VMware® Infrastructure 3

Advanced Technical Design Guide

~and~

Advanced Operations Guide

Two books in one!

Ron Oglesby
Scott Herold
Mike Laverick

Chapter 8 - Managing the Environment

Often when you think of a chapter on managing an IT environment you envision a bunch of how to's and frequent tasks. In this book we are not going to go down that path. The second section of this book has plenty on configuring the alerts and setting up templates for VM deployment, etc. Instead we are going to focus on a number of items needed to manage you environment long term, such as:

- Monitoring and Alerting from an architectural perspective
- DataCenter and Resource Pool configurations
- Integration into change/configuration management
- Maintenance plans
- Costing/chargeback models
- Capacity planning
- Writing custom applications using VirtualCenter data

We see these as critical management components. While deploying VMs and setting up individual alerts are important, how you are going to architect the entire environment is the first step.

Monitoring and Alerting

It should be noted that while alerting is based (generally) on the monitoring tools in use, Alerting is a separate and equal partner in the design and should be treated as such. Monitoring (or monitor) in reference to computer science is defined as *"A program that observes, supervises, or controls the activities of other programs or systems."* So what are we observing or controlling? And how are we going to do that?

The first and most used monitoring program for VI3 environments is obviously VirtualCenter. VirtualCenter by its nature tracks performance of objects it manages such as VMs, Hosts, Resource Pools, and data centers. It is also a central

point of integration for most third party management tools that support ESX and in most cases use its data to report back to their consoles. You should also know that there are a couple of drawbacks to using VirtualCenter; first it is VMware centric. Meaning if you have other systems to monitor and are using other tools it doesn't know about them. Second it has been known to have some odd database/statistics issues. For a while VC would miss collecting data on ESX servers, during these "blank periods" it would put zeros in the data base for the missing samples. Of course when the data was reported on via the console or rolled up for weekly or monthly averages, it would average in all those zeros bringing the utilization average down and making the long term performance data almost useless.

Of course, the upside of using VirtualCenter is it is VMware centric and knows about / can see into the host level resource utilization which is the most accurate information you can get in the virtual world.

The time-drift issue…

One of the reasons you want to use an ESX/VMware aware monitoring too`l is the ever present time drift in a VM. To explain time drift we need to look at how counters work at a basic level. Essentially there are two types of counters, those that rely on time, and those that don't. To differentiate between the two at a counter level is pretty easy, anything that references X per second, or a % of utilization is time based. As an example, System Context Switches in Windows OS shows the number of Context Switches per second. Or Total CPU generally referenced as % of CPU utilization is also a time based essentially showing the number of cycles used as a % of the available cycles per second. In contrast non-time based counters such as Memory - Bytes Available, is a flat number. It tells you at any given point in time how many bytes are available but is not based on what happens second to second.

So why is all of this important? Well, the counters that are based on time are essentially based on the system clock of the VM. And the system clock is based on the number of ticks the VM sees from the processor. The issue comes along when a VM is de-scheduled from the processor and how the VM sees the "real" processor underneath. Meaning the VM sees a 3GHz processor and expects to see 3GHz worth of processor ticks every second either from active processes or the system idle process. When a VM is de-scheduled from the processor to al-

low other VMs to run, it essentially looses processor cycles and therefore the clock in the VM is not correct (at least at a sub-second level). So to the VM a second may be a second, or it may be a second and a half. And if this is true, then a counter that reports X number of IOps or Y number of context switches is not 100% accurate. So the very scheduling mechanisms that make it possible for multiple VMs to run on a single processor also have an effect on our monitoring.

Now this is not to say that high utilization still isn't high… As an example a VM may be reporting 95-100% CPU utilization. But if it is competing for time on the processor it is executing, it may be seeing 100% of the CPU it is allocated, but only really using 50, 60 or 70% of a physical CPU. Of course if a VM is reporting that it is about 90% over 3 minutes of course there is an issue whether the clock is right or not, but from a capacity planning perspective it may be irrelevant.

Confusing huh? Well let's get into some design options to understand what we are monitoring and why.

So what needs to be monitored?

Essentially you need to start from the bottom up and work from the hardware level, through the host software (ESX Server) then up to the VM and potentially inside the VM itself. We'll look at each of these individually and what you get from each level.

Hardware Monitoring/Agents

To use a common example I will talk about HP hardware here. Often HP customers will use SIM or System Insight Manager, to monitor their hardware. The SIM agents get installed into the OS and monitor everything from CPU Fan speed, to power supplies. They then report this back to a central console. The thing with SIM is that it is hardware specific; don't try to use this on Dell. But Dell has its own software as does IBM. But it requires that the OS you install on has direct access to the hardware, which the VMs do not. So in our ESX environments we need to install agents into the ESX Service Console.

Anyway, what you want is an agent that you can report hardware failures or pending failures, etc. As of this writing the supported ESX agent levels for major vendors are as follows:

- HP Insight Management Agents: 7.5.1.A
- Dell OpenManage Agent: Version 5
- IBM Director Agent: 5.10

All of these vendors do a pretty good job of integrating with ESX these days. In almost every environment you will want some type of hardware monitoring since the VMs themselves cannot see the hardware directly and are unaware of it and running without it could be dangerous seeing how you just stacked 20 or so VMs on a single piece of hardware.

