9 Advanced Features

Advanced features are what really add value to vSphere and help distinguish it from its competitors. The features covered in this chapter provide protection for virtual machines (VMs) running on ESX and ESXi hosts, as well as optimize resources and performance and simplify VM management. These features typically have many requirements, though, and they can be tricky to set up properly. So, make sure you understand how they work and how to configure them before using them.

High Availability (HA)

HA is one of ESX's best features and is a low-cost alternative to traditional server clustering. HA does not provide 100% availability of VMs, but rather provides higher availability by rapidly recovering VMs on failed hosts. The HA feature continuously monitors all ESX Server hosts in a cluster and detects failures, and will automatically restart VMs on other host servers in an ESX cluster in case of a host failure.

How HA Works

HA is based on a modified version of the EMC/Legato Automated Availability Manager (AAM) 5.1.2 product that VMware bought to use with VMware VI3. HA works by taking a cluster of ESX and ESXi hosts and placing an agent on each host to maintain a "heartbeat" with the other hosts in the cluster; loss of a heartbeat initiates a restart of all affected VMs on other hosts. vCenter Server does not provide a single point of failure for this feature, and the feature will continue to work even if the vCenter Server is unavailable. In fact, if the vCenter Server goes down, HA clusters can still restart VMs on other hosts; however, information regarding availability of extra resources will be based on the state of the cluster before the vCenter Server went down.

HA monitors whether sufficient resources are available in the cluster at all times in order to be able to restart VMs on different physical host machines in the event of host failure. Safe restart of VMs is made possible by the locking technology in the ESX Server storage stack, which allows multiple ESX Servers to have access to the same VM's file simultaneously. HA relies on what are called "primary" and "secondary" hosts; the first five hosts powered on in an HA cluster are designated as primary hosts, and the remaining hosts in the cluster are considered secondary hosts. The job of the primary hosts is to replicate and maintain the state of the cluster and to initiate failover actions. If a primary host fails, a new primary is chosen at random from the secondary hosts. Any host that joins the cluster must communicate with an existing primary host to complete its configuration (except when you are adding the first host to the cluster). At least one primary host must be functional for VMware HA to operate correctly. If all primary hosts are unavailable, no hosts can be successfully configured for VMware HA.

HA uses a failure detection interval that is set by default to 15 seconds (15000 milliseconds); you can modify this interval by using an advanced HA setting of `das.failuredetectiontime = 15000`. A host failure is detected after the HA service on a host has stopped sending heartbeats to the other hosts in the cluster. A host stops sending heartbeats if it is isolated from the network, it crashes, or it is completely down due to a hardware failure. Once a failure is detected, other hosts in the cluster treat the host as failed, while the host declares itself as isolated from the network. By default, the isolated host leaves its VMs powered on, but the isolation response for each VM is configurable on a per-VM basis. These VMs can then successfully fail over to other hosts in the cluster. HA also has a restart priority that can be set for each VM so that certain VMs are started before others. This priority can be set to either low, medium, or high, and also can be disabled so that VMs are not automatically restarted on other hosts. Here's what happens when a host failure occurs.

1. One of the primary hosts is selected to coordinate the failover actions, and one of the remaining hosts with spare capacity becomes the failover target.
2. VMs affected by the failure are sorted by priority, and are powered on until the failover target runs out of spare capacity, in which case another host with sufficient capacity is chosen for the remaining VMs.
3. If the host selected as coordinator fails, another primary continues the effort.
4. If one of the hosts that fails was a primary node, one of the remaining secondary nodes is promoted to being a primary.

The HA feature was enhanced starting with ESX 3.5, and now provides VM failure monitoring in case of operating system failures such as the Windows Blue Screen of Death (BSOD). If an OS failure is detected due to loss of a heartbeat from VMware Tools, the VM will automatically be reset on the same host so that its OS is restarted. This new functionality allows HA to also monitor VMs via a heartbeat that is sent every second when using VMware Tools, and further enhances HA's ability to recover from failures in your environment.

When this feature was first introduced, it was found that VMs that were functioning properly occasionally stopped sending heartbeats, which caused unnecessary VM resets. To avoid this scenario, the VM monitoring feature was enhanced to also check for network or disk I/O activity on the VM. Once heartbeats from the VM have stopped, the I/O stats for the VM are checked. If no activity has occurred in the preceding two minutes, the VM is restarted. You can change this interval using the HA advanced setting, `das.iostatsInterval`.

VMware enhanced this feature even further in version 4.1 by adding application monitoring to HA. With application monitoring, an application's heartbeat will also be monitored, and if it stops responding, the VM will be restarted. However, unlike VM monitoring, which relies on a heartbeat generated by VMware Tools, application monitoring requires that an application be specifically written to take advantage of this feature. To do this, VMware has provided an SDK that developers can use to modify their applications to take advantage of this feature.

Configuring HA

HA may seem like a simple feature, but it's actually rather complex, as a lot is going on behind the scenes. You can set up the HA feature either during your initial cluster setup or afterward. To configure it, simply select the cluster on which you want to enable HA, right-click on it, and edit the settings for it. Put a checkmark next to the Turn On VMware HA field on the Cluster Features page, and HA will be enabled for the cluster. You can optionally configure some additional settings to change the way HA functions. To access these settings, click on the VMware HA item in the Cluster Settings window.

The Host Monitoring Status section is new to vSphere and is used to enable the exchange of heartbeats among hosts in the cluster. In VI3, hosts always exchanged heartbeats if HA was enabled, and if any network or host maintenance was being performed, HA could be triggered unnecessarily. The Enable Host Monitoring setting allows you to turn this on or off when needed. For HA to work, Host Monitoring must be enabled. If you are doing maintenance, you can temporarily disable it.

The Admission Control section allows you to enable or disable admission control, which determines whether VMs will be allowed to start if, by doing so, they will violate availability constraints. When Admission Control is enabled, any attempt to power on a VM when there is insufficient failover capacity within the cluster will fail. This is a safety mechanism to ensure that enough capacity is available to handle VMs from failed hosts. When Admission Control is disabled, VMs will be allowed to be powered on regardless of whether they decrease the resources needed to handle VMs from failed hosts. If you do disable Admission Control, HA will still work, but you may experience issues when recovering from a failure event if you do not have enough resources on the remaining hosts to handle the VMs that are being restarted.

The Admission Control Policy section allows you to select a type of policy to use. The three available policies are described in the sections that follow.

