Chapter 1

Performance Design

In the early days of VMware virtualization, we were all subscribed to one core set of beliefs: virtualization was a great tool to run multiple instances of nonimportant workloads on a single server. The stories always tended to start like this: "We tried virtualization with our mission-critical and performing workloads years ago and it ran horribly, so we don't virtualize those." Not everyone is willing to state exactly what year it was, but in pretty much every case they're talking about the first couple of releases of VMware. This particular distinction is important for two reasons: perception is reality, and people don't forget.

Digressing from virtualization for a moment, let's take a trip down memory lane back to the track and field of 1954. Those days weren't all that much different, with one minor exception: breaking the 4-minute mile record was an impossibility. Hundreds of years had passed with the belief that running the distance of one mile took a minimum of 4 minutes. Then on May 6, 1954, Roger Bannister did the impossible, breaking the 4-minute mile barrier.

But what does this have to do with virtualization, let alone performance considerations of designing VMware systems? We've gone our entire careers with the understanding that virtualization and performance were at odds with each other, a sheer impossibility. The whole concept of virtualizing mission-critical applications was not even a possibility to be pondered. We tried it in 2005 and it didn't work, or we know someone who tried it and they said, "No, it doesn't work; it's impossible."

Here's the good news: those barriers have been broken—shattered in fact. Virtualization is now synonymous with performance. In fact, virtualization can help drive even further levels of performance the likes of which would cause your physical systems to whimper. This book will help you take those old beliefs to your peers, colleagues, associates, and bloggers and put it all into perspective and context.

In the following chapters, we'll go through the depth and breadth of these perceived notions of performance-limiting areas so that we dispel old beliefs about virtualization and performance and focus on the reality of today with VMware vSphere. Most important, with discrete lessons, examples, and valuable scenarios of how to achieve performance within your virtualization environment, you'll walk away with information you won't forget, enabling you to experience virtualization and all its wonders for your most miniscule and most performance-critical and mission-critical applications.

In this chapter we look at:

- Starting simple
- Establishing a baseline

- Architecting for the application
- Considering licensing requirements
- Integrating virtual machines
- Understanding design considerations

Starting Simple

When it comes to design, people often begin by focusing on how difficult it is and start by overcomplicating things. Designing for performance in a VMware vSphere environment is no different. As with any design, there are a number of components that when treated together can be seen as a complex model likely to leave one overwhelmed by the challenges at hand. But you'll find that when broken up into discrete components such as CPU, memory, network, and storage, the entire architecture and ultimately its performance can be far more manageable. But where do we get started?

Determine Parameters

The first challenge when it comes to designing your environment for performance is determining what the parameters of the environment require in order to fulfill your needs. This is often translated into performance service-level agreements (SLAs) but may carry with it a number of other characteristics. Poorly defined or nonexistent SLAs commonly provision the maximum available resources into virtual machines, which can result in wasted resources, ultimately impacting your performance and the ability to meet any established SLAs.

For example, the typical behavior when provisioning SQL Server in a virtual machine is to allocate two or four virtual CPUs (vCPUs); 4, 8, or 16 GB of RAM; a sufficient amount of disk space on a RAID 1 set; and multiple 1 Gb NICs or 10 Gb interfaces. This is considered acceptable because it's often how physical machines will be deployed and provisioned. With little regard for what the application profile is, this typical configuration will spread from vSphere cluster to vSphere cluster, becoming a baseline established and set forth by database administrators (DBAs).

Not to disregard applications that truly meet or exceed that usage profile, but that should not be the de facto standard when it comes to an application profile design. Based on the latest VMware Capacity Planner analysis of >700,000 servers in customer production environments, SQL Server typically runs on two physical cores with an average CPU utilization of <6 percent (with 85 percent of servers below 10 percent and 95 percent of servers below 30 percent). The average SQL Server machine has 3.1 GB of memory installed with only 60 percent used, using an average of 20 I/O operations per second, or IOPS (with over 95 percent of servers below 100 IOPS), and last, an average network usage of 400 kilobytes per second (KBps) in network traffic.

Suffice it to say you could comfortably get by with a majority of your SQL Server installations running with 1vCPU, 2 GB of RAM, and on SATA disk. This is not to say that all of your servers will meet these criteria, but most of them likely will. This becomes important as you start to set the criteria for the starting "default template" to work from for a majority of your application profiles.

Continuing on the theme of starting simple, there are a few lessons that can help you get started down the road to meeting and exceeding your performance needs without having to invest months and months into testing. When working with a particular application, start by

referring to vendor support policies, recommendations, and best practices. "Sure," you're thinking, "Isn't that what this book is for, to give me recommendations and best practices?" Yes and no. Vendor support and best practices can change, often in response to updates, new releases, announcements, advances in hardware, and so on. So the best practices and recommendations for an AMD Opteron processor may differ from those for the latest Intel Xeon processor. Be sure to use these principles as a guide to ensure that you're asking the right questions, looking down the right paths, and applying the right rules when it comes to your architectural design, and with that knowledge in hand, you can easily handle the latest update to a CPU or a network card.

Architect for the Application

The second lesson when it comes to starting simple is to architect for the application and not for the virtualization solution, but to keep it scalable as projects require. Today's architectural decisions impact future flexibility necessary for growth. We'll go into greater depth on this later in this chapter, but we want to stress how relevant and important it is to remember that virtualization without application is just data consolidation with no value. In the end you're virtualizing applications into discrete containers that carry with them their own use cases and requirements, not building a comprehensive virtualization factory and cluster to hopefully house whatever you put into it.

Assess Physical Performance

The third thing you'll need to do is to sit back and pretend you're doing everything physically, and then do it virtually. A lot of people blindly enter into virtualization either assuming the system will be slower because it's "virtual" or expecting it to operate the same as it would if it were physical, and they unintentionally undersize and underarchitect things. Both of these choices can lead you down a dangerous path that will result in hair loss and virtualization regret. If you are able to garner a certain level of performance out of your application when running it physically but you give it fewer resources virtually, you're likely to get diminished results. Thus, understand the profiles of your applications and your servers, what their actual requirements are, and how they perform before you virtualize them instead of relying on perception after the fact with few or no metrics to help quantify.