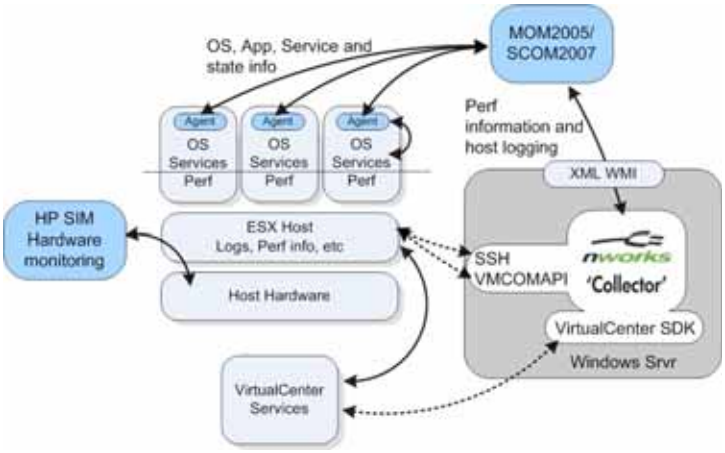
Host Monitoring

The next level of monitoring we need is at the host/ESX level. This is often handled by VirtualCenter in new environments. VirtualCenter will give you a pretty good look at an ESX host and does a decent job for daily operations. For long term capacity planning it is pretty much useless. But for host monitoring (is the processor utilization high, is memory over-committed, etc) it is pretty good. Its big lacking piece (as of VC 2.0.2) is a good way to handle alerting. Since VC's inception the ability to set alerts based on duration has not been there. Instead the alerts are based on a single threshold concept. As an example let's use processor usage:

Assume you want to get an alert when host processor usage is over 90%, but you don't want to catch spikes in utilization, you want 90% over a period of 5 or 10 minutes only. This is a dual threshold alert, it has to meet both a certain level of utilization AND for a certain number of samples/certain amount of time. The issue with Virtual Center is that it uses single threshold alarms. Meaning that it sets a threshold of say 90% on CPU, and one sample is 90% you get an alarm. Kind of a pain on the host, but very much a pain if you apply this to VMs. VMs are spiky by their nature (just like normal servers) and alerts triggered this easily can be annoying, but there is a solution, have VirtualCenter feed a more refined alerting system like MOM, Director, or HP Openview.

In most cases enterprise customers will already have a management tool in place something like Microsoft Operations Manager (or System Center as it's now called) or an HP Openview. The benefit of using one of these systems is that they tend to have more refined alerting systems that you may need, and it will limit the number of places that an alert will come from. Of course to use one of these you need a go-between type of system that will feed true ESX data to your monitoring system.

Figure 8- 1: Host Monitoring



Here I use nworks as an example. Nworks is one of the better known solutions that a number of environments use. It makes a go between collector for both HP Openview and MOM. Here we show a typical nworks setup where it collects information from VirtualCenter and feeds it back to the MOM console. The nice thing about this setup is that you get both the VirtualCenter / Host information but in the same console you can see you non-VM systems and have agents for apps and services inside the Virtual Machines.

Notice that the collector in the image uses the VirtualCenter SDK. This is the most common model for VMware partners building tools and add-ons. From a host perspective it can connect via the API or directly to the host via SSH. This type of software or something like it is an absolute must in any large environment.

Virtual Machine Monitoring

Finally you will want to watch your virtual machines at some point. In smaller shops often there is no host monitoring in physical servers, so they leave VMs the same way. But if you monitor your physical servers now for items like Up/Down state, free disk space, processor usage, service state, etc. you will probably want to do the same for VMs so lets look at a few important options.

Service state, Up/Down, and applications

Anything like this you can continue to do as you normally do. If you use something like MOM or NetIQ to monitor your AD or Exchange servers and are now making them VMs go ahead. If you have an agent you use to monitor service state or up/down of the server, keep them! They work just fine. No changes are needed. Granted you may see some odd things… I had a SQL guy tell me once that his read time on some obscure counter went from 3 ms to 4.5 ms average… I asked if performance changed for the end user? No, actually he didn't even know if the counter was important. Of course it was a TIME BASED counter so it was probably skewed, but that is another story. For general application stuff, keep them running.

Perfmon type metrics

Of course for serious counters/metrics you have to be a little smarter. Like we talked about before, time based metrics are inaccurate. Not completely wrong mind you, just not 100% right. As an example; I had a client that commonly monitored for servers at 90% cpu for longer than 5 minutes. While within the VM that counter might not be perfectly accurate from a whole CPU perspective, if the VM was using 90% of what it had access to it was still a lot. And if that was not normal they wanted an alarm triggered. It worked just fine, in some cases batch jobs would stall and spin up a process and it would signal the operations team so it still worked. But for capacity planning or very accurate trending time based metrics should not be relied upon.

For non-time based metrics, go ahead and set them up. Most VMware users will still monitor for things like disk space in use (it shows the logical space inside the VMDK) or free memory etc. To list each counter that works or not here would be a 10 page table. So instead remember the time based issues and make decisions intelligently.

Resource Pools

While this may seem an odd place to cover it, I wanted to touch on these items where it seemed most appropriate. Resource pools are used to manage resources in a cluster and often times throttle resources for different types of VMs. In most cases they are implemented over aggressively and tend to hurt the environment, hopefully we can put them into their proper place here and help your long term management of the environment.

Basically Resource Pools are an extension of the existing Share and reservation systems that VMware built to control resource allocation to VMs. Only when using Resource Pools you are grouping the VM together to share resources instead of applying these settings on an individual VM level (as was done with previous versions of ESX). So it is possible to cap or "limit" resources or to guarantee a minimum or "reservation" of either of resources (specifically CPU or memory) to a VM or a resource pool. Alongside these unchanging and hard-coded limits or reservations, we can have a more dynamic control over VM's usage of CPU, memory or disk. We can use VMware's proportional "share" system which responds to changes in resource demand relative to each VM and ESX host. At an extreme level it is possible to peg a resource to a VM with such features as CPU affinities – hard-coding a VM to have exclusive access to a CPU.

