Best Practices for Running VMware vSphere® on iSCSI

TECHNICAL MARKETING DOCUMENTATION
V 2.0A/JULY 2013

vmware®

Table of Contents

Introduction 4
1. iSCSI Overview 4
1.1 iSCSI Considerations 4
2. iSCSI Architecture 5
2.1 iSCSI Names 5
2.2 iSCSI Initiators and Targets 5
2.3 iSCSI Sessions and Connections 6
2.4 iSCSI Portals 6
3. iSCSI Implementation Options 7
3.1 Mixing iSCSI Options 8
4. Networking Settings 8
4.1 VMkernel Network Configuration 9
4.2 IPv6 Supportability Statements 9
4.3 Throughput Options 9
4.4 Minimizing Latency 9
4.5 Routing 10
4.6 Availability Options – Multipathing or NIC Teaming 10
4.7 Error Correction Digests 11
4.8 Flow Control 11
5. Security Considerations 11
5.1 Private Network 11
5.2 Encryption 11
5.3 Authentication 11
6. iSCSI Datastore Provisioning Steps 12
6.1 Software iSCSI Port Binding 13
6.2 Software iSCSI Multipathing Configuration Steps 14
7. Interoperability Considerations 16
7.1 Storage I/O Control 16
7.2 Network I/O Control 16
7.3 vSphere Storage DRS 17
7.4 vSphere Storage APIs – Array Integration 18
7.5 vCenter Site Recovery Manager/vSphere Replication 18
7.6 vSphere Storage vMotion 18

Table of Contents (continued)

8. Sizing Considerations. 19
8.1 Recommended Volume Size 19
8.2 Recommended Block Size 19
8.3 Maximum Number of Virtual Machines per Datastore 20
9. Booting a vSphere Host from Software iSCSI 20
9.1 Why Boot from SAN? 20
9.2 Compatible Network Interface Card 20
9.3 Configuring Host BIOS for iSCSI Boot 21
9.4 Installing a vSphere Host on an iSCSI LUN. 21
9.5 Booting from an iSCSI LUN 22
9.6 Troubleshooting Checklist When Booting from iSCSI. 22
10. Advanced Settings 22
10.1 LoginTimeout. 22
10.2 LogoutTimeout 22
10.3 RecoveryTimeout 22
10.4 NoopInterval and NoopTimeout 23
10.5 QFullSampleSize and QFullThreshold 23
10.6 Disk.DiskMaxIOSize 23
10.7 DelayedAck 23
11. Additional Considerations. 23
11.1 Disk Alignment 23
11.2 Microsoft Clustering Support 23
11.3 In-Guest iSCSI Support. 24
11.4 All Paths Down and Permanent Device Loss. 24
11.5 Read-Only File Systems on Linux Guest OS 24
11.6 Round Robin Path Policy Setting IOPS=1. 24
11.7 Data Center Bridging (DCB) Support 24
11.8 Tape Device Support 24
Conclusion 25
Acknowledgments. 25
About the Author 25

Introduction

VMware offers and supports a number of different storage technologies and protocols for presenting external storage devices to VMware vSphere® hosts. In recent years, the iSCSI protocol has gained popularity as a method for presenting block storage devices over a network to vSphere hosts. VMware has provided support for iSCSI storage since Virtual Infrastructure 3. This paper can help you understand the design considerations and deployment options for deploying vSphere infrastructures using iSCSI storage. It highlights trade-offs and factors to consider when deploying iSCSI storage to support vSphere environments. It is a complement to, not a replacement for, VMware® product documentation.

1. iSCSI Overview

iSCSI is a protocol that uses the TCP to transport SCSI commands, enabling the use of the existing TCP/IP networking infrastructure as a SAN. As with SCSI over Fibre Channel (FC), iSCSI presents SCSI targets and devices to iSCSI initiators (requesters). Unlike NAS, which presents devices at the file level, iSCSI makes block devices available via the network. Block devices are presented across an IP network to your local system. These can be consumed in the same way as any other block storage device.

1.1 iSCSI Considerations

For datacenters with centralized storage, iSCSI offers customers many benefits. It is comparatively inexpensive and it is based on familiar SCSI and TCP/IP standards. In comparison to FC and Fibre Channel over Ethernet (FCoE) SAN deployments, iSCSI requires less hardware, it uses lower-cost hardware, and more IT staff members might be familiar with the technology. These factors contribute to lower-cost implementations.

Although FC SANs represent the more established and mature technology in the storage world, iSCSI has become much more established in recent years.

One major difference between iSCSI and FC relates to I/O congestion. When an iSCSI path is overloaded, the TCP/IP protocol drops packets and requires them to be resent. FC communication over a dedicated path has a built-in pause mechanism when congestion occurs. When a network path carrying iSCSI storage traffic is oversubscribed, a bad situation quickly grows worse and performance further degrades as dropped packets must be resent. There can be multiple reasons for an iSCSI path being overloaded, ranging from oversubscription (too much traffic), to network switches that have a low port buffer.

Another consideration is the network bandwidth. Network bandwidth is dependent on the Ethernet standards used (1Gb or 10Gb). There are other mechanisms such as port aggregation and bonding links that deliver greater network bandwidth.

When implementing software iSCSI that uses network interface cards rather than dedicated iSCSI adapters, gigabit Ethernet interfaces are required. These interfaces tend to consume a significant amount of CPU Resource.

One way of overcoming this demand for CPU resources is to use a feature called a TOE (TCP/IP offload engine). TOEs shift TCP packet processing tasks from the server CPU to specialized TCP processors on the network adaptor or storage device. Most enterprise-level networking chip sets today offer TCP offload or checksum offload, which vastly improve CPU overhead.

iSCSI was considered a technology that did not work well over most shared wide-area networks. It has prevalently been approached as a local area network technology. However, this is changing. For synchronous replication writes (in the case of high availability) or remote data writes, iSCSI might not be a good fit. Latency introductions bring greater delays to data transfers and might impact application performance. Asynchronous replication, which is not dependent upon latency sensitivity, makes iSCSI an ideal solution. For example, VMware vCenter™ Site Recovery Manager™ may build upon iSCSI asynchronous storage replication for simple, reliable site disaster protection.

2. iSCSI Architecture

iSCSI initiators must manage multiple, parallel communication links to multiple targets. Similarly, iSCSI targets must manage multiple, parallel communications links to multiple initiators. Several identifiers exist in iSCSI to make this happen, including iSCSI Name, ISID (iSCSI session identifiers), TSID (target session identifier), CID (iSCSI connection identifier) and iSCSI portals. These will be examined in the next section.

2.1 iSCSI Names

iSCSI nodes have globally unique names that do not change when Ethernet adapters or IP addresses change. iSCSI supports two name formats as well as aliases. The first name format is the Extended Unique Identifier (EUI). An example of an EUI name might be *eui.02004567A425678D*.

The second name format is the iSCSI Qualified Name (IQN). An example of an IQN name might be *iqn.1998-01.com.vmware:tm-pod04-esx01-6129571c*.

2.2 iSCSI Initiators and Targets

A storage network consists of two types of equipment: initiators and targets. Initiators, such as hosts, are data consumers. Targets, such as disk arrays or tape libraries, are data providers. In the context of vSphere, iSCSI initiators fall into three distinct categories. They can be software, hardware dependent or hardware independent.

2.2.1 Software iSCSI Adapter

A software iSCSI adapter is VMware code built into the VMkernel. It enables your host to connect to the iSCSI storage device through standard network adaptors. The software iSCSI adapter handles iSCSI processing while communicating with the network adaptor. With the software iSCSI adapter, you can use iSCSI technology without purchasing specialized hardware.

