VMware vSphere®
Metro Storage Cluster
Case Study

VMware vSphere® 5.0

TECHNICAL MARKETING DOCUMENTATION
V 1.0 MAY 2012
# Table of Contents

Purpose and Overview ......................................................... 3  
Target Audience ................................................................. 3  
Interpreting This Document .................................................. 3  
Reference Material ............................................................... 3  
VMware vSphere Metro Storage Cluster Introduction ...................... 4  
Technical Requirements and Constraints .................................... 4  
Uniform Versus Nonuniform VMware vMSC Configurations ............... 5  
Infrastructure Architecture Case Study .................................... 7  
Infrastructure ................................................................. 7  
VMware vSphere Configuration ............................................... 9  
VMware vSphere 5.0 Update 1 Permanent Device Loss Enhancements .... 13  
VMware DRS ................................................................. 14  
Correcting Affinity Rule Violation ........................................... 16  
VMware vSphere Storage DRS ................................................. 17  
Failure Scenarios ............................................................... 18  
Single-Host Failure in Frimley Datacenter .................................. 19  
Single-Host Isolation in Frimley Datacenter ................................. 20  
Storage Partition .............................................................. 22  
Datacenter Partition ........................................................... 23  
Disk Shelf Failure in Frimley Datacenter .................................... 26  
Full Storage Failure in Frimley Datacenter .................................. 27  
Permanent Device Loss ....................................................... 28  
Full Compute Failure in Frimley Datacenter ................................ 29  
Loss of Frimley Datacenter .................................................... 30  
Summary ................................................................. 31  
About the Authors .............................................................. 31
Purpose and Overview

VMware vSphere® Metro Storage Cluster (VMware vMSC) is a new configuration within the VMware® Hardware Compatibility List (VMware HCL). This type of configuration is commonly referred to as a stretched storage cluster or metro storage cluster. It is implemented in environments where disaster/downtime avoidance is a key requirement. This case study was developed to provide additional insight and information regarding operation of a VMware vMSC infrastructure in conjunction with VMware vSphere. This paper will explain how vSphere handles specific failure scenarios and will discuss various design considerations and operational procedures. For detailed information about storage implementations, refer to documentation provided by the appropriate VMware storage partner.

Target Audience

This document is intended for individuals with a technical background who will be designing, deploying or managing a VMware vMSC infrastructure. This description includes but is not limited to technical consultants, infrastructure architects, IT managers, implementation engineers, partner engineers, sales engineers and customer staff. This solution brief is not intended to replace or override existing certified designs for VMware vMSC solutions. Instead, it is meant to supplement knowledge and provide additional information.

Interpreting This Document

The overall structure of this case study is, for the most part, self-explanatory. Throughout this document, however, there are key points of particular importance that will be highlighted for the reader. These points will be identified with the following label:

NOTE: General point of importance or to add further explanation on a particular section.

The authors of this document assume that the reader is familiar with VMware vSphere, VMware vCenter Server™, VMware vSphere® High Availability (vSphere HA), VMware vSphere® Distributed Resource Scheduler™ (VMware DRS), VMware vSphere® Storage DRS™ (VMware Storage DRS) and replication and storage clustering technology and terminology.

Reference Material

Several VMware knowledge base articles and documents have been used to develop this case study. It is recommended that readers familiarize themselves with the following reference materials:

• vSphere 5.0 support with NetApp MetroCluster
  http://kb.vmware.com/kb/2031038
• VMware vSphere 5 Availability
• VMware vSphere 5 Resource Management
VMware vSphere Metro Storage Cluster

Introduction

A VMware vSphere Metro Storage Cluster configuration is a VMware vSphere® 5 certified solution that combines synchronous replication with array-based clustering. These solutions typically are deployed in environments where the distance between datacenters is limited, often metropolitan or campus environments.

VMware vMSC infrastructures are implemented with the goal of reaping the same benefits that high-availability clusters provide to a local site, but in a geographically dispersed model with two datacenters in different locations. At its core, a VMware vMSC infrastructure is a stretched cluster. The architecture is built on the idea of extending what is defined as “local” in terms of network and storage. This enables these subsystems to span geographies, presenting a single and common base infrastructure set of resources to the vSphere cluster at both sites. In essence, it stretches network and storage between sites.

The primary benefit of a stretched-cluster model is that it enables fully active and workload-balanced datacenters to be used to their full potential. They acquire the capability to migrate virtual machines between sites with VMware vSphere® vMotion® and vSphere Storage vMotion, enabling on-demand and noninvasive mobility of workloads. The capability of a stretched cluster to provide this active balancing of resources should always be the primary design and implementation goal. Although often associated with disaster recovery, VMware vMSC infrastructures are not recommended as primary solutions for pure disaster recovery. This case study will not explain the difference between a disaster recovery and a disaster/downtime avoidance solution. For more details on this distinction, refer to Stretched Clusters and VMware® vCenter™ Site Recovery Manager™, “Understanding the Options and Goals,” located in the VMware Technical Resource Center at http://www.vmware.com/resources/techresources/10262.

