VMware® Infrastructure 3

Advanced Technical Design Guide

~and~

Advanced Operations Guide

Two books in one!

Ron Oglesby
Scott Herold
Mike Laverick

Chapter 8: Resource Monitoring

There are two very common mistakes that many administrators make when evaluating performance in VMware. Firstly, there is a tendency for an administrator to use the guest operating system's performance analysis tools such as Microsoft's Task Manager and Performance Monitor or Linux tools like VM's tat and top. Unfortunately, all of these tools have been designed for the guest operating system when it runs directly on physical hardware, not on virtual hardware. In the world of virtualization these tools have very limited usage. I still use Task Manager to identify crashed applications and use the "end task" button to kill processes. What I don't use is Task Manager's graphic chart to see what the percentage utilization is for the virtual CPU. All performance analysis is time critical- such as Page swaps/sec and CPU cycles/sec. The problem is that the VM does not see the CPU's physical clock – and therefore averages and such can be skewed. It is possible to run Performance Tools inside the VM *if* they have been written with the VMware Guest Operating System SDK (Software Development Kit). For example, Richard Garsthagen has two utilities called VM PerfMon and VM Time based on the VMware Guest OS SDK. These tools can be downloaded for free from the following site:

http://www.run-virtual.com/

Secondly, many administrators assume that poor performance is caused by the VM itself. However, a slow network response, for example, could be caused by configuration settings in the guest operating system. A couple of examples are bad DNS server search orders or poor settings in application software. All the tips and tricks you have learned to improve performance in the guest operating system still apply inside VM, such as disabling unneeded services. On its own, virtualization does not inherently improve performance or the reliability of your guest operating system – unless your VM is running on significantly better hardware than the previous existing physical systems.

Of course there is another way of measuring performance, and it's called "user experience." How fast or slow does a given system "feel" – and can you judge that against an agreed sense of "acceptable usage"? For example, what is an acceptable login time to a desktop PC against a domain controller running in VM? Is it 10 seconds, 30 seconds, 1 minute or 1 hour? Appealing though this

approach might be to a modern IT department wedded to the concept of being "user focused," the problem with this is manifold. Firstly, as my login example demonstrates, it is a highly qualitative approach and extremely subjective. Secondly, user expectations are constantly rising to whatever resources we are allocating – what is fast today is regarded as slow three months later. Lastly, it tells us nothing about the cause of the problem.

Identifying performance bottlenecks can be tricky in a virtualization environment. After all, we have many VMs, all executing on the same underlying hardware. How do you identify which VMs are the cause of poor performance and which are not? Many people choose to move VMs to an isolated ESX host and do their performance analysis there. This way you can see if it is that particular VM which is the source of the problem – or its relationship to others. If a VM is still performing badly when it has access to *all* the resources of an ESX host, then this might indicate the problem is within the guest operating system layer, rather than at the virtualization layer.

The key to fixing performance problems in any environment is knowing what tools are available to collect performance information. The only reliable, cast-iron guaranteed method of monitoring performance is using VMware or 3rd party tools that collect performance information *outside* of the VM. This will lead you to identifying the constraining resource. Invariably this will be one of the "four core;" CPU, Memory, Network, or Disk. It's important to be familiar with how to monitor the resources of VMs before we embark on making changes. If we don't know the source of our performance problems – how would we know what to change – and critically how would we verify that our changes have resolved our problem? Before we look at the tools for monitoring performance we need to understand how the VMkernel allocates resources to the VMs.

CPUs

VMkernel Load-balancing or Scheduling

When VMs run they execute their instructions on the physical CPUs of the ESX host. Within the ESX server the VMkernel is configured to monitor load on the CPUs, looking for a CPU that is doing less work than others. If the

VMkernel spots a significant disparity in the load on one physical CPU compared to another it will "schedule" that VMs threads to execute on a less busy CPU. This monitoring is configured at intervals of every 20 milliseconds and places a burden on the VMkernel. However, the performance gained by carrying out this analysis more than offsets the burden. The value of 20 milliseconds is configurable, and you may decide to make it less frequent if you feel your CPU load across an ESX host is relatively uniform. So if you don't experience large fluctuations in CPU activity you can change the frequency. The setting is called Cpu.MigratePeriod and is held under the Configuration Tab and Advanced Settings. The scheduler is designed to distribute CPU requests intelligently within the ESX host and reduce contention as much as possible. Contention is the word we use to describe a scenario where resources are scarce and two or more VMs "fight over" the resource, such as a CPU.

Single vCPUs or multiple vCPUs

Single vCPUs execute their threads on a single physical socket or core at any one time. In contrast, a dual or quad vCPU executes its instructions on more than one physical socket or core.

If you are using hyper-threading on Intel Xeon processors, the VMkernel treats each logical CPU in the Xeon chipset as if it was a physical processor. So a two socket processor with hyper-threading enabled would actually appear as if you had four physical processors. When it comes to dual or quad based VMs the VMkernel scheduler always makes sure it runs the VM on two different logical processors in two different sockets. There are some cases where hyper-threading actually degrades CPU performance, especially when the CPU I/O is exceptionally intensive. You might have already experienced this with resource-intensive products such as Oracle, Microsoft SQL, and Exchange. I have seen this personally in the terminal service environments with Citrix Presentation servers. For the most part, hyper-threading is a good feature to enable – but watch out for some high-end and processor intensive threads which would prefer to use the whole physical socket, rather than just a logical processor within it.

Here's a good analogy for hyper-threading. Imagine you have a narrow country lane which you would like enable for two-way traffic. Rather than widening the road you draw a line down the center of the road. This is what happens with

hyper-threading. The CPU package is able to take bi-directional communication. However, the actual width of the road or CPU hasn't widened. So although you might get two small cars (small CPU transactions) up and down this road – if a large vehicle comes along it will have to use both lanes anyway (a single large CPU transaction).

It is possible to have some per VM controls on how VM executes on a processor with hyper-threading available.

Controlling Hyper-threading Sharing on a VM

1. Right-click a VM, and Choose Edit Settings.
2. Click the Resource Tab, and Select Advanced CPU.

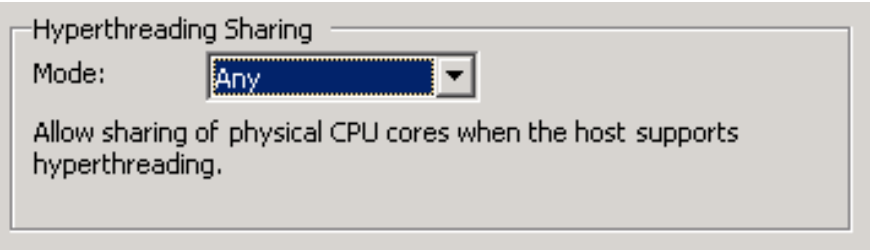
Figure 8.1 shows the ability to use three options:

Any: More than one VM can execute on the logical CPU.

None: VM receives all of the CPU not shared with other VMs.

Internal: VM with two vCPUs gets exclusive access to a CPU and its logical CPUs. This setting does not affect VMs with 4-vCPU's.

Figure 8.1



In conclusion, to get the benefit of virtual SMP you do need plenty of cores or sockets. There is little point in creating a dual based VM if you only have two sockets or two cores. All this will do is give the VMkernel's scheduler more

work to do and increase the chances of two VMs competing for a resource. Very simply, the more sockets or cores you have, the more opportunities the VMkernel schedule has of finding a CPU not in use or not heavily used – which then makes the VM perform better. The term that is sometimes used is having plenty of "hardware execution contexts," or in another way, plenty of sockets or cores to run a dual or quad vCPU.