Start with Defaults

And last, in the effort of keeping things simple, the fourth step to getting started is defaults, defaults, defaults, and best practices. They exist for a reason: garden variety simplicity. Designing with both in mind will help prevent unnecessary support cases. Unless you have a critical application that requires very unique and specific optimization that you gleaned by following the best practices and recommendations from lesson one, start with defaults. You can always modify the defaults and go from there when you have an unknown and untested application or use case. Operating within the realm of defaults enables you to begin to establish a baseline of the performance characteristics of your cluster and of your design. Don't hesitate to test things, but it shouldn't take you months to go through those iterative tests.

We cannot stress enough that establishing a baseline of predictability to meet your SLAs is important for your application profiles, but start small and work your way up larger as opposed to deploying large and trying to scale back from there.

Establishing a Baseline

The preceding sections provided examples of guidelines used for establishing a baseline. What exactly is a baseline, though, when it comes to VMware vSphere? A baseline is a series of predictable characteristics based upon what infrastructure you have in place for your CPU, memory, network, and storage.

Something you may realize quickly after designing and architecting your application is that once you start to establish a baseline, you've overprovisioned. Don't let the fact that you will likely overprovision 99 percent of your environment discourage you; the fact that you're reading this book is a sure sign that you're trying to get help for this!

There is no right or wrong baseline for your virtual machine and your applications. The exception is when you have anomalous conditions that should be treated as just that, an anomaly that, depending upon the characteristics, this book should help you identify and resolve. So without further ado, let's get down to the business of your baseline.

Baseline CPU Infrastructure

From an infrastructure standpoint, you are typically choosing your servers based upon the number of cores and CPUs it will support and the number of memory slots it can handle. From these architectural decisions you're able to dictate the maximum number of vCPUs your VMware vSphere cluster can support as well as the maximum amount of memory that will be available to your virtual machines. It will be these decisions that enable you to establish what the maximum configurations of a virtual machine will support.

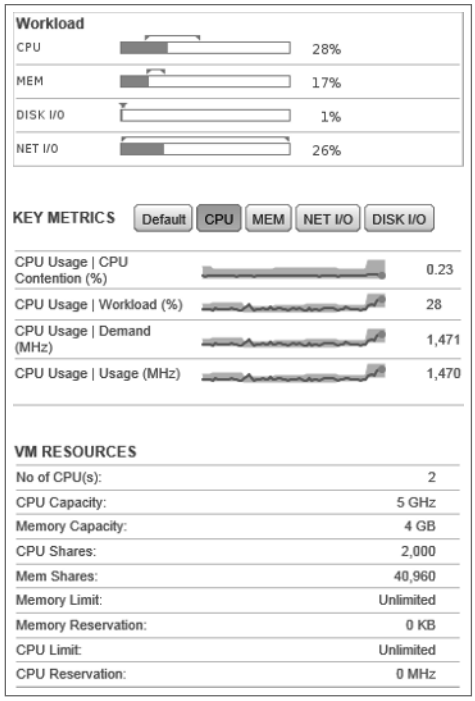
As an example, if you're repurposing older equipment that has a maximum of four cores available per server and is maxed out at 16 GB of RAM, you can reliably guarantee that you'll be unable to provision a single virtual machine with 32 vCPUs and 64 GB of RAM. So quickly, at a glance, your virtual configuration can only fall short of the resources available in your largest vSphere host—remember, vSphere and the overhead it reserves for VMs requires available resources as well.

Once you've established the maximum you're able to support, you'll quickly start to realize that your acceptable minimum is typically leaps and bounds below what you believe it to be. In the process of establishing a baseline of your applications, you will more often than not find that the "manufacturer suggested best practices" are conservative at best—and downright fraudulent at worst!

Let's take this example down into the weeds a moment if we can. How many times have you been told that an application requires a certain amount of resources because it is going to be ever so busy, only to find when you dive down into it that those resources go underutilized?

So does this imply that there is a best configuration when it comes to defining how many vCPUs you should assign to your application? Let's dig a little deeper into what exactly the characteristics of vCPU data can mean. You may notice in Figure 1.1 that the primary relevant data surrounding CPUs are number of CPUs, CPU capacity, CPU shares, CPU limit, and CPU reservations. Left to its own devices, your infrastructure can comfortably focus on the number of CPUs you assign a host and whether you specify any shares or limits; in most cases you can get by entirely on paying attention to the number of CPUs you assign the host and its workload.

Figure 1.1
Workload overview



Analyzing CPU usage, contention, and workload can help to establish your application baseline. It is an important distinction to point out that in fact a watched CPU *will* spike, so it is best to have data collected passively over time using a monitoring tool like esxtop, vCenter Operations Manager, or the built-in performance tool in vCenter. We've seen many users report that their application is underprovisioned because they loaded up Windows Task Manager and it showed the CPU spike; not only is that bound to happen, it is expected.

Now the most important part of monitoring your CPU utilization is going to be sitting back and *not* tweaking, modifying, or touching the number of shares or cores you have allocated. We know that's like being offered a cookie and being told to wait, but it is important to be patient in order to understand just what your application's use-case baseline is.

Author Christopher Kusek shared a little story about how he went about testing out these very steps to baseline an application. As engineers and architects we like to be extremely conservative in our estimates, but by the same token we want to use only the resources that are required. He had an application that can be pretty intensive, a monitoring tool that would collect data from his over 100 vCenter servers and the thousands of ESXi hosts that operated within those datacenters. Because it was fairly hefty, he felt that the initial two vCPUs and 8 GB of memory he allocated it was not up to snuff. So as this was an IT tool that could afford an outage whenever he deemed fit, he increased the number of vCPUs to eight and the memory up to 16 GB.

What this enabled was the ability to see at what peak the system would operate over a period of time (establishing the baseline) without impacting the performance of the application. He chose to take advantage of historical data collection and reporting, leveraging vCenter

Operations Management and the Performance tab of vCenter to collect this data. Something VMware vSphere is particularly good about is figuring out what it needs and sticking with it. In a short amount of time with this unpredictable application, it was established that the system would operate efficiently with a maximum of five vCPUs and 12 GB of memory.