Stretched cluster solutions offer the following benefits:

- Workload mobility
- Cross-site automated load balancing
- Enhanced downtime avoidance
- Disaster avoidance

Technical Requirements and Constraints

Due to the technical constraints of an online migration of virtual machines, there are specific requirements that must be met prior to consideration of a stretched-cluster implementation. These following requirements are also listed in the VMware Compatibility Guide:

- Storage connectivity using Fibre Channel, iSCSI, SVD and FCoE is supported.
- The maximum supported network latency between sites for the VMware® ESXi™ management networks is 10ms round-trip time (RTT).
- 10ms of latency for vMotion is supported only with VMware vSphere® Enterprise Plus Edition™ licenses (Metro vMotion).
- The maximum supported latency for synchronous storage replication links is 5ms RTT.
- A minimum of 622Mbps network bandwidth, configured with redundant links, is required for the ESXi vMotion network.
The storage requirements are slightly more complex. A VMware vMSC requires what is in effect a single storage subsystem that spans both sites. In this design, a given datastore must be accessible (able to be read and written to) simultaneously from both sites. Further, when problems occur, the ESXi hosts must be able to continue to access datastores from either array, transparently and without impact to ongoing storage operations.

This precludes traditional synchronous replication solutions, because they create a primary/secondary relationship between the active (primary) LUN, where data is being accessed, and the secondary LUN, which is receiving replication. To access the secondary LUN, replication is stopped (or reversed) and the LUN is made visible to hosts. This now “promoted” secondary LUN has a completely different LUN ID and is essentially a newly available copy of a former primary. This type of solution works for traditional disaster recovery–type configurations, because it is expected that virtual machines must be started up on the secondary site. The VMware vMSC configuration must have simultaneous, uninterrupted access that enables live migration of running virtual machines between sites.

The storage subsystem for a VMware vMSC must be able to be read from and write to the two locations simultaneously. All disk writes are committed synchronously at the two locations to ensure that data is always consistent regardless of the location from which it is being read. This storage architecture requires significant bandwidth and very low latency between the sites involved in the cluster. Increased distances or latencies cause delays to writing to disk, making performance suffer dramatically, and disallow successful vMotion instances between the cluster nodes that reside in different locations.

Uniform Versus Nonuniform VMware vMSC Configurations

VMware vMSC solutions are classified in two distinct categories, based on a fundamental difference in how hosts access storage. It is important to understand the different types of stretched storage solutions because this will influence your design considerations. The following two main categories are as described on the VMware Hardware Compatibility List:

• Uniform host access configuration – When ESXi hosts from both sites are all connected to a storage node in the storage cluster across all sites. Paths presented to ESXi hosts are stretched across distance.
• Nonuniform host access configuration – ESXi hosts in each site are connected only to storage node(s) in the same site. Paths presented to ESXi hosts from storage nodes are limited to the local site.

We will describe the two categories in depth to fully clarify what both mean from an architecture/implementation perspective.

With the uniform configuration, hosts in datacenter-A and datacenter-B have access to the storage systems in both datacenters. In effect, the storage-area network is stretched between the sites, and all hosts can access all LUNs. NetApp MetroCluster is an example of this. In this configuration, read/write access to a LUN takes place on one of the two arrays, and a synchronous mirror is maintained in a hidden, read-only state on the second array. For example, if a LUN containing a datastore is read/write on the array at datacenter-A, all ESXi hosts access that datastore via the array in datacenter-A. For ESXi hosts in datacenter-A, this is local access. ESXi hosts in datacenter-B that are running virtual machines hosted on this datastore send read/write traffic across the network between datacenters. In case of an outage, or operator-controlled shift of control of the LUN to datacenter-B, all ESXi hosts continue to detect the identical LUN being presented, except that it is now accessed via the array in datacenter-B.

The ideal situation is one in which virtual machines access a datastore that is controlled (read/write) by the array in the same datacenter. This minimizes traffic between datacenters and avoids the performance impact of reads’ going across the interconnect.
The notion of “site affinity”—sometimes referred to as “site bias” or “LUN locality”—for a virtual machine is dictated by the read/write copy of the datastore. For example, when a virtual machine has site affinity with datacenter-A, its read/write copy of the datastore is located in datacenter-A. This will be explained in greater detail in the “vSphere Distributed Resource Scheduler” section of this paper.

With the nonuniform configuration, hosts in datacenter-A have access only to the array within the local datacenter. This array (as well as its peer array in the opposite datacenter) is responsible for providing access to datastores in one datacenter from ESXi hosts in the opposite datacenter. Nonuniform configurations typically leverage the concept of a “virtual LUN.” This enables ESXi hosts in each datacenter to read and write to the same datastore/LUN. The clustering solution maintains the cache state on each array, so an ESXi host in either datacenter detects the LUN as local. Even when two virtual machines reside on the same datastore but are located in different datacenters, they write locally without any performance impact on either of them. A key point in this configuration is that each of the LUNs/datastores has “site affinity” defined. In other words, if anything happens to the link between the sites, the storage system on the preferred site for a given datastore is the only remaining one that has read/write access to it, thereby preventing any data corruption in the case of a failure scenario.
Because uniform configurations are currently the most commonly deployed, our test case uses uniform storage. Many of the design considerations, however, also apply to nonuniform configurations. We will point out the exceptions when this is not the case.

**Infrastructure Architecture Case Study**

In this section, we will describe the architecture deployed for this case study. We also will discuss some of the basic configuration and performance aspects of the various vSphere features. For an in-depth explanation of each feature, refer to the vSphere 5.0 Availability Guide and the vSphere 5.0 Resource Management Guide. We will make specific recommendations based on VMware best practices and will provide operational guidance where applicable. It will be explained in our failure scenarios how these best practices prevent or limit downtime.

**Infrastructure**

The infrastructure used for this case study consists of a single VMware vSphere 5.0 Update 1 cluster with four ESXi hosts. These hosts are managed by a VMware vSphere vCenter Server, Revision 5.0 Update 1. It was decided to use VMware vSphere 5.0 Update 1 to enable testing of the improved handling of permanent device loss (PDL) scenarios. These enhancements were introduced primarily for stretched-cluster environments. We will discuss this in greater detail in the “vSphere HA” subsection of this document.