To really leverage the value of vSMP you need four sockets/cores for dual VMs and eight sockets/cores for quad based VMs. Remember, Microsoft does not officially support "downgrading" an instance of Windows from dual or quad to a single CPU VM. Additionally, not all applications and services are multiprocessor (or multithreaded) aware – so sometimes adding an additional vCPU could make no difference to performance at all. Due to these issues I would recommend the decision to use vSMP should be taken with care, on a case-by-case basis. One way of finding out the impact of such a change could be to clone an existing uni-processor VM and upgrade the duplicate to be a dual or quad VM. You could then decide to delete or roll-back to the old VM should your test show to improve the situation or make it worse.

Lastly, in years to come this kind of concern might become less significant. Intel has, in their R&D labs, already created a multi-core processor that has up to 100 cores in a single socket under its "Tera-scale" computing research program. This CPU was created for experimental purposes only and will probably never see production market. It might not be so long until each VM executes on a dedicated core of its own.

CPU, Guest Operating System Kernel Updates and Idling

Excessive CPU activity can be caused with VM after P2V process. When you P2V a multi-processor physical machine, it is possible to "downgrade" to uni-processor VM. For this to work properly you would need to carry out a kernel update. In the world of Microsoft Windows this means changing the ACPI Function from an ACPU Multi-processor to an ACPI Uni-processor within the "Device Manager" tool. This replaces the "Hardware Abstraction Layer" file (hal.dll) and the ntoskernel.exe files. The incorrect HAL or ACPI functionality can cause a VM to make CPU demands even when it is not doing any legitimate CPU requests. This is because without the correct HAL the VMkernel cannot

"idle" the VM correctly. In other words the VMkernel doesn't know how to manage the VM correctly when it is in an idle or inactive state.

Microsoft has made moves in recent releases of Windows to inhibit even the administrator's ability to change the HAL. Ostensibly this has been done to "protect" novice administrators from changing the HAL, as it can cause a "blue screen of death" (BSOD) which puts the operating system in a state from which it cannot (without a great deal of effort and skill) be recovered. Fortunately, you can side step such "protection" using CLI tools freely available from Microsoft. One such excellent hardware management CLI is "DevCon." You will find it as part of the Driver Development Kit (DDK) from the mdsn.com part of Microsoft's website.

For example, this DevCon script would forcibly downgrade a multi-processor HAL Windows 2003 or Windows XP VM to use just a uni-processor HAL. The part that does the work is "devcon update %windir%\inf\hal.inf ACPIPIC_UP" which instructs the Windows kernel with HAL to use:

```
@echo off
cls
rem Author:       Mike Laverick
rem URL:    http://www.rtfm-ed.co.uk
echo ================================================
echo ==Downgrading ACPI to Uni-
Processor==================
echo ================================================
echo.
echo Please Wait
devcon sethwid @ROOT\PCI_HAL\0000 := !E_ISA_UP !ACPI-
PIC_UP !ACPIAPIC_UP !ACPIAPIC_MP !MPS_UP !MPS_MP
!SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP  > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := !E_ISA_UP !ACPI-
PIC_UP !ACPIAPIC_UP !ACPIAPIC_MP !MPS_UP !MPS_MP
!SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP
devcon sethwid @ROOT\PCI_HAL\0000 := +ACPIPIC_UP
devcon sethwid @ROOT\ACPI_HAL\0000 := +ACPIPIC_UP
devcon update %windir%\inf\hal.inf ACPIPIC_UP
echo Done!
echo.
echo ========================================
echo ==Script Completed==============================
echo ========================================
```

```
echo.
echo ============================================
echo ==Press any key to reboot the Virtual Ma-
chine===============
echo ============================================
pause > nul
devcon reboot
```

In the case of Windows 2000 the HAL references in the INF file are slightly different so to downgrade the HAL for it would require a slightly different script:

```
@echo off
cls
rem Author:       Mike Laverick
rem URL:    http://www.rtfm-ed.co.uk
echo ============================================
echo ==Downgrading ACPI to Uni-
Processor==================
echo ============================================
echo.
echo Please Wait
devcon sethwid @ROOT\PCI_HAL\0000 := !E_ISA_UP !ACPI-
PIC_UP !ACPIAPIC_UP !ACPIAPIC_MP !MPS_UP !MPS_MP
!SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP
devcon sethwid @ROOT\ACPI_HAL\0000 := !E_ISA_UP !ACPI-
PIC_UP !ACPIAPIC_UP !ACPIAPIC_MP !MPS_UP !MPS_MP
!SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP
devcon sethwid @ROOT\PCI_HAL\0000 := +ACPIAPIC_UP
devcon sethwid @ROOT\ACPI_HAL\0000 := +ACPIAPIC_UP
devcon update %windir%\inf\hal.inf ACPIAPIC_UP
echo Done!
echo.
echo ============================================
echo ==Script Completed================================
echo ============================================
echo.
echo ============================================
echo ==Press any key to reboot the Virtual Ma-
chine================
echo ============================================
pause > nul
devcon reboot
```

You can find copies of these scripts on my website. So if you find yourself in the unenviable position of downgrading the HAL you can do the normal "without warranty" disclaimers application and of course ensure backup, or snapshot your VMs before using them. They are under the section labeled "Sample P2V Post-Configuration Scripts for P2V and Sample ISO File."

http://www.rtfm-ed.co.uk/?page_id=8

Memory

Transparent Page Sharing (TPS)

ESX server deploys a number of memory management techniques to boost the amount of available RAM – and to dynamically manage the system should physical memory become scarce. The first and most important of these is "Transparent Page Sharing." If you run more than one copy of Windows or Linux on an ESX host, the VMkernel can identify that very similar information is likely to be duplicated in memory. So in the case of two instances of Windows there is likely to be more than one copy of files such as explorer.exe, svchost.exe, lsass.exe, spools.exe, and so on. The same scenario exists for every guest operating system supported by ESX in a VM. The VMkernel spots these duplicates and produces a single read-only copy. The read-only attribute is important from a security perspective, as it prevents the possibility of one VM modifying the memory contents of another VM. If a VM needs to modify the contents of its memory – the VMkernel seamlessly generates a read-write copy of the file – and instructs the VM where to find the file in memory. This is all done without the guest operating system or the VM realizing it is it taking place. In other words, the sharing of pages of memory is invisible (or transparent) to the guest operating system. It can be achieved because it is the VMkernel that is really in charge of the hardware, not Windows or Linux. VMware's own research has shown that around 30% of the guest operating system memory is duplicated between VMs when they are running on ESX. The values for the other guest operating systems are somewhat lower (because of Windows' systemic memory hungriness) but *all* VMs benefit from TPS to a greater or lesser degree.

The net result is a massive savings in the amount of RAM required to run a VM. In my own experience I have seen savings of 1-1.5GB on a server with as little as 2GB of physical RAM. The TPS feature also helps to "offset" the memory wasted by actually having a virtualization layer in the first place. After all, 272MB of RAM is lost when the Service Console loads. The VMkernel is round about 25MB in memory**.** Running one VM with one vCPU consumes 54MB of memory, with 64MB of memory consumed for a dual-VM. TPS helps offset this "virtualization overhead," sometimes even cancelling it out altogether.