2.2.2 Dependent Hardware iSCSI Adapter

This hardware iSCSI adapter depends on VMware networking and iSCSI configuration and management interfaces provided by VMware. This type of adapter can be a card that presents a standard network adaptor and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP and MAC addresses, as well as other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI licensed Broadcom 5709 NIC.

2.2.3 Independent Hardware iSCSI Adapter

This type of adapter implements its own networking and iSCSI configuration and management interfaces. An example of an independent hardware iSCSI adapter is a card that presents either iSCSI offload functionality only or iSCSI offload functionality and standard NIC functionality. The iSCSI offload functionality has independent configuration management that assigns the IP address, MAC address, and other parameters used for the iSCSI sessions. An example of an independent hardware adapter is the QLogic QLA4052 adapter.

This section examines the features and issues connected with each of these technologies.

2.3 iSCSI Sessions and Connections

iSCSI initiators and targets use TCP to create relationships called sessions. These sessions are identified by iSCSI session IDs (ISIDs). Session IDs are not tied to the hardware and can persist across hardware swaps. The initiator sees one logical connection to the target, as shown in Figure 1.

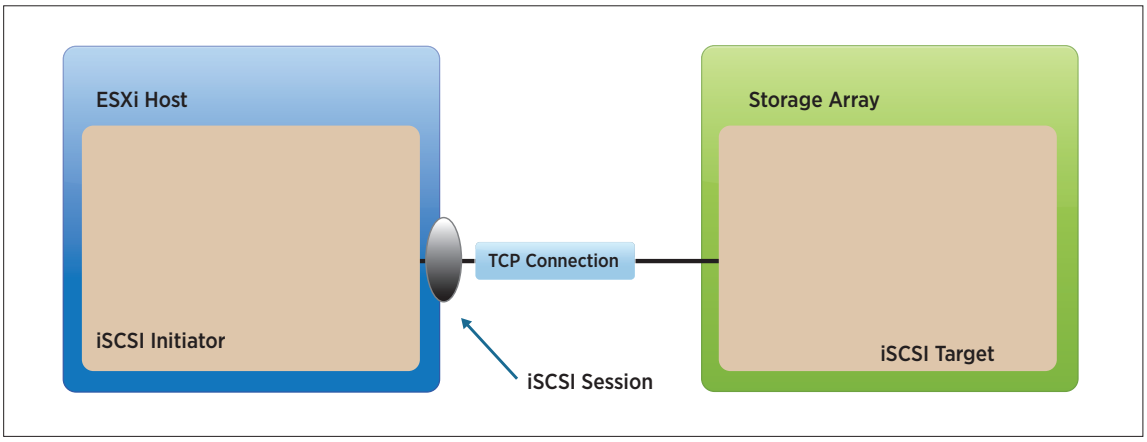

Figure 1. iSCSI Sessions

An iSCSI session might also contain multiple logical connections. From a vSphere host perspective, the sessions might also be thought of in term of paths between the initiator and target. Having multiple connections per session enables the aggregation of bandwidth and can also provide load balancing. An example of multiple logical connections to the target (identified by connection IDs, or CIDs) is shown in Figure 2.

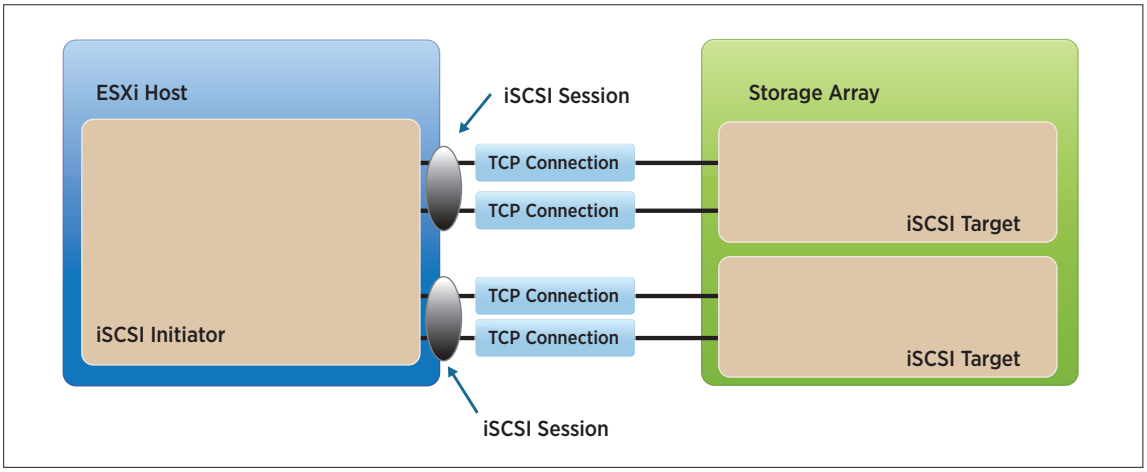

Figure 2. Multiple Connections per Sessions

However, a vSphere host does not support multiple connections per session at this time.

2.4 iSCSI Portals

iSCSI nodes keep track of connections via portals, enabling separation between names and IP addresses. A portal manages an IP address and a TCP port number. Therefore, from an architectural perspective, sessions can be made up of multiple logical connections, and portals track connections via TCP/IP port/address, as shown in Figure 3.

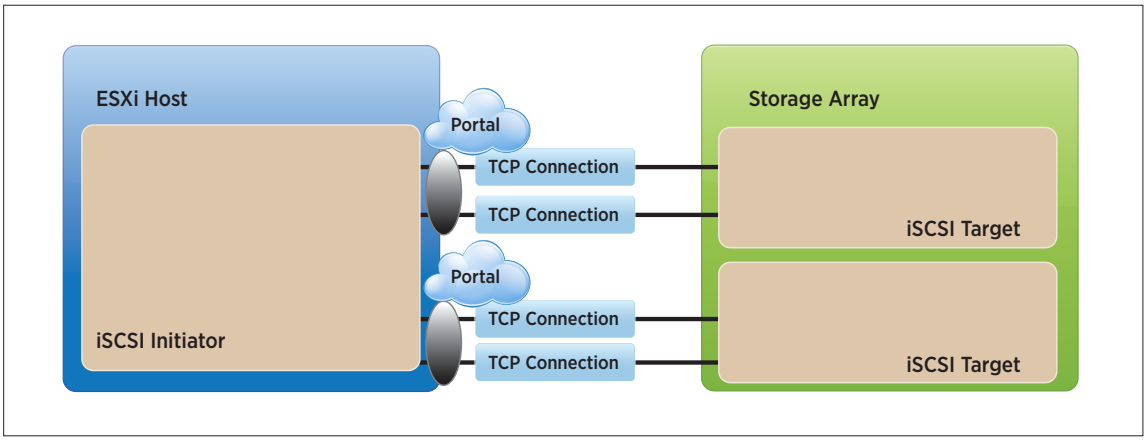

Figure 3. iSCSI Portals

In earlier versions of vSphere, the VMware iSCSI driver sent I/O over one portal only (a single session per connection) and only when that failed did the vSphere host try to use other portals in a Round Robin fashion.

In more recent versions, this behavior changed so that the driver now logs in to all the portals that are returned in the **SendTarget** discovery response. The reason for this enhancement was to enable support for new active/passive iSCSI arrays that required support. With active/passive arrays, the vSphere host storage stack was required to recognize each of the portals as different paths (targets) to effectively do multipath failovers.

NOTE: Not all iSCSI arrays behave like this. Some arrays still require an administrator to add additional paths manually.

3. iSCSI Implementation Options

VMware supports iSCSI with both software initiator and hardware initiator implementations. The software initiator iSCSI plugs into the vSphere host storage stack as a device driver in just the same way as other SCSI and FC drivers. This means that it implicitly supports the flagship file system of VMware, VMware vSphere VMFS, and also Raw Device Mappings (RDMs).

