate for virtualization.

- System\Context Switches—Mirroring the requirements of System Threads, the number of Context Switches is the number of times the processor needs to switch out what it is doing in order to perform work on some other activity. Context Switches are what makes multitasking possible within modern computing systems, but too many context switches are a drag on system performance. Systems with a high thread count often also have a high Context Switches count as the processor attempts to give equal priority to many needy processes, and the resource overhead with the switching has a tendency to reduce performance. A high count for Context Switches can indicate that a particular system is not a candidate for virtualization.

- System\Processor Queue Length—Processor Queue Length refers to the number of work items waiting for attention by the processor. Typically, a high count for Processor Queue Length indicates either that the processor is not fast enough to keep up with the load placed upon it or that the applications installed on that server are asking too much out of the processor. When considering candidacy for a server based on a high Processor Queue Length, consider the current processor of the system. If the current hardware is insufficient for the needs of its installed applications, that server may still be a virtualization candidate. However, if the server has a high Processor Queue Length and a good processor, the reverse is likely true.

- Processor\% Processor Time—% Processor Time indicates how much time the system processor is actively doing work. A high count for % Processor Time is not necessarily a bad thing—that just means that your processor is efficiently performing useful work. However, in terms of virtualization candidacy, a high % Processor Time indicates that, once virtualized, that server may not be able to collocate well with other systems on the host. A high % Processor Time will tend to reduce your consolidation.
- Physical Disk\Current Disk Queue Length—Current Disk Queue Length is the same for the disk subsystem in a system as Processor Queue Length is for the processor in a system. When Current Disk Queue Length is high, this indicates that the processing of a server is requesting more reads and writes from the disk subsystem than that subsystem can handle. Due to the highly disk-based nature of virtualization, virtualization hosts tend to have high disk subsystem requirements just to swap out resource requests from the hosted virtual machines. If one of those virtual machines performs too many operations to disk, this can impact the performance of the other hosted virtual machines. When Current Disk Queue Length is high, check to see whether disk operations can be offloaded to other disks—perhaps a different volume using different spindles, or another disk array entirely. If not, this system may not be a good candidate for virtualization.
- Memory\Pages/Sec—Systems that incur a high count for Pages/Sec are those where the types of processing done by the system are ever changing. When the system cannot find what it needs among the items already stored in RAM memory, it needs to page to disk to find the data it requires. Any pages to disk involve high latency due to the transfer itself and the differences in speeds between memory and disk. If a system has a very high count for Pages/Sec, it may not be a good candidate for virtualization.
- Disk\% Disk Time—A high % Disk Time indicates that the system requires a lot of attention from its disk subsystem. Virtualization hosts already require a high % Disk Time just to support the needs of the onboard virtual machines, so a system with a high % Disk Time may impact the performance of those operations and make that server less of a candidate for virtualization.
- Memory\Available Mbytes—Memory is the least “shareable” of the resources within a virtual host due to the fact that only one piece of data can reside within a memory space at a time. When a system has a low Available MBytes and a lot of memory assigned to it, more RAM will need to be installed to the host to support that virtual machine. Where this counter is most effective, though, is in finding the correct amount of memory to assign to a new virtual machine. With physical machines, it is usually cheaper to add RAM at the time of purchase; physical machines are often configured with too much RAM. By subtracting the total quantity of RAM from the Available MBytes, we get a good idea of how much RAM that system actually needs in order to perform its function.

Determining Candidacy

Once performance data has been collected over the appropriate period of time, you must then take the time to analyze the data across systems. As a starting point, consider plotting each systems' counters against each other using a spreadsheet tool. Then look for counters that are greater than one standard deviation away from the averages of other counters. Counters that are one-off's from others may indicate that a server is not a candidate for virtualization.

Additionally, analyze your hardware and software inventory to look for deviations from other systems. If a counter is far out of specification from the others but is related to a physical server that is very underpowered for the workload assigned to it, that server may still be considered a virtualization candidate once moved onto hardware that can support its need.

It is important here to note that this process is not completely objective. There is some subjective analysis associated to the determination of candidacy that will take experience and a keen eye to locate. When in doubt, consider the system a candidate and deprioritize the system on the list of those to move to virtualization. It is always a good idea during the follow-on step when moving systems into the virtualization environment to move the highest-risk and those least likely to be virtualization candidates last in the list.

> We'll talk more about this process in Chapter 3 when we discuss the virtualization implementation.

Identifying the Best-Fit Virtualization Applications

Lastly, while looking at your suite of supported applications—and more importantly your suite of supported OSs—consider the various types of virtualization software options available on the market. As we've seen earlier in this chapter, there are a number of options available, and not all are those seen the most in the media. In the next chapter, we'll take a hard look at all the options and help you come to an informed decision about which work best in your particular environment.

The Critical Virtualization Assessment

There is a lot of hype in the marketplace on virtualization. The information that surrounds much of that hype can drive you towards a particular solution for your environment. However, that solution may not necessarily be a best-fit for your needs. The goal of this chapter, and indeed this entire guide, is to help steer you towards selecting the right virtualization solution for your needs. The first step in that process involves fully understand your current environment—its hardware inventory, software composition, and performance characteristics. The next chapter will take what we've learned here about the "as-is" environment and analyze it along with some known feature sets, strengths, and weaknesses of a number of solutions to help you find that best fit.

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.