VMware® Infrastructure 3

Advanced Technical Design Guide

~and~

Advanced Operations Guide

Two books in one!

Ron Oglesby
Scott Herold
Mike Laverick

Chapter 10: VMotion, DRS, and HA

VMotion

How does VMotion work?

To say we are moving a VM from one ESX host to another is essentially a bit of a lie. In fact, we very rarely "move" data around at all. If you cut and paste a file in a folder to another folder what actually happens is the operating system copies the file to the new location. Once this is completed the original is deleted. Essentially, this is what happens in VMotion. The critical thing to stress is that the VM's files are *not* copied, but its *memory* contents are.

The VM on ESX1 is duplicated on ESX2, and the original is deleted. During VMotion what ESX creates is an initial pre-copy of memory from the VM running on ESX1 into a VM on ESX2. During the copy, a log file is generated to track any changes during the initial pre-copy process (it is referred to as a memory bitmap).

Once the VMs are practically at the same state, this memory bitmap is transferred to ESX2. Before the transfer of the bitmap file the VM on ESX1 is put into what is called a "quiesced" state. This quiesce process massively reduces the amount of activity occurring inside the VM that is being "moved." It allows the memory bitmap file to become so small that it can be transferred very quickly. This state also allows for rollback if a network failure takes place. In this respect VMotion has a transactional quality. VMotion is either successful or unsuccessful – what we don't get is two VMs on two different ESX hosts who believe they are the same. After the memory bitmap has been transferred the end-user is switched to using the VM on ESX2 – and the original VM is removed from ESX1.

During this switch from one ESX host to another there *may* be some dropped packets. Although the VM's physical MAC address does *not* change the physical MAC address on the ESX host does. The VMotion process triggers a Reverse Address Resolution Protocol (RARP) packet to make sure all network devices

on the same subnet as the VM are aware that packets destined for the VM IP address should be directed towards a new MAC address. In my experience these one or two lost packets are not enough to disconnect even a real-time networking system like Microsoft's Remote Desktop Protocol (RDP) or Citrix's Independent Computing Architecture (ICA) protocol. In some cases where I have been pinging a VM with ping –t (a constant ping) I've seen packets merely being delayed rather than dropped. Fundamentally, you will probably see more packets dropped daily on your WAN than you ever will with VMotion.

VMotion Requirements on the ESX Host

VMotion has a number of requirements for it to function both on the VM and between ESX hosts (source and destination) – a lack of these requirements can be a good reason to resort to a cold migration instead.

Here is a comprehensive list of ESX Host requirements:

- Shared storage/LUN Visibility between the source and destination ESX hosts (SAN, iSCSI or NAS) - this includes the VMFS volume where the virtual disk(s) are and any RDM LUNs.
- A VMkernel port group on vSwitch configured with 1000mps on the VMotion network- this VMkernel port group requires an IP address and a subnet mask.
- Access to the same "production" network.
- Consistently labeled virtual switch port groups (case-sensitive).
- Compatible CPUs.

If any one of these requirements cannot be met then VMotion could fail. Fortunately, most of the requirements can be purchased (shared storage and gigabit bandwidth) or re-configured (same production network and consistently labeled switch port groups). The requirement for 1000mps networking is not necessarily a hard one. I have successfully carried out VMotion with 100mps at full-duplex – and I have also seen it fail. In the main, these have been test and development environments where full VMware support is not required. If you want full VMware support you will need 1000mps networking. The cause of the failure with 100mps is a network timeout, merely which the amount of

memory in VM is large and frequently changing – and that 100mps pipe cannot bring the state of the two VMs close enough to trigger the VMotion.

There is one show stopper and that is compatible CPUs. Remember, although 99% of a VM is isolated from physical hardware – the one exception is that vCPUs do see specific *attributes* of the CPU. One reason to do a cold migration then is because your servers do not share CPU compatibility. New CPUs are constantly being released so it's inevitable that you will face some CPU incompatibility at some stage in your use of VMotion. Of course, many people attempt to purchase the same make and model of server, and indeed I've heard stories that some hardware vendors will even keep in stock CPUs that match your server's specification. I think it's likely that in the future the attributes that cause CPU incompatibilities will grow – as you purchase new hardware and as VMware exposes more of these features to a VM to improve performance, stability, and security. Here's a current list of attributes that would cause VMotion to fail due to CPU incompatibilities:

- **Processor Vendor: Intel Vs AMD**
- **Family: PIII Vs PIV, Opteron Family Numbers**

  We cannot carry out VMotion events from Intel processors to AMD processors. Within a given vendor there are "family" differences that would prevent VMotion from Intel Pentium III and Intel IV; similarly, within AMD there are family differences within the Opertons that would prevent VMotion.

- **SSE3 Instructions**

  SSE3 stands for "Streaming SIMD Extensions," and SIMD stands for "Single Instruction Multiple Data." These allow for improvements in processing for multi-media applications and have their roots in the MMX (Multi-Media eXtensions) feature found in some early Intel Pentium II processors.

- **Hardware Assist: Intel VT and AMD-V**

  These are recent enhancements that allegedly improve performance specifically for virtualization activity. They represent the first stages by Intel and AMD to create processors designed for virtualization. Right now the jury is out on whether they make such a massive impact for operating systems like ESX, but they are a step in the right direction and do represent a VMotion barrier.

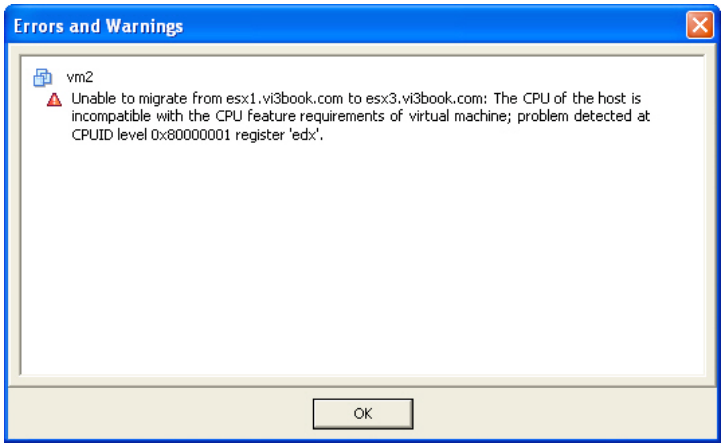
- **Execution Disable: Intel XD and AMD NX**

  These attributes are designed to secure processors from attacks and exploits used by hackers.

There are, of course, other processor differences – such as the number of cores, sockets, clock speed, and amount of onboard cache; these attributes are ignored in VMotion, so are not a concern.

You will soon know you have some kind of CPU incompatibility if you attempt a VMotion where one exists. Before attempting the VMotion, ESX "validates" the destination. If a CPU incompatibility is discovered then VMotion is not allowed to even start. Figure 10.1 shows the dialog box warning that I receive when I attempt a VMotion between one of my Dell PowerEdge Servers with an Intel Processor to a HP Proliant DL385 with AMD Opteron Dual-Core Processors.

Figure 10.1



VMotion Requirements on the VM

There are some requirements for the VM. Fortunately, these are configurable and unlikely to cause you many headaches once you have resolved them. In dialog boxes like the one above you get two types of messages – warnings and errors. Warnings can be bypassed; they are merely cautions that a problem *could* occur, whereas errors must be resolved before continuing.

Here is a definite list of errors and the reasons why:

- **Active connections to use an internal switch**

  By "active" we mean that the VM is configured and connected to the internal switch. VMotion cannot guarantee that the same internal switch exists on the destination host, and would offer the same uninterrupted connectivity to the VM.

- **Active connection to use a CD-ROM or floppy which is not on shared storage**

  By "active" we mean that the VM is configured and connected to the CD or Floppy Device. VMotion cannot guarantee that the CD or Floppy will still be accessible on the destination host.

- **CPU Affinities**

  VMotion cannot guarantee that the VM will be able to continue to run on the specific CPU number on the destination.

  GOTCHA:

  CPU affinities are disabled in DRS for this reason. CPU affinities can also cause problems with DRS. If you setup CPU affinities on VM on a stand-alone ESX host and later join it to a DRS cluster you will find it cannot be VMotion-ed. There is a workaround to "loosing" your control over CPU affinities in DRS without necessarily powering off the VM, entering maintenance mode and removing the ESX host from the DRS cluster. The loss of configuration options over CPU affinities only applies in the "Fully Automated" mode. It is possible to temporarily switch the cluster to "Partially Automated" or "Manual." This will re-enable the CPU affinity feature on the VM which will then allow you to remove the CPU affinity problem. This is very irritating, and so I would rec-

ommend avoiding CPU affinities unless you have a totally compelling reason to use them.

- **VMs in a Cluster Relationship**

  Firstly, the virtual disks and RDMs used in VM clustering must be on local storage for VMware Support; as we have seen, shared storage is a requirement for VMotion. Secondly, if you think about it, the uptime is delivered by the VM cluster. If you want to move a VM cluster you could power off one node and cold migrate it and then power it back on. If you repeated this practice with the second node effectively you would have moved the cluster without downtime (of course, the reliability of this approach is only as good as your clustering software). Thirdly, because there is a potential loss of packets on the heartbeat network during VMotion there could be unwanted cluster failover. Lastly, clustering software is highly dependent on SCSI reservations to lock storage to decide which node is active and which node is passive. If you attempted a VMotion event in active node, the SAN array would have received a SCSI reservation on the quorum disk from the WWN of the *source* host. Once the VMotion had been successful, the array would receive a renewal for that lock from the WWN of the *destination* host. It will reject that reservation because the destination node didn't have the reservation in the first place.

  Perhaps one solution to this issue would be I/O virtualization, where a WWN "alias" could be presented directly to the VM. In this case the lock would be created by the VM, *not* by our physical ESX host with a physical fiber channel card. Emulex is working on adapters that have this functionality.

- **RDMs**

  This is where the destination host does *not* have visibility to the RDM LUN. I would like to make it clear that RDMs are not incompatible per se with VMotion. In fact, one of the major reasons they were introduced in ESX 2.5.0 was to allow VMotion of VMs that were natively accessing storage, as previously the mechanism which was used to allow native access "broke" VMotion. The issue here is of LUN visibility and VMotion's requirement of shared storage. That means visibility of VMFS *and* Raw LUNs to both ESX hosts.

Here's a definitive list of warnings and the reasons why:

- **Configured to an internal switch**

  By "configured" we mean that the VM is set to use the internal switch. However, on the VM under Network Adapter and Device Status, Connected and Connected at Power On are not enabled.

- **Configured to use a CD-ROM or floppy which is not on shared storage**

  By "configured" we mean that the VM is set to use the CD or Floppy, but that on the VM under CD/DVD Drive or Floppy Drive and Device Status Connected and Connected at Power On are not enabled.

- **Snapshots**

  There could be warnings when deleting or reverting snapshots when the VM is moved. Personally, I have carried out VMotions with snapshots engaged – and never had a problem. After all, the files that make up a snapshot would be on shared storage. Nonetheless, the Vi-Client does warn you about having snapshots applied.

- **Lack of a heartbeat signal**

  This can happen if you have failed to install VMware Tools. It can also happen if you have just powered on a VM, and the VMware Tools service/daemon has yet to start. Additionally, I have found that this might temporarily interrupt the heartbeat signal if you have recently carried out a VMotion on the VM.

Figures 10.2 and 10.3 show you all the errors and warnings in single dialog boxes.