As previously mentioned, hardware iSCSI adapters fall into two categories – hardware dependent and hardware independent. Booting from iSCSI is also supported for both software and hardware iSCSI. Figure 4 shows the basic differences between an iSCSI hardware and iSCSI software implementation.

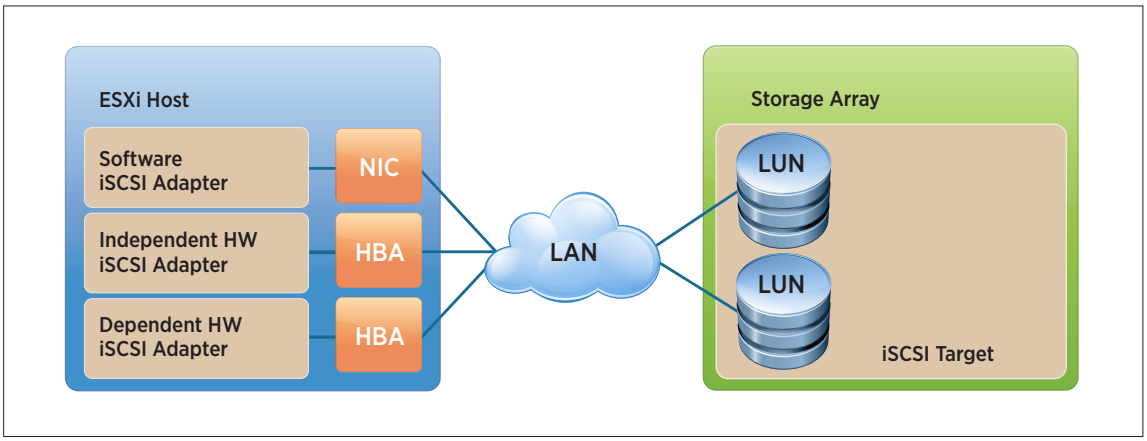

Figure 4. Software and Hardware iSCSI Initiators

With the hardware-initiator iSCSI implementation, the iSCSI HBA provides the translation from SCSI commands to an encapsulated format that can be sent over the network. A TCP offload engine (TOE) does this translation on the adapter.

The software-initiator iSCSI implementation leverages the VMkernel to perform the SCSI to IP translation and requires extra CPU cycles to perform this work. As mentioned previously, most enterprise-level networking chip sets offer TCP offload or checksum offloads, which vastly improve CPU overhead.

3.1 Mixing iSCSI Options

Having both software iSCSI and hardware iSCSI enabled on the same host is supported. However, use of both software and hardware adapters on the same vSphere host to access the same target is not supported. One cannot have the host access the same target via hardware-dependent/hardware-independent/software iSCSI adapters for multipathing purposes. The reason for this support statement is that the different adapter types relate primarily to performance and management. For example, each adapter can generate different speeds.

Also, vSphere manages the software iSCSI adapters, but the hardware adapters have different management interfaces.

Finally, there can be differences in the offloading mechanism whereby the hardware adapters can offload by default, but for software iSCSI it will depend on the NIC. You might or might not have offload capabilities.

It's similar in many ways to presenting the same LUN from the same array via iSCSI and FC. You can see it over multiple paths and you can send I/O to it over multiple paths, but it would not be supported due to the differences highlighted previously.

However, different hosts might access the same iSCSI LUN via different methods. For example, host 1 might access the LUN using the software iSCSI adapter of VMware, host 2 might access it via a hardware-dependent iSCSI adapter and host 3 might access it via a hardware-independent iSCSI adapter.

4. Networking Settings

Network design is key to making sure iSCSI works. In a production environment, gigabit Ethernet is essential for software iSCSI. Hardware iSCSI, in a VMware Infrastructure environment, is implemented with dedicated HBAs.

iSCSI should be considered a local-area technology, not a wide-area technology, because of latency issues and security concerns. You should also segregate iSCSI traffic from general traffic. Layer-2 VLANs are a particularly good way to implement this segregation.

Beware of oversubscription. Oversubscription occurs when more users are connected to a system than can be fully supported at the same time. Networks and servers are almost always designed with some amount of oversubscription, assuming that users do not all need the service simultaneously. If they do, delays are certain and outages are possible. Oversubscription is permissible on general-purpose LANs, but you should not use an oversubscribed configuration for iSCSI.

Best practice is to have a dedicated LAN for iSCSI traffic and not share the network with other network traffic. It is also best practice not to oversubscribe the dedicated LAN.

Finally, because iSCSI leverages the IP network, VMkernel NICs can be placed into teaming configurations. Alternatively, a VMware recommendation is to use port binding rather than NIC teaming. Port binding will be explained in detail later in this paper but suffice to say that with port binding, iSCSI can leverage VMkernel multipath capabilities such as failover on SCSI errors and Round Robin path policy for performance.

In the interest of completeness, both methods will be discussed. However, port binding is the recommended best practice.

4.1 VMkernel Network Configuration

A VMkernel network is required for IP storage and thus is required for iSCSI. A best practice would be to keep the iSCSI traffic separate from other networks, including the management and virtual machine networks.

4.2 IPv6 Supportability Statements

At the time of this writing, there is no IPv6 support for either hardware iSCSI or software iSCSI adapters in vSphere 5.1. Whereas some experimental support has existed in previous releases, and although all the plumbing exists to run software iSCSI over IPv6, this has not yet been adequately tested by VMware. Users who choose to run software iSCSI over IPv6 are not supported by VMware and do so at their own risk.

4.3 Throughput Options

There are a number of options available to improve iSCSI performance.

1. 10GbE – This is an obvious option to begin with. If you can provide a larger pipe, the likelihood is that you will achieve greater throughput. Of course, if there is not enough I/O to fill a 1GbE connection, then a larger connection isn't going to help you. But let's assume that there are enough virtual machines and enough datastores for 10GbE to be beneficial.

2. Jumbo frames – This feature can deliver additional throughput by increasing the size of the payload in each frame from a default MTU of 1,500 to an MTU of 9,000. However, great care and consideration must be used if you decide to implement it. All devices sitting in the I/O path (iSCSI target, physical switches, network interface cards and VMkernel ports) must be able to implement jumbo frames for this option to provide the full benefits. For example, if the MTU is not correctly set on the switches, the datastores might mount but I/O will fail. A common issue with jumbo-frame configurations is that the MTU value on the switch isn't set correctly. In most cases, this must be higher than that of the hosts and storage, which are typically set to 9,000. Switches must be set higher, to 9,198 or 9,216 for example, to account for IP overhead. Refer to switch-vendor documentation as well as storage-vendor documentation before attempting to configure jumbo frames.

3. Round Robin path policy – Round Robin uses an automatic path selection rotating through all available paths, enabling the distribution of load across the configured paths. This path policy can help improve I/O throughput. For active/passive storage arrays, only the paths to the active controller will be used in the Round Robin policy. For active/active storage arrays, all paths will be used in the Round Robin policy. For ALUA arrays (Asymmetric Logical Unit Assignment), Round Robin uses only the active/optimized (AO) paths. These are the paths to the disk through the managing controller. Active/nonoptimized (ANO) paths to the disk through the nonmanaging controller are not used.

Not all arrays support the Round Robin path policy. Refer to your storage-array vendor's documentation for recommendations on using this Path Selection Policy (PSP).

4.4 Minimizing Latency

Because iSCSI on VMware uses TCP/IP to transfer I/O, latency can be a concern. To decrease latency, one should always try to minimize the number of hops between the storage and the vSphere host. Ideally, one would not route traffic between the vSphere host and the storage array, and both would coexist on the same subnet.