The vmmemctl driver

When you install VMware Tools into a VM, alongside an improved network and mouse driver the VM has a memory control driver installed as well. Its file name is vmmemctl. It is normally referred to by VMware as the "balloon driver." This is because they use the analogy of a balloon to explain how this driver works. The most important thing to know about vmmemctl is that it is only engaged when memory is scarce and VMs are "fighting over" that resource – in other words, when contention is occurring. By itself it doesn't "fix" the problem of a lack of resources or unexpected peak demands for memory. In fact, its biggest use is as an indicator that there is a memory problem, so it is useful as a counter which draws the administrator's attention to potential problems. It is the symptom of a problem, not the source.

How does the vmmemctl driver work? Well, during normal operation where memory is plentiful and VMs are not in contention, the driver does nothing. It sits there inside the VM, deflated like a saggy balloon at the end of the party. However, when memory is scarce and the VMs are fighting over the resource, the vmmemctl driver begins to inflate. In other words, it begins to make demands for pages of memory. This generally occurs in a VM which you have marked as having a low priority on the system. The guest operating system obeys its internal memory management techniques – freeing up RAM by flushing old data to its virtual memory (page file or swap partition) to give the vmmemctl driver ranges of memory. Next comes the clever bit – rather than hanging on to this newly allocated memory, the vmmemctl driver hands over its memory to the VMkernel. The VMkernel in turn hands over this memory to the other VMs that really need it. When memory demands return to normal and are no longer scarce, the balloon driver deflates and gracefully hands back the memory it claimed to the guest operating system. On its own the

vmmemctl driver doesn't fix the problem which is a lack of memory. It does, on the other hand, give us a clear indicator of a potential problem. Additionally, it allows us to configure (along with other tools covered in the next chapter) the system for worse case scenarios by offering guaranteed levels of service to VMs that need it if memory becomes low.

The VMkernel VM Swap File

There are a number of approaches for handling the allocation of memory to a VM. These will be covered in more detail in the next module. It is possible to configure a VM to guarantee that a VM always runs in memory – and never uses virtual memory. To do this you would need quite a large amount of memory, depending on how much you allocate to a VM when you create it. It is also possible to configure a VM for what is referred to as "memory overcommitment." This is where we allocate more memory to VMs than is actually *physically* present on an ESX host. This allows for very high VM to ESX ratios without the need to buy more memory. The difference between what we allocate and how much physical memory we have is made up by VMkernel, VM swap file. Naturally, some anxieties surround this idea, not least the issue of performance. We all know that despite caching on SAN controllers and the increases in disk spindle speeds – memory is faster because there are no moving parts. It is important to know that in the world of ESX the VMkernel VM Swap File is only used as a "last resort." In other words, in order to get repeated read-write activity on the swap file, *all* memory would have been used on the ESX host. This is significantly different to the way our guest operating systems use their swap files and partitions. In Windows it not unusual to see page faults and swaps even on a server with lots of physical RAM. This is because Windows always has seen physical RAM and swap space as if it were one single block of memory. In the case of ESX, one would only expect to see swap activity in the extreme case where all RAM had been depleted. As with the balloon driver, swap activity is a symptom rather than the cause. It is an indication that an ESX is low on memory or VMs have been poorly configured with more memory reservations than they actually need.

Disks and Networking

Performance considerations on these resources have been covered earlier in this book. In the storage chapter I showed how correctly configuring the SAN or iSCSI for multi-pathing and selecting the correct RAID level can greatly affect performance. In the networking chapter I discussed the merits of "Traffic Shaping." I don't wish to restate those recommendations again, but I will cover the metrics and counters that would be used to diagnose bottlenecks in these resources.

Identifying Resource Constraints

CPU

Beside the overall amount of CPU used and the amount each VM is using – there is actually a much more revealing measure of actual VM performance called the "ready" value. The ready value means the VM is "ready" to execute processes and is waiting for the CPU to allocate a slice of CPU time. It is perfectly fine for a VM to be using 99% CPU, as long as its ready value remains low. A high CPU used value merely indicates a VM is processing – it doesn't necessarily mean the VM is performing badly. It could be benign CPU cycles caused by an application which is inherently CPU intensive. If however, the % ready value grows then this usually indicates the VM is ready to run but the CPU is not ready to supply the CPU time it demands. Does VMware give any guidelines on what appropriate CPU ready values should be? Yes, they say anything constantly over 5% should be looked at as a potential bottleneck.

Often there is a close relationship (an inverse proportion in some case) between the % of CPU used by a VM and the % ready value. As a VM is allocated more CPU time, its ready value should go down, and as the % of CPU time allocated value goes down, the ready value should grow. For this reason many people say that the "ready" value should be called the wait value – as it indicates how long a VM is waiting for the CPU to respond.

If you wish to see very high CPU ready values, take two VMs doing a very CPU intensive task and use the "processor affinity" options behind a VM and peg them to the same physical CPU. The net effect of this would cause "conten-

tion" as the two VMs fought over the same resource. VM processor affinity is a tool used in resource management which we will cover in the next module – as you can see, it could have a catastrophic impact on performance if configured incorrectly. CPU affinities also constitute a VMotion barrier.

Memory

Memory problems in a VM could be caused by an application or guest operating system mishandling its memory allocation. It's not uncommon for malign applications or services to demand an allocation of memory – and not gracefully hand them back. We label such applications or services as having a "memory leak." The source and cause of these memory problems lay not with the VM but poorly written operating systems and more frequently, even more poorly written application code. Despite the existence of transparent page sharing (TPS) and ballooning – it is beyond the scope of any virtualization platform to make bad code good. In my experience the most badly written code tends to come from application code developed in-house rather than from ISVs.

As stated earlier the ESX VMkernel prefers to use physical RAM in all cases. Only when physical RAM is scarce and VMs are fighting over access to RAM, does the per-VM swap file and the "ballooning" driver become engaged. Once you know this you can use the charts or ESXTOP to look for any persistent swap I/O and for any ballooning. Any swap or balloon driver activity indicates a memory problem.

Network

Before you blame a VM or an ESX host for slow network response, it's worth asking yourself – would this system be as slow if it was running on a physical host? Frequently the source of network bottlenecks exists in your physical network or at the slowness node of, for instance, a branch office. Additionally, networking problems could easily be the cause of mis-configuration of network settings within the VM. Perhaps your VM has the wrong default gateway or DNS settings which are causing long round trips or long DNS queries. Occasionally, when I teach a course I watch people troubleshoot networking problems. They check all the settings of the VM, but sometimes fail to even use the tools they are very familiar with such as ipconfig, ping, and so on. Network

troubleshooting should be begin with the basics first, before we over-complicate the process by checking the VM's settings.

It is legitimate to capture, analyze and measure the amount of packets sent or received to a VM. A network packet sent from a VM or received from a VM is unchanged (unless Traffic Shaping is engaged).

There is a close relationship between network activity and physical CPU usage. This is because the load-balancing mechanism of IP Hash (covered in Chapter 2) in itself causes the VMkernel to use the physical CPU. Additionally, lots of small TCP transactions inside a VM can cause the physical CPU to be busy – as the VMkernel works harder to remove packets from the physical NICs to the virtual NICs via the vSwitch. As a consequence, increasing physical CPU availability will help increase network performance.