For the purpose of our tests, we have simulated a customer environment with two sites, Frimley and Bluefin (referring to the cities in which VMware has offices in the United Kingdom). The network between the Frimley and Bluefin datacenters is a stretched layer 2 network with a minimal distance between them, as is typical in campus cluster scenarios.

Each site has two ESXi hosts. The vCenter Server is configured with VMware DRS affinity to the hosts in the Bluefin datacenter. In a stretched-cluster environment, only a single vCenter Server instance is used. This is different from a traditional VMware Site Recovery Manager configuration, in which a dual vCenter Server configuration is required. The configuration of virtual machine–host affinity rules is discussed in greater detail in the “vSphere DRS” subsection of this document. In our scenario, iSCSI is the main protocol. When using a NetApp MetroCluster, an iSCSI connection is configured to a particular virtual IP address that enables ESXi hosts...
to connect to a given storage controller. In case of a failure, this IP address shifts to the opposite storage controller, enabling seamless access to storage without requiring reconfiguration of the target storage IP address.

Eight LUNs have been created. Four of these are accessed through the virtual iSCSI IP address active in the Frimley datacenter and four are accessed through the virtual iSCSI IP address active in the Bluefin datacenter.
The VMware vSphere 5.0 Update 1 cluster is connected to a NetApp MetroCluster in a fabric configuration in a uniform device access model, which means that every host in the cluster is connected to both storage heads. Each of the heads is connected to two Brocade switches, which are connected to two similar switches in the secondary location. For any given LUN, one of the two storage heads presents the LUN as read/write via iSCSI. The opposite storage head maintains the replicated, read-only copy that is effectively hidden from the ESXi hosts. This configuration is described in depth in NetApp Technical Report TR-3548, *Best Practices for MetroCluster Design and Implementation*. The following diagram from that report depicts how the storage environment is connected:

**VMware vSphere Configuration**

In this case study, our focus is on vSphere HA, VMware DRS and VMware Storage DRS in relation to stretched-cluster environments, because design and operational considerations regarding vSphere are commonly overlooked and underestimated. Much emphasis traditionally has been placed on the storage layer, with little thought having been applied as to how the workloads will be provisioned and managed.

Workload balance and disaster avoidance are key drivers for using a stretched cluster. How do we ensure that our environment is properly balanced without impacting availability or severely increasing operational expenditure? How do we build the requirements into our provisioning process and periodically validate that we still meet them? Ignoring the requirements makes the environment confusing to administer and less predictable during the various failure scenarios in which these requirements would be looked to for help.
Each of the following three vSphere features has very specific configuration requirements and can enhance the resiliency of your environment and availability of your workload. Architectural recommendations will be made throughout this section. They are based on our findings from the testing of the various failure scenarios, each of which is documented in one of the following sections.

**vSphere HA**

The environment has four hosts and a uniform stretched-storage solution. Because a full-site failure is one scenario that must be taken into account in a resilient architecture, VMware recommends enabling admission control. And because workload availability is the primary driver for most stretched-cluster environments, it is recommended that sufficient capacity be allotted for a full-site failure. Further, because such hosts are equally divided across the two sites, and to ensure that all workloads can be restarted by vSphere HA, configuring the admission control policy to 50 percent is advised.

VMware recommends using the percentage-based policy because it offers the most flexibility and reduces operational overhead. Even when new hosts are introduced to the environment, there is no need to change the percentage and there is no risk of a skewed consolidation ratio due to the possible use of virtual machine–level reservations. For more details about admission control policies and the associated algorithms, refer to the *vSphere 5.0 Availability Guide* and the *vSphere 5.0 HA Best Practices Guide*.

The following screenshot shows a vSphere HA cluster configured with Admission Control set on Enable and using the percentage-based Admission Control Policy set at 50%:

vSphere HA uses heartbeat mechanisms to validate the state of a host. There are two heartbeat mechanisms, namely network heartbeating and datastore heartbeating. Network heartbeating is the primary mechanism for vSphere HA, used to validate the availability of a host. Datastore heartbeating is the secondary mechanism for vSphere HA, used to determine the exact state of the host after network heartbeating has failed.
If a host is not receiving any heartbeats, it uses a fail-safe mechanism to detect whether it is merely isolated from its master node or is completely isolated from the network. It does this by pinging the default gateway. In addition to this mechanism, one or more isolation addresses can be specified manually to enhance the reliability of isolation validation. VMware recommends specifying a minimum of two additional isolation addresses and that each of these addresses be site local.

In our scenario, one of these addresses physically resides in the Frimley datacenter and the other physically resides in the Bluefin datacenter. This enables vSphere HA validating for complete network isolation, even in the case of a connection failure between sites. The following screenshot shows an example of how to configure multiple isolation addresses. The vSphere HA advanced setting used is das.isolationaddress. More details on how to configure this can be found in VMware knowledge base article 1002117.

For vSphere HA datastore heartbeating functionality to operate correctly in any type of failure scenario, VMware recommends increasing the number of heartbeat datastores from two to four. The minimum number of heartbeat datastores is two and the maximum is five. Four is recommended in a stretched-cluster environment because that would provide full redundancy in both locations. Selecting four specific datastores—two from one site and two from the other—as preferred heartbeat datastores is also recommended. This enables vSphere HA to heartbeat to a datastore even in the case of a connection failure between sites. Subsequently, it enables vSphere HA to determine the state of a host in any scenario.