Technically, I really should mention that VMware Distributed Resource Service (DRS) is an invaluable resource management tool. DRS integrates very closely with VMware High Availability (HA) so it was decided to show these two products together in the "High Availability" chapter of this book.

During this chapter I want to use an analogy to explain how resources are managed in ESX. This analogy is about an airline. Each plane will represent an ESX host, a fleet of aeroplanes a VMware DRS cluster. Every plane has fixed capacity in terms of the number of passengers it can comfortably accommodate. Each of the "seats" will represent a VM. If the airline has 200 seats per plane and ten planes that is a total of 2000 seats. But the 200 seats per plane do represent a fixed limit on capacity. Fundamentally, though these 2000 seats exist as a logical capacity, a passenger can only board one plane – just as an VM can only currently execute one server, not a cluster. Access to the plane is controlled by unique system of reservations which is intended to guarantee not that you fly – but that a certain quality of travel like business class will be provided. In con-

trast, economy passengers do not make reservations. They merely turn up and hope there will be capacity to fly.

Setting Limits

As stated a moment ago it is possible to impose limits upon a VM or resource pool. This can be done for CPU resources (by Mhz) and for memory (by MB). In fact, limits are imposed on VM from the perspective of memory when you first define a VM. When you create a VM you set the maximum amount of memory it can have during the "New Virtual Machine" wizard. If a VM demands more memory it cannot exceed this amount allocated to it – even if free physical memory exists. At first glance this seems quite restrictive, however, from an architecture point of view, it has to happen. If we have poorly written applications and operating systems we would want to avoid the situation where a VM was able to drain a physical host of all its available RAM – thus causing a reboot of the physical server. If we use our analogy it's not possible for a single passenger on the plane to take up all the space at the expense of the other passengers. Additionally, there are some fixed limits we cannot exceed. If there are only 200 seats on the plane we cannot allow 201 passengers to board. Similarly, if the ESX host or pool resources are totally consumed – a limit has been reached. We cannot magicly out of nowhere find additional resources- with one exception. When all memory has been depleted is possible for a VM to use its VMkernel swap file.

In contrast there are no default limits on CPU usage. If a VM demands CPU time, and that CPU time is available, then this is allocated to a VM. One reason to cap or limit a VM usage of CPU could be because you know you have a poorly written application that regularly crashes. Perhaps when it crashes it "hogs" the CPU of the ESX host. Using CPU limits is one method (amongst many) to control these kinds of VM's.

Lastly, a word about terminology - in the ESX 2.x product the word "maximum" was used instead of the word limits. VMware changed the phraseology to "limits" as this is more meaningful and more clearly describes the feature. Additionally, VMware has moved away from allocating limits on CPU's by a percentage value and now prefers to use Mhz. Mhz is a more accurate measure of CPU usage than percentages, which can be very misleading. In the context

of CPU's 10% of a 1.44Mhz processor is decidedly different from 10% of 2.6Mhz processor.

Setting Reservations

In addition to limits, we can also use CPU or memory "reservations." One analogy that can help when thinking of reservations is to compare them to the plane reservations. These reservations are supposed to *guarantee* a resource in the form of a seat. Similarly a CPU or memory reservation in Mhz or MB is intended to *guarantee* the resource to the VM or resource pool. In this case there are no *default* reservations for CPU or memory, unlike the default limit on memory set when creating the VM. We can regard these reservations or guarantees as offering us a way of ensuring we meet certain performance levels. Perhaps you could even regard them as a way of meeting "service level agreements" (SLA).

Just like with a plane or hotel you must "meet" your reservation in order to board or check into a hotel. So, if you say that the memory reservation on VM is 256MB of RAM – and that amount of RAM is physically not available – you will be unable to power on the VM. VMware refers to this as "Admission Control." Similarly, if we configured a situation where a VM must get 1000Mhz of CPU time, but the physical host can only offer 500Mhz of CPU time – that VM would not power on.

Using our airplane analogy, if a passenger makes a reservation for 10 seats in business class those seats must be there. Our airline imposes a very *special* definition of customer care that states if a business class reservation cannot be met – rather than pushing our customer in economy class, we refuse admission to the flight. Here we can see the weakness of all analogies, they don't work perfectly in all circumstances. Nonetheless, I want to persist with this analogy as it is helpful in most cases.

As with CPU's in ESX 2.x VMware used to call "reservations" minimums. Again, the label change was introduced to be more meaningful.

Putting the concerns to one side for a moment there is very interesting and useful relationship between memory limits and reservations and the VMKernel

swap file. The difference between the reservation subtracted from the limit – determines the size of the VMkernel swap file. The usage of the VMkernel Swap file was mentioned in the previous chapter as an indicator of potential performance problems – but here I wish to delve a little deeper into different ways of using it. Explaining how to use the VMKernel swap file is perhaps best done with a couple of examples.

Example 1: Difference between Limit and Reservation

I had a VM with a 512MB limit and 256MB reservation – on power on the VM would create a 256MB VMkernel swap file (512-256) and guarantee that the VM would receive 256MB of RAM.

Example 2: No difference between limit and reservation

If I set the limit to 512MB and the reservation also as 512MB – and powered on the VM, ESX would not create a VMkernel swap file at all. It would run the VM entirely in a memory reservation of 512MB.

Example 3: Big difference between limit and reservation

If on the other hand the VM was given a 16GB limit, and the default of 0MB was used for the reservation – a 16GB VMkernel swap file would be created.