If network performance is poor in the VM then confirm four configuration settings. Firstly, confirm that VMware Tools has been installed. VMware Tools replaces the driver for the virtual NIC with one which is "idealized" for virtual networking. Secondly, that the physical NICs selected for that VM's vSwitch are 1000mps interfaces. Thirdly, if you are using 100mps ensure that you have set the correct speed and duplex. Lastly, ensure that no operator has incorrectly engaged the "traffic shaper" module. The traffic shaper module is designed to throttle a VM which malignly wishes to "hog" the network pipe at the expense of other VMs.

Disk

Generally, we use disk reads and writes to check what volume of disk activity is occurring – and disk queue lengths as an indication that storage is a bottleneck. In this case it is reasonable to use guest operating system tools to monitor disk activity – as MB written to a virtual disk is no different than MB written to a physical disk. VMFS is such a light, low-weight file-system that it can be disregarded as the source of a bottleneck.

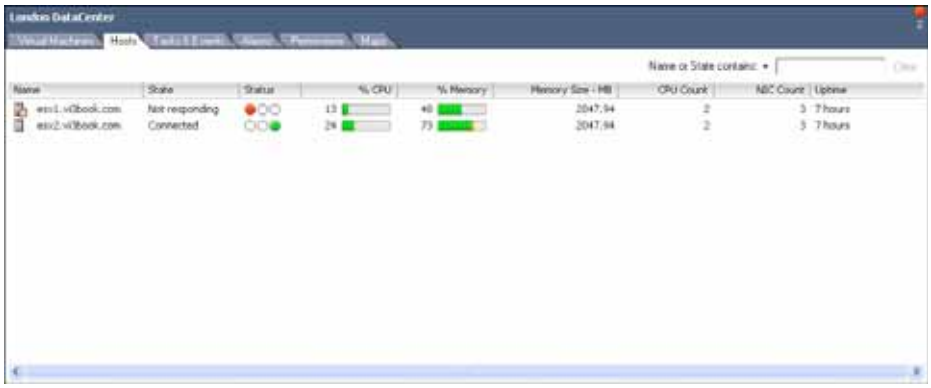
Before you consider any changes to the VM's settings, confirm that you have correctly configured and optimized your storage. In chapter 3 we discussed the importance of correctly setting the multi-pathing. You may also wish to ex-

periment with different RAID levels as these can have an impact on storage performance. Additionally, you may wish to double-check that the guest operating system has been correctly configured. Perhaps the VM is running out of disk space and is thrashing the disk looking for free space. Alternatively, a lack of memory inside the VM will force it to engage its virtual memory which increases disk activity.

Using VirtualCenter Charts

I've spoken at some length of the various counters that could expose a bottleneck in the system. Now let's look at where you can view these values and parameters. The best way of assessing performance generally is using the Performance tab in the VirtualCenter inventory. There are other places where you will see more high-level information. For instance, if you select your datacenter, and select either Virtual Machine tab or the Host Tab, you will get a view of information like CPU, Host Memory usage, VM memory usage – and information about your ESX host's resources such as amount memory, number of NICs and uptimes. Figure 8.2 shows this basic information.

Figure 8.2



However, if you really want to "drill down" and see performance stats in detail you will find yourself going to the performance tab.

In this section I am going to focus exclusively on CPU charts. Once you have the principle of how the interface works you will be able to find the data you are seeking for any resource. The last thing to mention is that the performance tab only appears on certain object types in the Inventory. These are:

- DRS and HA Clusters (Covered in Chapter 10)
- ESX Hosts
- Resource Pools (Covered in the next Chapter)
- Individual VMs

The default counters used vary depending on which of the 4 objects you select – but these can be customized. Below is a brief summary:

- DRS and HA Cluster – shows CPU Usage in MHz using 1 counter.
- ESX Host – shows CPU statistics using 4 counters.
- Resource Pools – uses the same counter as DRS and HA Clusters.
- VMs – shows CPU statistic using 3 counters.

All of these counters collect information every 20 seconds. This is the fastest refresh that charts can offer and is referred to, slightly confusingly, as "Real Time." This currently cannot be made any faster, and I would have thought collecting performance stats any quicker than this rate would constitute a performance hit in its own right! Information can be collected over a much longer period than this for purpose of tracking endemic trends; the defaults allow you to view information over a day, week, month, and a year. Additionally, the custom option allows you to set your own time frame; for instance, you could use it if you collect and view performance trends on a quarterly basis.

GOTCHA:

All performance data is time critical. It's absolutely imperative that time is correctly set on both the ESX host *and* the VirtualCenter Server. Failure to do so can result in the performance tab failing to return any information at all. The message you receive is:

"Performance data is currently not available for this entity."

In Chapter 12: *ESX on the Command-Line*, I will show you how to setup an NTP server to be the source of time, configured to speak to a publicly available NTP server – and then how to configure an ESX host to be a client of your internal NTP server. Alternatively, you can manually correct the time settings on your

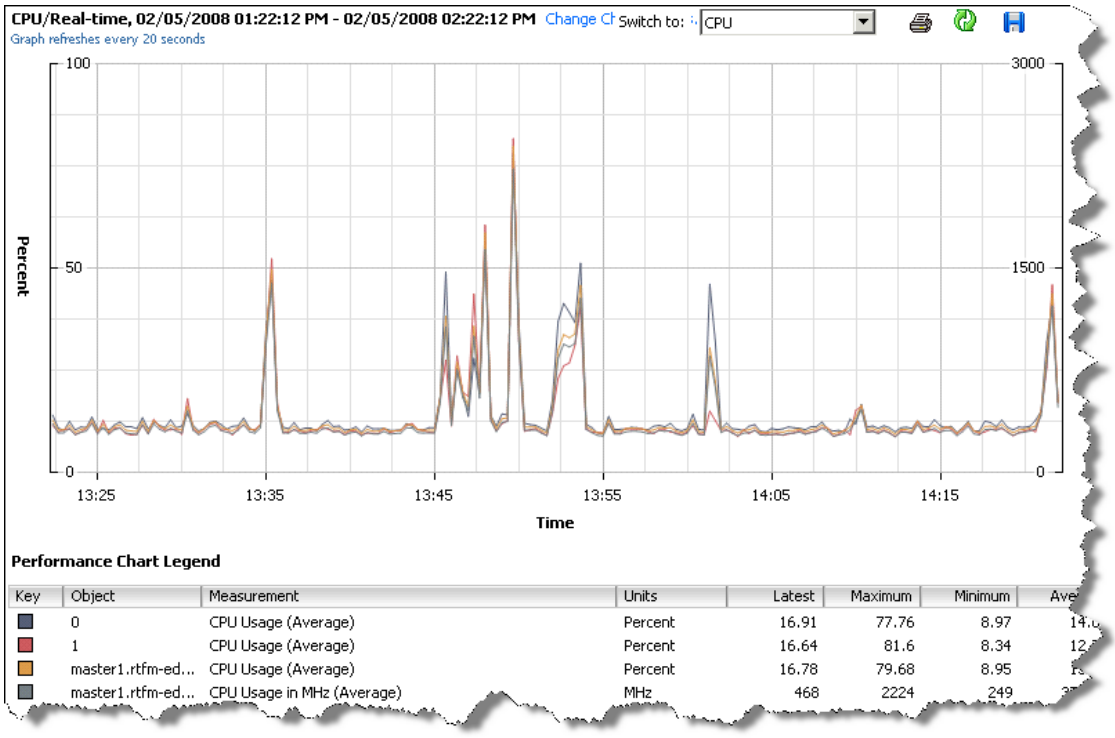
ESX hosts using the date command at the Service Console. If you have to correct time latency issue, do remember to wait to allow time for statistics to be collected.