Figure 10.2 – All the Errors

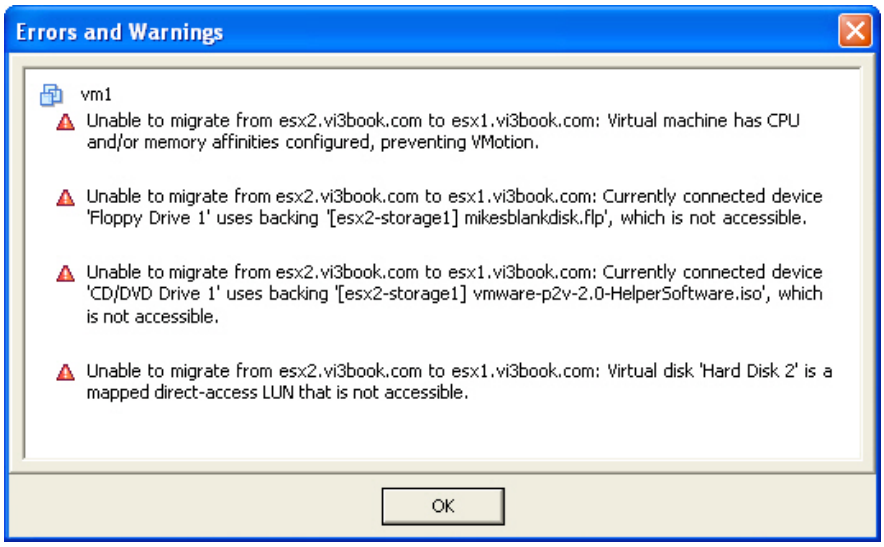
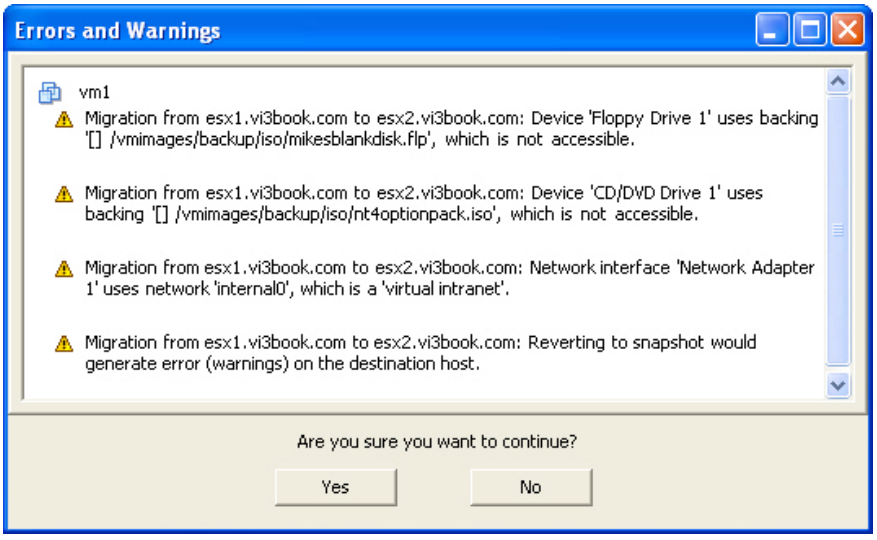


Figure 10.3 – All the Warnings



GOTCHA:

Just as these errors, warnings, and requirements can stop a manual VMotion, they can also stop automatic VMotion generated by VMware DRS, so it's important to resolve this wherever possible.

Configuring and Using VMotion

VMotion does require a VMKernel Port Group with a valid IP address and subnet mask for the VMotion network. A default gateway entry is not required as VMware does not support VMotion across routers or WANs. Earlier in this book, I configured a VMotion switch as an example of a VMkernel Port Group. In case you missed that part or have changed your configuration since then, I will repeat these instructions again, to save you having to flip back to the networking chapter. If you are running out of network cards on your server(s) then you could just create an additional VMkernel Port Group on an existing vSwitch.

- **Select your ESX host**.
- Click the **Configuration Tab.**
- In the **Hardware Pane**, select **Networking.**
- Click the **Add Networking…** link.
- Choose **VMKernel**, and **Click Next.**
- In the **Port Groups Properties** dialog, type a friendly name for this connection, such as **vmotion.**
- Enable **X Use this port group for VMotion.**
- Set an **IP Address** and **Subnet mask** for **VMotion**.
- You may then receive this message;

  "*There is no default gateway set. You must set a default gateway before you can use this port group. Do you want to configure it now*"

  Note:

  This message can be a little misleading because of the reference to the word "must." At this stage I've been told VMware has no intention of allowing VMotion across routers, so really the dialog box should say "may need."

  Having said this, a VMKernel port group like this could be used to access iSCSI and NAS/NFS devices. In that case, you must set a default gateway to cross the router.

VMotion by Drag-And-Drop

To initiate VMotion you can use drag-and-drop. This even allows you to drop your VM to the correct resource pools if you have them. You can also drag-and-drop multiple VMs by using shift+click to select multiples. The VMotion will do each VM in series (one after another) to preserve bandwidth on the VMotion network and prevent network timeouts.

1. **Select your VM** and **drag-and-drop** to the **ESX Host/Resource Pool.**
2. Choose to **Keep virtual machine files and virtual disks in their current location.**
3. Then **select** either **High** or **Low Priority**.

   Note:

   You get two priority settings – high and low. These do not control how quick the VMotion event is, but rather they set controls for the VM's availability. High only allows the VMotion to occur if there is no chance of the users being disconnected from the VM. In contrast, low allows the VMotion to go ahead even if a possible disconnect occurs. I would use high on a sensitive VM which is stateful, whereas I would use low on a non-mission critical stateless VM. Stateful services are ones which have almost continual IP communications such as Terminal Services, Citrix MetaFrame, or Voice-Over-IP. Stateless applications are ones which have only periodic IP communications such as databases, email, and web servers.

VMotion without Drag-and-Drop

You can start a VMotion without using drag-and drop either by right-clicking a VM and choosing Migration, or in the Summary Tab and Command Pane using the option "Migrate to New Host." This method asks more questions; you will be asked to select an ESX host as the destination and also where you wish to add it to a resource pool. Personally, I prefer to use the drag-and-drop method as you are asked less questions in the migration wizard.

Discovering CPU Incompatibilities

The Vi-Client does a very good job of stopping what could be a catastrophic event – that of moving a VM to one ESX host to another – where CPU incompatibilities exist. If this "validation" check wasn't done before the VMotion stopped then a VM would probably crash when it arrived at the destination. The VI Client fails by not telling you in a meaningful way what these CPU incompatibilities are. All the VI Client will tell you in the Summary tab of an ESX host are things like the number of CPUs, their clock speed, Vendor, and Family. Critically, it doesn't tell you anything about the incompatibilities that exist *within* the physical CPU such as SSE3, NX/XD, Intel-VT, or AMD-V.

Buy for Compatibility

One of the easiest ways to avoid CPU incompatibilities is to buy for compatibility. Simply put, this means being careful in your purchases to ensure that each ESX host has identical CPUs. This is attractive to organizations that have the purchasing power to buy blocks of servers. It's inevitable that over time you will be not able to buy the same hardware as two years ago. If this happens, we can see these new servers as representing a new "cluster" of ESX hosts that share common attributes. It's not an approach that will help in a company which is used to buying hardware on an as-needed basis.

Read the Manual

There are number of ways of finding out the attributes of your CPUs and whether your server hardware possesses compatibility issues. In recent months, both Dell and HP have released "compatibility documents" that will allow you to compare your hardware. I've yet to see an IBM document on this topic – but I dare say there will be a Redbook on the subject shortly.

For Dell Visit:

http://www.dell.com/downloads/global/solutions/vmotion_compatiblity_matix.pdf

For HP Visit:

ftp://ftp.compaq.com/pub/products/servers/vmware/vmmotion-compatibility-matrix.pdf

While these are very useful, and are the first step, they don't really help if you already have a mix of hardware vendors who use the same CPUs. Perhaps your organization deliberately does not buy from the same server vendor for strategic reasons. Perhaps there was a recent shift from purchasing hardware from HP to Dell or IBM to HP. It's entirely possible for there to be compatibility between these vendors if the chipset you are using is the same.

Using CPU Vendor Tools

Both AMD and Intel have their own tools for reporting the CPU types present in a system. I've chosen to mention these here for completeness. The downside of these tools is that they may flag attributes that are not a problem with VMotion. You can download the relevant tool from the following links.

Intel:

http://www.intel.com/support/processors/tools/piu/

AMD:

http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_871_9706,00.html

Using cupid.iso

Located on the ESX CD in the /images directory is a file called cupid.iso. This can be attached using an ILO or RAC board via "virtual media" or burned to a physical CD. The cupid.iso file is bootable and will show you the CPU characteristic of your processor. The iso is also freely available on VMware's website if you do not have access to the ESX media. You will find it under "CPU Compatibility Tools."

http://www.vmware.com/download/vi/drivers_tools.html

Figure 10.4 is a capture of the information from a HP Proliant DL385 with 1 AMD dual-core processor fitted.

Figure 10.4

```
Test: 56983: CPUID CHANGE: 340063
Reporting CPUID for 2 logical CPUs..

All CPU's are identical

Family: of model: 21 Stepping: 2

Vendor: AMD
Processor Cores: 2
SSE Support: SSE3
Supports NX/ED: Yes
Supports 64-bit Longmode: Yes
Supports 64-bit VMware: Yes
```

From this image we can see that "All the CPUs are identical" within the ESX host. This is interesting because there are some rare cases, such as after a reseller CPU upgrade, that one physical server may have different CPU types. Additionally, we can see the vendor is AMD, and my single socket contains two processor cores. There is full support for NX/XS and full support for 64-bit guest operating systems. The reference to "longmode" is actually an Intel mode. Only Intel 64-bit chips with VT Technology are supported for 64-bit guest operating systems – Intel uses the term "longmode" to describe this type of CPU.

Using 3rd party Tools – VMotion Info

Richard Garsthagen is currently Technical Marketing Manager for VMware in EMEA. He's formally a VMware Certified Instructor (VCI); in fact, he was the first instructor in EMEA for VMware. In his spare time, Richard is an enthusiastic blogger (www.run-virtual.com) and evangelist for the VirtualCenter Software Development Kit (SDK). The VirtualCenter SDK allows anyone to develop their own tools for VirtualCenter in practically any programming language they like. Richard recently wrote an application called VMotionInfo, which uses the SDK to unveil the CPU attributes of your server hardware. The really cool aspect of Richard's application is that it can be run against existing ESX hosts, without having to reboot them – as we have to with the cupid.iso method.

Figure 10.5 shows a screen grab of Richard's application taken from his website and Figure 10.6 shows a screen grab of my servers.

Figure 10.5

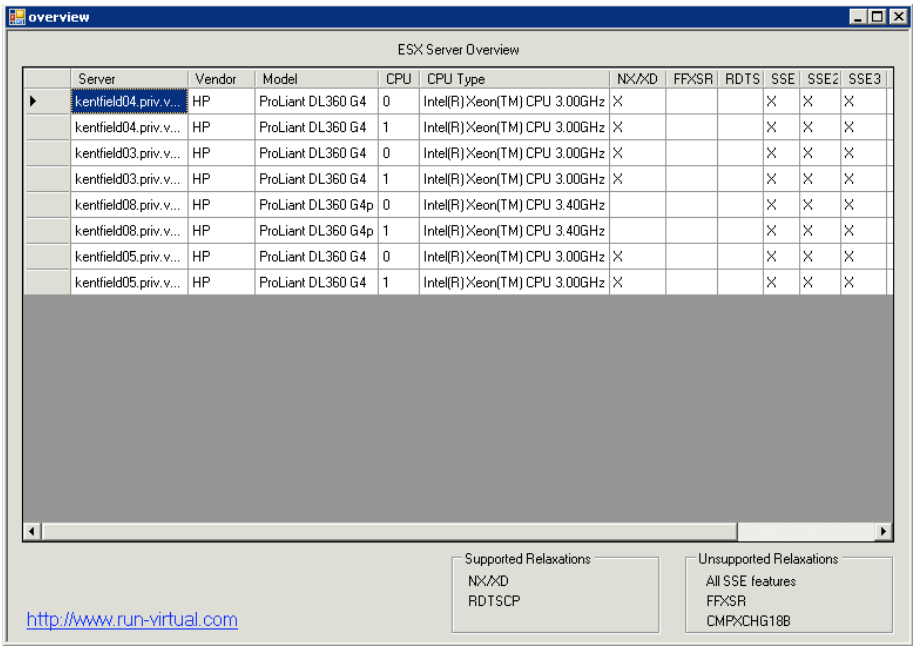
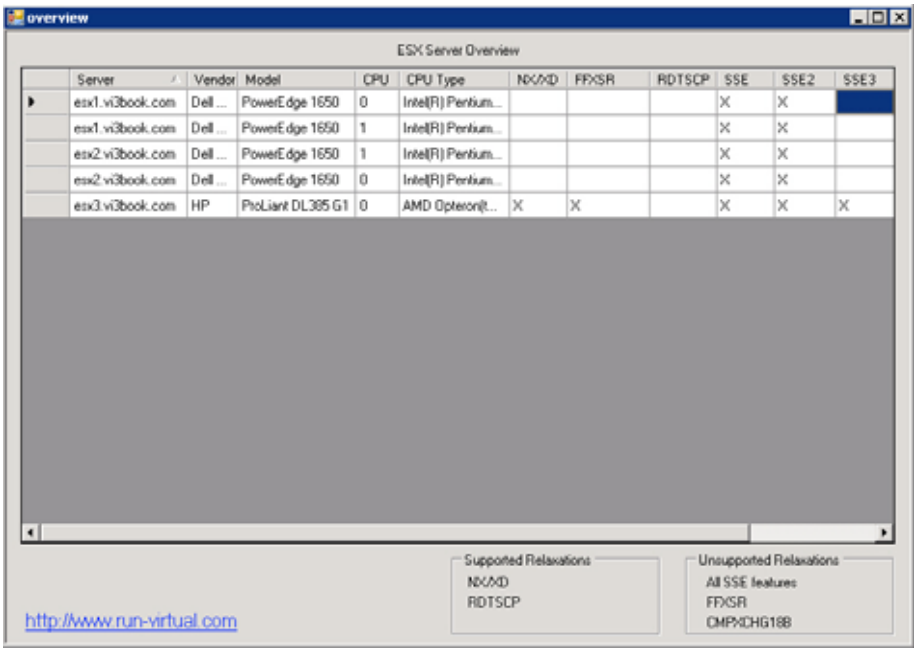