With example 1 if I had an ESX host with 2GB of physical RAM I could run at least 8 VM's before running out of memory (2048MB/256MB). I would not be able to power on a 9th VM because there would be insufficient memory to meet the reservation guarantee. If all the VM's simultaneously wished to use memory up to the limits (512MB*8) I would find I would get swap activity. What I hope is that this would be such an unlikely event – that it would be "safe" to configure the system this way.

Perhaps you find this "memory over-commitment" a bit scary. You are concerned about the negative aspects of swap activity, and you wish to have a cast-iron guarantee that your VM's will *always* run in memory. If this was the case you could use example 2. By setting the VM's limit and reservation to be the *same* value no swap file is created – and the VM is guaranteed to always run in

memory. However, on a 2GB system the effect of this policy would be very significant. I would only be able to run 4 VM's not 8 (2048MB/512MB). If I tried to create a 5th VM and power it on – it wouldn't, as all my memory would have been reserved for use by my other VM's.

I could imagine example 1 being configured by someone who is optimistic and is looking for very high VM to ESX host ratios or someone trying to run as many VM's with as few resources as possible. Alternatively, we could see example 2 as someone who is perhaps pessimistic or conservative, or someone who has so many resources there is no need to use the VMkernel swap file.

The last example, example number 3 is a warning. If you set extremely high values on memory, with no reservations – the ESX host will generate an extremely large swap file. Just like with memory reservations, you need the physical MB disk space to create the VMkernel swap file. In example 3, if you didn't have 16GB of free disk space in the LUN where the VM is stored, it would not power on as there would be insufficient resources to guarantee the *difference* between the limit and reservation.

The Share System

Between the gap between limits and reservations another system is at play. It is called the "Proportional Share" system. Shares allow you indicate that when a resource is scarce that one VM or resource pool is more important than another. To use our analogy it's like the airline treating a Hollywood star more importantly than the average guy on the street. Share values can be applied on per-VM basis or on resource pools. Unlike limits or reservations which are fixed and unchanging – shares on the other hand react dynamically to resource demands. The share value can specified by a number (usually in multiples of 1,000) or by user-friendly text value of normal, high, and low. The important thing to remember about the share value is that it is only relevant when the resource is scarce and contention is occurring. If the resource is plentiful or VM's do not have to compete over resources – the share value does nothing at all.

In my discussions with VMware I've been told that many customers do not use the proportional share system as much as VMware might like. Why might this be? Firstly, because customers frequently don't understand how shares work, and secondly, because shares only take effect when things are performing badly

– a great many people try to configure their ESX hosts and VM's so this never happens. Personally, I am a big fan of the shares system. What especially appeals to me is its dynamic nature, its ability to react to changes.

If you or your customer is still struggling with the concept of shares you might like to try a couple of other analogies. You could see the share value like shares in company that are quoted on the global stock exchanges. The amount of shares a company chooses to issue is up to it – what matters is the number or portion of your shares. The greater amount of shares you hold in a company the more of its resources you own. Therefore a big share owner of 5000 shares has much more influence than a shared holder with just a thousand shares. Perhaps you own as much as ¼ of the company or 25% of its shares.

Here's another analogy I use regularly on courses. Imagine you have 3 children – one is a baby, the next a 5 year-old and the last a teenager. When you come home after a hard day's work they all demand your time. Here your time is like the CPU, and each of your children are pesky VM's making demands on your tired brain. Being a particularly cruel parent you decide to take a permanent marker – write 3000 on the baby's forehead, 2000 on the toddler's forehead, and lastly 1000 on the teenager's forehead. You decide you're like one of the ESX hosts you manage at work, and this is your parental strategy from now on!

In this scenario when you are faced with contention (say when you come home from work) you would give the baby ½ (3000/6000), the toddler $^1/_3$ (2000/6000) and your teenager just $^1/_6$ (1000/6000) of your valuable time. Now when it's mid-evening you decide that the baby is tired enough to go to sleep. You're in luck tonight as he's out for the count in seconds – you now can give $^2/_3$ of your time to the toddler (2000/3000) and $^1/_3$ (1000/3000) of your time to the teenager. When the toddler goes to bed (after much crying and wailing normally) you are facing no contention at all. Just as you're settling down to watch your favorite sit-com the teenager comes down from her bedroom – perhaps the internet connection has failed or her games console has broken. She now decides this will be an opportune time to discuss why college is a waste of time and how she should really follow her favorite drug taking band around the country. Now you can give all of your time to teenager – 1000/1000 – in persuading her that while a life of drunken debauchery might have its appeal, it won't lead her to a prosperous career in IT like yours. Finally, everyone goes to bed – contention is over and you get the opportunity to get some well-earned z's. But then at 3 a.m. a sound is heard from the baby's room which grows into

crying. You're out of luck, and it is your turn and not your partner's to feed the baby. However, rest assured as long as you get to the baby quickly, it will not wake the others – and you are able to give 100% of your time to getting back in bed as quickly as possible!

All joking aside, the analogy does illustrate some points. Firstly, that share value adjusts depending on the level of contention. Secondly, that when there is no contention the share value does nothing at all. Thirdly, that when you become a parent you will have no time to yourself whatsoever!

Lastly, you should know that there is another way of setting the share value – which is by using friendly labels of "High, Normal, and Low." These offer novices a more intuitive way of dividing up resources. You will have seen these whenever you create a new VM. You can use these text labels on a VM and also in resource pools. If you are going to use these you should know what actual settings apply.

High
Allocates 2000 shares per virtual CPU

20 shares for every 1MB allocated to the VM

Normal
Allocates 1000 shares per virtual CPU

10 shares for every 1MB allocated to the VM

Low
Allocates 500 shares per virtual CPU

5 shares for 1MB allocated to the VM

As you can see, high is twice as much as normal and four times as much as low. There is also another assumption at play here. VMware assumes that the more memory you assign to a VM the more sensitive it is to a lack of memory. So when contention takes place the VM "wins" a greater slice of memory re-

sources. This assumption might not always be the case (although it frequently is). You could have a memory intensive application that is not business critical.