CPU Charts

I'm going to explain how to use the Vi Client charting features, using CPUs as the resource. Learn the principles here, and you can apply them to any other resource as well, such as memory, network, and disk.

The best view to see overall CPU utilization is on the properties of an ESX Host. Figure 8.3 shows the performance tab for an ESX host. Let me walk you through some of the common options.

Figure 8.3



Description:

- In the top left hand corner, we have the resource being monitored (CPU) and its frequency (in real-time every 20 seconds) and period (12.52pm to 1:52pm).
- The blue "Change Chart options" allows the changing of the resource from CPU to be CPU, Disk, Memory, Network, System, or Cluster Services. System shows how the ESX VMkernel is operating and Cluster Services only appears if the ESX host is in a DRS or HA Cluster.
- The blue floppy disk icon allows you to save the chart data in a Microsoft Excel .XLS format – the XLS file contains raw data in an Excel chart that is also embedded in the spreadsheet.
- The icon to the left of the blue floppy disk icon allows you to manually refresh the data – and the icon to the left allows you to detach the chart from the Vi Client into a separate Window.
- Below that is the chart itself. Moving the mouse over it will tell you the values for each of the options, in this case as a percentage. This is the small pop-up box in the center of the chart. As you move your mouse across the chart these statistics update.
- Lastly, below that we have the Performance Chart Legend. If you select an object in the list (like I have with CPU Usage in MHz) it then highlights that counter in the chart. This is why it is it is slightly darker than the rest. If you know Windows well, this is like pressing ctrl+h while in Performance Manager.

If you want to see how the VM is behaving you need to select your VM in the inventory and choose the Performance tab. Within the Vi Client this is the only place the all important %ready value appears on the properties of the VM, not the ESX host. This is a bit of shame as it would allow you to see the %CPU and %ready values for many VMs.

Note:

In the next chapter we will look at the command-line performance tool called esxtop. Esxtop does allow you to see the %ready value on VMs and compare and contrast them.

To see the %ready value on a VM:

1. **Select a VM** in the list.
2. Choose the **Performance Tab**.
3. Click the **Change Chart Options…**
4. **On the right-hand side** of the dialog box under **Objects and Description**,
5. **de-select the check box next to the VM's name**, and select the **vCPU underneath**.
6. Next, under **Counters and Descriptions**,
7. **scroll down the list of counters**, and select **CPU Ready**.
8. Click **OK**.

If you want to generate some large CPU activity for testing purposes we can do this by using Microsoft Calculator. Yes, Microsoft Calculator can be a very CPU intensive application depending on what you're calculating.

- Open Microsoft Calculator in the VM.
- Change to the scientific view.
- Type 9999999.
- Then click the n! button.

Every so often Windows complains that this calculation will take a long time – and would you like to stop the process or continue the calculation. The calculation continues until you stop it or it is completed.

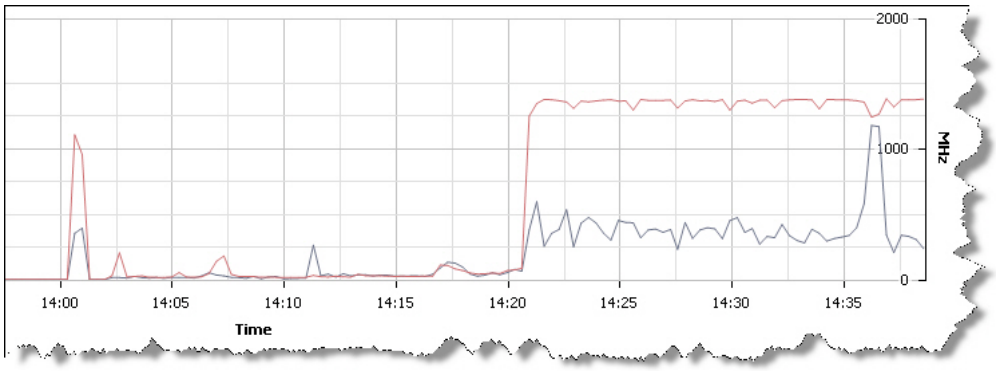
The n! button calculates factorials. A factorial is the product of a positive integer and all positive integers less than itself. For example, the factorial of a 4, written 4!, is $4 \times 3 \times 2 \times 1 = 24$. So calculating the factorial of 999999 results in a lot of calculations!

I picked up on this method of quickly generating CPU activity from Brian Madden's article "Citrix buys Application Performance Management vendor Reflectent. Is this a YAM?", which you can read online at:

http://www.brianmadden.com/content/content.asp?ID=589

Figure 8.4 shows my VM running with a calculator working on a factorial. The spike at 2 p.m. happened when I just powered on the VM. After 2:20 p.m. I ran calculator.

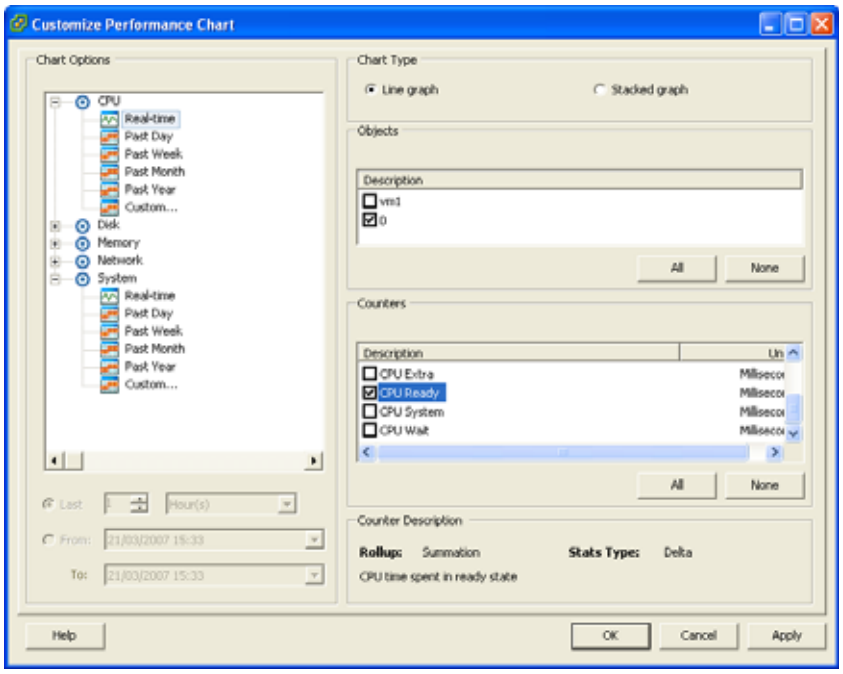
Figure 8.4



The Chart Options Dialog Box

A few moments ago we were in the chart options dialog box. That's where we added in the %ready value for the VM, focused on its vCPU. This is quite a "busy" dialog box with lots of options. Let me walk you through the settings, just like I walked you through the chart. Figure 8.5 shows us the chart settings on an individual VM.