Figure 10.6



Managing CPU Incompatibilities – CPU Masks

There are some CPU incompatibilities that we can do nothing about – such as the difference between an AMD CPU and an Intel CPU. Beyond this there are tools from VMware we can use to enforce compatibility at the expense of the CPU attribute. The term we use is a "CPU Mask." A CPU mask allows us to "hide" or "mask" attributes of the physical CPU from the VM. In this scenario we might mask the Intel-VT attribute or the AMD NX attribute to allow VMotion to occur between two ESX hosts that don't share the same CPU attributes. You can see the CPU mask like putting a pair of blinders on the VM, as you would on a horse. If the VM cannot see the NX attribute, for example, it will not use it.

CPU masks are property of VM and can be found under Edit Settings, the Options Tab, and Advanced. As Figure 10.7 shows, it is possible to Disable Acceleration (Intel-VT or AMD-V) and Hide the Nx flag from the guest. The advanced button allows you to create custom CPU masks (say to hide the SSE3 attribute) for the VM specified in hexadecimal. Currently, there is little or no documentation on the Advanced button. It was enabled in the VI Client reluctantly under user pressure.

At VMworld 2006, there was a useful presentation delivered by Matthias Hausner entitled "Migrating between Apples and Oranges with VMware VMotion in VMware Infrastructure 3." It contained some useful material about the Advanced button and emerging CPU incompatibilities.

http://download3.vmware.com/vmworld/2006/tac1356.pdf

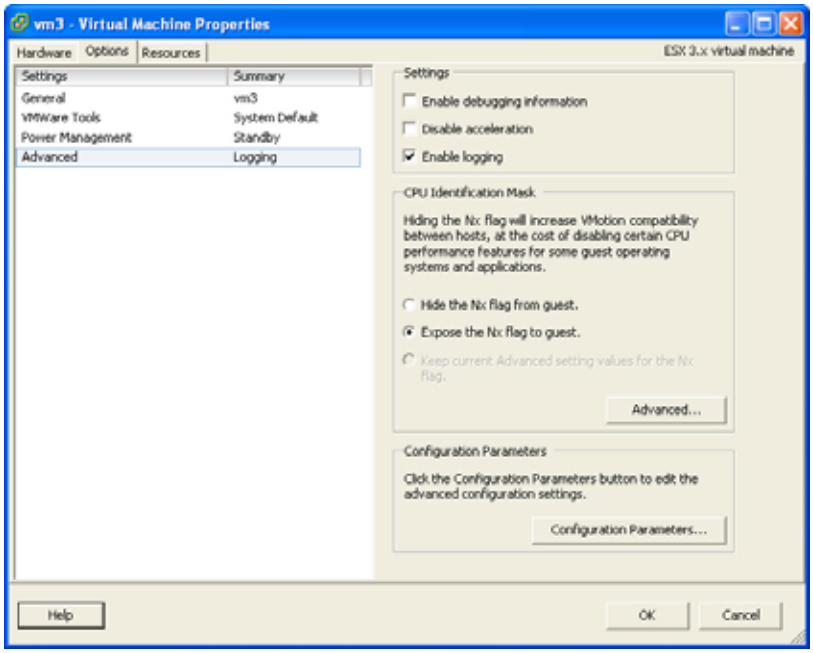
In May of 2007, VMware updated a knowledge based article surrounding this long running issue of CPU attributes. This KB article (1993) has been improved to offer more information about custom CPU masks.

http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=1993

Lastly, there is a thread where forum members discuss their relative successes and failures at creating their own custom CPU masks.

http://www.vmware.com/community/thread.jspa?threadID=50828

Figure 10.7



Resolving VMotion Errors and Warnings

Now that we are fully familiar with the typical VMotion errors and warnings you can receive, hopefully you should be well on the way to fixing these issues when they arise. However, I thought it would be useful to round-up these in one easy place so you know the most efficient ways.

The most popular errors are ones concerning active connections to removable devices. So the simplest way to turn these errors (which cannot be bypassed) into warnings (which can be bypassed) is to disconnect them. Most likely you will want to remove all errors and warnings.

Removing CD-ROM and Floppy Errors & Warnings

The best way to remove all CD-ROM floppy errors and warnings is to disconnect the devices within the VM, and then set them to use a “client device.” This removes any path statements to either local resources such as /dev/fd0 or

/dev/cdrom and also any path statements to ISOs or flp files held on a local datastore such as /vmfs/volumes/storage1. To configure this:

1. Right-click the VM.
2. Choose Edit Settings.
3. Select Floppy Device 1.
4. Remove any ticks next to Connected and Connected at power on.
5. Select Client Device.
6. Select CD/DVD Drive 1.
7. Remove any ticks next to Connected and Connected at power on.
8. Select Client Device.
9. Click OK to Virtual Machine Properties dialog.

Removing CPU Affinities

You should really remove any CPU affinities you have on VM prior to joining the ESX host it resides on into DRS Cluster. CPU Affinities and DRS clusters are incompatible with each other, and the configuration options for CPU affinities are removed from the interface in a fully-automated mode. If you join your ESX host to DRS cluster and then build your VMs you will discover that you cannot configure the CPU affinity feature at all. The root of this compatibility with DRS stems from the incompatibility with VMotion. To disable CPU affinities and return your VM to being able to execute on any CPU, change the configuration this way:

1. Right-click the VM.
2. Choose Edit Settings.
3. Select the Resources Tab.
4. Choose Advanced CPU in the dialog.
5. Select the option of No Affinity.
6. Click OK to the Virtual Machine Properties dialog.

Removing Internal Switch Errors and Warnings

If you have configured VMs to use virtual switches that are internal, there will be errors and warnings with those VMs. Remember, the goal of VMotion is to move a VM while powered *and* while users are connected. If you remember, one of the requirements of VMotion is access to the same networks for both the VM network and the VMotion network. There are two "workarounds" to this issue. Firstly, you could temporarily configure the VM to a "production" portgroup where communication would be enabled. Secondly, you could temporarily disconnect the VM from the internal switch, carry out the VMotion, and then reconnect it to a portgroup at the destination. This temporary disconnection produces a warning, rather than a hard error – and does allow you to click next to continue the VMotion.

Both of these workarounds are more than likely to disconnect users and therefore do not *strictly* meet the requirements for a true VMotion. This said, you might prefer these workarounds compared to the alternative, which is to shutdown the VM (which most definitely disconnects users!) and then "cold migrate" the VM to the new ESX host. Whatever your approach, you are likely to have to reconfigure the VM's networking and confirm that users can still connect as normal after the move has been completed. Where possible I would avoid internal switches if VMotion and DRS are important to you as they create more problems than they resolve in this aspect of the product.

To temporarily disconnect a VM from an internal switch use the following configuration:

1. Right-click the VM.
2. Choose Edit Settings.
3. Select the Network Adapter.
4. Remove any ticks for Connected and Connected at Power On.

Moving Virtual Machines - Cold Migration

When all hope is lost and you simply cannot work around the VMotion requirements, there is always cold migration. Cold migration has none of the stringent requirements of VMotion. The only requirement is that both ESX

hosts reside in the *same* datacenter. If both ESX hosts have visibility to the same storage then cold migration can be incredibly quick and the VM downtime kept to the minimum. If the two ESX hosts do not share storage then a cold migrate can take a much longer time. In the worst case scenario, where only local storage is available, it would generate network traffic on vSwitch0 as a cold migrate would use the Service Console network interface to move the VM's file from one host to another. In the best scenario your cold migrate might be only throttled by the speed of your SAN as it moves the VM's file from one SAN LUN to another in the same disk array.

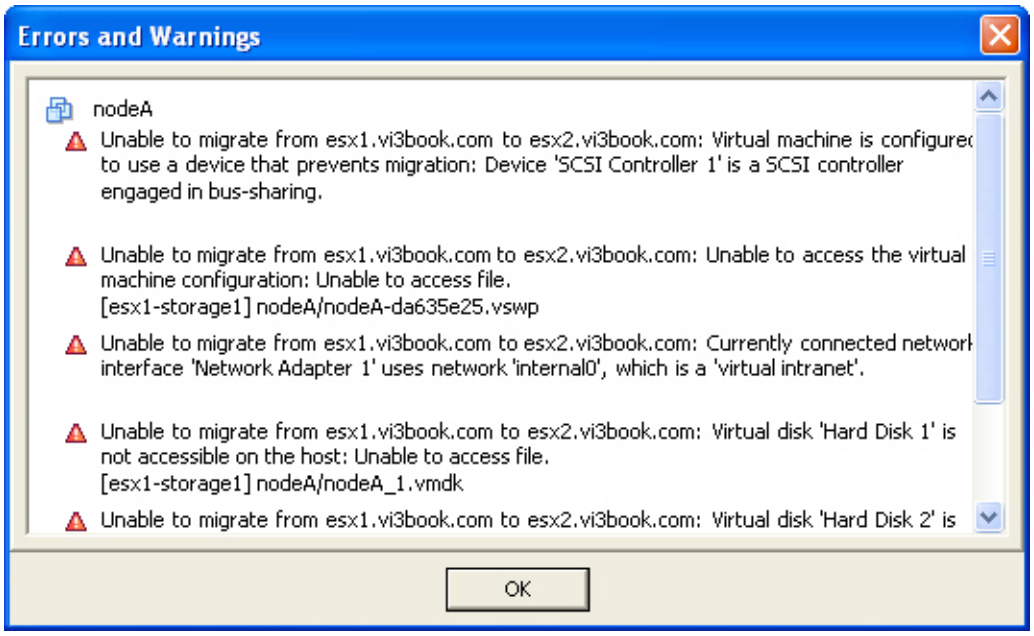
Another compelling reason to use cold migrate would be if you have a VM restriction which is not reconfigurable. For example, you might wish to move a VM cluster. As this requires that VMDKs and RDMs are stored locally for full VMware Support, then VMotion is impossible. In this case you would enact the following:

- Shutdown the secondary node in the cluster.
- Temporarily remove (but not delete) the quorum and shared RDMs. This would have to be done, otherwise the cold migrate would attempt to move them also, and they would be "locked" by other VMs in the cluster group.
- Cold migrate the VM.
- Re-add the quorum and shared RDMs.
- Power the VM Cluster back on.

As long as there is at least one cluster node up at any one time you would still achieve the VM uptime you require. To do this successfully, the quorum and shared disks would have to be on shared storage. So a Cluster-In-A-Box scenario, where all the quorum and shared virtual disks are possibly held on local storage, would have to be completely powered off and cold migrated.