CPU Affinities

One extreme method of controlling the VM's access to CPU resources is to "peg" it to a specified CPU. As was mentioned in the previous chapter, internally to physical server the VMkernel dynamically moves the VM across to work on the best CPU inside the ESX host. We can switch off this feature using CPU affinities on the properties of a VM. I would this regard this configuration as a last resort. Firstly, configuring it is very administration intensive. Not only to have to configure the VM in question to use only CPU3 for example, you also have to configure every other VM *not* to use CPU3 – to truly dedicate a VM to given CPU. Secondly, CPU affinities are incompatible with VMotion. Thirdly, as DRS is effectively an automated VMotion for performance – CPU affinities also break DRS. Removing CPU affinities on a VM running on a ESX host which is already a member of DRS cluster is possible, but very convoluted. Therefore, I consider CPU affinities a very last resort.

Resource Pools or VM Settings

There are two main ways to apply many of these settings – limits, reservations, and share values. Resource pools only affect CPU and memory resources – so if your goal is to control disk and network activity, these must be done on the properties of the VM or a vSwitch respectively. Despite this limitation, CPU and memory resources are very critical, and resource pools offer a much more effective way of applying limits, reservations, and share values. Right-clicking each VM and setting these values is by its nature very administration intensive, whereas dragging and dropping a VM to the correct VM is an easy task. If you are trying to calculate the total share value it is easier to compare a small number of resource pools, rather comparing a large number of VM's.

Resource pools can be live in two main places – hanging off a stand-alone ESX server which divides up the resources of a single host into smaller units or pools, or alternatively, resource pools can be created on VMware Cluster which divides the total resources of many servers into pools. Once you got the concept of resource pools and the way they function – they are the same if they are on a stand-alone or a cluster, but they really come into play when applied to

clusters where resrouces needs and workload movement become more important. You could create resource pools based by department (sales, accounts or distribution), function (web servers, database, or file servers), or by IT Infrastructure (test, development, production).

If we apply our analogy to this we can see each of the airplanes as representing an ESX host. A DRS cluster represents the collective capacity of all my airplanes, but fundamentally a passenger can only fly on one airplane – not on two simultaneously. Our air-traffic controllers are very clever guys who can detect some airplanes stretched to capacity and move passengers from one plane to another while they are flying. This would be called VMotion in ESX.

In the Operations Guide section of this book much of this information will be repeated. We feel this topic cannot be stressed enough as these "settings" are often misunderstood. As with all of our information if you want specifics on how to configure these settings, go ahead and jump forward to the Operations Guide.

Integration into Change/Configuration Management

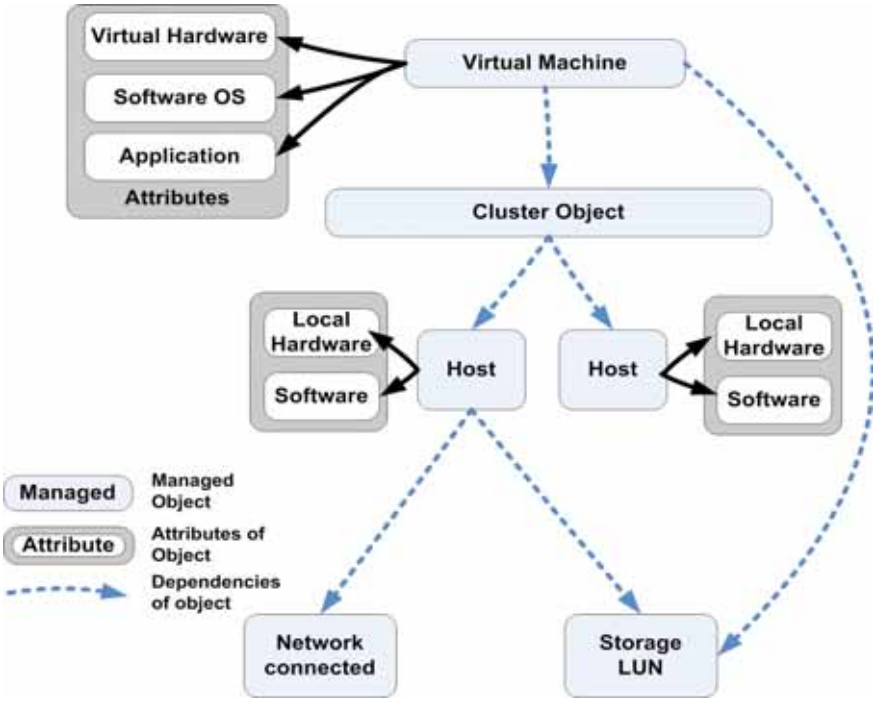
One of the big items forgotten in most designs is how to integrate the new virtual environment into existing change control or configuration management systems. Virtual Machines are sometimes treated like physical servers, which can limit the inherent abilities of the new system, or the new technology creates 'fear' that makes the change control for the environment overly conservative. In either case it is really a lack of understanding that causes this. So here we will point out common functions and where they often fit in change control procedures. But before we do this we will first overview the idea of a cluster and Virtual Machine from it components perspective for configuration mgmt.