The number of heartbeat datastores can be increased by adding an advanced setting called das.heartbeatDsPerHost, which is illustrated in the following screenshot.

NOTE: vSphere HA advanced settings can be configured via the following steps:

- Edit settings on your cluster object.
- Click vSphere HA.
- Click Advanced Options.
VMware recommends using **Select any of the cluster datastores taking into account my preferences.** This enables vSphere HA to select any other datastore if the four designated datastores that we have manually selected become unavailable. This setting is shown in the following screenshot:
VMware vSphere 5.0 Update 1 Permanent Device Loss Enhancements

As of VMware vSphere 5.0 Update 1, enhancements have been introduced to enable an automated failover of virtual machines residing on a datastore that has a PDL condition. A PDL condition, as will be shown in one of our failure scenarios, is one that is communicated by the array controller to ESXi via an SCSI sense code. This condition indicates that a device (LUN) has become unavailable and is likely to be permanently unavailable. An example scenario in which this condition would be communicated by the array is when a LUN is set offline. This condition is used in nonuniform models during a failure scenario, to ensure that ESXi takes appropriate action when access to a LUN is revoked. When a full storage failure occurs, it is impossible to generate the PDL condition because there is no chance of communication between the array and the ESXi host. This state will be identified by the ESXi host as an all paths down (APD) condition.

The following settings apply only to a PDL condition and not to an APD condition. In the failure scenarios, we will demonstrate the performance difference for these two conditions.

Two advanced settings have been introduced in VMware vSphere 5.0 Update 1 to enable vSphere HA to respond to a PDL condition. The first setting, `disk.terminateVMonPDLDefault`, is configured on a host level in `/etc/vmware/settings` and should be set to `True` by default. This is a per-host setting, and the host requires a reboot for it to take effect. This setting ensures that a virtual machine is killed when the datastore on which it resides enters a PDL state. The virtual machine is killed as soon as it initiates disk I/O on a datastore that is in a PDL condition and all of the virtual machine files reside on this datastore. If virtual machine files do not all reside on the same datastore and a PDL condition exists on one of the datastores, the virtual machine will not be killed. VMware recommends placing all files for a given virtual machine on a single datastore, ensuring that PDL conditions can be mitigated by vSphere HA. VMware also recommends setting `disk.terminateVMonPDLDefault` to `True`. A virtual machine is killed only when issuing I/O to the datastore. Otherwise, it remains active. A virtual machine that is running memory-intensive workloads without issuing I/O to the datastore might remain active in such situations.

The second setting is a vSphere HA advanced setting called `das.maskCleanShutdownEnabled`. It was introduced in VMware vSphere 5.0 Update 1 and is not enabled by default. It must be set to `True` on vSphere HA cluster(s). This setting enables vSphere HA to trigger a restart response for a virtual machine that has been killed automatically due to a PDL condition. This enables vSphere HA to differentiate between a virtual machine that was killed due to the PDL state and a virtual machine that has been powered off by an administrator.
VMware recommends setting `das.maskCleanShutdownEnabled` to `True` to limit downtime for virtual machines residing on datastores in a PDL condition. When `das.maskCleanShutdownEnabled` is not set to `True` and a PDL condition exists while `disk.terminateVMonPDLDefault` is set to `True`, virtual machine restart will not occur after virtual machines have been killed. This is because vSphere HA will determine that these virtual machines have been powered off or shut down manually by the administrator.

**VMware DRS**

VMware DRS is used in many environments to distribute load within a cluster. It offers many features that can be very helpful in stretched environments. VMware recommends enabling VMware DRS to allow for load balancing across hosts in the cluster. Its load balancing calculation is based on CPU and memory use. As such, care must be taken with regard to storage and networking resources as well as traffic flow. To prevent storage and network traffic overhead in a stretched-cluster environment, VMware recommends implementing VMware DRS affinity rules to enable a logical separation of virtual machines. This will subsequently help improve availability. It will also help by ensuring separation of these services across sites for virtual machines that are responsible for infrastructure services such as Microsoft Active Directory and DNS.

VMware DRS affinity rules also help prevent unnecessary downtime and storage and network traffic flow overhead by enforcing desired site affinity. VMware recommends aligning vSphere virtual machine–host affinity rules with the storage configuration, that is, setting virtual machine–host affinity rules so that a virtual machine tends to run on a host at the same site as the array that is configured as the primary read/write node for a given datastore. For example, in our test configuration, virtual machines stored on the Frimley-01 datastore are set with virtual machine–host affinity to favor hosts in the Frimley datacenter. Because virtual machine–host affinity rules are designed to ensure that virtual machines stay local to the storage that is primary for their datastore, this ensures that in the case of a network connection failure between sites, virtual machines will not lose connection with that particular storage system. This coincidentally also results in all read I/O staying local.

**NOTE:** Different storage vendors use different terminology to describe the relationship of a LUN to a particular array or controller. For the purpose of this document, we will use the generic term “storage site affinity,” which means the preferred location for access to a given LUN.

VMware recommends implementing “should rules,” because these are violated by vSphere HA in the case of a failure. Availability of services should always prevail over performance. In the case of “must rules,” vSphere HA does not violate the rule set. This might potentially lead to service outages. In the scenario where a full datacenter fails, “must rules” make it impossible for vSphere HA to restart the virtual machines, because they do not have the required affinity to be allowed to start on the hosts in the other datacenter. VMware DRS communicates these rules to vSphere HA, and these are stored in a “compatibility list” governing allowed startup. VMware DRS, under certain circumstances such as massive host saturation coupled with aggressive recommendation settings, can also violate “should” rules. Although this is very rare, we recommended monitoring for violation of these rules, because a violation can possibly impact the availability and performance of your workload.