Host Failures Cluster Tolerates

This is used to ensure that there is sufficient capacity among the remaining host servers to be able to handle the additional load from the VMs on failed host servers. Setting the number of host failures allowed will cause the cluster to continuously monitor that sufficient resources are available to power on additional VMs on other hosts in case of a failure. Specifically, only CPU and memory resources are factored in when determining resource availability; disk and network resources are not. You should set the number of host failures allowed based on the total number of hosts in your cluster, their size, and how busy they are.

vCenter Server supports up to four host failures per cluster; if all five primaries were to fail simultaneously, HA would not function properly. For example, if you had four ESX hosts in your cluster, you would probably only want to allow for one host failure; if you had eight ESX hosts in your cluster, you might want to allow for two host failures; and if you had a larger cluster with 20 ESX hosts, you might want to allow for up to four host failures. This policy uses a slot size to determine the necessary spare resources to support the number of host failures that you select. A slot is a logical representation of the memory and CPU resources that satisfy the requirements for any powered-on VM in the cluster. HA automatically calculates slot sizes using CPU and memory reservations, and then the maximum number of slots that each host can support is determined. It does this by dividing the host's CPU resource amount by the CPU component of the slot size, and rounds down the result. The same calculation is made for the host's memory resource amount. The two numbers are then compared, and the lower number is the number of slots that the host can support. The failover capacity is computed by determining how many hosts (starting from the largest) can fail and still leave enough slots to satisfy the requirements of all powered-on VMs. Slot size calculations can be confusing and are affected by different things. For more information on slot sizes, see the vSphere Availability Guide at www.vmware.com/pdf/vsphere4/r40/vsp_40_availability.pdf.

Percentage of Cluster Resources Reserved As Failover Capacity

Instead of using slot sizes, HA uses calculations to ensure that a percentage of the cluster's resources are reserved for failover. It does this by calculating the total resource requirements for all powered-on VMs in the cluster. Next, it calculates the total number of host resources available for VMs. Finally, it calculates the current CPU failover capacity and current memory failover capacity for the cluster, and if they are less than the percentage that is specified for the configured failover capacity, admission control will be enforced. The resource requirements for powered-on VMs comprise two components, CPU and memory, and are calculated just like slot sizes are. The total number of host resources available for VMs is calculated by summing the host's CPU and memory resources. The current CPU failover capacity is computed by subtracting the total CPU resource requirements from the total host CPU resources and dividing the result by the total host CPU resources. The current memory failover capacity is calculated similarly. This method is a bit more balanced than specifying host failures, but it is not as automated because you have to manually specify a percentage.

Specify a Failover Host

This method is the simplest, as you are specifying a single host onto which to restart failed VMs. If the specified host has failed, or if it does not have enough resources, HA will restart the VMs on another host in the cluster. You can only specify one failover host, and HA will prevent VMs from being powered on or moved to the failover host during normal operations to ensure that it has sufficient capacity.

If you select a cluster in the vSphere Client and then choose the Summary tab, you can see the cluster's current capacity percentages. It is important to note that when a host fails, all of its VMs will be restarted on the single ESX host that has the lightest workload. This policy can quickly overload the host. When this occurs, the Distributed Resource Scheduler (DRS) kicks in to spread the load across the remaining hosts in the cluster. If you plan to use HA without DRS, you should ensure that you have plenty of extra capacity on your ESX hosts to handle the load from any one failed host. Additionally, you can set restart priorities so that you can specify which VMs are restarted first, and even prevent some VMs from being restarted in case of a failure.

The Virtual Machine Options section is for the cluster default settings that will apply to all VMs in the cluster by default, as well as individual VM settings. The cluster default settings apply to each VM created in or moved to the cluster, unless the VM is individually specified. The first setting is VM Restart Priority, which is the priority given to a VM when it is restarted on another host in case of a host failure. This can be set to High, Medium, Low, or Disabled. Any VM set to Disabled will not be restarted in case of a host failure.

The second setting is Host Isolation Response, which is used to determine what action the failed host that is isolated should take with the VMs that are running on it. This is used if a host is still running but has a failure in a particular subsystem (e.g., a NIC or host bus adapter [HBA] failure) or a connectivity problem (e.g., cable or switch) and is not completely down. When a host declares itself isolated and the VMs are restarted on other host, this setting dictates what happens on the failed host. The options include leaving the VM powered on, powering off the VM (hard shutdown), and shutting down the VM (graceful shutdown). If you choose to shut down the VM, HA will wait five minutes for the VM to shut down gracefully before it forcefully shuts it down. You can modify the time period that it waits for a graceful shutdown in the Advanced Configuration settings. It is usually desirable to have the VM on the failed host powered off or shut down so that it releases its lock on its disk file and also does not cause any conflicts with the new host that powers on the VM.

One reason you may choose to leave the VM powered on is if you do not have network redundancy or do not have a reliable network. In this case, you may experience false triggers to HA where the ESX host is okay but has just lost network connectivity. If you have proper network redundancy on your ESX hosts, HA events should be very rare. This setting will not come into play if the failed host experiences a disastrous event, such as completely losing power, because all the VMs will be immediately powered off anyway. In two-host configurations, you almost always want to set this to leave the VM powered on.

The last section is for Virtual Machine Monitoring (and Application Monitoring in vSphere 4.1), which restarts on the same host VMs that have OS failures. You can enable/disable this feature by checking the Enable VM Monitoring field and then using the slider to select a sensitivity from Low to High; if you want to customize your settings, you can click the Advanced Options button (vSphere 4.1) or check the Custom field (vSphere 4.0) and customize the settings on the screen that appears. In vSphere 4.1, instead of checking to enable VM monitoring, you can choose Disabled, VM Monitoring, or VM & Application Monitoring. Here are the VM monitoring advanced/custom options that you can set.

- **Failure Interval—**This declares a VM failure if no heartbeat is received for the specified number of seconds (default is 30).
- **Minimum Uptime—**After a VM has been powered on, its heartbeats are allowed to stabilize for the specified number of seconds. This time should include the guest OS boot time (default is 120).
- **Maximum per-VM Resets—**This is the maximum number of failures and automated resets allowed within the time that the maximum resets time window specifies. If no window is specified, then once the maximum is reached, automated reset is discontinued for the repeatedly failing VM and further investigation is necessary (default is 3).
- **Maximum Resets Time Window—**This is the amount of time for the specified maximum per-VM resets to occur before automated restarts stop (default is one hour).

You can also set individual VM settings that are different from the cluster defaults.

vSphere 4.1 added another new feature to HA that checks the cluster's operational status. Available on the cluster's Summary tab, this detail window, called Cluster Operational Status, displays more information about the current HA operational status, including the specific status and errors for each host in the HA cluster.

Advanced Configuration

Many advanced configuration options can be set to tweak how HA functions. You can set these options through the Advanced Options button in the HA settings, but you have to know the setting names and their values to be able to set them. These options are not displayed by default, except for the options that are set if you enable Virtual Machine Monitoring, and you must manually add them if you wish to use them. To see a list of the many HA advanced options that you can set, visit http://vsphere-land.com/vinfo/ha-advanced-options.