At this point, Christopher tuned the system down to those requirements, and with his baseline set, he knew not only what the operational workload was, but also what the expected peaks were. He was then able to expect, without undersizing and impacting performance and without oversizing and wasting resources, what it really took to run this particular application.

What you will find is that allocation of vCPUs can be one of the most important aspects of your virtualization environment, especially when you consider the CPU scheduler covered in Chapter 4, "CPU." This is definitely one area not to skimp on in terms of aggregate MHz but at the same time not to be wasteful in terms of number of provisioned logical vCPUs (coscheduling complexity of >1 or 2 vCPUs = +++%RDY%) because it truly is a finite asset that cannot be "shared" without performance implications.

Memory

Memory is an infinite resource that we never have enough of. When it comes to establishing your memory baseline, you're typically going to see a pretty consistent pattern. All things being equal, you will find that if your application, workload, and use case do not significantly change, the amount of memory consumed and required will begin to be predictable.

Knowing the importance of memory in an infrastructure, VMware has made numerous investments over the years. Whether through memory overcommit, compression, or ballooning, this is one resource that is designed to be allocated. But it bears mentioning that just because you can allocate 128 GB of RAM on a system with only 64 GB of RAM doesn't mean you always should. What this means for your applications is that establishing baseline memory is a delicate balance. If you prescribe too little memory, your system ends up swapping to disk; if you prescribe too much memory, you end up overprovisioning the system significantly. This delicate balance is often seen as the sweet spot of memory allocation. For most people this tends to be pretty arbitrary, depending upon the application and the operating system. This was pretty easily done when running 32-bit applications because the system would be unable to address beyond 3 to 4 GB of RAM, encouraging a fairly consistent design of 4 GB of memory being allocated.

When it comes to 64-bit operating systems and applications capable of using large amounts of memory, there is a tendency to design as if you were running it physically and assign arbitrary amounts of resources. As a result, a virtual machine that may only require 512 MB or 768 MB of RAM will often be allocated with 1, 2, 4, or more GB of RAM. A step further than that when it comes to overarchitected and overprescribed applications like Exchange 2010, the minimum may come in at 12, 24, or even 36 GB of RAM.

Figure 1.2 shows a sample workload of an Exchange 2003 server with 4 GB of memory allocated to it.

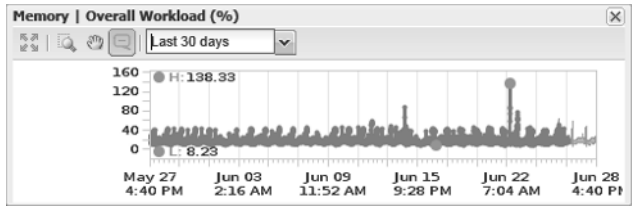


FIGURE 1.2 Memory overall workload percentage

While analyzing and establishing the baseline of this application over the course of 30 days, the low at ~327 MB and an average high would peak at approximately 1.6 GB of memory allocated. All workloads may experience a "spike" as this system did, demanding well over 5.6 GB of the available 4 GB, but anomalies are just that, and they can be expected to sometimes be well outside of the norms.

Fortunately for us, the method in which VMware employs its memory enhancements (see Chapter 5, "Memory") allows us to take an unexpected spike, as it did on June 22 (see Figure 1.2), without it having a devastating impact upon the operations of the virtual machine and the underlying application.

One area VMware does not skimp on is providing you with more than enough insight into whether you're making the right decisions in your architecture and what the metrics of those decisions are. Within vCenter, you're able to get down to the details of just how much memory the application you're running is using, as shown in Figure 1.3.

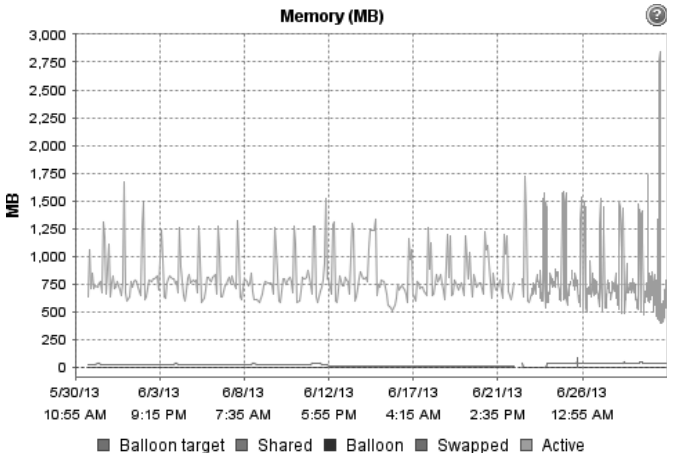


Figure 1.3
vCenter memory usage

At first glance you'd be able to pretty readily tell that your application over the prior 30 days has been running well under 2 GB of usage even at its high points. Use these tools at your disposal, such as the Performance tab, on your VMs within vCenter to get a comfortable feel for your baseline. Unlike having physical servers, where procurement is required to make critical changes with virtualization, you can simply shut down your host, add additional memory, and bring it back online if you happened to underprovision it in the first place.

It is important to identify that you can never be "wrong" when it comes to allocating your memory to your virtual machines. You might provision too little, or you might considerably overprovision the memory to a particular guest, but none of those decisions is set in stone. If you need to go back and either increase or decrease the amount of memory you provided, that should be the least of your worries.

Network

Knowing what your limits are when it comes to networking is especially important. It's extremely important to start with a solid foundation when it comes to your network. Taking advantage of features such as VLAN trunking (802.1q) and static link aggregation (802.3ad) when possible will help you keep your network infrastructure more virtual and reliable.

Whether you're building your networking infrastructure from scratch or repurposing existing gear, we cannot stress enough the importance of knowing your limits. If your application is latency sensitive, throwing more virtual NICs at the problem may not solve it as much as co-locating the servers that are communicating with each other on the same cluster and vSwitch. Networking can often make or break an infrastructure due to misconfiguration or general misunderstanding. Know what the aggregate potential of your VMware vSphere cluster is, including what the lowest common denominator is. You can readily establish a network baseline of "1 Gb links" if you find that the majority of your workloads are barely using less than 1 Mb, let alone a full 1 Gb link.