NOTE: If iSCSI port bindings are implemented for the purposes of multipathing, you cannot route your iSCSI traffic.

4.5 Routing

A vSphere host has a single routing table for all of its VMkernel Ethernet interfaces. This imposes some limits on network communication. Consider a configuration that uses two Ethernet adapters with one VMkernel TCP/IP stack. One adapter is on the 10.17.1.1/24 IP network and the other on the 192.168.1.1/24 network. Assume that 10.17.1.253 is the address of the default gateway. The VMkernel can communicate with any servers reachable by routers that use the 10.17.1.253 gateway. It might not be able to talk to all servers on the 192.168 network unless both networks are on the same broadcast domain.

4.5.1 The VMkernel TCP/IP Routing Table

Another consequence of the single routing table affects one approach you might otherwise consider for balancing I/O. Consider a configuration in which you want to connect to iSCSI storage and also want to enable NFS mounts. It might seem that you can use one Ethernet adapter for iSCSI and a separate Ethernet adapter for NFS traffic to spread the I/O load. This approach does not work because of the way the VMkernel TCP/IP stack handles entries in the routing table.

For example, you might assign an IP address of 10.16.156.66 to the VMkernel adapter you want to use for NFS. The routing table then contains an entry for the 10.16.156.x network for this adapter. If you then set up a second adapter for iSCSI and assign it an IP address of 10.16.156.25, the routing table contains a new entry for the 10.16.156.x network for the second adapter. However, when the TCP/IP stack reads the routing table, it never reaches the second entry, because the first entry satisfies all routes to both adapters. Therefore, no traffic ever goes out on the iSCSI network, and all IP storage traffic goes out on the NFS network.

The fact that all 10.16.156.x traffic is routed on the NFS network causes two types of problems. First, you do not see any traffic on the second Ethernet adapter. Second, if you try to add trusted IP addresses both to iSCSI arrays and NFS servers, traffic to one or the other comes from the wrong IP address.

4.5.2 Using Static Routes

As mentioned before, for vSphere hosts, the management network is on a VMkernel port and therefore uses the default VMkernel gateway. Only one VMkernel default gateway can be configured on a vSphere host. You can, however, add static routes to additional gateways/routers from the command line. Instruction on how to do this is documented in VMware Knowledge Base (KB) article 2001426.

4.6 Availability Options – Multipathing or NIC Teaming

To achieve high availability, the local-area network (LAN) on which the iSCSI traffic runs must be designed with availability, downtime avoidance, isolation and no single point of failure (SPOF) in mind. Multiple administrators must be involved in designing for high availability. These are the virtualization administrator and the network administrator (and maybe the storage administrator). This section outlines these steps and investigates several options, which can be utilized to make your iSCSI datastores highly available.

In both cases that follow, at least two network interface cards are required. Whereas 1Gb interfaces will meet the requirements for a highly available network, 10Gb network adaptors will also improve performance.

4.6.1 NIC Teaming for Availability

A best practice for iSCSI is to avoid the vSphere feature called teaming (on the network interface cards) and instead use port binding. Port binding introduces multipathing for availability of access to the iSCSI targets and LUNs. If for some reason this is not suitable (for instance, you wish to route traffic between the iSCSI initiator and target), then teaming might be an alternative.

If you plan to use teaming to increase the availability of your network access to the iSCSI storage array, you must turn off port security on the switch for the two ports on which the virtual IP address is shared. The purpose of this port security setting is to prevent spoofing of IP addresses. Thus many network administrators enable this setting. However, if you do not change it, the port security setting prevents failover of the virtual IP from one switch port to another and teaming cannot fail over from one path to another. For most LAN switches, the port security is enabled on a port level and thus can be set on or off for each port.

4.6.2 iSCSI Multipathing via Port Binding for Availability

Another way to achieve availability is to create a multipath configuration. This is a more preferred method over NIC teaming, because this method will fail over I/O to alternate paths based on SCSI sense codes and not just network failures. Also, port bindings give administrators the opportunity to load-balance I/O over multiple paths to the storage device. Additional advantages around port binding will be discussed later in this paper.

4.7 Error Correction Digests

iSCSI header and data digests check the end-to-end, noncryptographic data integrity beyond the integrity checks that other networking layers provide, such as TCP and Ethernet. They check the entire communication path, including all elements that can change the network-level traffic, such as routers, switches and proxies.

Enabling header and data digests does require additional processing for both the initiator and the target and can affect throughput and CPU use performance.

Some systems can offload the iSCSI digest calculations to the network processor, thus reducing the impact on performance.

4.8 Flow Control

The general consensus from our storage partners is that hardware-based flow control is recommended for all network interfaces and switches.

5. Security Considerations

The following items comprise a list of considerations from a security perspective when implementing iSCSI.

5.1 Private Network

iSCSI storage traffic is transmitted in an unencrypted format across the LAN. Therefore, it is considered best practice to use iSCSI on trusted networks only and to isolate the traffic on separate physical switches or to leverage a private VLAN. All iSCSI-array vendors agree that it is good practice to isolate iSCSI traffic for security reasons. This would mean isolating the iSCSI traffic on its own separate physical switches or leveraging a dedicated VLAN (IEEE 802.1Q).

5.2 Encryption

ISCSI supports several types of security. IPSec (Internet Protocol Security) is a developing standard for security at the network or packet-processing layer of network communication. IKE (Internet Key Exchange) is an IPSec standard protocol used to ensure security for VPNs. However, at the time of this writing IPSec was not supported on vSphere hosts.

5.3 Authentication

There are also a number of authentication methods supported with iSCSI.

- Kerberos
- SRP (Secure Remote Password)
- SPKM1/2 (Simple Public-Key Mechanism)
- CHAP (Challenge Handshake Authentication Protocol)

At the time of this writing (vSphere 5.1), a vSphere host does not support Kerberos, SRP or public-key authentication methods for iSCSI. The only authentication protocol supported is CHAP. CHAP verifies identity using a hashed transmission. The target initiates the challenge. Both parties know the secret key. It periodically repeats the challenge to guard against replay attacks. CHAP is a one-way protocol, but it might be implemented in two directions to provide security for both ends. The iSCSI specification defines the CHAP security method as the only must-support protocol. The VMware implementation uses this security option. Initially, VMware supported only unidirectional CHAP, but bidirectional CHAP is now supported.

6. iSCSI Datastore Provisioning Steps

Before a vSphere host can utilize iSCSI storage, the following configuration steps must be taken:

1. Create a new VMkernel port group for IP storage on an already existing virtual switch (vSwitch) or on a new vSwitch when it is configured. The vSwitch can be a vSphere Standard Switch (VSS) or a VMware vSphere Distributed Switch™.
2. Ensure that the iSCSI initiator on the vSphere host(s) is enabled.
3. Ensure that the iSCSI storage is configured to export a LUN accessible to the vSphere host iSCSI initiators on a trusted network.

For network connectivity, the user must create a new VMkernel port group to configure the vSwitch for IP storage access. The user must also populate the network access information, including any VLAN tag associated with the storage network.