Figure 8.5



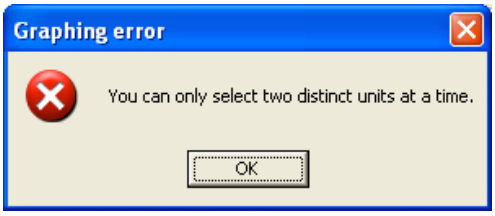
On the left-hand side of the dialog box, under chart options, we have the core resources of CPU, Disk, Memory, and Network. Expand one of these and then select an interval (Real-Time, Past Day, Past Week, Past Month, or Past Year). Unfortunately, it's not currently to possible show CPU, Disk, Memory, and Network together in four little charts for comparison purposes. This would be useful for working out something like whether disk activity was actually a symptom of excessive swap activity within the guest caused by a lack of memory. Selecting the interval of custom activates the "Last" and "From" options in the dialog box.

On the properties of a VM we also have the "System" options. In the context of a VM this allows you to see its average uptimes and the regularity of its heartbeat. These can be taken as a very general measure of the "availability" of a given VM. Clearly long uptimes with regular heartbeats mean our VM is alive and healthy – poor uptimes and an irregular heartbeat, or no heartbeat at all, is generally a sign that your VM is feeling a bit poor – and is need of intensive care or resuscitation. One reason for not receiving a heartbeat at all from a VM is because VMware Tools Service has failed to start or not been installed at all.

Over on the right-hand side of the dialog box, under chart types we can see the chart's appearance. On individual VM we only get two types (Line Graph and Stacked Graph). If you select the ESX host and Change Chart options you have three types (Line Graph, Stacked Graph, and Stacked Graph [Per VM]). Personally, I've always thought line graphs are clearer and easier to interpret.

Under objects we have description. Every resource CPU has objects – usually either the entire VM or its individual vCPUs. In turn every object has attributes or "counters" which tells the performance statistics that we have. Sometimes both of these options cannot be simultaneously displayed. So if we have VM1 and 0 selected together and also try to select CPU ready – you will receive the error message in Figure 8.6

Figure 8.6



Under counters we can select additional metrics to gather more data about performance. There are many, many counters on every single resource – far too many for us to outline here. You do get very brief information in the counter description pane but these aren't always very helpful. For example, the dialog box in Figure 8.5 defines CPU Ready as being "CPU Time spent in Ready State." I would recommend consulting VMware documentation and learning them on a case-by-case basis dependent on your performance challenges.

Configuring Alarms and Alerts

Alarms and alerts are built-in to VirtualCenter. Unfortunately, they are few in number and you cannot define new condition criteria of your own. There are built-in Alarms and Alerts which are defined at the top of the inventory in "Host and Clusters." These settings are inherited down the inventory and applied to all ESX hosts and VMs. If you wish to modify these "defaults" you must modify them at the point at which they were inherited. If you wish to

create new alarms and alerts with different settings you must delete the built-in ones and create ones of your own on the relevant folder.

Modifying existing alarms

A good example of modifying an existing alarm might be the built-in alarm on ESX hosts called "Host connection state." Currently the default settings on this alarm only contain one condition (red) which occurs when an ESX host becomes "Not responding;" "disconnected" is an additional condition. These messages can occur when rebooting an ESX host or if you have a network failure. Initially, an ESX host will enter a "Not responding message," and after a timeout it enters a "disconnected" state. You can force an ESX host to enter the disconnected state forcibly – by right-clicking the ESX host and choosing disconnect.

1. In the Inventory, select "Hosts and Clusters."
2. Select the Alarm Tab.
3. Click the "Definitions" button.
4. Double-click at the Alarm called "Host connection state."
5. Select the Triggers Tab.
6. Change Warning to be "Not Responding."
7. Change Alert to be "Disconnected."
8. Click OK.

Setting Custom Alarms for ESX Hosts

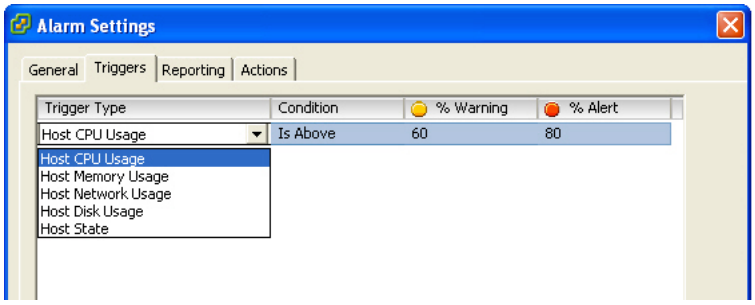
In my case I have a mix of different servers with different levels of memory and CPU resources. One of my problems is that default values for an alarm or alert at 75% and 90% are being triggered too late for my hardware. Within my London DataCenter I have created a folder for my Intel servers and AMD servers. I will apply custom alarms for each folder.

1. In the Inventory choose Hosts and Clusters.
2. Select the Alarm Tab.

3. Click the Definitions button.
4. Right-click Host CPU usage, and Choose Remove.
5. Right-click Host Memory Usage, and Choose Remove.
6. Select the folder that contains your ESX host.
7. Select the Alarms tab.
8. Click the Definitions button.
9. Right-click and choose New Alarm.
10. In the Alarm name dialog field type in a friendly name such as "Host CPU usage – Intel."
11. Click the Triggers tab, and select Add.

    Figure 8.7 shows that by default this adds "Host CPU Usage" – but if you click next to it you will get a pull-down list of additional triggers including some that are not created by default during installation ("Host network usage" and "Host disk usage").

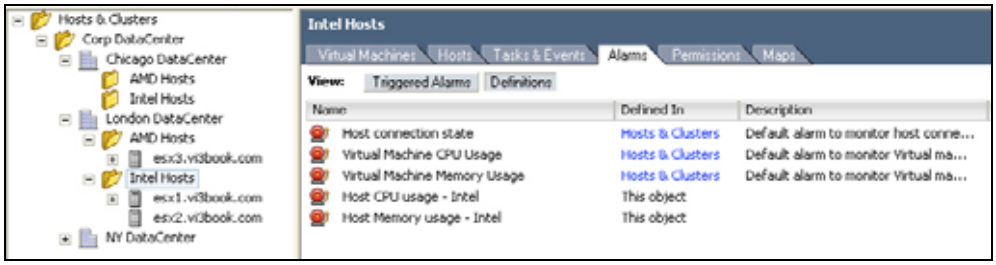
Figure 8.7



12. Adjust the initial warning and alert to be the appropriate values – in my case I lowered the values to 60% and 80%.

I repeated the same task with my AMD folder as well. The resulting configuration is outlined in Figure 8.8 which shows that "Host connection state," "Virtual Machine CPU Usage" and "Virtual Machine Memory Usage" are inherited from "Hosts & Clusters" (highlighted in blue). In contrast, my new definitions are inherited from the Intel Hosts folder level.

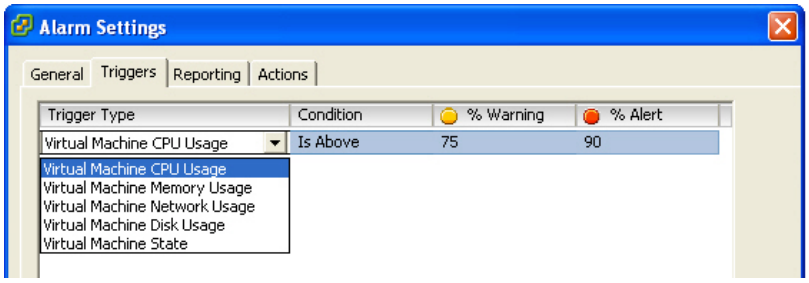
Figure 8.8



Note:

I can repeat this configuration, removing the default alarms from "Host & Clusters" for Virtual Machines. Instead I can define new ones just for the London DataCenter, leaving Ron and Scott freedom to do what they wish for the NY and Chicago DataCenters. Again, Figure 8.9 shows that when you add in custom alarms and alerts you will find additional conditions are available including network, disk usage, and the VM state (where a VM is powered on, off, and suspend). To see this, change the Alarm Type from "Monitor a host" to "Monitor a Virtual Machine."