Configuration Management

Configuration management requires an understanding of the entire system, of what components are dependent on each other, and how they are all interre-

lated. Below I have an image showing cluster from a component overview. Let's take a look at this:

Figure 8- 2: Configuration Management



Basically a VM is not directly dependent on a specific host. From a configuration and change mgmt perspective it is really dependent on a cluster, this in turn is dependent on the hosts that are part of that cluster. Finally the hosts are dependent on the storage and networks they are connected. The So the VM is really not dependent on any specific host. So from a configuration mgmt and even change control perspective you can see the logical links being created in the environment. If you want to shut down one host, and not the whole cluster, you can do this without interruption to any VMs.

From a configuration management perspective this diagram is interesting since it shows all the levels in the VM environment and samples of the attributes you can track. Let's look at a few specific items/objects and their possible configuration items you want to track:

Virtual Machines

From an individual VM perspective you no longer need things like asset tags. This is sometimes a bit confusing as asset management databases often are tied to configuration management, so you may have to find a workaround like using the logical cluster name as the VM's asset tag. Anyway, some of the attributes we commonly see in configuration databases for VMs are:

- VM hardware configuration
- Guest Operating System information
- Storage/LUN VM is stored on
- VM cost center/info on VM charges (see the VM cost section later in this chapter)

Obviously the OS is something used whether physical or virtual. The biggest change is how hardware is account for, asset tags and of course the Storage the VMDKs are located on. This is often used to determine impacts of changes to do with SANs but can be very useful once your environment is large and you do storage changes, migrations, or upgrades.

Hosts

On the host these are often labeled as infrastructure like domain controllers or name servers. In addition another issue with configuration information is how some network teams require that you associate an IP or specific host with a specific network switch port. Of course on the hosts you may have several nics that have 10 or 20 VMs on them. Additional items you may see for the hosts include:

- Local hardware config and asset information
- Software versions and patch level
- Rack information
- Cluster membership
- Network port/switch information
- SAN port / switch information

Another alternative is to create a cluster object in the config management database, and manage the hosts as part of that. Personally I like keeping the hosts separate, and making their cluster an attribute of each host.

Network and Storage

Network and Storage attributes often don't need any additional information for VMs, but instead need a change in how servers are added. As mentioned previously often network and SAN teams will require individual host information or IPs to assign the port to your servers. SAN teams are usually ok with the concept of multiple servers connected to a single lun (from other cluster technologies) it is the Network team that often sees this as a foreign concept so make sure you cover it.

Change Management

One of the most frequently ask questions I get from clients around change control is "Should we do a change request for a VMotion". I find this funny because of all the new things going on in the environment and all the moving parts, the one that gets the most attention is VMotion. Of course I understand this since it is often a pretty mis-understood piece of technology, but also I try to point out there is a lot more to worry about than VMotion. In the VirtualCenter and cluster design chapter we went over DRS. And if you plan on using that it pretty much rules out change controls for VMotion, so let's look at some items in the environment may need change control, either from a pre-approved stand point or a change that may require more review.

Below is a list of common changes that you will need to consider (that are new to the environment):

- VMotion – Movement of a VM between hosts, no downtime, no change in storage – no impact to VM
- Host configuration change – Prior to failure or maintenance, uses VMotion to move VMs to other hosts, so that the originating server can be repaired (maintenance mode)- no impact to VMs

- Patches for ESX hosts, host hardware maintenance, etc.- no impact to VMs if VMs are moved first
- DRS – Automatic load-leveling on hosts using VMotion, could occur daily or hourly, no downtime for VMs or hosts
- Cluster change – Addition of LUNs, no downtime for VMs, rescan of storage
- Cluster change – Removal of LUNs, only affects the VMs stored on those LUNs
- Cluster change – Upgrade of hosts – Potential impact major – Downtime of VMs not always required – VMTools upgrades (during major host upgrade) requires VM restart on installation of new tools
- Addition of host to cluster – No impact to VMs.
- VirtualCenter updates – no VM changes, but possible loss of access to VMs via Virtual center

Maintenance Plans

Maintenance plans can be as varied as ESX or Windows Server builds are between different companies, but the common thread is there has to be one. Organizations without a minimal maintenance plan will often fall into the "I'm two service packs and one major rev behind" problem. With that known we believe in using weekly and monthly maintenance plans for ESX environments. For the most part these servers just run, but you still need to check things out / clean things up as needed.

Weekly Maintenance Tasks

For your VI 3 environment I would recommend starting with at least 5 and possibly 6 weekly tasks. These tasks are basics that review the existing environments logs, check volumes and look for open snap shots. Here is my basic list:

- *Review host logs for each server and note errors or possible issues for troubleshooting*. Review forums and VMware documentation for any errors to determine if this is a benign error or something that needs correcting.
- Review VirtualCenter logs for issues or errors.
- *Review VMFS volumes for space in use and available capacity*. Any volumes with less than 10% available space should be looked at further and VM deployment should be stopped for that lun. (can be scripted or viewed in VC)
- *Review environment for VMs with open snapshots*. Snapshots that are not applied over time can grow huge amounts and cause performance issues and lock ups on the VM when applied. Weekly reviews will ensure snapshots are not 'forgotten'. (Can be scripted)
- Check local drive space on hosts for partitions filling up. Clear log files or temporary storage as needed.
- If you have "temporary VMs" used for Test/Dev check them to determine if they can be decommissioned and are not forgotten about and left to run forever.

A number of these tasks can be automated for the environment. Items like checking for disk space either locally on the hosts or on the VMFS volumes can be done using a shell script to output to a text file, then these files can simply be reviewed. How you accomplish tasks like this is not important, the importance is in actually doing them. Hopefully you can start with these tasks and add your own. The next section looks at a monthly checklist that focuses more on capacity planning and upgrades, but is no less important.