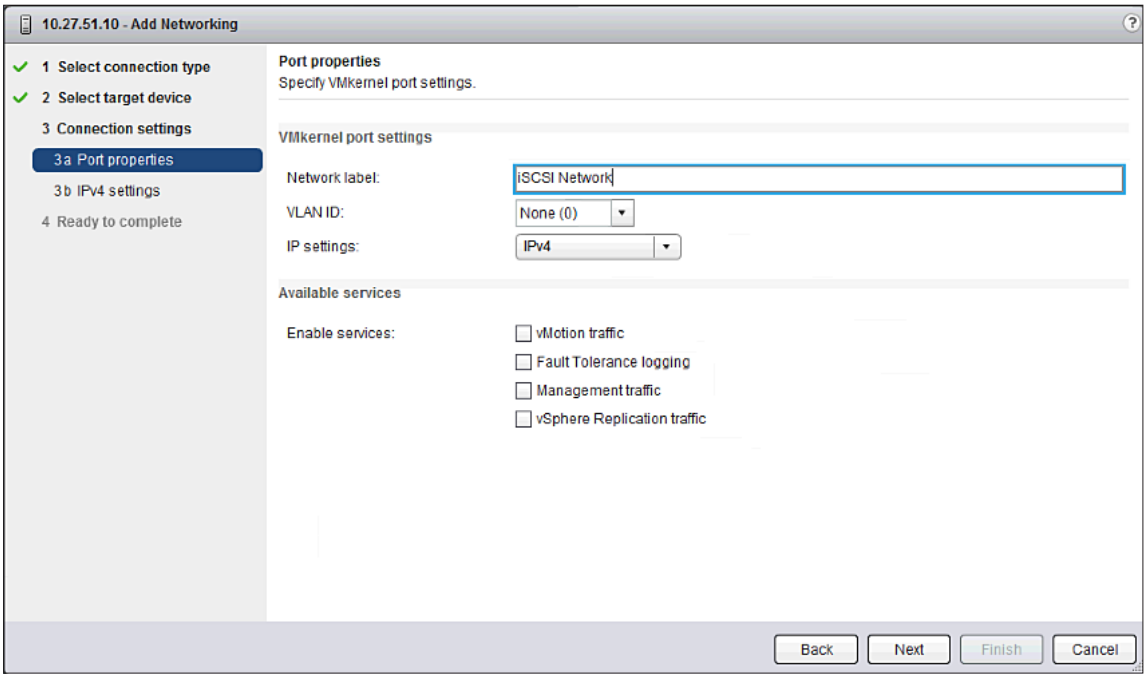

Figure 5. VMkernel Connection Settings

To enable the iSCSI initiator, additional details such as the IP address of the target array and any CHAP-related credentials must also be added.

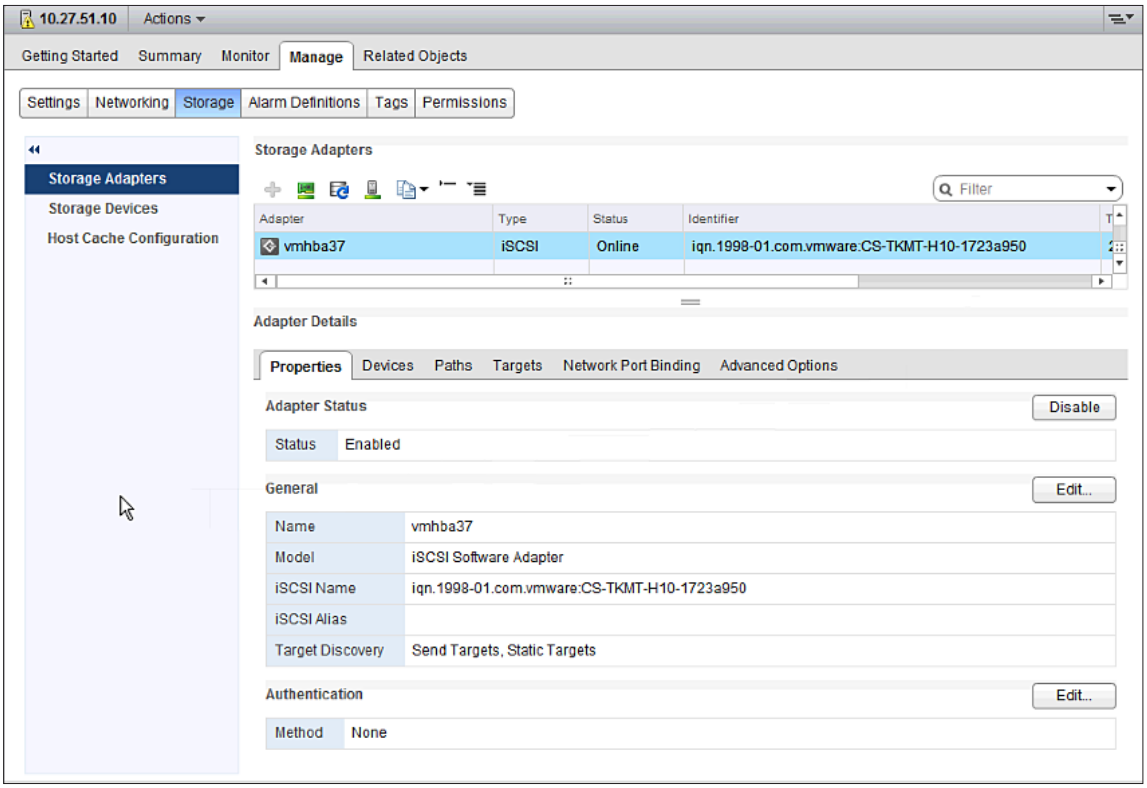

Figure 6. Enable Software iSCSI Adapter

At this point, port binding details might also be added for multipathing purposes.

6.1 Software iSCSI Port Binding

In vSphere 5.0, VMware introduced a new user interface (UI) to make it much easier to configure multipathing for software iSCSI. The UI is a major change from vSphere 4.x, which required administrators to use the command line to get an optimal multipath configuration with software iSCSI.

The UI enables one to select different network interfaces for iSCSI use, check them for compliance and configure them with the software iSCSI adapter. This multipathing configuration is also referred to as iSCSI port binding or network port binding.

6.1.1 Why Use iSCSI Multipathing?

The primary use case of this feature is to create a multipath configuration with storage that presents only a single storage portal, such as the DELL EqualLogic and the HP LeftHand. Without iSCSI multipathing, this type of storage would have one path only between the VMware ESX® host and each volume. iSCSI multipathing enables us to multipath to this type of clustered storage.

Another benefit is the ability to use alternate VMkernel networks outside of the vSphere host management network. This means that if the management network suffers an outage, you continue to have iSCSI connectivity via the VMkernel ports participating in the iSCSI bindings.

NOTE: VMware considers the implementation of iSCSI multipathing over NIC teaming a best practice.

6.2 Software iSCSI Multipathing Configuration Steps

In this example, I have configured a software iSCSI adapter, vmhba32.

At this point, no targets have been added, so no devices or paths have been discovered. Before implementing the software iSCSI bindings, I must create a number of additional VMkernel ports (vmk) for port binding to the software iSCSI adapter.

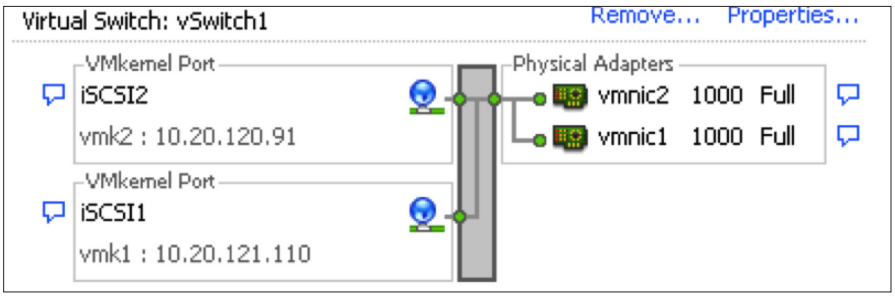

Figure 7. Single VSS with Two VMkernel Ports

For port binding to work correctly, the initiator must be able to reach the target directly on the same subnet – **iSCSI port binding in vSphere 5.0 does not support routing**. In this configuration, if I place my VMkernel ports on VLAN 74, they can reach the iSCSI target without the need of a router. This is an important point and requires further elaboration because it causes some confusion. If I do not implement port binding and use a standard VMkernel port, then my initiator can reach the targets through a routed network. This is supported and works well. It is only when iSCSI binding is implemented that a direct, nonrouted network between the initiators and targets is required. In other words, initiators and targets must be on the same subnet.