Figure 8.9



GOTCHA:

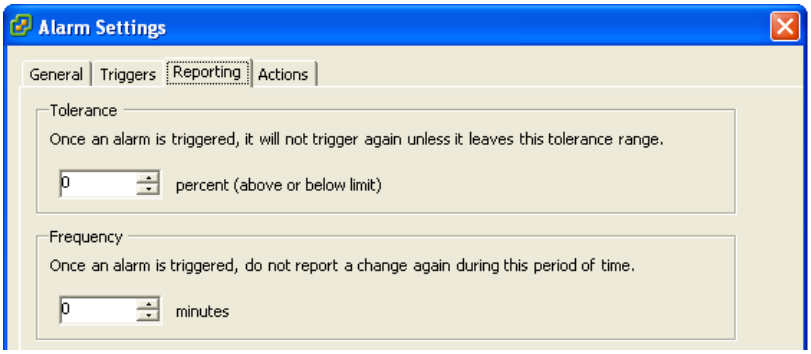
Be careful when using Virtual Machine state with powered on or powered off. If you do this you can create yellow or red alerts on the VMs which are not a problem. The system is doing its job giving you an alarm even when a condition is met. As most VMs are powered in a production environment on it's not a terribly useful alarm condition. The same can happen if you set an alarm on

VMs that are powered off. This generates an alarm on VM which isn't even powered on. This could be an irritation in test and development environments where many VMs are powered on when needed.

Configuring Tolerances and Frequency

Tolerance and Frequency settings in the Reporting Tab allow you to control how "chatty" an alarm can be. Figure 8.10 shows this dialog box. This can be useful to stop unwanted SMTP/SNMP alerts. Set in the reporting tab, tolerance affects alarm conditions that are set by a percentage (as opposed to Boolean conditions which are logical such as VM or Host State conditions). So it is possible to send an alert when a warning reaches 75% but not send another warning until it changes by another 5% below or above 75%.

Figure 8.10

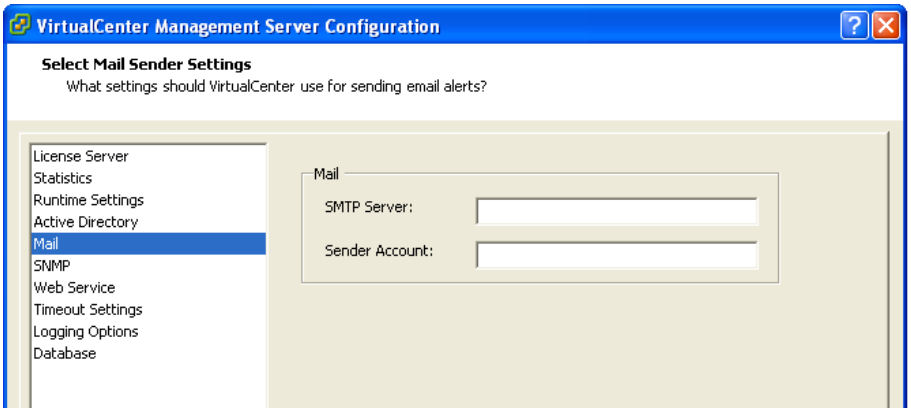


Frequency allows a user to modify how often VirtualCenter checks on an alarm that has already been triggered. So we can ignore an alarm for, say, 60 minutes before it sends out another.

Configuring Email Alerts

Configuring VirtualCenter to issue emails generated by alarms requires VirtualCenter to know the SMTP detail's server name and both the sender's (From:) and recipient's (To:) email addresses. These settings are configured in the VirtualCenter Management dialog box. Incidentally, it's from this dialog box that some global VirtualCenter settings can be configured. Figure 8.11 shows the dialog box in question.

Figure 8.11



1. In the menu choose Administration and VirtualCenter Management Server Configuration.
2. Select the **Mail** section of the dialog box.
3. In the **SMTP Server** field, type the name of your mail server, smtp1.vi3book.com for example.
4. In the **Sender Account** field, type the name of account used to send emails, virtualcenter@vi3book.com for example.

   GOTCHA:

   Due to the absence of a password field, the SMTP requires authentication switched off. In fact, for this to function correctly your SMTP server has to be configured as an "Open Relay." If this SMTP server is internet facing it will be a magnet for spammers, so make sure whatever you set here is an internal-only email system.
5. Next open an alarm such as "Host State Connection."
6. Choose the **Actions** tab.
7. Click the **Add** button (the default is "Send a notification trap").
8. Click into the **Value** column, and type the "To:" recipient email address.

Note:

You can also enable this to send an alarm from Green to Yellow (our "Not Responding" alarm) as well as from Yellow to Red (our "Disconnected" alarm).

Configuring SNMP Alerts

SNMP is the Simple Network Management Protocol. Generally, large corporations make the investment into commercially available SNMP such as HP OpenView or Computer Associates UniCenter applications. If you are merely wishing to test that SNMP is correctly configured or use free software, there is a whole host of free SMNP management tools. This demonstration shows you how to use a free tool called Trap Receiver from Network Computing Technologies to test the SNMP configuration. You can download the Trap Receiver software at:

http://www.ncomtech.com/

1. Download and Install Trap Receiver to the VirtualCenter Server.
2. Run the Trap Receiver Application from the Start Menu choosing the Start the Service button.
3. Open an Alarm definition. In my case I choose my alarm called Virtual Machine CPU usage in the London DataCenter.
4. Select the Action Tab.
5. Click the Add button.
6. Choose Send a notification trap, and Click OK.

Note:

Notice that you can set more actions for VMs than you can for an ESX host – including such responses as run as script, power on or off a VM, suspend a VM, or reset a VM. Theoretically, we could have an alarm that said if a VM went to 99% of its vCPU that it would be reset.

Note:

To generate an alarm I ran a cpu intensive application within one of my VMs. This generated responses in the trap receiver application. Figures 8.12 and 8.13 show the responses.