Often, networking problems can be overlooked when troubleshooting or understanding some troubles. Author Christopher Kusek recalls that he once worked with a VMware cluster where backups were running quickly except for on a few virtual machines used to back up large databases. A request was made to provision additional backup servers because the backups were running slowly and the client wanted to split the load up, assuming it would make it faster. It turned out that virtual machines being backed up that were co-located with the backup server were moving "super fast" because they were operating across the same vSwitch and able to transfer at 10 Gb, but the other virtual machines resided on separate nodes in the cluster and had to go across a slower 1 Gb link.

When it comes to networking in VMware, your network will often be limited by and only as fast as its weakest link. For example, due to misconfiguration, you may end up using a management network interface at 1 Gb or slower or a networking configuration on an uplink switch. However, if you follow a few standard rules and practices, you can prevent these problems on your network.

Networking tends to differ pretty heavily from decisions you make around CPU and memory because you are usually deciding how many MHz and MB you're assigning an application from a "pool" of available computing power. The decision around networking is only how many interfaces to assign an application. Unless your application requires access to multiple networks with different routes and VLANs, the answer to that will almost always be one interface.

What this translates into, as seen in Figure 1.4, is that unlike your memory and CPU workloads, which in peak moments can exceed 100 percent, network has a hard ceiling after which everything leveraging the network in the VM will either slow down or just drop the packets. Simply throwing more NICs at the virtual machine will usually not resolve this, especially if it is network bound by the ESXi host.

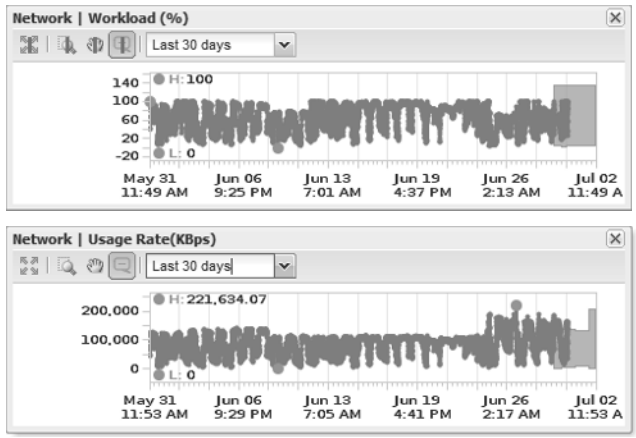


FIGURE 1.4
Network workload percentage and network usage rate

This particular file server was under pretty significant network stress continuously. When additional network links were introduced on the ESXi host system, the virtual machine was not only able to relieve some of that stress, it also had real-world user implications. At peak times, network bandwidth would drop to less than 1 KBps for end users, after providing the additional links even under the same peak stress as before the performance would soar to MB/s speeds, as you can see in Figure 1.5.

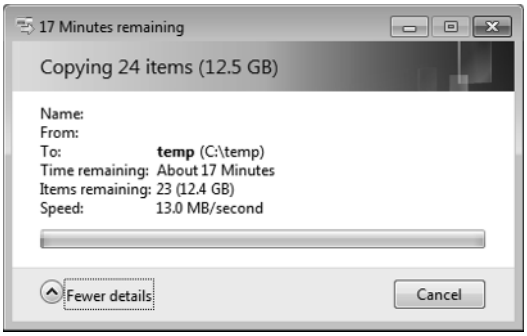


Figure 1.5
Networking file copy example

Networking will continue to be a challenge we all suffer through, usually limited more by our physical architecture than we ever will by our virtual design decisions. All you can do is make the best of what you have and identify what will benefit your applications best. In Chapter 6, "Network," we'll show you how to better make those decisions on how to work with what you have and how to identify situations where taking into account locality of reference and co-location of virtual machines will serve you better than merely throwing more hardware at the situation.

Storage

When it comes to establishing the baseline of a virtualized environment, one of the most overlooked areas is often storage. One of the greatest misconceptions is that because you switched from physical servers to virtual servers you should be able to skimp on the number of spindles required to handle the IO profile. Quite the opposite is typically the case.

While most physical servers will use less than 5 percent of their CPU potential and only a portion of their physical memory, and only touch the surface of their network cards, if a physical server was using 1,900 IOPS to perform its workload, it will continue to use 1,900 IOPS when it is switched to the virtual. Establishing your baseline when it comes to storage is even more important. Identify just how many IOPS you were using before, decide if there are characteristics of your applications that have particular special needs, and make sure that is reflected in the datastores supporting the storage for the apps.

While many of the characteristics of your configuration may change as you make design considerations for virtualization, how you design for storage isn't likely to change nearly as much as you think.

The same is true if your application was using 10 IOPS when it was physical; it will continue to use just as few in the virtual world. This also encourages you to cram a whole bunch of low-I/O and low-utilization previously physical machines into even fewer virtual machines. With the exception of some aggressive applications and workloads like databases, you'll come to

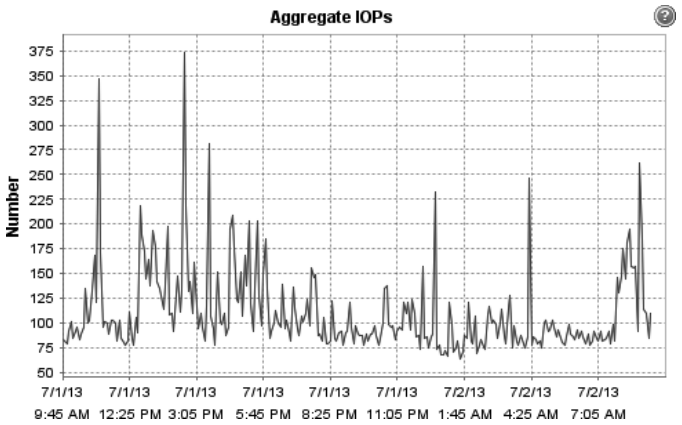
find the demands of the majority of applications are often more space constrained than IOPS constrained.