There is another important point to note when it comes to the configuration of iSCSI port bindings. On VMware standard switches that contain multiple vmnic uplinks, each VMkernel (vmk) port used for iSCSI bindings must be associated with a single vmnic uplink. The other uplink(s) on the vSwitch must be placed into an **unused** state. This is only a requirement when there are multiple vmnic uplinks on the same vSwitch. If you are using multiple VSSs with their own vmnic uplinks, then this is not an issue.

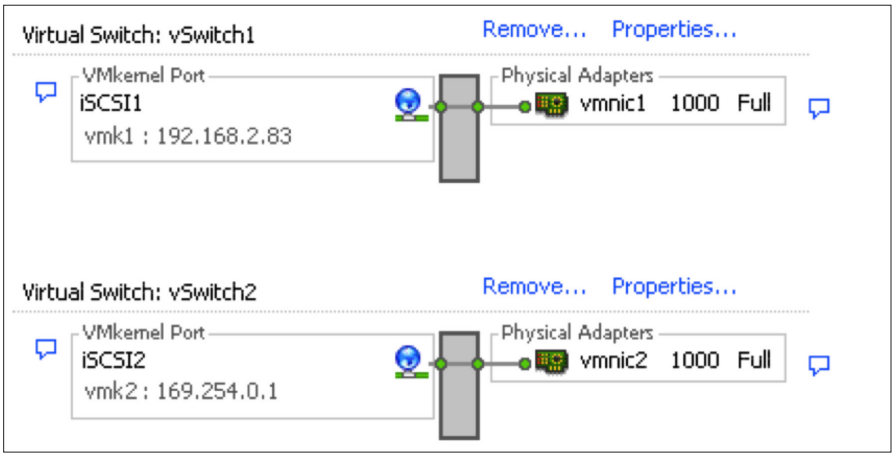

Figure 8. Multiple VSS with One VMkernel Port

Continuing with the network configuration, a second VMkernel (vmk) port is created. Now there are two vmk ports, labeled iSCSI1 and iSCSI2. These will be used for the iSCSI port binding/multipathing configuration. The next step is to configure the bindings and iSCSI targets. This is done in the properties of the software iSCSI adapter. Since vSphere 5.0, there is a new **Network Configuration** tab in the Software iSCSI Initiator Properties window. This is where the VMkernel ports used for binding to the iSCSI adapter are added.

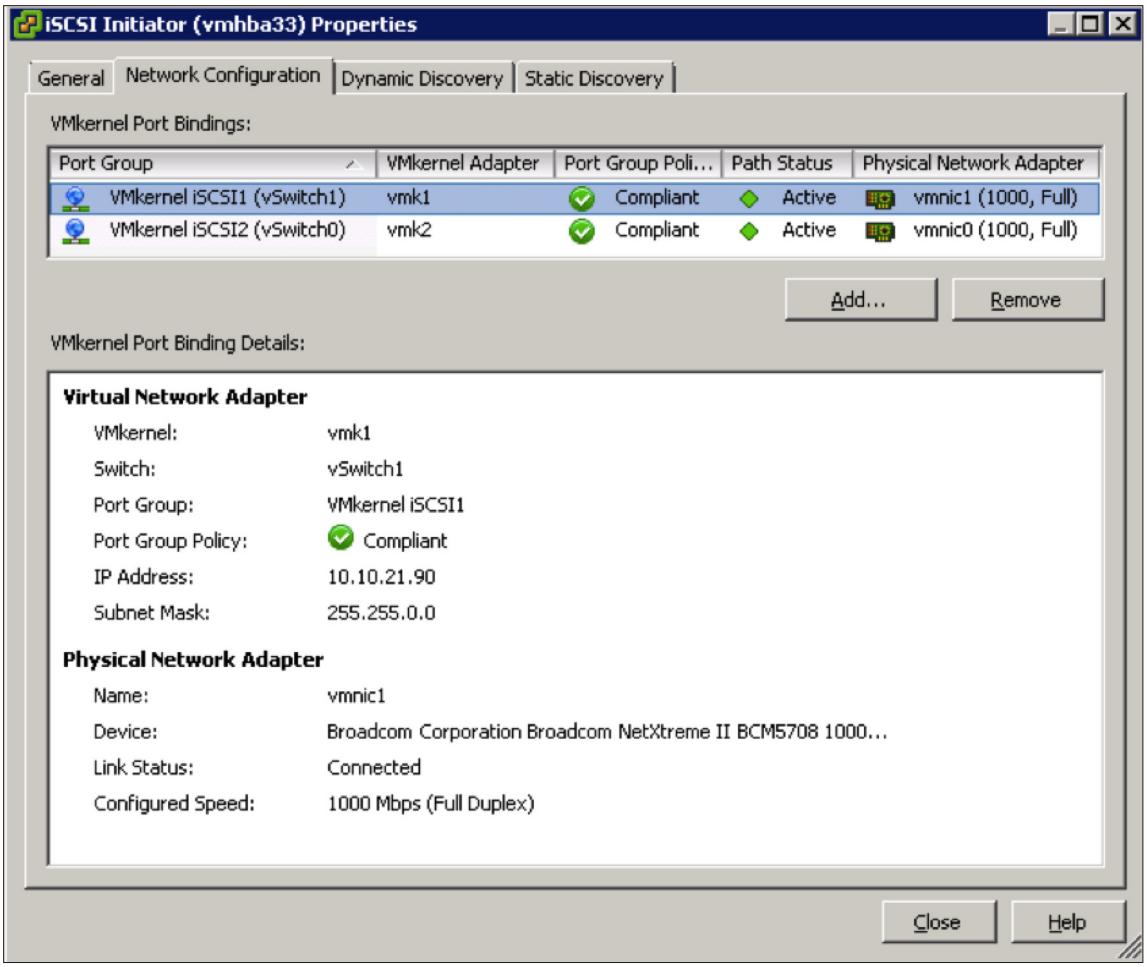

Figure 9. Network Configuration Tab

After selecting the VMkernel adapters for use with the software iSCSI adapter, the **Port Group Policy** tab will tell you whether or not these adapters are compliant for binding. If you have more than one **active** uplink on a vSwitch that has multiple vmnic uplinks, the vmk interfaces will not show up as compliant. Only one uplink should be **active**. All other uplinks should be placed into an **unused** state.

The next step is to proceed to the **Dynamic Discovery** tab, where the iSCSI targets can now be added. Because port binding is being used, iSCSI targets must be reachable by the software iSCSI adapter through a nonroutable network. In other words, the storage controller ports are on the same subnet as the VMkernel NICs.

At this point, there are two VMkernel ports bound to the software iSCSI adapter and connected to an iSCSI target or targets (which are in effect array controllers). These targets are all going to the same storage array, so a LUN presented out on all targets will be visible across multiple paths.

This new UI for iSCSI bindings makes multipathing for the software iSCSI adapter much easier. But do keep in mind the requirements for a nonroutable network between the initiator and target and the fact that VMkernel ports must have only a single active vmnic uplink in VMware standard switches having multiple vmnic uplinks.

Consult the detailed white paper entitled *Multipathing Configuration for Software iSCSI Using Port Binding* on the VMware Technical Resource Center for further information.

7. Interoperability Considerations

This section discusses features that are in some way related to storage, and iSCSI storage in particular. Most of the interoperability features are tried and tested with iSCSI, but areas that might be cause for additional consideration are highlighted when there is contention for I/O.