Figure 10.8 shows the errors generated if a VMotion of "Cluster-In-A-Box" was attempted.

Figure 10.8



Cold Migration triggered by CPU Incompatibility

In this example I am going to do a cold migrate from one of my Dell Intel servers to my HP AMD server. The VM will remain on the same storage. Occasionally, this produces a warning about possible changes in the VM as it moves from one processor type to another. Figure 10.9 shows this dialog box. This is a benign change and should not be a cause for concern.

Figure 10.9

This was caused by moving by VM from an old Intel server to a new AMD server. New attributes such as NX/XD and Hardware Assist could be exposed to the VM whereas previously they were not present.

1. Shutdown the guest operating system in the VM.
2. Drag-and-Drop the VM to the destination ESX Host/Resource Pool, and confirm that Validation has succeeded.
3. Choose to Keep virtual machine configuration files and virtual disks in their current location.

   Note:

   If your VM is located on storage which is not shared you will find this option is unavailable. You will be only be able to choose the second option to "Move virtual machine configuration files and virtual disks."
4. Choose High Priority.
5. Once the move is complete, power on the VM.

Cold Migration for Storage Relocation

The second "hidden" usage of cold migrate is as a file management tool. It is possible to use the migration wizard to keep the VM registered on the same

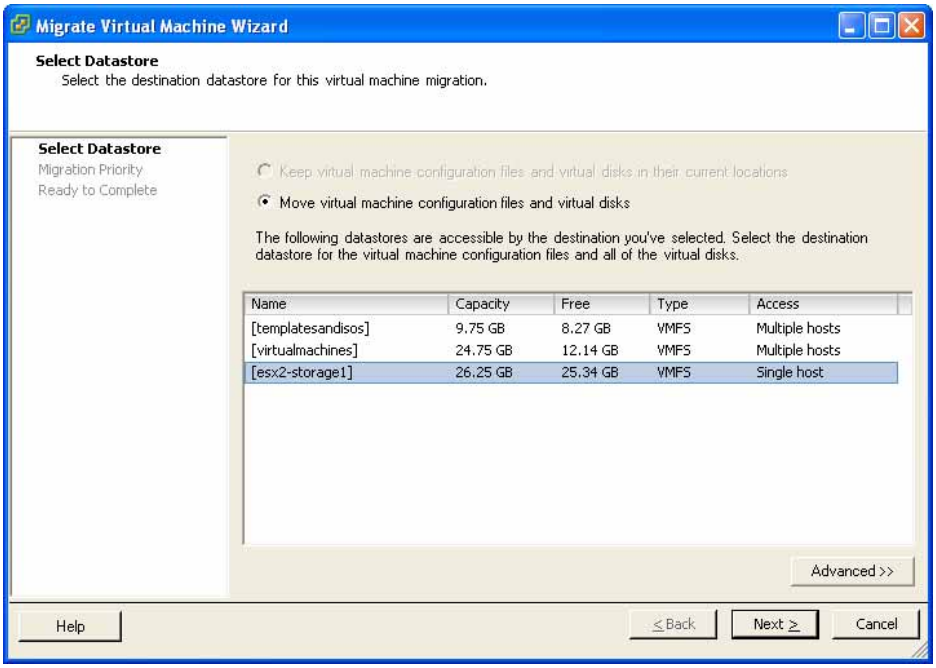
ESX host, but move the files from one datastore to another. Perhaps one of your LUNs is full and you wish to free up space. You could move a VM from one LUN to another without necessarily using it to move the VM from one ESX host to another.

1. **Shutdown your VM**. In my case, I am using nodeA.
2. **Drag-and-drop your VM to the destination ESX host/Resource Pool**.

   Note:

   You might have to confirm various warning messages if your VM cluster is configured for an internal switch as would be the case in a Cluster-In-A-Box scenario.
3. Figure 10.10 shows how the use of local storage prevents both VMotion and the "Keep virtual machine configuration files and virtual disks in their current location" in the Cluster-In-A-Box scenario.

Figure 10.10



4. **Select the required datastore** – in my case, esx2-storage1.
5. Click **Next**.
6. Choose **High Priority**.

Data-Motion

One format of moving a VM occurs between ESX 2.x.x and ESX 3.x.x and is sometimes referred to as Data Motion, or DMotion, amongst community forum members. It is hoped by many of us in the VMware Community that DMotion will be enabled *between* ESX 3.x.x hosts.

Incidentally, "DMotion" is not an official VMware term. One method of upgrading from ESX 2/VirtualCenter 1 to ESX 3/VirtualCenter2 is by moving a VM from an ESX2 VMFS2 storage to ESX3 with VMFS3 storage. This can be done without shutting down the VM. The process moves the VM from one ESX host (Version 2) to another (Version 3) as well as moving the VM's files. This is achieved by engaging an ESX 3.x.x snapshot on the VMFS3 volume, which then unlocks the disks stored in the VMFS2 files system so they can be copied to the new storage. Even though "DMotion" is not an official term, we can see its origins in the name given to snapshot delta files created during this data motion.

DMotion-scsi0:00_vm2-delta.vmdk

DMotion-scsi0:00_vm2.vmdk

Configuring Data-Motion

As you might imagine this necessitates quite a number of requirements:

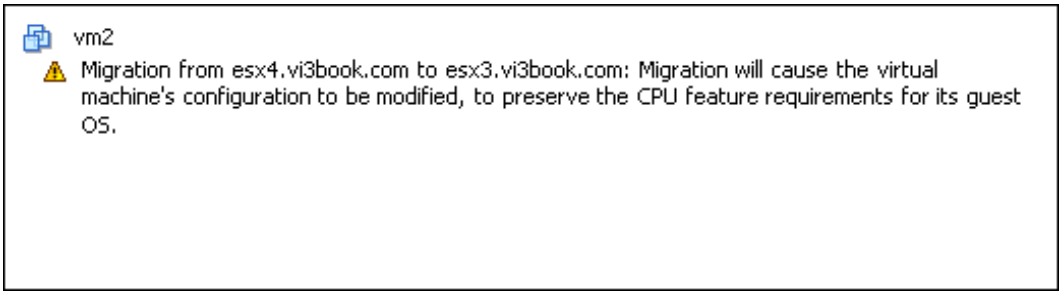
- Meet all the requirements for VMotion on both hosts
    - VMotion enabled on ESX2 and ESX3
    - vSwitch labels

- Visibility to LUNs
- CPU Compatibility
- Resolve any VM based errors and warnings (connected CD-ROM, configured for internal switches and so on)

- Software Requirements
  - ESX 3.0.1 or higher
  - VirtualCenter 2.0.1 or higher

When you come to do the move you will be warned that you may not be able to move the VM back to the source ESX server. Additionally, despite the fact your ESX hosts will have to have CPU compatibility, you will receive a warning shown in Figure 10.11.

Figure 10.11

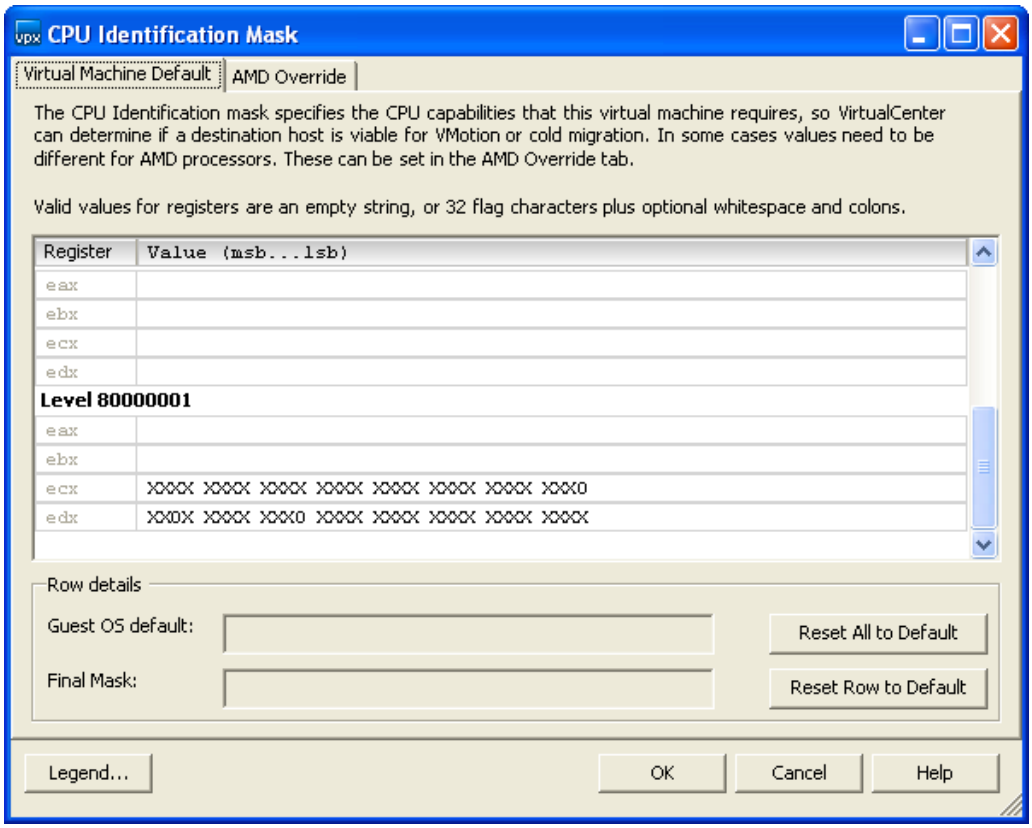


This happens because a VM running on an ESX 2.x.x host sees *less* CPU attributes than an ESX 3.x.x. This message can also appear even when you use a cold migration of the VM. After the DMotion process has completed, the VMX file is modified to include CPU masks. Below is an example of the CPU mask introduced when I DMotion'd a VM from ESX 2.5.4 to ESX 3.0.1 on two identical HP Proliant DL385s.

```
cpuid.1.eax = "----xxxxxxxx--------------------"
cpuid.80000001.ecx = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx0"
cpuid.80000001.edx = "xx0xxxxxxx0xxxxxxxxxxxxxxxxxxx"
```

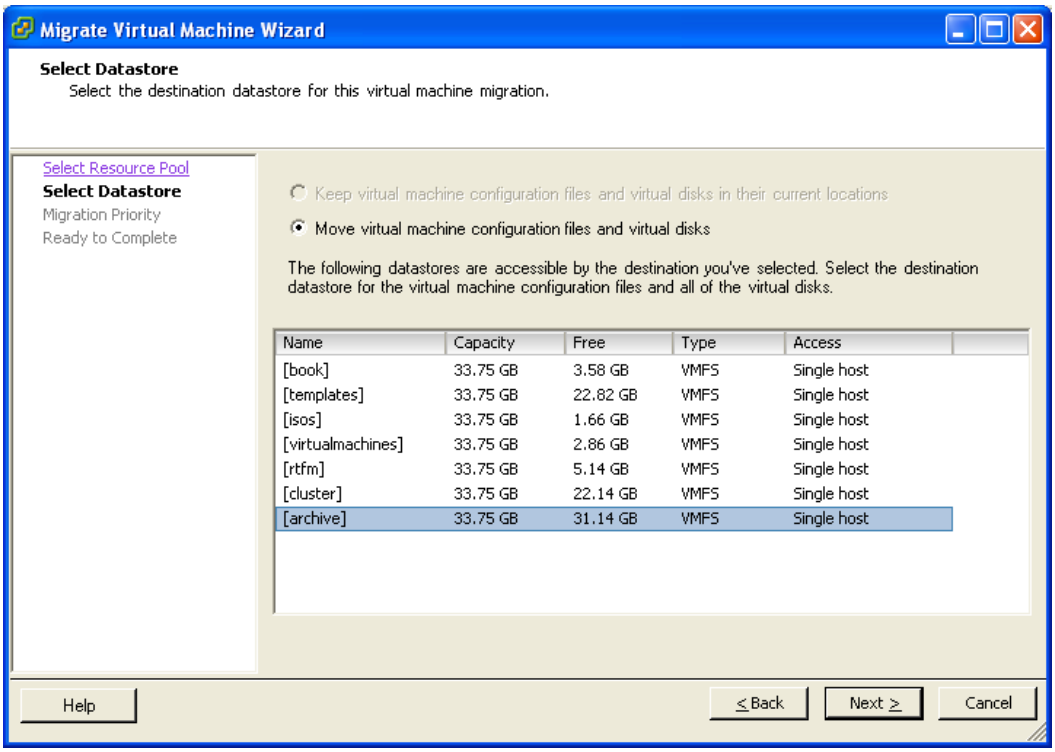
These CPU masks can be also viewed in the VI Client. Figures 10.12b shows the CPU Identification Mask on this VM (Edit Settings, Options Tab, Advanced, and Advanced Button).