Additional Resources

Understanding the mechanics of HA can be confusing, but fortunately some good information is available on the Internet that can help you with this. Check out the following resources to find out more about the HA feature:

- vSphere Availability Guide (http://vmware.com/pdf/vsphere4/r40_u1/vsp_40_u1_availability.pdf)
- HA Deepdive (www.yellow-bricks.com/vmware-high-availability-deepdiv/)
- HA: Concepts, Implementation, and Best Practices (www.vmware.com/files/pdf/VMwareHA_twp.pdf)

Distributed Resource Scheduler (DRS)

DRS is a powerful feature that enables your virtual environment to automatically balance itself across your ESX host servers in an effort to eliminate resource contention. It utilizes the VMotion feature to provide automated resource optimization through automatic migration of VMs across hosts in a cluster. DRS also provides automatic initial VM placement on any of the hosts in the cluster, and makes automatic resource relocation and optimization decisions as hosts or VMs are added to or removed from the cluster. You can also configure DRS for manual control so that it only provides recommendations that you can review and carry out.

How DRS Works

DRS works by utilizing resource pools and clusters that combine the resources of multiple hosts into a single entity. Multiple resource pools can also be created so that you can divide the resources of a single or multiple hosts into separate entities. Currently, DRS will only migrate VMs based on the availability and utilization of the CPU and memory resources. It does not take into account high disk or network utilization to load-balance VMs across hosts.

When a VM experiences increased load, DRS first evaluates its priority against the established resource allocation rules and then, if justified, redistributes VMs among the physical servers to try to eliminate contention for resources. VMotion will then handle the live migration of the VM to a different ESX host with complete transparency to end users. The dynamic resource allocation ensures that capacity is preferentially dedicated to the highest-priority applications, while at the same time maximizing overall resource utilization. Unlike the HA feature, which will still operate when vCenter Server is unavailable, DRS requires that vCenter Server be running for it to function.

Configuring DRS

Similar to HA, the DRS feature can be set up in a cluster either during its initial setup or afterward. To configure DRS, simply select the cluster on which you want to enable DRS, right-click on it, and edit its settings, or select the cluster and in the Summary pane click the Edit Settings link. Put a checkmark next to the Enable VMware DRS field on the General page, and DRS will be enabled for the cluster. You can optionally configure some additional settings to change the way DRS functions. To access these settings, click on the VMware DRS item in the Cluster Settings window.

Once you enable DRS, you must select an automation level that controls how DRS will function. You can choose from the following three levels.

- **Manual**—Initial placement is the host on which DRS was last located; migration recommendations require approval.
- **Partially Automated**—Initial placement is automated; migration recommendations require approval.
- **Fully Automated**—Initial placement is automated; migration recommendations are automatically executed.

When considering an automation level, it is usually best to choose Fully Automated and let DRS handle everything. However, when first enabling DRS, you might want to set the automation level to Manual or Partially Automated so that you can observe its recommendations for a while before turning it loose on Fully Automated. Even when selecting Fully Automated, you can still configure individual VM automation levels, so you can specify certain VMs to not be migrated at all (disabled) or to be set to Manual or Partially Automated. To configure individual VM automation levels, click on Virtual Machine Options, located under DRS. Usually, the default three-star level is a good starting point and works well for most environments. You should be careful when choosing more aggressive levels, as you could have VMs moving very frequently between hosts (i.e., VM pong), which can create performance issues because of the constant VMotions which cause an entire LUN to be locked during the operation (i.e., SCSI reservations).

DRS makes its recommendations by applying stars to indicate how much the recommendation would improve the cluster's performance. One star indicates a slight improvement, and four stars indicates significant improvement. Five stars, the maximum, indicates a mandatory move because of a host entering maintenance mode or affinity rule violations. If DRS is set to work in Fully Automated mode, you have the option to set a migration threshold based on how much it would improve the cluster's performance. The lowest threshold, which is the most conservative, only applies five-star recommendations; the highest threshold, which is very aggressive, applies all recommendations. There are also settings in between to only apply two-, three-, or four-star recommendations.

You can configure affinity rules in DRS to keep certain VMs either on the same host or on separate hosts when DRS migrates VMs from host to host. These affinity rules (not to be confused with CPU affinity) are useful for ensuring that when DRS moves VMs around, it has some limits on where it can place the VMs. You might want to keep VMs on the same host if they are part of a tiered application that runs on multiple VMs, such as a web, application, or database server. You might want to keep VMs on different hosts for servers that are clustered or redundant, such as Active Directory (AD), DNS, or web servers, so that a single ESX failure does not affect both servers at the same time. Doing this ensures that at least one will stay up and remain available while the other recovers from a host failure. Also, you might want to separate servers that have high I/O workloads so that you do not overburden a specific host with too many high-workload servers.

Because DRS does not take into account network and disk workloads when moving VMs, creating a rule for servers that are known to have high workloads in those areas can help you to avoid disk and network I/O bottlenecks on your hosts. In general, try to limit the number of rules you create, and only create ones that are necessary. Having too many rules makes it more difficult for DRS to try to place VMs on hosts to balance the resource load. Also watch out for conflicts between multiple rules which can cause problems.

Once you have DRS enabled, you can monitor it by selecting the cluster in vCenter Server and choosing the Summary tab. Here you can see load deviations, the number of faults and recommendations, and the automation level. By clicking the resource distribution chart you can also see CPU and memory utilization on a per-VM basis, grouped by host. Additionally, you can select the DRS tab in vCenter Server to display any pending recommendations, faults, and the DRS action history. By default, DRS recommendations are generated every five minutes; you can click the Run DRS link to generate them immediately if you do not want to wait. You can also click the Apply Recommendations button to automatically apply the pending recommendations.

DISTRIBUTED POWER MANAGEMENT (DPM)

DPM is a subcomponent of DRS and is a green feature that was introduced in VI 3.5 that will power down hosts during periods of inactivity to help save power. All the VMs on a host that will be powered down are relocated to other hosts before the initial host is powered down. When activity increases on the other hosts and DPM deems that additional capacity is needed, it will automatically power the host back on and move VMs back onto it using DRS. DPM requires that the host has a supported power management protocol that automatically powers it on after it has been powered off. You can configure DPM in either manual or automatic mode. In manual mode, it will simply make recommendations similar to DRS, and you will have to manually approve them so that they are applied. You can also configure DPM so that certain host servers are excluded from DPM, as well as specify that certain hosts are always automatic or always manual.

HOW DPM WORKS

Although DPM existed in VI3, it was not officially supported and was considered experimental. This was because it relied on Wake On LAN (WOL) technology that exists in certain network adapters but was not always a reliable means for powering a server on. Being able to power up servers when needed is critical when workloads increase, so a more reliable technology was needed for DPM to be fully supported. For this, VMware turned to two technologies in vSphere: Intelligent Platform Management Interface (IPMI) and HP's Integrated Lights-Out (iLO).