VMware recommends manually defining “sites” by creating a group of hosts that belong to a site and adding virtual machines to these sites based on the affinity of the datastore on which they are provisioned. In our scenario, only a limited number of virtual machines are provisioned. VMware recommends automating the process of defining site affinity by using tools such as VMware vCenter™ Orchestrator™ or VMware vSphere® PowerCLI™. If automating the process is not an option, using a generic naming convention is recommended, to enable simplifying the creation of these groups. VMware recommends that these groups be validated on a regular basis to ensure that all virtual machines belong to the group with the correct site affinity.

The following screenshots depict the configuration used for this case study. In the first screenshot, all the virtual machines that should remain local to the Bluefin location are added to the Bluefin virtual machine group.
Next, a Bluefin host group is created, containing all hosts residing in this location.

Next, a new rule is created that is defined as a “should run on” rule. It links the host group and the virtual machine group for the Bluefin location.
This should be done for both locations, which should result in two rules.

**Correcting Affinity Rule Violation**

VMware DRS assigns a high priority to correcting affinity rule violations. During invocation, the primary goal of VMware DRS is to correct any violations and generate recommendations to migrate virtual machines to the hosts listed in the Cluster Host Group. These moves have a higher priority than load-balancing moves and will be started before load-balancing moves.

VMware DRS is invoked every 5 minutes by default, but it also is triggered if the cluster detects changes. For instance, when a host reconnects to the cluster, VMware DRS is invoked and generates recommendations to correct the violation. Our testing has shown that VMware DRS generates recommendations to correct affinity rules violations within 30 seconds after a host reconnects to the cluster. VMware DRS is limited by the overall capacity of the vMotion network, so it might take multiple invocations before all affinity rule violations are corrected.
VMware vSphere Storage DRS

VMware Storage DRS enables aggregation of datastores into a single unit of consumption from an administrative perspective, and it balances virtual machine disks when defined thresholds are exceeded. It ensures that sufficient disk resources are available to your workload. VMware recommends enabling VMware Storage DRS.

VMware Storage DRS uses Storage vMotion to migrate virtual machine disks between datastores within a datastore cluster. Because the underlying stretched storage systems use synchronous replication, a migration or series of migrations have an impact on replication traffic and might cause the virtual machines to become temporarily unavailable due to contention for network resources during the movement of disks. Migration to random datastores might also potentially lead to additional I/O latency in uniform access configurations if virtual machines are not migrated along with their virtual disks, from a site perspective. For example, if a virtual machine resident on a host in site A has its disk migrated to a datastore in site B, it will continue operating but with potentially degraded performance. The virtual machine's disk reads will now be subject to the increased latency associated with reading from the virtual iSCSI IP at site B, and reads will be subject to intersite latency instead of being satisfied by a local target.

To control when and if migrations occur, VMware recommends that VMware Storage DRS be configured in manual mode. This enables human validation per recommendation and allows recommendations to be applied during off-peak hours while gaining the operational benefit and efficiency of the initial-placement functionality.

VMware recommends creating datastore clusters based on the storage configuration with respect to storage site affinity. Datastores with a site affinity for site A should not be mixed in datastore clusters with datastores with a site affinity for site B. This enables operational consistency and eases the creation and ongoing management of VMware DRS virtual machine–host affinity rules. It is recommended that all VMware DRS virtual machine–host affinity rules be updated accordingly when virtual machines are migrated via Storage vMotion between datastore clusters and when crossing defined storage-site affinity boundaries. To simplify the provisioning process, VMware recommends aligning naming conventions for datastore clusters and virtual machine–host affinity rules.

The naming convention used in our testing gives both datastores and datastore clusters a site-specific name to provide ease of alignment of VMware DRS host affinity with virtual machine deployment in the correlate site. The following graphic illustrates the site-specific storage layout in our Bluefin and Frimley sites. vCenter Server map functionality cannot be used to view site affinity regarding storage, because it currently does not display datastore cluster objects.
Failure Scenarios

There are many failures that can be introduced in clustered systems. But in a properly architected environment, vSphere HA, VMware DRS and the storage subsystem will not detect many of them. We will not address the zero-impact failures, such as the failure of a single network cable, because they are explained in depth in the documentation provided by the storage vendor of your chosen solution. We will discuss the following “common” failure scenarios:

• Single-host failure in Frimley datacenter
• Single-host isolation in Frimley datacenter
• Storage partition
• Datacenter partition
• Disk-shelf failure in Frimley datacenter
• Full storage failure in Frimley datacenter
• Full compute failure in Frimley datacenter
• Full compute failure in Frimley datacenter and full storage failure in Bluefin datacenter
• Loss of complete Frimley datacenter

We will also examine scenarios in which specific settings are incorrectly configured. These settings will determine the availability and recoverability of your virtual machines in a failure scenario. Therefore, it is important to understand the following examples of the impact of misconfiguration:

• Incorrectly configured virtual machine-host affinity rules
• Incorrectly configured heartbeat datastores
• Incorrectly configured isolation address
• Incorrectly configured PDL handling
• vCenter Server split-brain scenario

All of these scenarios have been extensively tested. The following are our findings per scenario and possible recommendations where applicable.
Single-Host Failure in Frimley Datacenter

In this scenario, we simulated the complete failure of a host in Frimley datacenter by pulling the power cables on a live system. This scenario is depicted in the following diagram:

Result
All virtual machines were restarted successfully by vSphere HA in accordance with host affinity rules.