7.1 Storage I/O Control

Storage I/O Control (SIOC) prevents a single virtual machine residing on one vSphere host from consuming more than its share of bandwidth on a datastore that it shares with other virtual machines residing on other vSphere hosts.

Historically, the disk shares feature can be set up on a per–vSphere host basis. This works well for all virtual machines residing on the same vSphere host sharing the same datastore built on a local disk. However, this cannot be used as a fairness mechanism for virtual machines from different vSphere hosts sharing the same datastore. This is what SIOC does. SIOC modifies the I/O queues on various vSphere hosts to ensure that virtual machines with a higher priority get more queue entries than those virtual machines with a lower priority, enabling these higher-priority virtual machines to send more I/O than their lower-priority counterparts.

SIOC is a congestion-driven feature. When latency remains below a specific latency value, SIOC is dormant. It is triggered only when the latency value on the datastore rises above a predefined threshold.

SIOC is recommended if you have a group of virtual machines sharing the same datastore spread across multiple vSphere hosts and you want to prevent the impact of a single virtual machine's I/O on the I/O (and thus performance) of other virtual machines. With SIOC you can set shares to reflect the priority of virtual machines, but you can also implement an IOPS limit per virtual machine. This means that you can limit the impact, in number of IOPS, which a single virtual machine can have on a shared datastore.

SIOC is available in the VMware vSphere Enterprise Plus Edition™. More details on SIOC can be found in the *Storage IO Control Technical Overview and Considerations for Deployment* white paper.

7.2 Network I/O Control

The Network I/O Control (NIOC) feature ensures that when the same network interface cards are used for multiple traffic types, other traffic types on the same network interface cards do not impact iSCSI traffic. It works by setting priority and bandwidth using priority tags in TCP/IP packets. With 10GbE networks, this feature can be very useful, because there is one pipe that is shared with multiple other traffic types. With 1GbE networks, you have probably dedicated the pipe solely to iSCSI traffic. This means that NIOC is congestion driven. NIOC takes effect only when there are different traffic types competing for bandwidth and the performance of one traffic type is likely to be impacted.

Whereas SIOC assists in dealing with the noisy-neighbor problem from a datastore-sharing perspective, NIOC assists in dealing with the noisy-neighbor problem from a network perspective.

Using NIOC, one can also set the priority levels of different virtual machine traffic. If certain virtual machine traffic is important to you, these virtual machines can be grouped into one virtual machine port group and lower-priority virtual machines can be placed into another virtual machine port group. NIOC can now be used to prioritize virtual machine traffic and ensure that the high-priority virtual machines get more bandwidth when there is competition for bandwidth on the pipe.

SIOC and NIOC can coexist and in fact complement each other.

NIOC is available in the vSphere Enterprise Plus Edition. More details on NIOC can be found in the *Network I/O Control Best Practices* white paper.

7.3 vSphere Storage DRS

VMware vSphere Storage DRS™, introduced with vSphere 5.0, fully supports VMFS datastores on iSCSI. When you enable vSphere Storage DRS on a datastore cluster (group of datastores), it automatically configures balancing based on space usage. The threshold is set to 80 percent but can be modified. This means that if 80 percent or more of the space on a particular datastore is utilized, vSphere Storage DRS will try to move virtual machines to other datastores in the datastore cluster using VMware vSphere Storage vMotion® to bring this usage value back down below 80 percent. The usage statistics of the datastores are checked on an ongoing basis.

If the cluster is set to the **automatic** mode of operation, vSphere Storage DRS uses vSphere Storage vMotion to automatically migrate virtual machines to other datastores in the datastore cluster if the threshold is exceeded. If the cluster is set to **manual**, the administrator is given a set of recommendations to apply. vSphere Storage DRS will provide the best recommendations to balance the space usage of the datastores. After you apply the recommendations, vSphere Storage vMotion, as seen before, moves one or more virtual machines between datastores in the same datastore cluster.

Another feature of vSphere Storage DRS is that it can balance virtual machines across datastores in the datastore cluster based on I/O metrics, and specifically latency.

vSphere Storage DRS uses SIOC to evaluate datastore capabilities and capture latency information regarding all the datastores in the datastore cluster. As mentioned earlier, the purpose of SIOC is to ensure that no single virtual machine uses all the bandwidth of a particular datastore. It achieves this by modifying the queue depth for the datastores on each vSphere host.

In vSphere Storage DRS, its implementation is different. SIOC, on behalf of vSphere Storage DRS, checks the capabilities of the datastores in a datastore cluster by injecting various I/O loads. After this information is normalized, vSphere Storage DRS can determine the types of workloads that a datastore can handle. This information is used in initial placement and load-balancing decisions.

vSphere Storage DRS continuously uses SIOC to monitor how long it takes an I/O to do a round trip. This is the latency. This information about the datastore is passed back to vSphere Storage DRS. If the latency value for a particular datastore is above the threshold value (the default is 15 milliseconds) for a significant percentage of time over an observation period (the default is 16 hours), vSphere Storage DRS tries to rebalance the virtual machines across the datastores in the datastore cluster so that the latency value returns below the threshold. This might involve one or more vSphere Storage vMotion operations. In fact, even if vSphere Storage DRS is unable to bring the latency below the defined threshold value, it might still move virtual machines between datastores to balance the latency.

When evaluating vSphere Storage DRS, VMware makes the same best practice recommendation made for vSphere Storage DRS initially. The recommendation is to run vSphere Storage DRS in manual mode first and then monitor the recommendations that vSphere Storage DRS is surfacing, ensuring that they make sense. After a period of time, if the recommendations make sense and you build a comfort level using vSphere Storage DRS, consider switching it to automated mode.

There are a number of considerations when using vSphere Storage DRS with certain array features. Check your storage vendor's recommendation for using vSphere Storage DRS. There might be specific interaction with some advanced features on the array that you want to be aware of. VMware has already produced a very detailed white paper regarding the use of vSphere Storage DRS with array features such as tiered storage, thin provisioning and deduplication. More details regarding vSphere Storage DRS interoperability with storage-array features can be found in the *VMware vSphere Storage DRS Interoperability* white paper.

7.4 vSphere Storage APIs – Array Integration

Many storage arrays now support a number of VMware vSphere Storage APIs – Array Integration primitives. This API enables the vSphere host to offload certain storage operations to the storage array rather than consuming resources on the vSphere host by doing the same operations.

For block storage arrays, no additional VIBs need to be installed on the vSphere host. All software necessary to use vSphere Storage APIs – Array Integration is preinstalled on the hosts.

The first primitive to discuss is Extended Copy (XCOPY), which enables the vSphere host to offload a clone operation or template deployments to the storage array.

NOTE: This primitive also supports vSphere Storage vMotion.

The next primitive is called Write Same. When creating VMDKs on block datastores, one of the options is to create an Eager Zeroed Thick (EZT) VMDK, which means zeroes get written to all blocks that make up that VMDK. With the Write Same primitive, the act of writing zeroes is offloaded to the array. This means that we don't have to send lots of zeroes across the wire, which speeds up the process. In fact, for some arrays this is simply a metadata update, which means a very fast zeroing operation.

Our final primitive is Atomic Test & Set (ATS). ATS is a block primitive that replaces SCSI reservations when metadata updates are done on VMFS volumes.

Thin provisioning (TP) primitives were introduced with such vSphere 5.0. features as the raising of an alarm when a TP volume reached 75 percent of capacity at the back end, TP-Stun and, of course, the UNMAP primitive.

vSphere Storage DRS leverages 75 percent of capacity event. After the alarm is triggered, vSphere Storage DRS no longer considers those datastore as destinations for initial placement or ongoing load balancing of virtual machines.