IPMI is a standard that was started by Intel and is supported by most major computer manufacturers. It defines a common set of interfaces that can be used to manage and monitor server hardware health. Since IPMI works at the hardware layer and does not depend on an operating system, it works with any software or operating system that is designed to access it. IPMI relies on a Baseboard Management Controller (BMC) that is a component in server motherboards and monitors many different sensors on the server, including temperature, drive status, fan speed, power status, and much more. IPMI works even when a server is powered off, as long as it is connected to a power source. The BMC is connected to many other controllers on the server and administration can be done using a variety of methods. For instance, out-of-band management can be done using a LAN connection via the network controller interconnects to the BMC. Other out-of-band management options include remote management boards and serial connections.

For HP servers, DPM uses HP's proprietary iLO remote, out-of-band management controllers that are built into every HP server. HP has used its iLO technology under several different names for years, and has only recently begun to also embrace the IPMI standard. Dell's Remote Access Card (DRAC) remote management controllers provide the same functionality as HP's iLO, but Dell fully supports IPMI. Typically on Dell DRAC boards you need to enable IPMI via the server BIOS to be able to use it.

In addition to the IPMI and iLO power management protocols, vSphere also now fully supports WOL. However, if multiple protocols exist in a server, they are used in the following order: IPMI, iLO, WOL.

Configuring DPM

DPM requires that the host be a member of a DRS-enabled cluster. Before you can configure DPM in vSphere, you typically have to enable the power management protocol on whatever method you are using. If you are using WOL, this is usually enabled in your host server's BIOS. Depending on the server for IPMI, you can usually enable this also in the server BIOS or in the web-based configuration utility for the server out-of-band management board (e.g., Dell DRAC). This setting is usually referred to as IPMI over LAN. For HP's iLOs, make sure the Lights-Out functionality is enabled in the iLO web-based configuration utility; it usually is by default. Both IPMI and iLO require authentication to be able to access any of their functionality; WOL does not.

You can determine whether a NIC supports the WOL feature and that it is enabled in the vSphere client by selecting a host, choosing the Configuration tab, and then, under Hardware, selecting Network Adapters. All of the host's NICs will be displayed and one of the columns will show whether WOL is supported. Once you have configured the protocol that you will use with DPM, you can configure it in vSphere by following these steps.

- For IPMI/iLO, select the host in the vSphere Client, and on the Configuration tab under Software, select Power Management. Click Properties and enter the username/password of a user account that can log in to the management board, as well as the IP address and MAC address of the board. You can usually find the MAC address for iLO boards on the NIC tab of the System Information screen on the iLO web page.
- For WOL, there is nothing to configure, but you must make sure the NIC that DPM uses is the one assigned to the VMkernel virtual switch (vSwitch). You may need to rearrange your NICs so that you have one that supports WOL in the VMkernel vSwitch. Additionally, the switch port that the NIC is connected to must be set to Auto-negotiate, as many NICs support WOL only if they can switch to 100MBps or less when the host is powered off.

Once DPM is enabled and configured properly, you will want to test it before you use it. To test it, simply select the host in the vSphere Client, right-click on it, and select Enter Standby Mode, which will power down the host. You will be prompted if you want to move powered-off/suspended VMs to other hosts. Powered-on VMs will automatically be migrated using VMotion; if they are not capable of using VMotion, they will be powered off. Your host will begin to shut down and should power off after a few minutes. Verify that your host has powered off, and then, in the vSphere Client, right-click on the host and select Power On. If the feature is working, the host should power back on automatically.

Once you have verified that DPM works properly, you need to enable DPM. To do this, edit the settings for your cluster; next, under the DRS category, select Power

Management. You can then select either the Off, Manual, or Automatic option. The Manual option will only make recommendations for powering off hosts; the Automatic option will enable vCenter Server to automatically execute power management-related recommendations. You can also set a threshold for DPM, as you can with DRS, which will determine how aggressive it gets with powering off hosts. The DRS threshold and the DPM threshold are essentially independent. You can differentiate the aggressiveness of the migration and host power state recommendations they respectively provide. The threshold priority ratings are based on the amount of over- or underutilization found in the DRS cluster and the improvement that is expected from the intended host power state change. Priority-one recommendations are the biggest improvement, and priority-five the least. The threshold ranges from conservative, which only applies priority-one recommendations, to aggressive, which applies priority-five recommendations.

If you select Host Options, you can change the Power Management settings on individual hosts to have them use the cluster default, always use Manual, or always use Automatic; you can also disable the feature for them.

DPM Considerations

Although DPM is a great technology that can save you money and that every large datacenter should take advantage of, you should be aware of the following considerations before you use it.

- If you are using a monitoring system to monitor that your ESX servers are up or down, you will trigger an alert whenever a host is shut down. Having servers going up and down automatically can generate a lot of confusion and false alarms. In addition, many datacenters measure uptime statistics on all the servers. Having hosts going down during periods of inactivity can significantly skew those numbers. If you use DPM, you should consider adjusting your operational procedures to exclude the ESX hosts from monitoring and instead only monitor the VMs.
- If you are using HA in your cluster, be aware that DRS and DPM maintain excess powered-on capacity to meet the HA settings. Therefore, the cluster may not allow additional VMs to be powered on and/or some hosts may not be powered down, even though the cluster may appear to be sufficiently idle. Additionally, if HA strict admission control is enabled, DPM will not power off hosts if doing so would violate failover requirements. If HA strict admission control is disabled, DPM will power off hosts even if doing so violates failover requirements.

- Similar to DRS, the DPM feature works best if all hosts are in Automatic mode, which gives DPM more flexibility in selecting hosts to power on and off. If hosts are in Manual DPM mode, DPM must wait for approval to perform its action, which can limit its effectiveness. DPM is more biased toward hosts in Automatic mode than Manual mode because of this. Consider using Automatic mode whenever possible, and disabling DPM on hosts that you do not want to be powered off.
- DPM will consider historical demand when determining how much capacity to keep available, and keeps some extra capacity available for changes in demand.
- Having servers power off and on automatically requires a lot of trust in the technology. To protect against a situation in which a host is not powered on when necessary, set an alarm in vCenter Server to be alerted if this happens. You can create a host alarm for the event "DRS cannot exit the host out of standby mode" so that you can be alerted when this happens and can take care of the situation.

VMOTION

VMotion is a powerful feature that allows you to quickly move an entire running VM from one ESX host to another without any downtime or interruption to the VM. This is also known as a "hot" or "live" migration.

HOW VMOTION WORKS

The entire state of a VM is encapsulated and the VMFS filesystem allows both the source and the target ESX host to access the VM files concurrently. The active memory and precise execution state of a VM can then be rapidly transmitted over a high-speed network. The VM retains its network identity and connections, ensuring a seamless migration process as outlined in the following steps.