Explanation
If a host fails, the failure is detected by the cluster’s vSphere HA master node because network heartbeats from it are no longer being received. After the master node has detected that network heartbeats are absent, it will start monitoring for datastore heartbeats. Because the host has failed completely, it cannot generate datastore heartbeats, so these too will be detected as absent by the vSphere HA master node. During this time, a third availability check is performed by pinging the management addresses of the failed hosts. If all of these checks return as unsuccessful, the master node will declare the missing host dead and will attempt to restart all the protected virtual machines that were running on the host before the master node lost contact with the host.

The VMware DRS virtual machine–host affinity rules defined on a cluster level are “should rules”; as such, the virtual machine can potentially be restarted on a host in the other datacenter. In our testing, we witnessed this type of operation multiple times, where virtual machines started on any available host in the cluster, including hosts at Bluefin datacenter. VMware DRS will then attempt to correct any violated affinity rules at the first invocation and will automatically migrate virtual machines in accordance with their affinity rules, to bring virtual machine placement in alignment. VMware recommends manually invoking VMware DRS to ensure that all virtual machines are placed on hosts in the correct location, to prevent possible performance degradation due to
misplacement. In our scenario, misplacement would lead to increased latency, because the virtual machine would be accessing storage in the other location. The following screenshot depicts how to manually run VMware DRS from the DRS tab on a vCenter Server cluster object within the VMware vSphere Cient:

---

**Single-Host Isolation in Frimley Datacenter**

In this scenario, we isolated a single host in Frimley datacenter from the rest of the network by disconnecting all network links.

---

**Result**

Virtual machines remained running because isolation response was configured to Leave Powered On.

**Explanation**

When a host is isolated, the isolation is detected by the vSphere HA master node because network heartbeats from it are no longer being received. When the master node detects that network heartbeats are absent, it starts monitoring for datastore heartbeats. Because the host is isolated, it generates datastore heartbeats for the secondary vSphere HA detection mechanism. Detection of valid host heartbeats enables the vSphere HA master node to determine that the host is running but is isolated from the network. Depending on the isolation response configured, the impacted host might choose to Power Off or Shut Down virtual machines, or alternatively to leave the virtual machines powered on. The isolation response is triggered 30 seconds after the host has detected that it is isolated.
VMware recommends aligning the isolation response to business requirements and physical constraints. From a best practices perspective, Leave Powered On is the recommended isolation response setting for the majority of environments. Isolated hosts are a rare event in a properly architectured environment, given the built-in redundancy of most modern designs. In environments that use network-based storage protocols, such as iSCSI and NFS, the recommended isolation response is Power Off. With these environments, it is more likely that a network outage that causes a host to become isolated will also affect the host’s ability to communicate to the datastores.

If an isolation response other than the recommended Leave Powered On is selected and a Power Off or Shut Down isolation response is triggered, virtual machines will be restarted by the vSphere HA master node on the remaining nodes in the cluster. The VMware DRS virtual machine–host affinity rules defined on a cluster level are “should rules”; as such, the virtual machine can potentially be restarted on a host in the other datacenter. In our testing, we have witnessed this type of operation multiple times. VMware recommends manually invoking VMware DRS following host isolation and the associated cluster response. This ensures that all virtual machines are placed on hosts in the correct location, and it prevents possible performance degradation due to misplacement. In our scenario, misplacement would lead to increased latency because the virtual machine would be accessing storage in the other location.
Storage Partition

In this scenario, a failure was simulated on the storage network between datacenters, as depicted in the following diagram:

![Diagram showing storage partition scenario]

**Result**

Virtual machines remained running, with no impact.

**Explanation**

Each LUN has storage site affinity defined and VMware DRS rules aligned with this affinity. Therefore, none of the virtual machines were impacted, because their storage remained available within the site.

If for any reason the affinity rule for a virtual machine has been violated and the virtual machine is running on a host in Frimley datacenter while its disk resides on a datastore that has affinity with Bluefin datacenter, the virtual machine would not be able to successfully issue I/O following an intersite storage partition. This is because the datastore would be in an APD condition. Because the vSphere HA master node would still be receiving network heartbeats from all hosts in the cluster, it would not take any action. Any virtual machines that had violated their site affinity would not be restarted by vSphere HA. These virtual machines must be powered off manually and then powered on manually to be fully functional again.

To prevent unnecessary downtime in an APD scenario, VMware recommends monitoring compliance of VMware DRS rules. Although VMware DRS is invoked every 5 minutes, this does not guarantee that all affinity rule violations will be resolved. Therefore, rigid monitoring is recommended, enabling quick identification of anomalies while preventing unnecessary downtime.
Datacenter Partition

In this scenario, we fully isolate Frimley datacenter from Bluefin datacenter, as depicted in the following diagram:

Result
Virtual machines remained running, with no impact.

Explanation
In this test, the two datacenters were fully isolated from each other. This scenario is similar to both the storage partition and host isolation scenarios. Virtual machines were not impacted by this failure because VMware DRS rules were correctly implemented and no rules were violated.

vSphere HA follows a logical process to determine which virtual machines require restarting during a cluster partition. The vSphere HA master node running in Frimley datacenter detects that all hosts in Bluefin datacenter are unreachable. The vSphere HA master node first detects that no network heartbeats are being received. After that, it validates whether there are any storage heartbeats being generated; this check will not detect storage heartbeats, because the storage connection between sites has also failed and the heartbeat datastore will be updated only “locally.” Because the virtual machines with affinity to the remaining hosts are still running, no action is needed for them. Next, vSphere HA validates whether a restart can be attempted. However, the read/write versions of the datastores located in Bluefin are not accessible by the hosts in Frimley; as such, no attempt will be made to start the missing virtual machines.