VMware has made a number of investments and development into storage over the years, recognizing its extreme importance to the delivery of operational workloads. Features such as Storage DRS (SDRS), Storage vMotion, VAAI, VASA, VSA, vFlash, VSAN, Storage I/O Control (SIOC), and multipathing policies take simple SAN or NAS provisioned disks to the next level for your virtual infrastructure.

With these virtualization-enhanced storage capabilities, you're able to maximize on-demand and online modifications of your virtual machine environment. What this truly enables you to do is establish your lowest tier of storage as the "default" for virtual machines and then move up a tier on a disk-by-disk basis if required by taking advantage of Storage vMotion online without application downtime.

When it comes to identifying the baseline of your application, operating systems tend to have a much lower set of requirements than performance-driven applications. Take the graph in Figure 1.6, for example. The 21 operating systems running against this single disk tend to be averaging around 10 IOPS apiece, running as high as 16 IOPS at their peak. That's hardly anything to be overly concerned about when it comes to storage design to meet the performance needs of these application OS disks.

FIGURE 1.6
OS aggregate IOPS



But the story tends to get a bit murkier when it comes to the baseline of applications that the business is reliant upon. In the graph shown in Figure 1.7, the average lows are below those of the operating systems seen in Figure 1.6, but the peak workload is significantly higher, requiring an extensive storage architecture to be able to respond to the demands of the application.

The architecture and design of storage is very similar to networking because these investments are usually not made lightly. Storage architecture, whether designed well or not, will usually stick with you for a minimum of three to five years, depending upon your organization's depreciation and refresh cycle.

Fortunately, the intelligence of the storage capabilities within VMware and guidance identified in Chapter 7, "Storage," can help you to take your storage architecture to the next level, whether through redesign and re-architecture or by simply making some slight modifications to take advantage of your existing investment.

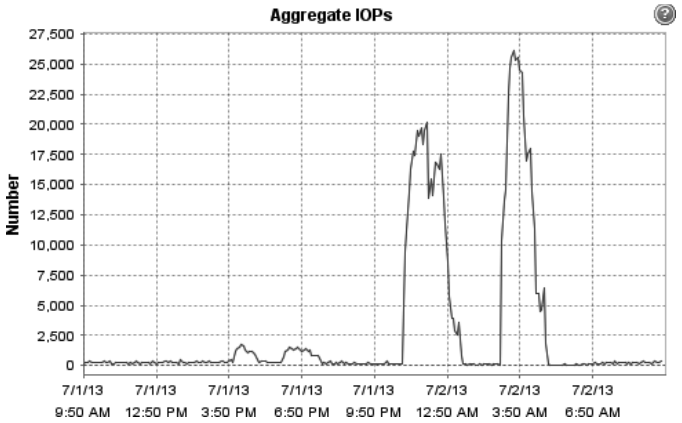


Figure 1.7
Application aggregate IOPS

Architecting for the Application

You've gone through the effort to build a VMware vSphere cluster and have established a baseline of the capabilities of your ESXi servers. Now you're ready to start populating it, right? Not exactly.

It is important at this point that you ensure that your architecture and even your templates are designed with the applications in mind and not the solution. The reason you do not architect solely based on your architecture is that given the ability, your application owners will request the maximum available that your solution supports. In other words, if they knew they could get a 32-vCPU server with 1 TB of RAM and 64 TB of disk space, even if only to host a 32-bit server that can't support more than 4 GB of RAM, they will. Then the requests would never stop coming in and your system would collapse in inefficiency.

At this point it becomes extremely important to define the applications and start characterizing their workloads. This matters whether you're deploying a web server, a highly performing database server, a utility server, or an AppDev vApp consumed by your development team; the characteristics of performance can be predictable and the expectation of the template standardized upon.

To begin, people will often establish a catalog of the services they offer to their end users, similar to that mentioned previously. Then from within this catalog a breakdown is established to meet the most likely characteristics as needed by the user community in the form of CPU, memory, network, and storage needs. As necessary, some of these workloads will be broken up into subsets of small, medium, large, or custom, as in the following examples:

Small: 1 vCPU, 2 GB of RAM
Medium: 2 vCPU, 4 GB of RAM
Large: 4 vCPU, 8 GB of RAM
Custom: Up to 64 vCPU with 1 TB of RAM

The rules about characterizing your applications and workloads aren't set in stone, but should be determined through your design and architectural considerations. Using tools like VMware Capacity Planner, Microsoft Assessment and Planning Toolkit, VMware vCenter

Operations Manager, and some of the native tools like the vCenter Performance tab and Perfmon, can help you take these characteristics from smoke and mirrors or app owners' mad dreams to cold, hard operational facts.

It is important to remember that if you undersize a virtual machine and an application—whether because you were unsure of the workload or because the number of users of a system increased, demanding additional resources—you can correct that by simply taking the system down and adding additional resources. We've yet to meet an app owner who complained when we visited them and said, "I noticed your application is actually underperforming and could use additional memory or another vCPU. Would you mind if I shut down your server and provide you with more resources?" They usually jump at the chance and establish a hard and fast downtime window for you to make those changes, if not let you do it immediately.

Yet, visit that same application owner and try to reclaim six of those eight vCPUs you allocated them and not only will they never find the time to shut down the system, they'll stop taking your calls! To head this issue off at the pass, you may want to hot-add on all of your virtual machines. Unfortunately, not all operating systems support hot-add of CPU and memory, and there are numerous caveats to consider, covered in Chapter 4 and Chapter 5, respectively.

Considering Licensing Requirements

The first thing you might be thinking of is what does licensing have to do with the design phase of your virtual infrastructure? Licensing has everything to do with the architecture and design of your vSphere environment. As you make decisions about how many virtual CPUs you want to allocate to your templates and ultimately to your applications, this can have a direct impact on how much you'll be paying in licensing to support that particular application.