Figure 10.12



DMotion is very similar to Vmotion with the following exception; because ESX 2 and ESX 3 do not share a common file-system your only option in the migration wizard is relocate the VM's storage. Figure 10.13 shows how the "Keep virtual machine configuration files and virtual disks their current locations" option is unavailable. Consequently the event is labeled in the VI Client as "Relocate Virtual Machine storage" event.

Figure 10.13



At the end of the process you should find that the VM has been moved to the new storage location, and that the snapshot files have been committed. The process will also convert the ESX 2.x.x VM into a format such that all the VM's files are held in a single directory.

VMware DRS

DRS Overview

In the simplest case, all VMware DRS is an automated VMotion. This is triggered by the system recognizing an imbalance in the resources used on each ESX host. DRS re-balances the "cluster" of ESX hosts. A lot of people mistakenly believe that what they should see is an "even" number of VMs on each ESX host. This is not the intention of DRS. After all, different VMs create different amounts of resource demands. What we are looking for is a relatively even load on ESX hosts. VMware has tested up to 32 ESX hosts in a single DRS cluster. They recommend not exceeding 16 ESX hosts. As we will see

later, VMware HA has much lower tested values, tested to 16 ESX hosts. As you are likely to use both DRS and HA together, it's perhaps worth settling for a limit of no more than 16 hosts in any DRS and HA cluster. DRS is very conservative and currently will not allow more than 60 VMotions per hour. DRS only checks for an imbalance in the cluster once every 5 minutes. Many customers worry that they may get a "DRS storm" when an ESX host fails. The argument goes something like this: an ESX host fails, triggering VMware HA which would then cause VMs to power on the remaining ESX hosts. This creates an imbalance in the cluster which then triggers a VMotion or DRS "storm." This simply does not happen, because DRS would wait at least 5 minutes before checking the cluster, and it would only offer recommendations based on your "migration threshold." This allows the administrator to control how aggressively it tries to rebalance the cluster.

Another major feature of DRS, apart from dynamic load-balancing, is "initial placement." This allows DRS to decide where to place or power on a VM for the first time. These two features of DRS closely integrate with VMware's HA software. So if an ESX host crashes say because of hardware failure – HA is in charge of detecting the crash and making sure VMs are started on other nodes in the cluster. In turn, DRS will re-balance the cluster. If the failed server comes back online again and re-joins the DRS cluster then its free capacity will be recognized, and VMotion events will be triggered to utilize the server.

DRS possesses three different levels of automation:

- **Manual**

  The Administrator is offered recommendations of where to place a VM.

  The Administrator is offered recommendations of whether to VMotion a VM.

- **Partially Automated**

  DRS decides where a VM will execute.

  The Administrator is offered recommendations of whether to VMotion a VM.

- **Fully Automated (Default)**

  DRS decides where a VM will execute.

  DRS decides whether VMotion or not, based on a threshold parameter, obeying any rules or exclusions created by the Administrator.

  Note:

  Setting to DRS manual and partial does not break VMware HA. If an ESX host fails, your VM gets powered on without asking where to power it on. If an ESX host failure occurs, the VM is powered on – and only later do you get recommendations to rebalance the cluster.

At first glance, many Administrators would choose manual as they prefer to be "in control" of their systems. However, this might not necessarily be a good decision. Firstly, if you want to power on 10 or 20 VMs simultaneously answering 10 or 20 "initial placement" dialog boxes can become irritating. Secondly, will you always have the VI Client open to see recommendations and then act on them? Thirdly, if VMotion events are included as part of change management requests then you would waste time waiting for such requests to be processed. By the time you get approval for the VMotion the performance will have changed, rendering the request invalid. Perhaps it's time to learn to lose control and have VMware move VMs for you. If you have particularly politically sensitive VMs that shouldn't be moved without prior approval, we can exclude them from the DRS process.

In addition to these 3 different levels of automation, we can set a "Migration Threshold." This allows you to say how aggressive level DRS will balance the cluster of ESX hosts. You have 5 threshold levels beginning with "conservative" and ending with "aggressive" – with the default being in the middle of these two extremes. When VM is selected as a candidate for VMotion by DRS, it will be given a "star" rating – and the threshold level ties in directly with this. So level 5 is conservative and will only trigger VMs which have a 5-star recommendation.

- **Level 1 – Conservative**

  Triggers a VMotion if the VM has a 5-star rating. 5-star recommendations are rare, and you receive this recommendation when

one of your "affinity rules" is breached. Alternatively, another cause of a 5-star recommendation could be putting an ESX host into "maintenance mode."

We will discuss affinity rules and maintenance mode shortly.

- **Level 2 – Moderately Conservative**

  Triggers a VMotion if the VM has 4 or more stars.

- **Level 3 – Default**

  Triggers a VMotion if the VM has 3 or more stars.

- **Level 4 – Moderately Aggressive**

  Triggers a VMotion if the VM has 2 or more stars.

- **Level 5 – Aggressive**

  Triggers a VMotion if the VM has 1 or more stars.

  DRS Rules and Regulations

As stated earlier, while DRS Automation Levels allow you to specify a global rule for the cluster such as "Full Automation," it is possible to have per-VM exceptions to this rule. This allows us to flag sensitive VMs as requiring administrator intervention. It is also possible to completely exclude VM's from DRS because of incompatibility reasons. A classic example of this is excluding a VM cluster because it lacks compatibility with VMotion and therefore DRS, too.

These automation levels which affect the entire cluster allow us to impose some rules and regulations. The first of these are referred to as "Affinity" and "Anti-Affinity" rules. This allows us to configure a scenario which states that two or more VMs must either be "kept together" (affinity) or "separated" (anti-affinity). So perhaps two VMs are very network-intensive and should have affinity so they remain on the same vSwitch in the same ESX host. Why? Well, because when two VMs communicate on the same vSwitch no Ethernet collisions occur – and networking is as fast as the VMKernel can manage. Perhaps you decide to keep two CPU or memory intensive VMs separate from each other so that they do not compete for those resources. Another reason to keep VMs apart is that they share the same role. It doesn't make much sense to have two identical VMs on the same ESX host which could fail – but to distribute them across many ESX hosts does make sense. This would stop an "eggs in

one basket scenario" where all your Domain Controllers, Web Servers, or Citrix Servers ended up on the same ESX host.

GOTCHA:

It is possible to configure affinity rule conflicts. The VI Client will allow you to configure a rule where VM1 loves VM2, and VM2 loves VM3, but that VM3 hates VM1. This becomes like a plot line in a soap-opera. Fortunately, the VI Client will warn you that you are creating a logical impossibility.

Resource Pools and DRS Clustering

Just as stand-alone ESX hosts can have resource pools, so can DRS clusters. In fact, it probably makes it more compelling to use resource pools in a VMware cluster, as you are more likely to want to carve up the aggregate of many ESX hosts in a cluster into small pools of resources. When you add an existing ESX stand-alone host with resource pools into a DRS cluster you will be asked what you would like to do with them. You have two options – to remove and start again or to "graft" them to the DRS cluster. If you choose this second option you will see in the DRS Cluster the name of the resource pool followed by "Grafted from esx1.vi3book.com…" indicating where the resource pool was originally created. Personally, I like to remove existing resource pools and define new ones. After all the limits, reservations, and share values imposed on a stand-alone ESX host are unlikely to be relevant to a cluster of ESX hosts that provide six times the resources. If a resource pool on stand-alone ESX host does not contain any VMs, it is not "grafted" to the DRS cluster – it is simply removed.

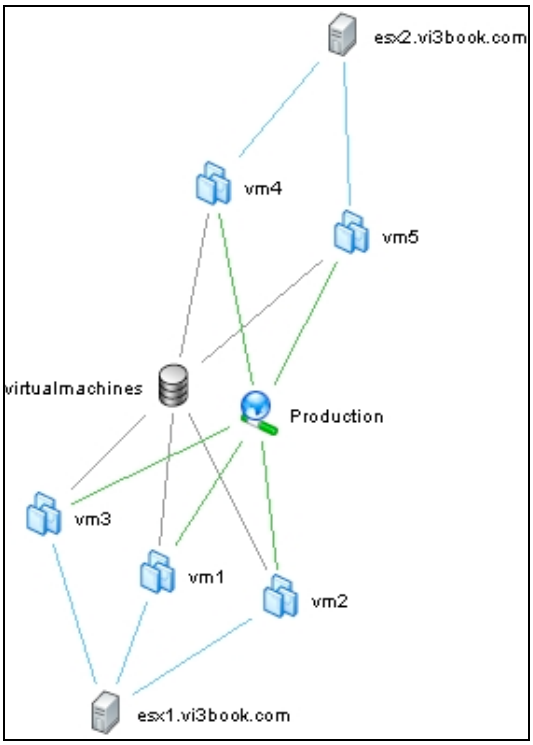
GOTCHA:

Make sure you have VMotion configured correctly before you begin. A good test is to check you can VMotion every single VM in your system. DRS currently makes no checks for VMotion whatsoever when it is enabled. In fact, you can even set up DRS without VMotion being enabled on a VMkernel port group! The most common mistake I have seen is the simplest – forgetting to put a tick in the check box to "Enable VMotion" on a VMkernel Port Group. You can confirm VMotion is enabled by looking at the Summary tab of each of

your ESX hosts. Lastly, if you ever have problems with DRS or even HA a manual VMotion will at least check that you have fulfilled the requirements of all three features.

A good way to check if you have the basic relationships in place – shared networking and shared storage - is using the maps feature. Maps allow you to see a graphical representation of your system. Figure 10.14 shows my two ESX hosts – both have access to the same storage and networking. Maps can be saved in a JPEG, BMP, or EMF format for documentation purposes using in the VI Client under File, Export, and Export Maps.

Figure 10.14



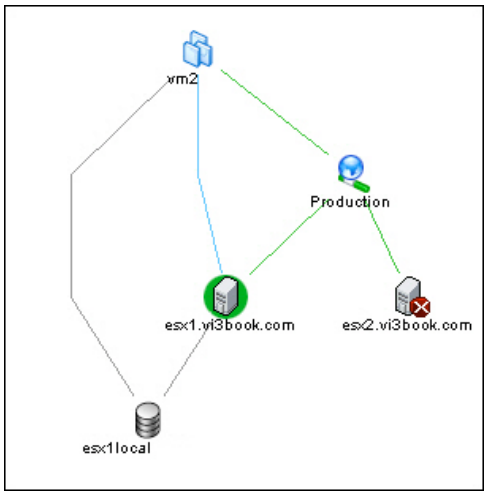
Note:

While currently VMware offers no method of converting maps into a Microsoft Visio format, there are companies who have software that will. Veeam software has recently released a Reporter tool which does precisely that.

http://www.veeam.com/veeam_reporter.asp

Figure 10.15 shows a situation where a VM is not on shared storage in the DRS cluster. Notice the red X next to the ESX host.

Figure 10.15



Setting up a DRS Cluster – Manual

In this scenario, we will initially setup the cluster in manual mode. This is so you can experience the recommendation system and have full control before switching to fully automated mode. I won't be showing the partially automated mode in this chapter as it is merely a hybrid of manual and fully automated.

I will be creating my DRS cluster with the Intel Host folder. This makes sense because only my Intel Server has the CPU compatibility required for VMotion and DRS.

Creating the DRS Cluster

1. In the Inventory view of Hosts & Clusters.
2. Right-click and choose New Cluster.
3. Type in a friendly name for your cluster, such as Intel Cluster and select VMware DRS.
4. Change the automation level to Manual.

5. Click Next and Finish.

Adding ESX hosts

1. Drag and Drop your first ESX host into the cluster.
2. Accept the default which removes any existing host-based resource pools.
3. Next, drag-and-drop your second ESX host into the cluster.