Figure 8.12

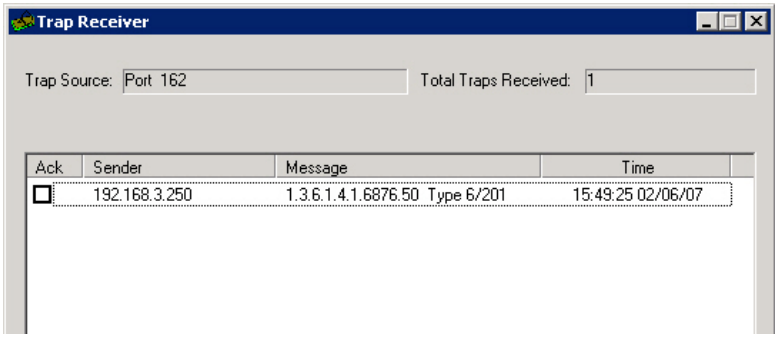
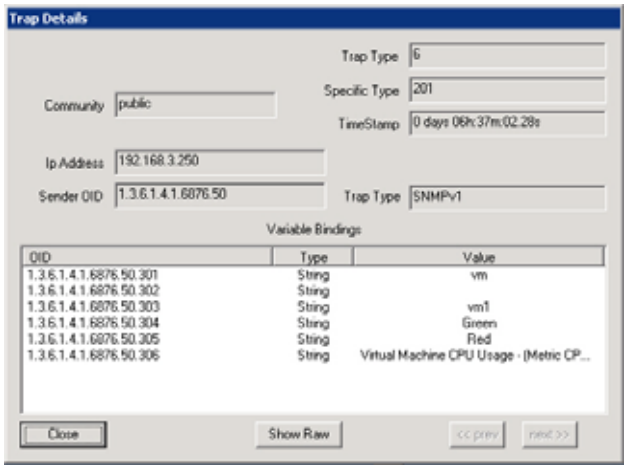


Figure 8.13



7. **Note:**

In reality it is unusual to run an SMNP Management tool on the VirtualCenter server itself but instead run it on a separate system altogether. If you wish to do this you need to modify VirtualCenter's default SNMP management settings held in the VirtualCenter

Management Server Configuration dialog box (accessible from the Administration menu).

Disabling Alarms

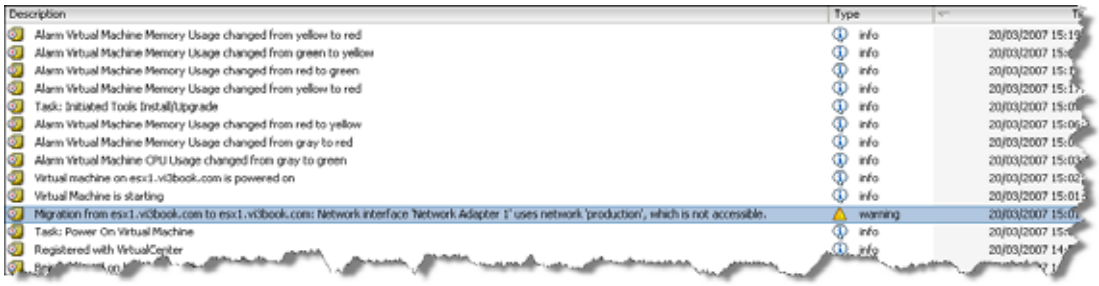
There is no easy way to temporarily turn off an alarm for an individual VM. The most we can do is temporarily disable an alarm if we feel that it is incorrectly generating unwarranted alerts.

1. Locate an alarm; for example, I went back to the London DataCenter where my VM alarms were held and modified an alarm I created called Virtual Machine State.
2. Double-click the **Alarm**.
3. Under the **General** Tab remove the tick next to **Enable this Alarm**.

Events and Tasks

As you probably have noticed already, the Vi Client's "Recent Task" pane is useful in showing ongoing processes and in a multi-administrator environment allows you to see tasks triggered by other VirtualCenter administrators. A record of all tasks and events is kept in VirtualCenter, and the tab called "Tasks & Events" appears at every point in the Inventory. As you navigate down the inventory you gradually see less and less focused information on a particular VM. Figure 8.14 shows a type event view on a VM. Notice the warning, which indicates an attempt to carry out a VMotion was tried but failed because of a networking problem. This is probably a vSwitch Port Group naming problem.

Figure 8.14



If a warning message appears on the Vi Client which you fail to make a note of you will generally find it in Tasks & Events.

Scheduled Tasks

VirtualCenter does have the ability to automate some repetitive tasks without the need to learn a scripting language. Scheduled tasks are configured with a point-and-click interface and all you do to automate is:

- Power State of a VM
- Clone a VM
- Deploy a VM from a Template
- Create a New VM (not from a template)
- VMotion a VM
- Relocate a VM (Cold Migrate and/or Move its files)
- Snapshot a VM
- Add an ESX Host

Scheduled tasks are very easy to configure, so I won't even show them here. I just wanted you to know they were available. The only "gotcha" with scheduled tasks is how to express the time for event to happen. Most users would expect the system to use a 24-hour format. Instead you type in a time, say 10:00. You must follow this by manually typing whether this is in the morning or evening with 10:00 a.m. or 10:00 p.m. in the interface. Another slight gotcha is you only add an ESX host to a datacenter object, not a folder within a datacenter with the scheduled tasks feature.

Log Files and Support Data

VirtualCenter and ESX both have logging information; this could be useful in troubleshooting scenarios and for liaising with VMware Support. Additionally, this can become a management issue – how long will logging data be retained and how frequently will the logs be rotated.

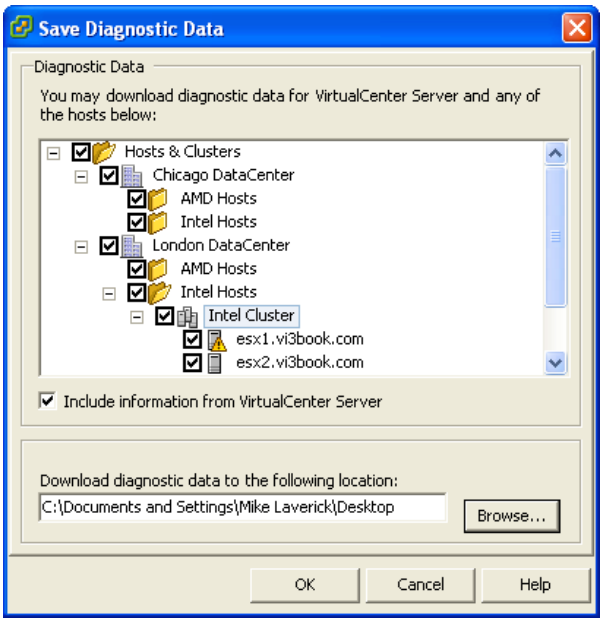
VirtualCenter Logs

Access and Export your VirtualCenter and ESX logs by:

1. Clicking **Admin button** in the Vi Client.
2. Select the **System Logs** tab.
3. Your logs for both VirtualCenter and ESX hosts can be exported and submitted to VMware Support by clicking the **Export Diagnostic Data button**.

Figure 8.15 shows me selecting in the Inventory that I want to collect. Use the browse button to save the export location.

Figure 8.15



Note:

This starts a task in the "Recent Tasks" view called "Generate Diagnostic Bundles." Additionally, you can create VMware Support log bundles directly at the VirtualCenter server itself if you click:

Start > Programs > VMware

Select **Generate VirtualCenter Server log bundle**.

This method generates a zip vcsupport-M-DD-YYY-HH-MM.zip on the VirtualCenter's desktop.

ESX Host specific Log File Generation

The main location for ESX logs is in /var/log. This location is used to store log files for the Service Console and the VMkernel. Files that begin VMK are VMkernel logs, whereas all other files are Service Console log files. Individual log bundles and VMware Support files can be generated at the service console using a script:

1. Logon to the Service Console as root.
2. Type cd to make sure you are /root.
3. Type:

   vm-support

   Note:

   This tool gathers up all your logs and configuration information – and zips it all up in a handy tgz file. These files are then uploaded to VMware's FTP (rather than being sent in emails) which is disclosed to you when you open a Service Request (SR) with VMware Support.

   Note:

   Log rotation is handled at the command-line on a per ESX host basis using the "logrotate" utility.