When you have an application that is licensed on a per-vCPU basis, if you're able to meet and exceed the SLAs of that application with fewer vCPUs, you will be saving hard dollars, which translates into a more cost-effective and efficient infrastructure. The following table gives you a sense of hard vCPU limits when it comes to license versions.

	VSPHERE ESSENTIALS KITS		VSPHERE WITH OPERATIONS MANAGEMENT ACCELERATION KITS		
	Essentials	Essentials Plus	Standard	Enterprise	Enterprise Plus
Includes					
vSphere	6 CPUs	6 CPUs	6 CPUs	6 CPUs	6 CPUs
vCenter Server	1 instance vCenter Server Essentials	1 instance vCenter Server Essentials	1 instance vCenter Server Standard	1 instance vCenter Server Standard	1 instance vCenter Server Standard
vSphere Data Protection Advanced				6 CPUs	6 CPUs
Entitlements per CPU License					
vCPU	8-way	8-way	8-way	32-way	64-way
Features					
Health Monitoring and Performance Analytics			•	•	•
Capacity Management and Optimization			•	•	•
Operations Dashboard and Root Cause Analysis			•	•	•
Hypervisor	•	•	•	•	•
vMotion		•	•	•	•
High Availability		•	•	•	•
Data Protection and Replication		•	•	•	•
vShield Endpoint		•	•	•	•
vSphere Storage Appliance		•			
Fault Tolerance (1 vCPU)			•	•	•
Storage vMotion			•	•	•
Distributed Resource Scheduler and Distributed Power Management				•	•
Storage APIs for Array Integration, Multipathing				•	•
Distributed Switch					•
Storage DRS and Profile-Driven Storage					•
I/O Controls (Network and Storage) and SR-IOV					•
Host Profiles and Auto Deploy					•

Source: `www.vmware.com/files/pdf/vsphere_pricing.pdf`

This can really start to make sense when you move beyond consolidation and simple virtualization of your nonessential applications and focus on your mission-critical apps, which have greater demands and needs. Consider the demands and needs of virtualizing your MSSQL, Exchange, Oracle, and SAP servers. You may be able to get away with fewer than eight vCPUs in some of those application profiles, and you wouldn't want to be without vMotion, HA, Data Protection, and vSphere Replication.

As your environment moves beyond simple consolidation and into a full-blown, highly available virtualized architecture, it is inherently beneficial to review and identify what business needs align with feature sets available in only the more advanced versions of vSphere. In later chapters you'll find discussions of features that are only available in certain versions of vSphere. This will help provide justification to align your business needs to the feature sets.

Integrating Virtual Machines

As we dive into the points of integration of virtual machine scalability, we take a deeper look at some of the technologies that have made VMware ESX a market leader for the past decade. Three key aspects of that success have been the use of the technologies VMware vMotion, Distributed Resource Scheduler (DRS), and High Availability (HA).

Virtual Machine Scalability

We've touched on the topic of virtual machine scalability in various sections as individual discussions. The topic of how many vCPUs and how much memory to assign and how the licensable implications and limitations will drive that ultimate scalability has been discussed briefly. In a majority of situations you'll find the topic of your virtual machines needing scalability insignificant. Where this tends to rear its ugly head is when an application has needs beyond what you're able to support or provide.

Often scalability is a demand under the circumstances of misconfiguration or negligence, such as, for example, a database server with poorly written queries that execute far more work than necessary or a mailbox server designed and architected to operate with 1,000 mailbox users and is overallocated to have 5,000 users. In some of these conditions, throwing more resources at the server may help, but there is no guarantee that solution will solve the problem.

Whether you're provisioning guests at the bare minimum required for them to operate or providing them the maximum resources available in a single ESXi host, the tools are in the box to allow you to grow and shrink as your applications demand. Some of the tools that make that a possibility are explained in the following sections.

vMotion

VMware vSphere's vMotion remains one of the most powerful features of virtualization today. With vMotion, you can perform various infrastructure maintenance tasks during business hours rather than having to wait until the wee hours of the morning or weekends to upgrade BIOS or firmware or do something as simple as add more memory to a host. vMotion requires that each underlying host have a CPU that uses the same instruction set, because after all, moving a running virtual machine (VM) from one physical host to another without any downtime is a phenomenal feat.

VMware VMs run on top of the Virtual Machine File System (VMFS) or NFS. Windows still runs on New Technology Filesystem (NTFS), but the underlying filesystem is VMFS-5 or VMFS-3. VMFS allows for multiple access, and that is how one host can pass a running VM to another host without downtime or interruptions. It is important to realize that even momentary

downtime can be critical for applications and databases. Zero downtime when moving a VM from one physical host to another physical host is crucial.

Unfortunately, there is no way to move from Intel to AMD or vice versa. In the past, there were even issues going from an older Intel CPU to a newer Intel CPU that since have been mitigated with the introduction of Enhanced vMotion Compatibility (EVC), shown in Figure 1.8.

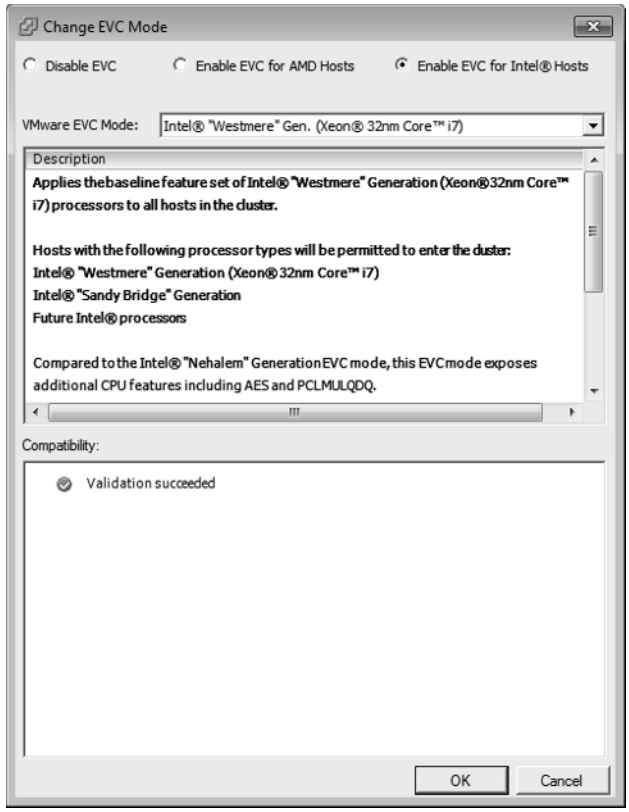


FIGURE 1.8
The Change EVC Mode dialog box

vMotion technology requires shared storage, but the virtual machine files do not move from that shared storage during the logical transition. If, for example, you have to change the virtual machine's physical location, you must first power down the VM and then "migrate" it from one logical unit number (LUN) or hard drive to another LUN or hard drive. Or you can use Storage vMotion, allowing the virtual machine to move between hosts and storage.