Likewise, the ESXi hosts in Bluefin datacenter detect that there is no master node available and initiate a master election process. After the master node has been elected, they will try to discover which virtual machines were running before the failure and will attempt to restart them. Because all virtual machines with affinity to Bluefin datacenter are still running in Bluefin datacenter, there is no need for a restart. Only the virtual machines with
affinity to Frimley datacenter are unavailable, and vSphere HA will not be able to restart these virtual machines because the datastores on which they are stored have affinity with Frimley datacenter and are unavailable in Bluefin datacenter.

If host affinity rules have been violated, that is, virtual machines had been running in a location where storage is not defined as read/write by default, performance changes. We intentionally violated the configured affinity rules to document the performance. We manually moved a virtual machine from Frimley datacenter to Bluefin datacenter; the virtual machine ran on a host in Bluefin datacenter but was accessing the datastore in Frimley datacenter. The following sequence was witnessed when the datacenters were isolated from each other:

1. The virtual machine with affinity to Frimley datacenter but residing in Bluefin datacenter was unable to reach its datastore. This resulted in its being unable to write to or read from disk.

2. In Frimley datacenter, this virtual machine was restarted by vSphere HA because the hosts in Frimley datacenter did not detect the instance running in Bluefin datacenter.

3. Because the datastore was available only to Frimley datacenter, one of the hosts in Frimley datacenter acquired a lock on the VMDK and was able to power on this virtual machine.

4. This created a scenario in which the same virtual machine was powered on and running in both datacenters.

Why is this possible?

- The network heartbeat from the host that is running this virtual machine is missing because there is no connection to that site.
- The datastore heartbeat is missing because there is no connection to that site.
- A ping to the management address of the host that was running the virtual machine fails because there is no connection to that site.
- The master node located in Frimley datacenter detects that the virtual machine was powered on before the failure. Because it’s unable to communicate after the failure with the virtual machine’s host in Bluefin datacenter, it will attempt to restart the virtual machine.
It is determined that the datastore in Bluefin datacenter is APD; as such, no action is taken. (As explained previously, the virtual machine will be automatically killed only with a PDL condition.)

If the connection between sites is restored, a classic "virtual machine split-brain scenario" will exist. For a short period of time, two copies of the virtual machine, each having the same MAC address, will be active on the network. Only one copy, however, will have access to the virtual machine files, and vSphere HA will detect this. As soon as it is detected, all processes belonging to the virtual machine copy that has no access to the virtual machine files will be killed, as depicted in the following screenshot:

In this example, the unnecessary downtime equates to a virtual machine's having to be restarted. This would not be necessary if site affinity were maintained correctly. Therefore, to prevent unnecessary downtime, VMware recommends closely monitoring that VMware DRS rules align with datastore site affinity.
Disk Shelf Failure in Frimley Datacenter

In this scenario, one of the disk shelves in Frimley datacenter has failed. Both Frimley01 and Frimley02 on Storage A are impacted.

Result
Virtual machines remained running, with no impact.

Explanation
In this scenario, only a disk shelf in Frimley datacenter failed. The storage processor recognized the failure and instantly switched from the primary disk shelf in Frimley datacenter to the mirror copy in Bluefin datacenter. There was no noticeable impact to any of the virtual machines, except for a short spike in terms of I/O response time. This scenario is fully recognized and handled by the stretched-storage solution. There is no need for a rescan of the datastores or the HBAs because the switchover is seamless and, from the ESXi perspective, the LUNs are identical.
Full Storage Failure in Frimley Datacenter

In this scenario, we tested a full storage system failure in Frimley datacenter, as depicted in the following diagram:

Result
Virtual machines remained running, with no impact.

Explanation
When the full storage system failed in Frimley datacenter, a takeover command had to be initiated manually. As described previously, we used a NetApp MetroCluster configuration. This takeover command is particular to NetApp environments. After the command was initiated, the mirrored, read-only copy of each of the failed datastores was set to read/write and was instantly accessible. We are describing this process on an extremely high level. For more details, refer to the storage vendor’s documentation.

From a virtual machine’s perspective, this failover is seamless. The storage controllers handle it, and no action is required from the vSphere or storage administrator. All I/O will now pass across the intrasite connection to the other datacenter because virtual machines will remain running in Frimley datacenter while their datastores are accessible only in Bluefin datacenter.

vSphere HA does not detect this type of failure. Although the datastore heartbeat might be lost briefly, vSphere HA will not take action because the vSphere HA master node checks for the datastore heartbeat only when the network heartbeat has not been received for 3 seconds. Because the network heartbeat remained available throughout the storage failure, vSphere HA was not required to initiate any restarts.
Permanent Device Loss

In this scenario, we tested a PDL condition. Because this scenario is uncommon in uniform configurations, we forced it by setting a LUN to offline. PDL conditions are more common in nonuniform VMware vMSC configurations.

Result
Virtual machines were killed by ESXi and restarted by vSphere HA.