Monthly Maintenance Tasks

Some organizations will change this schedule to a quarterly schedule. Personally if it is done monthly I find that environments stay well up to date. For some reason when these tasks are moved to quarterly they are pushed off more often and or forgotten about for a while. By establishing a routine for the first or last week of the month, these tasks will happen on a regular basis and keep your environment in good shape.

- *Create a capacity report for the environment and distribute to IT and Mgmt*. Capacity plans are detailed more in the next section of this chapter. But they are imperative as your environment grows.
- *Update your VM Templates with the latest hotfixes and patches approved for the environment*. This will help to ensure that your templates are always current and when deployed need little work done to get them online.
- *Review the VMware website for new patches or fixes for your infrastructure*. You should determine if you need the patches being deployed. Security fixes are often a must, but if the patch fixes something with linux guest tools, and you have no linux guests, it's something you can wait on or not deploy at all. Make sure you document WHY you need or don't need to deploy a specific patch, this will help with audits.
- Review patches for VirtualCenter or other supporting services like monitoring tools used in the VI environment.
- Test and deploy updates as needed.

Again, you may decide to add or remove items from this list or may find that you have special tasks that need to be done. I have a client that has 4 sensitive VMs in their environment that must be tracked constantly. They have added a monthly task here, to check the VM hardware specs, review its audit logs and the logs for the host they are on.

VM Costs/Chargeback

We have been deploying VMs on ESX for clients since 2003. In the past 4 years on thing has become apparent in each environment I have visited. The environments that work out their VM cost/chargeback strategy early on tend to be very successful very quickly, the ones that don't tend to run into capacity issues or over spend (not meeting ROI targets) very quickly.

I think the reason for this is twofold; 1: clients that implement a cost/chargeback solution often have their act together in other areas and this just happens to be one of them and 2: those that don't implement a cost model

often run out of capacity and run into performance and capacity issues since "VMs are free!!!".

The reality is simple economics, and while I know we in IT don't like to think like this since it gets finance involved, to be successful you have too. Look at it this way, if a project has budgeted $30K for hardware, $50K for software and $100K for consulting to implement a new piece of software, then you as the VMware guy give them 7 VMs for "free" so they free up that 30K, do you think they are just going to transfer it to you later for your next host? Nope, they are going to spend it on something else. Then when the VMware environment runs out of capacity and there are no more funds for servers or storage you are left holding the bag.

Cost of the VM

It really doesn't matter which one you implement, it's just important that you pick one. Based on your organization and the IT finance model you already have, this decision may be a foregone conclusion. But in either case, the cost model should include the following:

- Server cost
- Rack space
- UPS/Power distribution
- KVM
- ESX /Virtual Infrastructure licensing costs
- Supporting software costs (like 3rd party monitoring tools, or tools like ESX Ranger)
- SAN / NAS disk cost (if you are storing your VMs off the ESX host)
- Fiber and Network switch ports

From these basic numbers you can calculate an average VM's costs. Basically if you know all the component costs in a system, then the number of VMs you are going to host per server, simple division will give you an average VM cost (some organizations call this a slice).

The interesting thing in this model is that you have several ways to cost out the VMs to your customers. You can come up with an average cost, know that some VMs will have more disk space, or dual processors etc, but still charge everyone the same amount. This will result in a basic cost model, but seems to me to be a little "socialist". I mean if everyone is charged the same amount why would an application owner not ask for dual proc 4GB VMs? They would. So while this model can work, I believe in "up charges" for VM hardware upgrades.

To me you should attempt to drive the end-users behavior by costing/pricing VMs appropriately to their hardware allocated. This (like purchasing physical hardware) helps to drive them not to over allocate resources. Of course this model can also become very complex. The idea of up-charging for a dual processor VM over a single Processor VMs sounds nice, but its not as simple as doubling the price.

Let's assume you 'charge' $1000 for a single processor 512MB of RAM VM with 8GB of iSCSI based storage. Then a user asks for a dual processor server with the rest of the configuration the same. Essentially you have doubled one resource (processor) but changed nothing else. The proper way to do this is to understand your cost of ram, processors, storage space etc. That way when the next app owner only wants to go to 1.5 GB of RAM on a single processor system, you can properly charge for the 1 GB upgrade.

If you have a mature SAN environment in your organization you may already have a model like this in house. Storage is a shared service (like VMware) that has many options. Thirty GB of disk to the end user could be a mirror on the backend (60GB) or a RAID5 with a hot spare, or it could be replicated off site or mirrored locally based on user requirements. In any case the pricing reflects the actual hardware used and does not just double or triple based on simple sizing changes.

Finally, another reason to charge for VMs is to help stop VM sprawl. About 2 years ago Scott and I made a point to start pointing out that VM Sprawl was going to be an issue because; 1- VMs are so cheap, 2 –VMs that are free have no reason to be decommissioned, there is no incentive for app owners to go back and shut them down, thus continuing the cycle of using resources for no good reason.

In any case getting the right amount of money for the VMs and their underlying supporting infrastructure is extremely important. Most organizations handle VM purchases via inter department transfers. The best ones have accounts just for ESX shared infrastructure. The reason for this is to ensure that the money used to purchase VMs today, is allocated to purchasing the next set of capacity for tomorrow's VMs.

One-Time Cost vs Chargeback

We often get into long debates about 'chargeback' within client organizations. Some IT environments do not have a chargeback system and therefore see any type of IT "charge" as a chargeback, and thus push back strongly on charging for VMs. Frankly, whether you do an upfront cost or a recurring chargeback is up to you, but the reality is that you must do it for this technology in larger organizations.