   Note:

   Continue step 3 if you have additional ESX hosts which fulfill the VMotion requirements.

Viewing the Cluster Resources

Figure 10.16 shows the main summary tab for my Intel Cluster. Here we can see that DRS is enabled and that my two ESX hosts are offering 2 CPUs with a collective amount of 5GHz of CPU time and 4GB of RAM. Under the VMware DRS pane I can see that there are already some recommendations generated by DRS as there is a blue link stating "Migration Recommendations: 3." Below that, we have the DRS Resource Distribution. The first chart shows how balanced my hosts are. The two blue and yellow columns represent two ESX hosts that are imbalanced. One is not using much CPU or memory (it lies in the 0-10 and 10-20 range), and the other is using much more resources (it lies in the 50-60 and 60-70 range). It would be more ideal if these columns were closer together which indicates my two ESX hosts are consuming the same amount of resources. The second charts how many resources are available in the cluster. My VMs are mainly idle and therefore 90% of the cluster resources are not in use. It would be more desirable if these columns were further to the left – as this would indicate that I would be getting more VMs from the resources I possess.

Figure 10.16

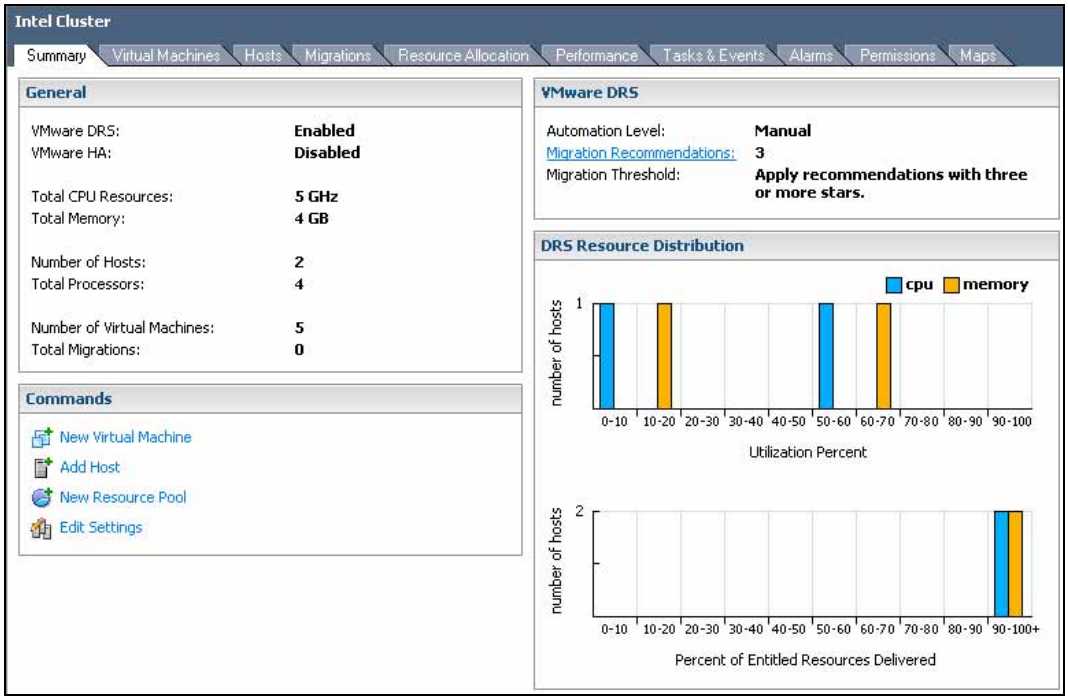
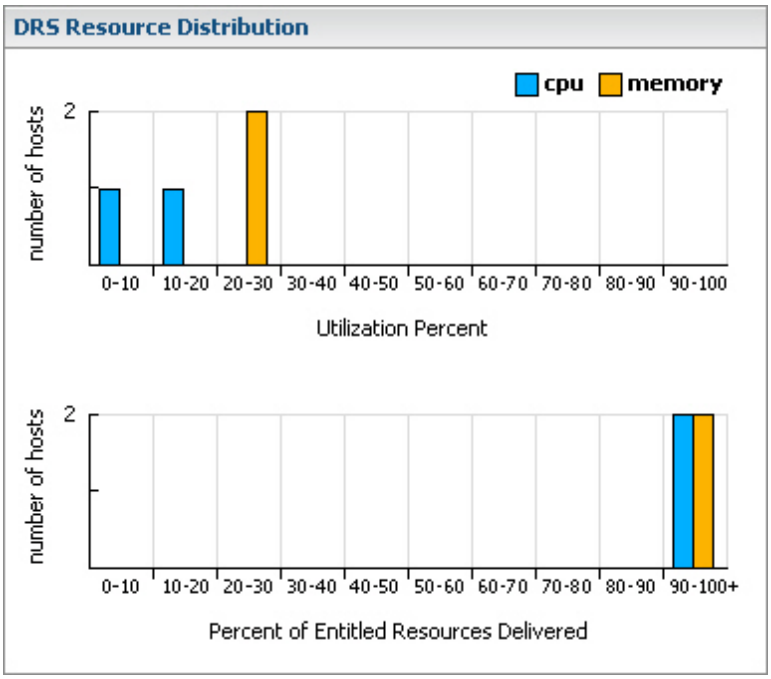


Figure 10.17 shows a DRS cluster where full automation has been enabled, and DRS has been allowed to balance the ESX hosts. Here we can see that both hosts are equally using RAM, but are not completely balanced for CPU usage.

Figure 10.17



Applying Recommendations

You can see recommendations under the "Migrations" tab on a DRS cluster. Alternatively, if you already have recommendations you can click the "Migration Recommendation" hyper-link as shown in Figure 10.18 above which will take you to the same location.

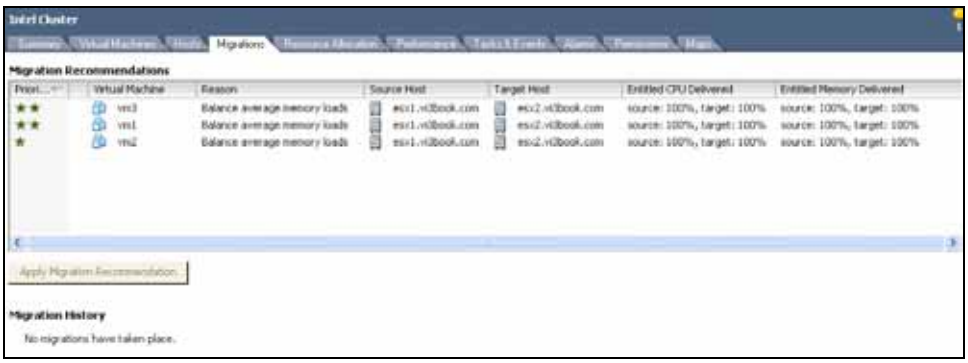
1. Click the Migration Tab.
2. Select a VM with the highest star rating and click the Apply Recommendation.

   TIP:

   If you are not seeing the migration tab you can sometimes trigger it by adjusting your threshold to be more aggressive. Alternatively, try running the cpubusy.vbs file in *one* of the VMs. Lastly, try using VMotion to overburden one of your ESX hosts to create an imbalance in the cluster.

Figure 10.18 shows a typical migration tab summary, with each VM flagged with a star-rating together with the reason, in this case "Balance average memory loads." Incidentally, it is possible to select multiple VMs and apply the migration recommendation. DRS will do each VMotion one after the other in series.

Figure 10.18



Initial Placement Questions

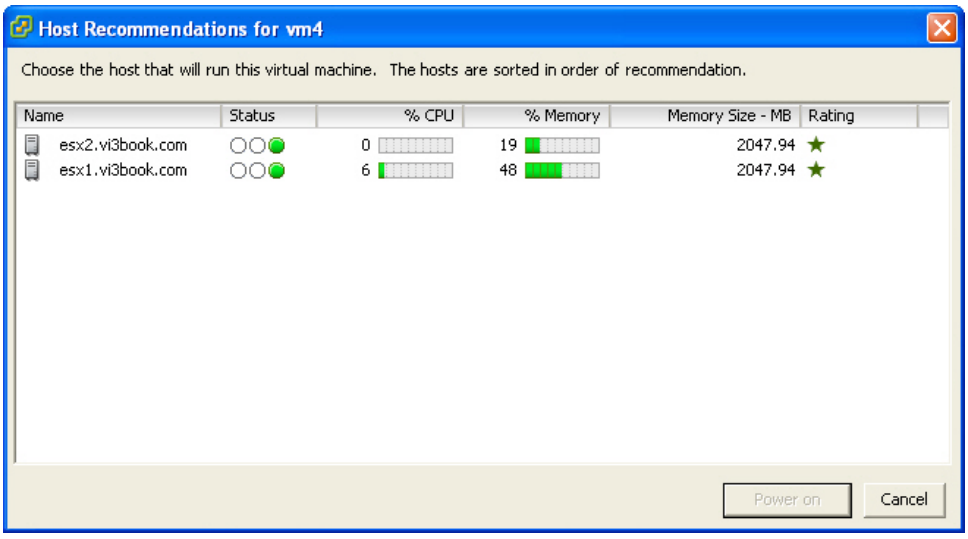
If you power off a VM, and then power it on again using either manual or partial automated modes, you should be confronted with a dialog box.

1. Power off and Power on a VM.
2. Select the ESX host you would prefer the VM to run on.
3. Click the Power On button.

   Note:

   The dialog box is sorted in order of recommendation – making the ESX host at the top of the list the most appropriate ESX host to power the VM on. Figure 10.19 shows that ESX2 is hardly using any CPU resources and is only using 19% of its memory, compared to ESX1 which is using 48% of its memory.

Figure 10.19



Configuring DRS Cluster Rules

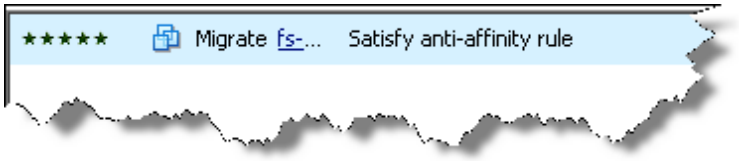
As stated before, we have two types of control on individual VMs. We can create affinity (keep VMs together) and anti-affinity (separate VMs) rules. Additionally, we can set a custom automation level on sensitive VMs. Remember, the only time you receive a 5-star recommendation is when one of your affinity rules is broken.

Creating an Anti-Affinity Rule:

1. **Right-click** the Cluster.
2. Choose **Edit Settings**.
3. Select **Rules**.
4. Click the **Add** button.
5. **Type in a friendly name for the rule**, like VM1 hates VM2.
6. Under the **Type** option choose "**Separate Virtual Machines**."
7. **Select VM1** and **VM2** and Click **OK**.
8. Click **OK** to create your rule.

Figure 10.20 shows a situation where VM1 and VM2 are residing on the same ESX host. This is in breach of my anti-affinity rule that stated they should be kept apart – so the reason for the migration is to "Satisfy anti-affinity rule." Of course, you will receive a similar recommendation if you have created an affinity rule, and the two VMs are not on the same ESX host.

Figure 10.20



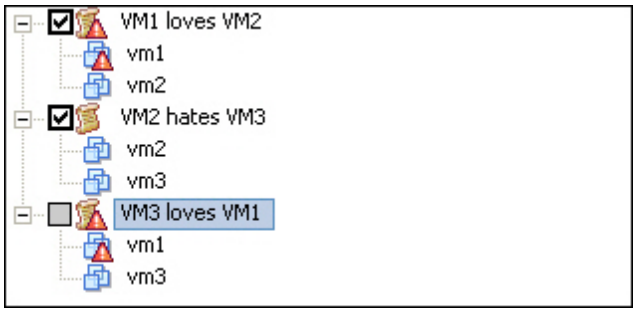
GOTCHA:

VirtualCenter *does* check your affinity rules when you first power on a VM. However, it *does not* check your affinity rules to either alert or stop the administrator from carrying out manual VMotion events which breach affinity rules. If an administrator carries out a manual VMotion it is entirely possible to accidentally trigger a 5-star recommendation by manually putting two VMs together or apart that should be kept separated or together, respectively. Interestingly, if you use fully automated mode this still happens, but what DRS does is automatically carry out another VMotion to undo this administrative "error."