Explanation
When the PDL condition was simulated, the virtual machines residing on the datastore were killed instantly. (More details about PDL can be found in the “VMware vSphere 5.0 Update 1 Permanent Device Loss Enhancements” section of this paper.) After being killed, the virtual machines were restarted by vSphere HA. The PDL and killing of the virtual machine world group can be observed by following the vmkernel.log file located in /var/log/. The following is an outtake of the vmkernel.log file in which a PDL is recognized and appropriate action is taken:


VMware recommends configuring advanced options disk.terminateVMOnPDLDefault and dasmaskCleanShutdown Enabled to True. If they are not configured—they are by default set to False—vSphere HA will not take any action and the virtual machines affected by a PDL might not be restarted. This is described in depth in the “VMware vSphere 5.0 Update 1 Permanent Device Loss Enhancements” section of this paper.
Full Compute Failure in Frimley Datacenter

In this scenario, we tested a full compute failure in Frimley datacenter by simultaneously removing power from all hosts within the site.

Result

All virtual machines were successfully restarted in Bluefin datacenter.

Explanation

The vSphere HA master node was located in Frimley datacenter at the time of the full compute failure there. After the hosts in Bluefin datacenter detected that no network heartbeats were being received, an election process was started. Within approximately 20 seconds, a new vSphere HA master node was elected from the remaining hosts. Subsequently, the new master node determined which hosts had failed and which virtual machines had been impacted by this failure. Because all hosts at the other site had failed and all virtual machines they held had been impacted, vSphere HA initiated the restart of all of these virtual machines. vSphere HA can initiate 32 concurrent restarts on a single host, providing a low restart latency for most environments. The only sequencing of start order comes from the broad High, Medium and Low categories for vSphere HA. This is a policy that must be set on a per–virtual machine basis. These policies were observed as being adhered to. High-priority virtual machines started first, followed by the medium- and low-priority virtual machines.

As part of the test, we powered on the hosts in Frimley datacenter again. As soon as VMware DRS detected that these hosts were again available, a VMware DRS run was invoked. This initial VMware DRS execution solves only the violated VMware DRS affinity rules. Because the initial VMware DRS run corrects only the VMware DRS rule violations, resource imbalance was not correct until the next full invocation of VMware DRS. VMware DRS is invoked by default every 5 minutes or when virtual machines are powered off or on through the use of the vSphere Client.
Loss of Frimley Datacenter

In this scenario, a full failure of Frimley datacenter is simulated.

**Result**

All virtual machines were successfully restarted in Bluefin datacenter.

**Explanation**

In this scenario, the hosts in Bluefin datacenter lost contact with the vSphere HA master node and elected a new vSphere HA master node. Because the storage system had failed, a takeover command was initiated on the surviving site, again due to a NetApp-specific process. After the takeover command had been initiated, the new vSphere HA master node accessed the per-datastore files that vSphere HA uses to record the set of protected virtual machines. The vSphere HA master node then attempted to restart those virtual machines that were not running on the surviving hosts in Bluefin datacenter. In our scenario, all virtual machines were restarted within 2 minutes after failure and were fully accessible and functional again.

*NOTE: vSphere HA will stop attempting to start a virtual machine after 30 minutes by default. If the storage team has not issued the takeover command within that time frame, the vSphere administrator must manually start virtual machines when the storage is available.*
Summary

When properly operated and architected, stretched clusters are excellent solutions that increase resiliency and offer intersite workload mobility. There has been some confusion regarding failure scenarios and the different types of responses from both the vSphere layer and the storage layer. In this case study, we have tried to explain how vSphere HA and VMware DRS respond to certain failures in a stretched-cluster environment. We have also offered recommendations regarding configuration of a vSphere cluster in this type of environment. This paper has highlighted the importance of site affinity, the role that VMware DRS rules and groups play in this process, how vSphere HA interacts with these rules and groups, and how customers must ensure that the logic mandated by these rules and groups is enforced over time if the reliability and predictability of the cluster are to be maintained.

About the Authors

Duncan Epping is principal architect in the Technical Marketing group at VMware. His primary focus is on cloud infrastructure management architecture. He was among the first VMware certified design experts (VCDX 007). Duncan is the co-author of several books, including best sellers vSphere 5.0 Clustering Technical Deepdive and VMware vSphere 4.1 HA and DRS Technical Deepdive. He is the owner and main author of the leading virtualization blog yellow-bricks.com.

- Follow Duncan on Twitter: @DuncanYB.

Frank Denneman is senior architect in the Technical Marketing group at VMware. His primary focus is on resource management and downtime avoidance. He specializes in VMware DRS, VMware Storage DRS, VMware vSphere vMotion, and VMware vSphere Storage vMotion. Frank is the co-author of multiple books, including best sellers vSphere 5.0 Clustering Technical Deepdive and VMware vSphere 4.1 HA and DRS Technical Deepdive. He is the owner and main author of one of the top virtualization and VMware blogs, frankdenneman.com.

- Follow Frank on Twitter: @frankdenneman.

Ken Werneburg is senior technical marketing manager at VMware. He works on business continuity and disaster recovery solutions in the Cloud and Infrastructure Management group. Ken has been with VMware since 2006.

- Follow Ken on Twitter: @vmKen.

Lee Dilworth is principal systems engineer with VMware. His primary focus is on availability and disaster recovery solutions. He has been with VMware since 2005.

- Follow Lee on Twitter: @LeeDilworth.

Stuart Hardman is a technical systems engineer with VMware. He is responsible for managing and maintaining the VMware Briefing Center in the VMware Frimley (UK) office. He works on partner solution stacks and their VMware integration points. His primary focus is on producing customer-facing collateral and supporting the field systems engineers. Stuart has spent most of his career in either the partner or reseller channel. He joined VMware late in 2010 after a stint with the British Royal Navy.

- Follow Stuart on Twitter: @Shard_man.