1. The migration request is made to move the VM from ESX1 to ESX2.
2. vCenter Server verifies that the VM is in a stable state on ESX1.
3. vCenter Server checks the compatibility of ESX2 (CPU/networking/etc.) to ensure that it matches that of ESX1.
4. The VM is registered on ESX2.
5. The VM state information (including memory, registers, and network connections) is copied to ESX2. Additional changes are copied to a memory bitmap on ESX1.

6. The VM is quiesced on ESX1 and the memory bitmap is copied to ESX2.
7. The VM is started on ESX2 and all requests for the VM are now directed to ESX2.
8. A final copy of the VM's memory is made from ESX1 to ESX2.
9. The VM is unregistered from ESX1.
10. The VM resumes operation on ESX2.

CONFIGURING VMOTION

VMotion requires shared storage for it to function (Fibre Channel [FC], iSCSI, or NFS), and also has some strict requirements to ensure compatibility of a VM moving from one ESX host to another, as outlined in the following list.

- Both the source ESX host and the destination ESX host must be able to access the same shared storage on which the VM is located; the shared storage can be either FC, iSCSI, or NFS. VMotion will also work with Raw Device Mappings (RDMs) as long as they are configured to work in virtual compatibility mode.
- ESX hosts must have a Gigabit Ethernet network adapter or higher to be configured on the VMkernel vSwitch used by VMotion; slower NICs will work, but they are not recommended. For best results, and because VMotion traffic is sent as clear text, it is best to have an isolated network for VMotion traffic.
- ESX hosts must have processors that are able to execute each other's instructions. Processor clock speeds, cache sizes, and number of cores can differ among ESX hosts, but they must have the same processor vendor class (Intel or AMD) and compatible feature sets. It is possible to override these restrictions for CPUs from the same vendor, but doing so can cause a VM to crash because it must access a CPU feature or instruction that the new ESX host does not support.

Here are some additional requirements for VMotion to function properly.

- vSwitch network labels (port groups) must match exactly (including case) on each ESX host.
- A VM cannot be using CPU affinity, which pins a VM to run on a specific processor(s) on an ESX host.
- A VM cannot be connected to an internal-only (no NICs assigned to it) vSwitch.

- Using jumbo frames is recommended for best performance.
- The source and destination hosts must be licensed for VMotion.
- A VM cannot have its virtual CD-ROM and floppy drives mapped to either a host device or a local datastore ISO file.

Before configuring VMotion on your host servers, you should make sure they meet the requirements for using it. Configuring VMotion is fairly simple; you must first set up the VMkernel networking stack on a vSwitch which is used for VMotion by creating a port group on the vSwitch. You can do this by editing the vSwitch that you want to use for VMotion, clicking the Add button, and selecting VMkernel. You then configure the port group properties and set the IP address for the VMotion interface. You can verify the network connectivity of the VMotion interface by using the vmkping Service Console utility to ping the VMkernel interface of other hosts.

VMOTION CONSIDERATIONS

Configuring VMotion is easy, but there are requirements and compatibility issues that you need to be aware of. Here are some considerations that you should know about when using and configuring VMotion.

- In versions prior to ESX 3.5, VMs that had their swap file (.vswp file) not located on shared storage could not be moved with VMotion. This was because the destination host would not be able to access the .vswp file that was located on the source host's local disk. Beginning with ESX 3.5, support for using VMotion on VMs that have local .vswp files was added. If a VM with a local .vswp file is VMotioned, the .vswp file is re-created on the destination host and the nonzero contents of the .vswp file are copied over as part of the VMotion operation. This can cause the VMotion operation to take slightly longer than normal due to the added .vswp copy operation in addition to the normal CPU and memory state copy operation. Using a local swap file datastore can be advantageous, as it frees up valuable and expensive shared disk space to be used for other things, such as snapshots and virtual disks.
- If your VMs have their CD-ROM drive mapped to either a host device or a local ISO datastore, they cannot be VMotioned, as the destination server will not have access to the drive. Additionally, if the CD-ROM is mapped to a shared ISO datastore, make sure all ESX hosts can see that shared ISO datastore. Consider using a shared ISO datastore on a VMFS volume, or alternately, on an NFS or Samba share instead.

- Using VMotion with VMs with running snapshots is supported, as long as the VM is being migrated to a new host without moving its configuration file or disks.
- It's very important to ensure that vSwitch network labels are identical (case-sensitive) across all hosts. If they are not, you cannot VMotion a VM between two hosts that do not have the same Network Labels configured on their vSwitches.
- CPU compatibility is one of the biggest headaches when dealing with VMotion because VMotion transfers the running architectural state of a VM between host systems. To ensure a successful migration, the processor of the destination host must be able to execute the equivalent instructions as that of the source host. Processor speeds, cache sizes, and number of cores can vary between the source and destination hosts, but the processors must come from the same vendor (either Intel or AMD) and use compatible feature sets to be compatible with VMotion. When a VM is first powered on, it determines its available CPU feature set based on the host's CPU feature set. It is possible to mask some of the host's CPU features using a CPU compatibility mask in order to allow VMotions between hosts that have slightly dissimilar feature sets. See VMware Knowledge Base articles 1991 (http://kb.vmware.com/kb/1991), 1992 (http://kb.vmware.com/kb/1992), and 1993 (http://kb.vmware.com/kb/1993) for more information on how to set up these masks. Additionally, you can use the Enhanced VMotion feature to help deal with CPU incompatibilities between hosts.
- It is a recommended security practice to put your VMotion network traffic onto its own isolated network so that it is only accessible to the host servers. The reason for this is twofold. First, VMotion traffic is sent as clear text and is not encrypted, so isolating it ensures that sensitive data cannot be sniffed out on the network. Second, it ensures that VMotion traffic experiences minimal latency and is not affected by other network traffic as a VMotion operation is a time-sensitive operation.

Enhanced VMotion Compatibility (EVC)

Enhanced VMotion Compatibility (EVC) is designed to further ensure compatibility between ESX hosts. EVC leverages the Intel FlexMigration technology as well as the AMD-V Extended Migration technology to present the same feature set as the baseline processors. EVC ensures that all hosts in a cluster present the same CPU feature set to every VM, even if the actual CPUs differ on the host servers. This feature will still not allow you to migrate VMs from an Intel CPU host to an AMD host. Therefore, you should only create clusters with ESX hosts of the same processor family, or choose a processor vendor and stick with it. Before you enable EVC, make sure your hosts meet the following requirements.

- All hosts in the cluster must have CPUs from the same vendor (either Intel or AMD).
- All VMs in the cluster must be powered off or migrated out of the cluster when EVC is being enabled.
- All hosts in the cluster must either have hardware live migration support (Intel FlexMigration or AMD-V Extended Migration), or have the CPU whose baseline feature set you intend to enable for the cluster. See VMware Knowledge Base article 1003212 (http://kb.vmware.com/kb/1003212) for a list of supported processors.
- Host servers must have the following enabled in their BIOS settings: For AMD systems, enable AMD-V and No Execute (NX); for Intel systems, enable Intel VT and Execute Disable (XD).