GOTCHA:

As stated before, rule conflicts are possible. Figure 10.21 shows this occurrence.

Figure 10.21



Custom Automation Levels

As you might remember, VM clustering in any of its forms is incompatible with VMotion and DRS. A good example of custom automation levels is completely disabling DRS on VMs that run some kind of clustering software.

1. **Right-click** the cluster.
2. Choose **Edit settings**.
3. In the dialog box, select **Virtual Machine options**.
4. Select **nodeA** and choose **Disabled**.
5. Select **nodeB** and choose **Disabled**.

Figure 10.22 shows nodeA and B disabled from the DRS functionality.

Figure 10.22

Virtual Machine	Automation Level
nodeA	Disabled
nodeB	Disabled
vm1	Default (Manual)
vm2	Default (Manual)
vm3	Default (Manual)
vm4	Default (Manual)

Note:

Even with this option to exclude the VM clusters from DRS, in a manual or partial automation mode, you are still asked which ESX host to power on nodeA or nodeB. You will find there is only one server on the list because the VMDK and RDM files are on local storage for VMware Support purposes. This isn't a problem, it's just an unnecessary dialog box.

DRS Automation Levels and Maintenance Mode

Maintenance mode is an option available next to shutdown and reboot on an ESX host. You have probably shutdown and rebooted your ESX host a couple of times since you started reading this book, so you might ask, "Why leave this topic until now?" Well, the main reason is that maintenance mode becomes really interesting and cool when combined with DRS. It was felt it might be relevant to cover what maintenance mode is for in the context of DRS.

Stated very simply, maintenance mode is an isolation state used whenever you need to carry out critical ESX host tasks such as firmware, ESX host, memory, and CPU upgrades. This isolation state will prevent other VirtualCenter users from creating new VMs and powering them on your ESX host. It will also stop any manual VMotion events created by an administrator or automatic VMotion events generated by DRS. Maintenance mode does survive reboots – this allow the administrator time to confirm that their changes have been effective (like adding a new hardware device such as a NIC or a HBA) before VM's can be execute on the ESX host.

How maintenance mode works will differ depending on whether or not the ESX host is in a cluster. If it is causing you a problem, maintenance mode can be cancelled at any time by right clicking the task in the task pane and choosing cancel.

Below is a list of what maintenance mode will do depending on your configuration:

- **Stand-alone ESX host** (Creates a pop-dialog box warning)

Administrator must VMotion all VMs to another ESX host manually. If VMotion is not available administrator must power off all VMs before maintenance mode is triggered.

- **Manual or Partial DRS Automation** (Creates a pop-dialog box warning)

  Generates recommendations which have to be applied to evacuate the ESX host of all currently running VMs. Figure 10.23 shows a migration recommendation triggered by maintenance mode.

Figure 10.23

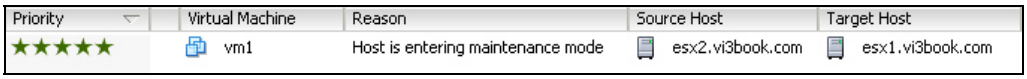


Figure 10.24 shows a very rare occurrence in a production environment. This is a recommendation triggered by maintenance mode where DRS has no choice but to break one or more affinity rules. It happens frequently on two node DRS clusters. Here, VM1 and VM2 must be kept separate on ESX1 and ESX2, but ESX1 is forced into maintenance mode. DRS has no other option than to provide a recommendation that would break an anti-affinity rule. The small icon in the 2nd column after the star rating (highlighted in the screen capture) indicates when this is happening.

Figure 10.24

- **Fully Automated DRS** (No Dialog Box Pop-ups)

  DRS automatically moves the VMs of the ESX host to other nodes in the cluster, obeying your rules and regulations where possible.

Configuring Full Automated DRS using Maintenance Mode

1. **Right-click** your cluster.
2. Choose **Edit Settings**.
3. **Change the Automation Level** to **Fully Automated**.
4. Click **OK**.
5. Now **select one of your ESX hosts in the cluster**, in the **Summary Tab** select **Enter Maintenance Mode**.

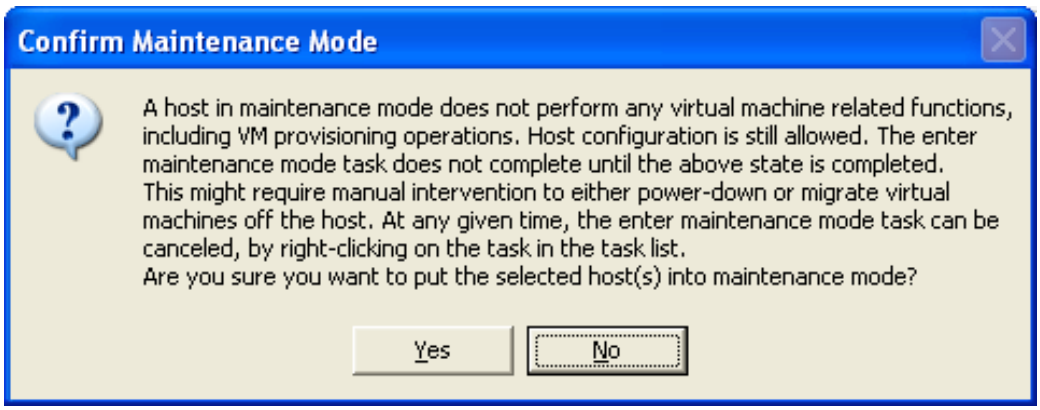
   Note:

   This should trigger an automatic VMotion of all your VM's from that host on to other ESX servers.

Maintenance Mode Hangs

One of the most common problems people experience in maintenance mode is that it hangs and does not complete. When using maintenance mode, either when an ESX host is in stand-alone mode or in a DRS/HA cluster mode, the VMs currently residing on the host must either be moved (via VMotion) or powered off. If a VM cannot be moved or powered off, the maintenance mode waits for you to resolve the problem. Unfortunately, the VI Client currently doesn't prompt you that maintenance mode is waiting nor does the VI Client tell you *why* maintenance mode is hanging. If you carefully read the maintenance mode dialog the VI Client does warn you about the requirements for maintenance mode to complete. Figure 10.25 is the dialog box that appears.

Figure 10.25



This is usually caused when DRS is in a fully automated mode but is unable to move all the VMs to other ESX hosts in the cluster. When this happens, maintenance mode just sits there – no warnings or pop-ups appear. If you try powering on a VM you will receive the message:

"The operation is not allowed in the current state."

I've seen this message in class, and I must admit the first time it happened, it had me stumped. It turned out that one of the students had attempted to enter maintenance mode earlier – and then carried on with other tasks. This meant that at first glance the "Task" pane view at the bottom of the VI Client did not show the message "Entering maintenance mode…."

While in the process of entering maintenance mode you will find you cannot carry out many tasks except for ones that would resolve the problem like the following:

- Manual VMotion
- Powering off a VM
- Cancelling Maintenance Mode

Nine times out of ten you will find that there is a property of the VM which stops the automated VMotion triggered by attempting maintenance mode. One tip is to try a manual VMotion, as this should create a meaningful pop-up message that will lead to resolving the problem and finally completing the entry into maintenance mode. Typically DRS' inability to move a VM is caused by the VM errors mentioned earlier that are listed here:

- Connected removable to local storage or devices
- Connected internal switches
- RDM's to LUN not presented to other ESX hosts
- VM Clustering
- CPU Affinities

Where are my VMs running?

There is a short answer to this – we don't know. Once you have engaged fully automated mode you will not really know (depending on how conservative or aggressive you have configured DRS in the settings) from one hour to the next where your VM will be running. For some people this is a difficult concept as they are so tied to the physical world. Remember ESX is not grid or parallel processing – so fundamentally a VM only executes on one host at one time. If you want to know on which ESX host VM is currently running there are two main methods.

Firstly, on any object that contains your VM (root container, datacenter, folder, or cluster) select the tab called "Virtual Machines;" right-click the descriptive names of one of the columns such as "name," and enable the option "host." Secondly, if you find your VM in the inventory in the Summary tab and the General pane, there should be a field that specifies its state and on which ESX host it is currently executing.

Resource Pools on a DRS Cluster

Of course, it's possible to create resource pools on a DRS cluster. In fact, it probably makes more sense that you would want to carve up the resources of a cluster into smaller pools. Resource pools on a DRS cluster work in exactly the same way as an individual host. Although the VI Client gives the impression that the Resource Pools "hang off" the cluster what happens is that these resource pools get created on each ESX host. Fundamentally, a VM is executed on an ESX host, not across ESX hosts. This would require some type of "grid computing" hardware which currently is cost prohibitive for most organizations. Therefore, when a VM demands its reservation for memory, for example, that reservation must be found physically on an ESX host. Remember that resource pools and DRS clusters represent a logical grouping of resources – we are still constrained by the physical limits of each of our servers. Lastly, when a VM is moved from one ESX host to another in DRS, its resource pool membership remains the same. Again, many people commonly think that the resource pool represents a physical location – when in fact it is merely a software concept that allows allocating resources and controlling performance of our VMs.

When you create resource pools in DRS cluster what actually happens is the resource pool is created on each ESX host. After creating resource pools on DRS cluster, if you subsequently open the VI Client on each ESX host (which is not recommend as it could cause integrity issues with the cluster) you will see every host in the DRS cluster shares the same resource pool names.

Resource pools are used to manage internal ESX processes as well. On every ESX host there is a system "root resource" pool. Contained within this resource pool are child resource pools used to manage the VMkernel tasks. There is no practical usage of this within the ESX host for a production environment. But for completeness it was felt you should know about them. You can see the ESX host root resource pool by doing the following.

1. Choose your ESX host.
2. Select the **Configuration** Tab.
3. Select **System Resource Allocation**.
4. Click the **Advanced** option.

Figure 10.26 shows the child resource pools. The nursery resource pool has a special function. It's where baby processes are born and reared, and they move to other resource pools once they are old enough to play with the grown-ups.

Figure 10.26

VMware HA

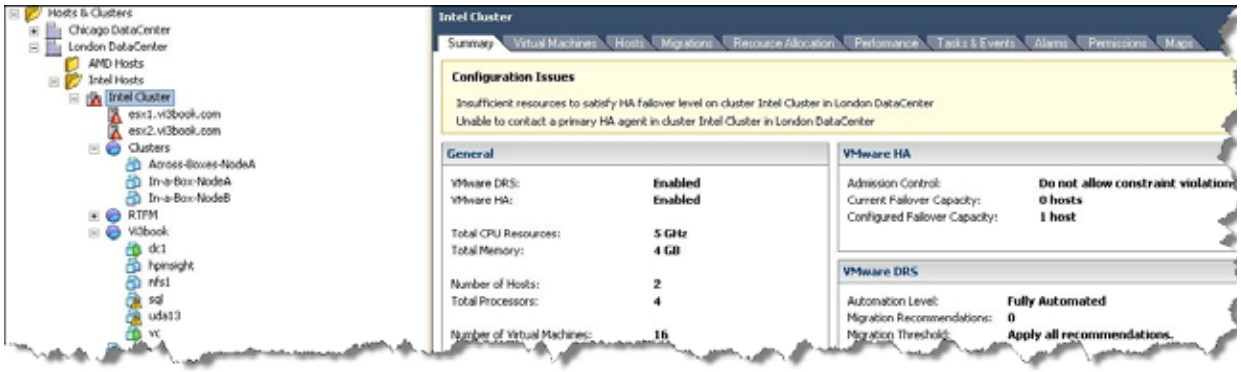
VMware HA is not actually an original VMware product. VMware procured a license for Legato's Automated Availability Management software and re-engineered it to work with VMs. Legato is now owned by VMware's parent company EMC and has been re-branded as EMC Autostart. However, the directories in /opt still retain the original Legato folder structure. VMware has made substantial changes to AAM and has access to the source code, so we expect improvements in VMware HA in the future.