A caveat to vMotion is that traditional intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) may not work as originally designed. Part of the reason for this is that the traffic of VMs that are communicating with one another inside a host never leaves the host and therefore cannot be inspected. Virtual appliances are developed to address this concern. They have the ability to run side-by-side VMs.

Since uptime is important, VMware developed Storage vMotion so that the physical location of a running virtual machine's storage can be changed, again without any downtime and without losing any transactional information. Storage vMotion is very exciting because one of the reasons that virtualization is the hottest technology in IT today is the flexibility and mobility it

brings to applications in the datacenter (compared with running servers the traditional way in a physical environment).

There are other ways to leverage the technology. Virtual machines can be moved on the fly from shared storage to local storage if you need to perform maintenance on shared storage or if LUNs have to be moved to other hosts. Imagine moving a server with no downtime or sweat on your part by simply dragging a virtual machine onto another server in the cluster.

Made available in vSphere 5.1 was the ability to use vMotion without shared storage, with a few caveats and considerations:

- Hosts must be ESXi 5.1 or later.
- It does not work with DRS.
- It counts against the limits for both vMotion and Storage vMotion, consuming a network resource and 16 datastore resources.

Distributed Resource Scheduler

Distributed Resource Scheduler (DRS) helps you load-balance workloads across a vSphere cluster. Advanced algorithms constantly analyze the cluster environment and leverage vMotion to migrate a running VM from one host to another without any downtime. You can specify that DRS perform these actions automatically. Say, for instance, that a VM needs more CPU or memory and the host it is running on lacks those resources. With the automatic settings you specify, DRS will use vMotion to move the VM to another host that has more resources available. DRS can be set to automatically make needed adjustments any time of day or night or to issue recommendations instead. Two circumstances that often trigger such events are when an Active Directory server is used a lot in the morning for logins and when backups are run. A DRS-enabled cluster shares all the CPU and memory bandwidth as one unified pool for the VMs to use.

DRS is extremely important because in the past, VMware administrators had to do their best to analyze the needs of their VMs, often without a lot of quantitative information. DRS changed the way the virtualization game was played and revolutionized the datacenter. You can now load VMs onto a cluster and the technology will sort out all the variables in real time and make necessary adjustments. DRS is easy to use, and many administrators boast about how many vMotions their environments have completed since inception (see Figure 1.9).

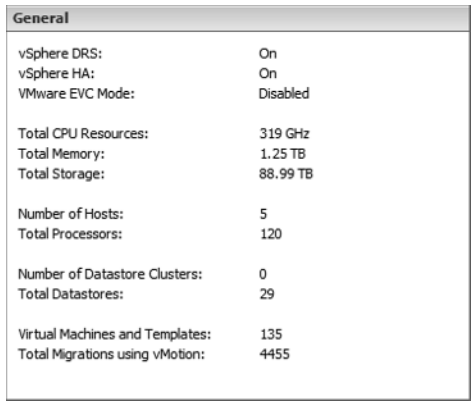


Figure 1.9
This depicts all vMotions, including those invoked by DRS

For example, let's say an admin virtualizes a Microsoft Exchange server, a SQL server, an Active Directory server, and a couple of heavily used application servers and puts them all on one host in the cluster. The week before, another admin virtualized several older Windows servers that were very lightweight; because those servers used so few resources, the admin put them on another host. At this point, the two hosts are off-balanced on their workloads. One has too little to do because its servers have low utilization and the other host is getting killed with heavily used applications. Before DRS, a third admin would have had to look at all the servers running on these two hosts and determine how to distribute the VMs evenly across them. Administrators would have had to use a bit of ingenuity, along with trial and error, to figure out how to balance the needs of each server with the underlying hardware. DRS analyzes these needs and moves VMs when they need more resources so that you can attend to other, more pressing issues.

High Availability

When CIOs and management types begin learning about virtualization, one of their most common fears is "putting all their eggs in one basket." "If all our servers are on one server, what happens when that server fails?" This is a smart question to ask, and one that VMware prepared for when it revealed the HA, or High Availability, feature of VMware Infrastructure 3. A virtual infrastructure is managed by vCenter, which is aware of all of the hosts that are in its control and all the VMs that are on those hosts. vCenter installs and configures HA, but at that point, the ESXi hosts monitor heartbeats and initiate failovers and VM startup. This is fundamentally important to understand because vCenter can be one of the VMs that has gone down in an outage and HA will still function, providing a master HA host, aka failover coordinator, is still available.

VMware recommends a strategy referred to as an N+1 (as a minimum, not an absolute), dictated by architectural requirements. This simply means that your cluster should include enough hosts (N) so that if one fails, there is enough capacity to restart the VMs on the other host(s). Shared storage among the hosts is a requirement of HA. When a host fails and HA starts, there is a small window of downtime, roughly the same amount you might expect from a reboot. If the organization has alerting software, a page or email message might be sent indicating a problem, but at other times, this happens so quickly that no alerts are triggered. The goal of virtualization is to keep the uptime of production servers high; hosts can go down, but if servers keep running, you can address the challenge during business hours.

Understanding Design Considerations

In this last part of the chapter, we go on to look into what you've learned in the previous sections and apply those principles to choosing a server and determining whether you ought to scale up or scale out.