Once you are sure your hosts meet the requirements, you are ready to enable EVC by editing the cluster settings. There are two methods that you can use for doing this, as EVC cannot be enabled on existing clusters unless all VMs are shut down. The first method is to create a new cluster that is enabled for EVC, and then to move your ESX hosts into the cluster. The second method is to shut down all the VMs in your current cluster or migrate them out of the cluster to enable it.

The first method tends to be easier, as it does not require any VM downtime. If you choose the first method, you can simply create a new cluster and then move your hosts one by one to the cluster by first putting it in maintenance mode to migrate the VMs to other hosts. Then, once the host is moved to the new cluster, you can VMotion the VMs back to the host from the old cluster to the new one. The downside to this method is that you have to once again set up your cluster HA and DRS settings on the new cluster, which means you'll lose your cluster performance and migration history.

STORAGE VMOTION

Storage VMotion (SVMotion) allows you to migrate a running VM's disk files from one datastore to another on the same ESX host. The difference between

VMotion and SVMotion is that VMotion simply moves a VM from one ESX host to another, but keeps the storage location of the VM the same. SVMotion changes the storage location of the VM while it is running and moves it to another datastore on the same ESX host, but the VM remains on the same host. The VM's data files can be moved to any datastore on the ESX host which includes local and shared storage.

How SVMotion Works

The SVMotion process is as follows.

1. A new VM directory is created on the target datastore, and VM data files and virtual disk files are copied to the target directory.
2. The ESX host does a "self" VMotion to the target directory.
3. The Changed Block Tracking (CBT) feature keeps track of blocks that change during the copy process.
4. VM disk files are copied to the target directory.
5. Disk blocks that changed before the copy completed are copied to the target disk file.
6. The source disk files and directory are deleted.

SVMotion does more than just copy disk files from one datastore to another; it can also convert thick disks to thin disks, and vice versa, as part of the copy process. SVMotion can also be used to shrink a thin disk after it has grown and data has been deleted from it. Typically when you perform an SVMotion, you are moving the VM location to another storage device; however, you can also leave the VM on its current storage device when performing a disk conversion. SVMotion can be an invaluable tool when performing storage maintenance, as VMs can be easily moved to other storage devices while they are running.

Configuring SVMotion

You should be aware of the following requirements for using SVMotion.

- VM disks must be in persistent mode or be an RDM that is in virtual compatibility mode. For virtual compatibility mode RDMs, you can migrate the mapping file or convert them to thick-provisioned or thin-provisioned disks during migration, as long as the destination is not an NFS datastore. For physical compatibility mode RDMs, you can migrate the mapping file only.

- The VM must have no snapshots. If it does, it cannot be migrated.
- ESX/ESXi 3.5 hosts must be licensed and configured for VMotion. ESX/ESXi 4.0 and later hosts do not require VMotion configuration in order to perform migration with SVMotion. ESX/ESXi 4.0 hosts must be licensed for SVMotion (Enterprise and Enterprise Plus only).
- The host that the VM is running on must have access to the source and target datastores and must have enough resources available to support two instances of the VM running at the same time.
- A single host can be involved in up to two migrations with VMotion or SVMotion at one time.

In vSphere, SVMotion is no longer tied to VMotion; it is licensed separately and does not require that VMotion be configured to use it. No extra configuration is required to configure SVMotion, and it can be used right away as long as you meet the requirements outlined in the preceding list. In VI3, you needed to use a remote command-line utility (svmotion.pl) to perform an SVMotion; in vSphere, this is now integrated into the vSphere Client. To perform an SVMotion, you select a VM and choose the Migrate option; however, you can still use svmotion.pl to perform an SVMotion using the vSphere CLI. When the Migration Wizard loads, you have the following three options from which to choose.

- **Change Host—**This performs a VMotion.
- **Change Datastore—**This performs an SVMotion.
- **Change Host and Datastore—**This performs a cold migration for which the VM must be powered off.

FAULT TOLERANCE (FT)

Fault Tolerance (FT) was introduced as a new feature in vSphere to provide something that was missing in VI3: continuous availability for a VM in case of a host failure. HA was introduced in VI3 to protect against host failures, but it caused the VM to be down for a short period of time while it was restarted on another host. FT takes that to the next level and guarantees that the VM stays operational during a host failure by keeping a secondary copy of it running on another host server; in case of a host failure, that VM then becomes the primary VM and a new secondary is created on another functional host. The primary VM and secondary VM stay in sync with each other by using a technology called Record/Replay that was first introduced with VMware Workstation. Record/Replay works by recording the computer execution on a VM and saving it into a logfile; it can then take that recorded information and replay it on another VM to have a copy that is a duplicate of the original VM.

The technology behind the Record/Replay functionality is built into certain models of Intel and AMD processors, and is called vLockstep by VMware. This technology required Intel and AMD to make changes to both the performance counter architecture and virtualization hardware assists (Intel VT and AMD-V) that are inside their physical processors. Because of this, only newer processors support the FT feature; this includes the third-generation AMD Opteron based on the AMD Barcelona, Budapest, and Shanghai processor families; and Intel Xeon processors based on the Core 2 and Core i7 micro architectures and their successors. VMware has published a Knowledge Base article (http://kb.vmware.com/kb/1008027) that provides more details on this.

How FT Works

FT works by creating a secondary VM on another ESX host that shares the same virtual disk file as the primary VM, and then transfers the CPU and virtual device inputs from the primary VM (record) to the secondary VM (replay) via an FT logging NIC so that it is in sync with the primary and ready to take over in case of a failure. Although both the primary and secondary VMs receive the same inputs, only the primary VM produces output such as disk writes and network transmits. The secondary VM's output is suppressed by the hypervisor and is not on the network until it becomes a primary VM, so essentially both VMs function as a single VM. It's important to note that not everything that happens on the primary VM is copied to the secondary; certain actions and instructions are not relevant to the secondary VM, and to record everything would take up a huge amount of disk space and processing power. Instead, only nondeterministic events which include inputs to the VM (disk reads, received network traffic, keystrokes, mouse clicks, etc.) and certain CPU events (RDTSC, interrupts, etc.) are recorded. Inputs are then fed to the secondary VM at the same execution point so that it is in exactly the same state as the primary VM.