One-Time costs are nice as they show the end-user a cost reduction right away (you would have bought this Dell PE 1950 for $5K, but instead here is a VM for $1.9K). The issue (long-term) with this model is that it never expires. Meaning someone buys a VM in 2004, by 2007 the lease is up on your hosts and storage and you are ready to refresh. Well, if the VM doesn't have something like an expiration date the end-user has a never ending upgrade path for free. So it is imperative in the upfront models to have a time out/drop dead date for VMs to ensure that you can handle/pay for future refreshes.

Chargeback on the other hand is perfect for VM environments. Whether you charge monthly, quarterly or annually, someone is looking at that bill every time and is driven to keep costs down (as we all are). Chargeback models ensure that the app owners or departments continue to pay for that system and continue to maintain the hardware and software (through paying for the service).

You can make either system work for you, the trick is that you have to have a cost model for your VMs early in your adoption to be successful. The cost model should be built in such a way that it motivates users not to over allocate resources, and pushes them to help keep VM sprawl in control.

Capacity Planning

An interesting thing has been happening in VMware environments for the last few years, people have been going back to their old ways… By that I mean I have recently audited two environments that have begun to underutilize the hardware in the ESX environment. One was averaging about 20% processor and 40% memory across their 30 hosts and the other was using even less on a 52 host 5 cluster environment. When prodded as to why this was going on, the response in both was the same; "we don't want to push the servers since any performance hit will be seen as a VM issue". Of course in both environments the average VM had a single proc with 1 GB of ram. The hosts were 8 core machines so that any VM spinning at 100% would only use 1/8th (or 12%) of available CPU. Averaging 20% normally they would have to have 6 or 7 VMs spin out of control, on the same host simultaneously…

Of course, the issue really is that they don't know about their capacity, how to plan for it, or how to articulate what is being used or what is needed to their management. So the easy thing to do is to revert to the old ways, underutilize the hardware and ask for more servers. This is the worst thing you can do. The cost of VMs will go up, and the savings your CIO or VP of Infrastructure was looking for will not materialize.

To keep you from falling into this trap, let's talk about capacity in a VI environment and how to really look it, calculate it and articulate it.

It's about the VMs!

One of the funny things about a lot of admins doing capacity planning is that they tend to look at their VirtualCenter, take a look at the little charts or the % proc used on each host and its memory, then guess (that's right guess) at their available capacity. They may even print up the cluster or resource pool reports showing 51% average processor, 75% average memory usage, 8.3 MB of average disk through put etc, then present that as a capacity report to their manager or VP.

As a VP or Manager my first (and really only) question about this report would be "So how many more VMs can I deploy?" Obviously the answer is not in that

type of report. Managers and VPs don't care about the processor running at 40% and neither should you.

If a server runs at 40% average processor utilization, that means nothing. Not without a VM count to rationalize it. So if you had 4 VMs on the server running at 40% and you don't want your servers running over 80% at any given time, you essentially can deploy 4 more VMs before you reach your normal operating maximum. This is simplified but imagine, your environment ran at 40% average, and you had an average of 14 VMs on your servers. Now the average VM is using 2.85% of the processor (40/14=2.85) and if the operating maximum is 80% you can deploy 14 more VMs before hitting that limit. See why the percentages mean nothing without the VMs?

So what are you to do? Well, it's simple really. The idea is to find the bottleneck in one of the core four resources (Processor, Memory, Disk or Network). Each time you build a capacity report you have to determine which resource you will run out of first. Using the previous example of 40% processor utilization let's build on that with some memory stats and disk and network stats. We will simplify the disk and network stats here so that our Processor and Memory stats are what we are focused on.

Environment 1, 10 hosts 140 VMs:

- Processor Utilization average across hosts: 40%
- Physical Memory Utilization average across hosts: 70%
- Disk throughput average across hosts: 22.9 Mb
- Network throughput average across hosts: 31.5 Mb
- Target maximum on any resource: 80% of maximum physical resource

With these basic numbers we can come up with a capacity report that shows amount of VM capacity in use, available, and emergency capacity available. First you need to identify the average utilization of each resource for the average VM in your environment. Do this by taking the number of VMs and dividing it by the number of hosts to come up with your average VM per host count (14 in our case).

Once you have this number you can come up with averages for each of the core four resources per VM. So in our example environment you have the following:

Average VM in Environment 1

- 2.85% of processor utilization
- 5% of host memory utilization
- About 1.6 Mb of disk throughput
- About 2.25 Mb of network throughput

Knowing that we have multiple Gigabit nics in each system and 4GB fiber, we can see that disk and network are nowhere near their maxes. On the other hand the average host is at 70% memory utilized. This means that if current trends continue we can deploy 6 more VMs per host (5% memory per VM with 30% memory available average 30/5=6). Take that 6 per host average and multiple it by the number of hosts in the farm (10) and we can deploy about 60 new VMs before we run out of memory. Of course we have simplified this here, but you get the idea.

Beyond just baseline number, we also will ask VMware end users to identify their target maximums for capacity planning. So in most instances IT admins will say "we don't want to run above 80% normally". So for our previous example where memory was already at 70% the math would change slightly and instead of taking the available 30 and dividing by the VM average usage we should take 10% (80-70) and divide that by VM average usage to come up with 2 VMs per host or 20 more VMs.

I know this seems a little crazy to those new to capacity planning, but this is what is important from a management perspective, how many more VMs can you deploy? Answer that question with real numbers, and getting new capacity should be much easier.

Writing custom applications using VC data

Before releasing this book we had about a 52 page section on writing custom applications using VirtualCenter data. Sadly the book was just TOO BIG to even think about binding and we found that the section was much more about SQL queries and building an app than it was about VMware's VI3. So instead we are going to release this as a whitepaper on VMguru.com. RobZylowski (a friend and all around smart guy) was gracious enough to write all these pages on developing a VC app as an example and to not make his work public would just be a shame.