VMware's VirtualCenter Agent interfaces with a VMAP API which acts as an intermediary layer to the AAM software. VirtualCenter is required to *configure* HA but is not required for HA to function. So, even if your VirtualCenter is down or dead then HA will continue to do its job. The architecture of HA is a peer-to-peer with each ESX host in a "mesh" topology constantly checking each other for functionality. This check is done via the Service Console vSwitch.

VMware HA has a very simple goal: when an ESX host crashes so do the VMs on the failed host. However, within 15 seconds the other ESX hosts in the same HA cluster detect that the ESX host has failed and power on the VMs on the remaining hosts. If DRS is enabled it will then re-balance the cluster. DRS will also detect when the failed host is available again and re-balance the cluster. Critically, what VMware HA does not do is manage crashed VMs. To deal with that scenario you need either VM cluster (covered earlier in this chapter) or script triggered by an alarm to reboot a failed VM. When an ESX host fails the selection process currently goes alphabetically through the remaining servers and powers on a VM on the first server that has sufficient resources to run that VM. This is intended to get the VM powered on and back online as quickly as possible. DRS is very conservative in its checks. Don't expect just because an ESX host crashes that suddenly you would get a lot of VMotion events or recommendations. VMware plans to improve this selection by name first and capacity second in future versions of VirtualCenter. The long term plan is to improve the algorithm so selection is calculated by selecting the ESX with the most unreserved capacity.

As with VMotion and DRS, HA also requires shared storage and shared networking. The only thing that HA does not require is CPU compatibility. After all, the VM is powered off when the ESX host fails and powered on a new ESX host when HA detects the failure. HA requires DNS forward (name to IP address) name resolution. There is some documentation from VMware that states that reverse DNS name resolution is also a requirement. This is true but only if you add ESX hosts into VirtualCenter by the IP address. If you add your ESX hosts to VirtualCenter by FQDN then all you need to do is forward lookups. Perhaps the best practice is to cover your bases and ensure before forward and reverse that lookups are configured in DNS. As with the license server, sort out your DNS issue before you even begin, and the setup and configuration should be relatively painless. Lastly, there is currently a limit with DNS names longer than 29 characters. VMware has promised to fix this issue in future releases. As an experiment I switched off my DNS servers after the cluster had be configured. Figure 10.27 shows the effect of no DNS name resolution on VMware HA.

Figure 10.27



HA and Resource Management

HA also introduces some careful re-consideration of resource management. After all, if I have 7 ESX hosts in a cluster running 70 VMs – and then I lose an ESX host – will the remaining ESX host be able to run the *same* number of VMs on 6 ESX hosts rather than 7? One practical response is to design a system that has +1 redundancy. So instead of 7 servers, we would have 8. If one failed we would still be able to achieve the same performance with fewer nodes. Of course, the buck has to stop somewhere. Could we tolerate 2, 3, or 4 ESX host fails, and still run the same number of VMs? The HA software allows us to set such tolerances of ESX host failure during its configuration but the question is both an operational and design issue.

Another response to this question of resources is to only power on the remaining ESX hosts the right number of VM's needed to give "acceptable usage." Perhaps you have 2 domain controller VMs running across 10 ESX hosts in HA cluster. You know from testing and experience, that for acceptable usage you only need 10 up and running at any one time. VMware HA allows you to disable VMs from HA altogether and also set "priorities" for which VMs are started first.

HA and the "Split Brain" Phenomena

If you are experienced in the world of conventional Windows or Linux Clustering you might already be familiar with the term "split brain." It's a kind of clustering schizophrenia. It describes a situation where more than one node thinks

that other nodes have failed. This is like some people believing they are Napoleon, and everyone else is crazy; the host believes it is fine and that the others are the problem.

As mentioned earlier, the mesh topology that HA creates is driven by the Service Console vSwitch. If an ESX host in an HA cluster experiences an NIC failure or cable break then this can trigger the "split brain" scenario – this is sometimes referred to as the "isolated host." This isolated ESX host would mistakenly believe that the other 7 ESX hosts had crashed, while all the other ESX hosts believe that the bad ESX host has crashed as well. In fact, the VMs are running perfectly fine on all ESX hosts.

What is VMware HA's default behavior when split brain occurs? The default is that the isolated host (the ESX host with the failed Service Console network) powers off all its VMs. This causes the VM's files to be "unlocked" in the shared storage. This then allows the other ESX hosts to assume control, and power on the VMs that were previously running on the isolated host. The assumption in the default is, if split-brain occurs, then begin HA failover to the remaining ESX hosts. VMware HA does have an over-ride option for this default, allowing the Administrator to configure a VM to stay powered on, when the split brain event happens.

One way to protect yourself from the split brain phenomena (apart from regular trips to an expensive psychotherapist) is make sure that the ESX hosts have redundancy on the Service Console networking. One method would be to use a second NIC behind vSwitch0, and perhaps patch it to a different physical switch. This would protect HA from NIC, cable, and switch failures. Alternatively, you could add a second Service Console port group to a switch used for another aspect of your virtual infrastructure such as VMotion.

Setting up VMware HA

I'll assume you already have a DRS cluster setup and therefore all that HA needs is enabling on an existing cluster. If this is not the case then you will need to create a cluster and drag-and-drop your ESX hosts into it.

When you enable an ESX host cluster, VMware will trigger the AAM software on each host – one at a time. This can cause some benign alerts. Clearly, we cannot have an HA cluster with just one ESX host. Until the second ESX host is configured, the cluster will have alerts and warnings on it. These will not disappear until you have at least two ESX hosts in the cluster enabled for HA.

1. Right-click the Cluster.
2. Choose Edit Settings.
3. Click Enable VMware HA.

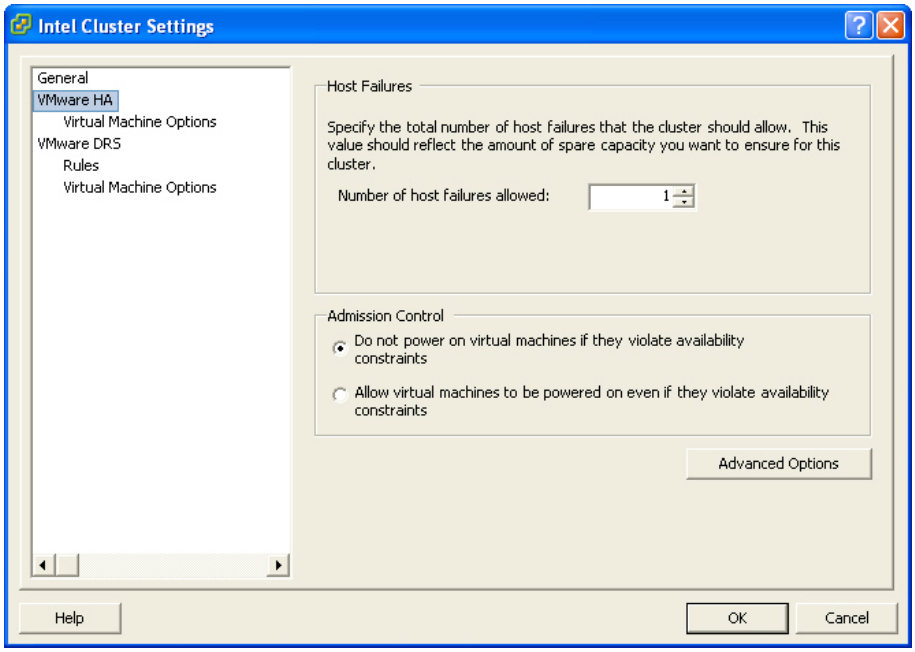
   Note:

   This will add an additional set of HA options in the pane to the left-hand side of the dialog box.

4. Under the option "General," select VMware HA. This will open some settings associated with the cluster.

Figure 10.28 shows the main configuration options for HA.

Figure 10.28



We can control the number of ESX host failures we tolerate until the HA clustering stops powering on VMs on the remaining hosts. The maximum number of hosts we can tolerate is 4. Remember VMware recommends no-more than 16 ESX hosts per HA cluster, so in reality this means we could tolerate a quarter of the ESX hosts failing. If a 5th ESX host failed, HA would not power up the lost VMs on the remaining 3 nodes in the cluster. You might ask why the number 4 is used here. In Legato's AAM, the system is based around primary and backup AAM servers. The maximum number of primaries is 4, and if one fails an election process would promote a backup to be a primary. If all 4 primaries were lost simultaneously (although this is an extremely unlikely event, I can see it happening only if you had a blade enclosure failure), then the AAM software would be broken.

The more hosts you have the better utilization you get from a resource perspective is another way of looking at this issue. If I have 2 nodes I can only load them to 45% each. If one failed the remaining ESX host would have to have 90% capacity (with 10% reserved for the Service Console and the virtualization overhead) to provide the same resources. If I have 4 nodes, I can load them at 65% each. If one failed the remaining three ESX hosts would each provide 25% of the resources needed to make up for the loss of one ESX host? The moral of the story is the more ESX hosts you have, the more you can load them – and still tolerate an ESX host failure. Remember though, at the end of the day, a VM executes on a given ESX host. If that VM has a memory reservation – it must be found in physical memory. So although the DRS cluster might have 1GB of free memory left in the cluster and 4 ESX hosts, this 1GB of memory is actually not completely available. The "spare capacity" in cluster is a logical representation of capacity, not a physical representation of where that memory actually resides.

Below the "number of host failures" allowed we have options associated with "Admission Control." These are the rather confusingly labeled options called "Allow virtual machines to be powered on even if they violate availability constraints" and "Allow virtual machines to be powered on only if they do not violate availability constraints." What do these very long sentences mean? Perhaps they are best explained with a scenario. If you had two ESX hosts with the "number of host failures allowed" set to 1 and one of them failed the default is set in such a way that you wouldn't be able to power on any new VMs. If the second option was engaged, then you would. The assumption is in the default.

There is little point in powering new VMs if you have had host failures as there would be fewer resources.

Under VMware HA we have Virtual Machine options. This allows us to set different start-up priorities for VMs and also configure the "isolation response" should an ESX host suffer from the split brain phenomena.

VMware HA and VM Clustering

To gain full support from VMware, the virtual disks and RDMs used with VM Clusters (using such software as Microsoft Clustering Service) must be on *local storage.* This effectively excludes them from being used with VMotion, DRS, and HA. The software is so compatible that you do not need to dedicate hardware to your VM clusters. You can still run them; they just won't benefit from these advanced VMware features.

Testing VMware HA

There are a couple of ways to test HA, but by far the most convincing is to remove the power from one of the ESX hosts. If you feel uncomfortable with a hard test, issuing a reboot (without maintenance mode) should cause the ESX host to emulate the same hard failure.

If you wish to simulate the isolation response where an ESX host appears to have failed because of lost of service console connectivity you can use the following command:

esxcfg-vswif -d vswif0

This disables the Service Console vswif interface. The command esxcfg-vswif -e vswif0 will re-enable it again.

Monitoring a DRS and HA Cluster

Both DRS and HA will give you status information about the integrity of the cluster. In fact, you may have already seen these notifications during the configuration of HA.

On the cluster icon itself you will find different icons to represent the state of the cluster. The red icon indicates a configuration problem, and the yellow icon indicates the server with the DRS or HA issue. In HA this happens every time you add a second host in the cluster. It takes some time for each of my ESX servers to be enabled for HA, and when you have an HA cluster of just one server then HA misreports this as the failure of the cluster.

The most common reason for the red icon on a DRS or HA cluster is administrators incorrectly using the VI Client. Once you have VMware clustering enabled you should not "point" the VI Client directly to the ESX host; this bypasses VirtualCenter. The only reason to do this is if your VirtualCenter environment has malfunctioned – and in that case you would be better served by resolving the VirtualCenter server.

If the cluster icon has a yellow exclamation mark this indicates that resources are scarce and reservations may not be met. In this case you could experience admission control style problems. The most common reason for this is a number of ESX hosts have become unavailable and as a result the cluster has experienced a drop in total capacity.

Summary

In this chapter we have looked at the many ways you can offer high availability to both the VM and the ESX host – by either VM clustering or HA clustering. We also looked at how DRS and HA are so closely integrated with each other you are unlikely to want to use them in isolation from each other. Lastly we looked at how correctly setting up VMotion and DRS can make it very easy to bring a physical server down for hardware maintenance while keeping your VMs online at all times.