The information from the primary VM is copied to the secondary VM using a special logging network that is configured on each host server. It is highly recommended that you use a dedicated gigabit or higher NIC for the FT logging traffic; using slower-speed NICs is not recommended. You could use a shared NIC for FT logging for small or dev/test environments and for testing the feature. The information that is sent over the FT logging network between the two hosts can be very intensive depending on the operation of the VM. VMware has a formula that you can use to determine the FT Logging bandwidth requirements:

$$\text{VMware FT logging bandwidth} = (\text{Avg disk reads (MB/s)} \times 8 + \text{Avg network input (Mbps)}) \times 1.2\ [20\%\ \text{headroom}]$$

To get the VM statistics needed for this formula you must use the performance metrics that are supplied in the vSphere Client. The 20% headroom is to allow for CPU events that also need to be transmitted and are not included in the formula. Note that disk or network writes are not used by FT, as these do not factor into the state of the VM. As you can see, disk reads will typically take up the most bandwidth, and if you have a VM that does a lot of disk reading, you can reduce the amount of disk read traffic across the FT logging network by adding a special VM parameter, `replay.logReadData = checksum`, to the VMX file of the VM; this will cause the secondary VM to read data directly from the shared disk instead of having it transmitted over the FT logging network. For more information on this, see the Knowledge Base article at http://kb.vmware.com/kb/1011965.

It is important to note that if you experience an OS failure on the primary VM, such as a Windows BSOD, the secondary VM will also experience the failure, as it is an identical copy of the primary. However, the HA VM monitor feature will detect this, and will restart the primary VM and then respawn a new secondary VM. Also note that FT does not protect against a storage failure; since the VMs on both hosts use the same storage and virtual disk file, it is a single point of failure. Therefore, it's important to have as much redundancy as possible, such as dual storage adapters in your host servers attached to separate switches (multipathing), to prevent this. If a path to the SAN fails on the primary host, the FT feature will detect this and switch over to the secondary VM, but this is not a desirable situation. Furthermore, if there was a complete SAN failure or problem with the LUN that the VM was on, the FT feature would not protect against this.

Because of the high overhead and limitations of FT, you will want to use it sparingly. FT could be used in some cases to replace existing Microsoft Cluster Server (MSCS) implementations, but it's important to note what FT does not do, which is to protect against application failure on a VM; it only protects against a host failure. If protection for application failure is something you need, a solution such as MSCS would be better for you. FT is only meant to keep a VM running if there is a problem with the underlying host hardware. If you want to protect against an operating system failure, the VMware HA feature can provide this also, as it can detect unresponsive VMs and restart them on the same host server. You can use FT and HA together to provide maximum protection; if both the primary and secondary hosts failed at the same time, HA would restart the VM on another operable host and respawn a new secondary VM.

Configuring FT

Although FT is a great feature, it does have many requirements and limitations that you should be aware of. Perhaps the biggest is that it currently only supports single vCPU VMs, which is unfortunate, as many big enterprise applications that would benefit from FT usually need multiple vCPUs (e.g., vSMP). But don't let this discourage you from running FT, as you may find that some applications will run just fine with one vCPU on some of the newer, faster processors that are available. VMware has mentioned that support for vSMP will come in a future release. Trying to keep a single vCPU in lockstep between hosts is no easy task, and VMware needs more time to develop methods to try to keep multiple vCPUs in lockstep between hosts.

Here are the requirements for the host.

- The vLockstep technology used by FT requires the physical processor extensions added to the latest processors from Intel and AMD. In order to run FT, a host must have an FT-capable processor, and both hosts running an FT VM pair must be in the same processor family.
- CPU clock speeds between the two hosts must be within 400MHz of each other to ensure that the hosts can stay in sync.
- All hosts must be running the same build of ESX or ESXi and be licensed for FT, which is only included in the Advanced, Enterprise, and Enterprise Plus editions of vSphere.
- Hosts used together as an FT cluster must share storage for the protected VMs (FC, iSCSI, or NAS).
- Hosts must be in an HA-enabled cluster.
- Network and storage redundancy is recommended to improve reliability; use NIC teaming and storage multipathing for maximum reliability.
- Each host must have a dedicated NIC for FT logging and one for VMotion with speeds of at least 1Gbps. Each NIC must also be on the same network.
- Host certificate checking must be enabled in vCenter Server (configured in vCenter Server Settings→SSL Settings).

Here are the requirements for the VMs.

- The VMs must be single-processor (no vSMPs).
- All VM disks must be "thick" (fully allocated) and not "thin." If a VM has a thin disk, it will be converted to thick when FT is enabled.
- There can be no nonreplayable devices (USB devices, serial/parallel ports, sound cards, a physical CD-ROM, a physical floppy drive, physical RDMs) on the VM.
- Most guest OSs are supported, with the following exceptions that apply only to hosts with third-generation AMD Opteron processors (i.e., Barcelona, Budapest, Shanghai): Windows XP (32-bit), Windows 2000, and Solaris 10 (32-bit). See VMware Knowledge Base article 1008027 (http://kb.vmware.com/kb/1008027) for more details.

In addition to these requirements, there are also many limitations when using FT, and they are as follows.

- Snapshots must be removed before FT can be enabled on a VM. In addition, it is not possible to take snapshots of VMs on which FT is enabled.
- N_Port ID Virtualization (NPIV) is not supported with FT. To use FT with a VM you must disable the NPIV configuration.
- Paravirtualized adapters are not supported with FT.
- Physical RDM is not supported with FT. You may only use virtual RDMs.
- FT is not supported with VMs that have CD-ROM or floppy virtual devices connected to a physical or remote device. To use FT with a VM with this issue, remove the CD-ROM or floppy virtual device or reconfigure the backing with an ISO installed on shared storage.
- The hot-plug feature is automatically disabled for fault tolerant VMs. To hot-plug devices (when either adding or removing them), you must momentarily turn off FT, perform the hot plug, and then turn FT back on.
- EPT/RVI is automatically disabled for VMs with FT turned on.
- IPv6 is not supported; you must use IPv4 addresses with FT.
- You can only use FT on a vCenter Server running as a VM if it is running with a single vCPU.
- VMotion is supported on FT-enabled VMs, but you cannot VMotion both the primary and secondary VMs at the same time. SVMotion is not supported on FT-enabled VMs.

- In vSphere 4.0, FT was compatible with DRS, but the automation level was disabled for FT-enabled VMs. Starting in vSphere 4.1, you can use FT with DRS when the EVC feature is enabled. DRS will perform initial placement on FT-enabled VMs and also will include them in the cluster's load-balancing calculations. If EVC in the cluster is disabled, the FT-enabled VMs are given a DRS automation level of "disabled". When a primary VM is powered on, its secondary VM is automatically placed, and neither VM is moved for load-balancing purposes.

You might be wondering whether you meet the many requirements to use FT in your own environment. Fortunately, VMware has made this easy for you to determine by providing a utility called SiteSurvey (www.vmware.com/download/shared_utilities.html) that will look at your infrastructure and see if it is capable of running FT. It is available as either a Windows or a Linux download, and once you install and run it, you will be prompted to connect to a vCenter Server. Once it connects to the vCenter Server, you can choose from your available clusters to generate a SiteSurvey report that shows whether your hosts support FT and if the hosts and VMs meet the individual prerequisites to use the feature. You can also click on links in the report that will give you detailed information about all the prerequisites along with compatible CPU charts. These links go to VMware's website and display the help document for the SiteSurvey utility, which is full of great information about the prerequisites for FT. In vSphere 4.1, you can also click the blue caption icon next to the Host Configured for FT field on the Host Summary tab to see a list of FT requirements that the host does not meet. If you do this in vSphere 4.0, it shows general requirements that are not specific to the host.