Choosing a Server

When it comes to choosing a server, there is no right or wrong answer, but hopefully with a little guidance you can take the appropriate steps to end up with the best solution for your infrastructure. The question of reusing versus replacing your servers will often come up, and the answer can entirely depend upon the age, warranty, and capability of the servers you plan

to reuse. Thus, here are some cardinal rules to follow when it comes to determining your virtual architecture:

- Stay within the same CPU family or risk losing performance.
- Just because you have hardware to use doesn't mean you should.
- If it's out of warranty or not supported on the HCL, replace it.

Earlier we mentioned CPUs and CPU families in relation to EVC, and the discussion of CPU-aware load balancing in Chapter 4 will express the importance of CPU considerations when it comes to nonuniform memory access (NUMA). Keeping your CPU family the same will enable you to have VMs vMotion throughout the cluster without any complication or efforts on your part. By reusing older hardware, which may require EVC, you may be introducing more problems and need to troubleshoot more issues in your environment than if you had a more uniform virtual infrastructure. There is by no means anything wrong with reusing older hardware; you just need to consider whether the benefit of repurposing outweighs that of replacing it in power, cooling, performance, and space.

It is understandable if you found an old cache of 4xPort 1 Gb NICs that you don't want to go to waste, but given the choice of a single 10 Gb converged network adapter (CNA) or 10 1 Gb interfaces, for numerous reasons you should adopt the CNA. As you find yourself ripping out and replacing older infrastructure and servers, you'll find that your requirements for the number of cables and cards will greatly diminish. Physical servers that were required for resiliency of storage and connectivity to have a minimum of two 4xPort 1 Gb NICs and two 2xPort Fibre Channel connected to separate fabrics can now be replaced with a single pair of CNAs to provide both storage and network connectivity. Not only are the implications of power and cooling greatly reduced with fewer ports drawing power, but you also significantly reduce the number of cables required to be run and come out of your servers.

Last, if it is out of warranty or no longer on the Hardware Compatibility List (HCL), just replace it. Chances are, by the time that hardware has had the opportunity to drop out of warranty, or if it's somehow no longer on the HCL, it is not going to be a suitable candidate to be running your mission-critical infrastructure. Yes, it may be an acceptable fit for your lab (as we'll discuss in Chapter 3, "The Test Lab"), but this is not the time to try to get by with something you wouldn't relegate as a replacement for an otherwise equivalent physical workload.

Scaling Up vs. Scaling Out

Scaling up and scaling out takes the decisions you make choosing your server to the next level, partly deciding how many baskets you want to keep your eggs in but at the same time deciding how many different kinds of eggs you're talking about. You can ask 10 different people their opinion on whether you should scale up or out and you'll get 77 different answers. And that's perfectly okay; they're all completely right and yet likely wrong at the same time.

Whether to scale up or out really will fall into architectural decisions that you've made, haven't made, and do not have under your control. If you have many extremely high CPU and memory performance applications that require large amounts of both, you'll want to lean toward scaling up, whereas if your workload is pretty easily met and you have a good balance and load across your cluster, you may want to consider scaling out. It's important to consider that the more you scale out the more network and storage ports you'll need available, and if

your environment is full or nearing full, the costs of scaling out might outweigh the benefits versus simply scaling up.

All things being equal though, consider the example of two clusters, one scaled up (Figure 1.10) and the other scaled out (Figure 1.11).

FIGURE 1.10
Scaled up

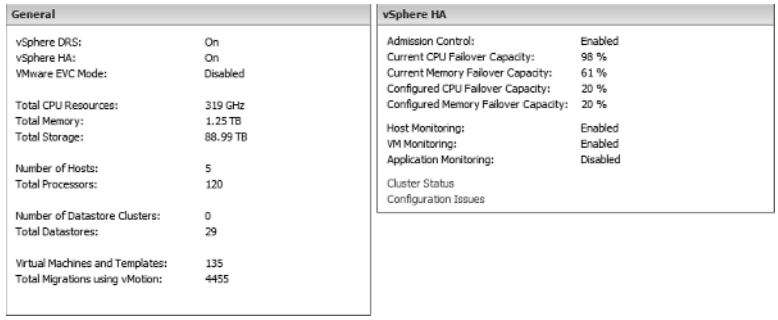
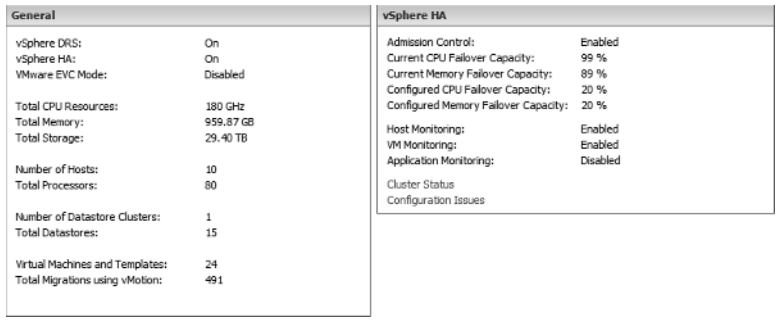


FIGURE 1.11
Scaled out



The availability of additional hosts provides a wealth of benefits, including having a greater percentage of CPU and memory failover capacity. While in this example the scaled-up system has nearly twice as much CPU resources, their memory requirements, due to the increased number of memory slots, enable having the same or an even larger pool of addressable memory.

Without truly understanding your usage profile, application use cases, and more, no one can make an educated decision about whether scaling up or scaling out is appropriate for you, though fortunately, today's availability of high compute and large memory systems will often mean you need not choose.

Summary

This chapter started out by discussing the implications of architecting and designing your virtualization environment for performance, but if you're not new to virtualization, chances are you build your environments with consolidation in mind. These two need not be mutually exclusive. In fact, what started out as an experiment in consolidating your infrastructure may

have accidentally evolved into a virtual infrastructure being used as a consolidation point, irrespective of the end state.

Hopefully, by this point you have a pretty stable foundation of what it will take to adapt your existing infrastructure or architect a new infrastructure to the point that it is capable of running your business's IT infrastructure. Over the next several chapters, we'll dive deeper into the specifics of what will make your infrastructure performance soar.