Another method for checking to see if your hosts meet the FT requirements is to use the vCenter Server Profile Compliance tool. To check using this method just select your cluster in the left pane of the vSphere Client, and then in the right pane select the Profile Compliance tab. Click the Check Compliance Now link and it will check your hosts for compliance, including FT. Before you enable FT, be aware of one important limitation: VMware currently recommends that you do not use FT in a cluster that consists of a mix of ESX and ESXi hosts. This is because ESX hosts might become incompatible with ESXi hosts for FT purposes after they are patched, even when patched to the same level. This is a result of the patching process and will be resolved in a future release so that compatible ESX and ESXi versions are able to interoperate with FT even though patch numbers do not match exactly. Until this is resolved, you will need to take this into consideration if you plan to use FT, and make sure you adjust your clusters that will have FT-enabled VMs so that they consist of only ESX or ESXi hosts and not both. See VMware Knowledge Base article 1013637 (http://kb.vmware.com/kb/1013637) for more information on this.

Implementing FT is fairly simple and straightforward once you meet the requirements for using it. The first step is to configure the networking needed for FT on the host servers. You must configure two separate vSwitches on each host: one for VMotion and one for FT logging. Each vSwitch must have at least one 1Gbps NIC, but at least two are recommended for redundancy. The VMotion and FT logging NICs must be on different network subnets. You can do this by creating a VMkernel interface on each vSwitch, and selecting "Use this port group for VMotion" on one of them and "Use this port group for Fault Tolerance logging" on the other. You can confirm that the networking is configured by selecting the Summary tab for the host; the VMotion Enabled and Fault Tolerance Enabled fields should both say Yes. Once the networking is configured, you can enable FT on a VM by right-clicking on it and choosing the Fault Tolerance item, and then Turn On Fault Tolerance.

Once enabled, a secondary VM will be created on another host; at that point, you will see a new Fault Tolerance section on the Summary tab of the VM that will display information including the FT status, secondary VM location (host), CPU and memory in use by the secondary VM, secondary VM lag time (how far behind it is from the primary, in seconds), and bandwidth in use for FT logging. Once you have enabled FT, alarms are available that you can use to check for specific conditions such as FT state, latency, secondary VM status, and more.

FT Considerations

Here is some additional information that will help you understand and implement FT.

- VMware spent a lot of time working with Intel and AMD to refine their physical processors so that VMware could implement its vLockstep technology, which replicates nondeterministic transactions between the processors by reproducing their CPU instructions. All data is synchronized, so there is no loss of data or transactions between the two systems. In the event of a hardware failure, you may have an IP packet retransmitted, but there is no interruption in service or data loss, as the secondary VM can always reproduce execution of the primary VM up to its last output.

- FT does not use a specific CPU feature, but requires specific CPU families to function. vLockstep is more of a software solution that relies on some of the underlying functionality of the processors. The software level records the CPU instructions at the VM level and relies on the processor to do so; it has to be very accurate in terms of timing, and VMware needed the processors to be modified by Intel and AMD to ensure complete accuracy. The SiteSurvey utility simply looks for certain CPU models and families, but not specific CPU features, to determine whether a CPU is compatible with FT. In the future, VMware may update its CPU ID utility to also report whether a CPU is FT-capable.
- In the case of split-brain scenarios (i.e., loss of network connectivity between hosts), the secondary VM may try to become the primary, resulting in two primary VMs running at the same time. This is prevented by using a lock on a special FT file; once a failure is detected, both VMs will try to rename this file, and if the secondary succeeds it becomes the primary and spawns a new secondary. If the secondary fails because the primary is still running and already has the file locked, the secondary VM is killed and a new secondary is spawned on another host.
- There is no limit to the number of FT-enabled hosts in a cluster, but you cannot have FT-enabled VMs span clusters. A future release may support FT-enabled VMs spanning clusters.
- There is an API for FT that provides the ability to script certain actions, such as disabling/enabling FT using PowerShell.
- There is a limit of four FT-enabled VMs per host (not per cluster); this is not a hard limit, but is recommended for optimal performance.
- The current version of FT is designed to be used between hosts in the same datacenter, and is not designed to work over WAN links between datacenters due to latency issues and failover complications between sites. Future versions may be engineered to allow for FT usage between external datacenters.
- Be aware that the secondary VM can slow down the primary VM if it is not getting enough CPU resources to keep up. This is noticeable by a lag time of several seconds or more. To resolve this, try setting a CPU reservation on the primary VM which will also be applied to the secondary VM and will ensure that both VMs will run at the same CPU speed. If the secondary VM slows down to the point that it is severely impacting the performance of the primary VM, FT between the two will cease and a new secondary will be created on another host.

- Patching hosts can be tricky when using the FT feature because of the requirement that the hosts have the same build level, but it is doable, and you can choose between two methods to accomplish this. The simplest method is to temporarily disable FT on any VMs that are using it, update all the hosts in the cluster to the same build level, and then reenable FT on the VMs. This method requires FT to be disabled for a longer period of time; a workaround if you have four or more hosts in your cluster is to VMotion your FT-enabled VMs so that they are all on half of your ESX hosts. Then update the hosts without the FT VMs so that they are the same build levels; once that is complete, disable FT on the VMs, VMotion the primary VMs to one of the updated hosts, reenable FT, and a new secondary will be spawned on one of the updated hosts that has the same build level. Once all the FT VMs are moved and reenabled, update the remaining hosts so that they are the same build level and then VMotion the VMs around so that they are balanced among all the hosts.
- FT can be enabled and disabled easily at any time; often this is necessary when you need to do something that is not supported when using FT, such as an SVMotion, snapshot, or hot-add of hardware to the VM. In addition, if there are specific time periods when VM availability is critical, such as when a monthly process is running, you can enable it for that time frame to ensure that it stays up while the process is running, and disable it afterward.
- When FT is enabled, any memory limits on the primary VM will be removed and a memory reservation will be set equal to the amount of RAM assigned to the VM. You will be unable to change memory limits, shares, or reservations on the primary VM while FT is enabled.

For more information on FT, check out VMware's Availability Guide that is included as part of the vSphere documentation (http://vmware.com/pdf/vsphere4/r40_u1/vsp_40_u1_availability.pdf).

Summary

In this chapter, we covered some of the more popular advanced features in vSphere. There is a lot to learn about these features, so make sure you read through the documentation and get as much hands-on experience with them as you can before implementing them in a production environment. VMware's Knowledge Base has a great deal of articles specifically about these features, so make sure you look there for any gotchas or compatibility issues as well tips for troubleshooting problems.