The vSphere Storage APIs – Array Integration primitive TP-Stun was introduced to detect out-of-space conditions on SCSI LUNs. If a datastore reaches full capacity and has no additional free space, any virtual machines that require additional space will be stunned. Virtual machines that do not require additional space continue to work normally. After the additional space has been added to the datastore, the suspended virtual machines can be resumed.

Finally, the UNMAP primitive is used as a way to reclaim dead space on a VMFS datastore built on thin-provisioned LUNs. A detailed explanation of vSphere Storage APIs – Array Integration can be found in the white paper, *VMware vSphere Storage APIs – Array Integration (VAAI)*.

NOTE: At the time of this writing, there was no support from vSphere Storage APIs – Array Integration for storage appliances. Support from vSphere Storage APIs – Array Integration is available only on physical storage arrays.

7.5 vCenter Site Recovery Manager/vSphere Replication

VMware vCenter Site Recovery Manager™ fully supports array-based replication on iSCSI datastores. VMware vSphere Replication fully supports the replication of virtual machines that reside on iSCSI datastores.

7.6 vSphere Storage vMotion

vSphere Storage vMotion has gone through quite a few architectural changes over the years. The latest version in vSphere 5.x uses a mirror driver to split writes to the source and destination datastores after a migration is initiated. This means speedier migrations because there is only a single copy operation now required, unlike the recursive copy process used in previous versions that leveraged Change Block Tracking (CBT).

One consideration that has been called out already is that vSphere Storage vMotion operations of virtual machines between datastores cannot be offloaded to the array without support from vSphere Storage APIs – Array Integration. In those cases, the software data mover does all vSphere Storage vMotion operations.

The only other considerations with vSphere Storage vMotion are relevant to both block operations and NAS. This is the configuration maximum. At the time of this writing, the maximum number of concurrent vSphere Storage vMotion operations per vSphere host is two and the maximum number of vSphere Storage vMotion operations per datastore is eight. This is to prevent any single datastore from being unnecessarily impacted by vSphere Storage vMotion operations.

NOTE: A new enhancement in vSphere 5.1 enables up to four VMDKs belonging to the same virtual machine to be migrated in parallel, as long as the VMDKs reside on unique datastores.

Check your storage vendor's recommendation for using vSphere Storage vMotion. There might be specific interaction with some advanced features on the array that you must be aware of when moving a virtual machine from one datastore to another datastore.

8. Sizing Considerations

8.1 Recommended Volume Size

When creating this paper, we asked a number of our storage partners if there was a volume size that worked well for iSCSI. All partners said that there was no performance gain or degradation depending on the volume size and those customers might build iSCSI volumes of any size, so long as it was below the array vendor's supported maximum. The datastore sizes vary greatly from customer to customer.

DELL recommends starting with a datastore that is between 500GB and 750GB for their Compellent range of arrays. However, because VMFS datastore can be easily extended on the fly, their general recommendation is to start with smaller and more manageable datastore sizes initially and expand them as needed. This seems like good advice.

Sizing of volumes is typically proportional to the number of virtual machines you attempt to deploy, in addition to snapshots/changed blocks created for backup purposes. Another consideration is that many arrays now have deduplication and compression features, which will also reduce capacity requirements. A final consideration is Recovery Point Objective (RPO) and Recovery Time Objective (RTO). These determine how fast you can restore your datastore with your current backup platform.

8.2 Recommended Block Size

This parameter is not tunable, for the most part. Some vendors have it hard set to 4KB and others have it hard set to 8KB. Block sizes are typically a multiple of 4KB. These align nicely with the 4KB grain size used in the VMDK format of VMware. For those vendors who have it set to 8KB, the recommendation is to format the volumes in the guest operating system (OS) to a matching 8KB block size for optimal performance. In this area, it is best to speak to your storage-array vendor to get vendor-specific advice.

The questions one should ask are as follows:

1. What is the volume block size on the array?
2. Is it tunable?
3. If so, what should I tune it to? Be sure to explain that the datastore will be used by virtual machines, so the workload will be random for the most part.
4. Are there any considerations when formatting the volumes in the guest OS?

8.3 Maximum Number of Virtual Machines per Datastore

The number of virtual machines that can run on a single datastore is directly proportional to the infrastructure and the workloads running in the virtual machines. For example, one might be able to run many hundreds of low-I/O virtual machines but only a few very intensive I/O virtual machines on the same datastore. Network congestion is an important factor. Users might consider using the Round Robin path policy on all storage devices to achieve optimal performance and load balancing. In fact, since vSphere 5.1 EMC now has the Round Robin path policy associated with its SATP (Storage Array Type Plug-in) in the VMkernel, so that when an EMC storage device is discovered, it will automatically use Round Robin.

The other major factor is related to the backup and recovery Service-Level Agreement (SLA). If you have one datastore with many virtual machines, there is a question of how long you are willing to wait while service is restored in the event of a failure. This is becoming the major topic in the debate over how many virtual machines per datastore is optimal.

The snapshot technology used by the backup product is an important question—specifically, whether it uses array-based snapshots or virtual machine snapshots. Performance is an important consideration if virtual machine snapshots are used to concurrently capture point-in-time copies of virtual machines. In many cases, array-based snapshots have less impact on the datastores and are more scalable when it comes to backups. There might be some array-based limitations to look at also. For instance, the number of snapshot copies of a virtual machine that a customer wants to maintain might exceed the number of snapshot copies an array can support. This varies from vendor to vendor. Check this configuration maximum with your storage-array vendor.

KB article 1015180 includes further details regarding snapshots and their usage. As shown in KB article 1025279, virtual machines can support up to 32 snapshots in a chain, but VMware recommends that you use only two or three snapshots in a chain and also that you use no single snapshot for more than 24–72 hours.

9. Booting a vSphere Host from Software iSCSI

VMware introduced support for iSCSI with ESX 3.x. However, ESX could boot only from an iSCSI LUN if a hardware iSCSI adapter was used. Hosts could not boot via the software iSCSI initiator of VMware. In vSphere 4.1, VMware introduced support making it possible to boot the host from an iSCSI LUN via the software iSCSI adapter.

NOTE: Support was introduced for VMware ESXi™ only, and not classic ESX.

Not all of our storage partners support iSCSI Boot Firmware Table (iBFT) boot from SAN. Refer to the partner's own documentation for clarification.

9.1 Why Boot from SAN?

It quickly became clear that there was a need to boot via software iSCSI. Partners of VMware were developing blade chassis containing blade servers, storage and network interconnects in a single rack. The blades were typically diskless, with no local storage. The requirement was to have the blade servers boot off of an iSCSI LUN using network interface cards with iSCSI capabilities, rather than using dedicated hardware iSCSI initiators.

9.2 Compatible Network Interface Card

Much of the configuration for booting via software iSCSI is done via the BIOS settings of the network interface cards and the host. Check the VMware Hardware Compatibility List (HCL) to ensure that the network interface card is compatible. This is important, but a word of caution is necessary. If you select a particular network interface card and you see iSCSI as a feature, you might assume that you can use it to boot a vSphere host from an iSCSI LUN. This is not the case.

To see if a particular network interface card is supported for iSCSI boot, set the I/O device type to **Network** (not iSCSI) in the HCL and then check the footnotes. If the footnotes state that **iBFT** is supported, then this card can be used for boot from iSCSI.

9.3 Configuring Host BIOS for iSCSI Boot

After it is verified that the network interface card is supported, the next step is to enter the BIOS of the network interface card and ensure that it is enabled for iSCSI boot. The next step is to get into the network interface card configuration. For example, if you were planning to boot from software iSCSI with a Broadcom NetXtreme network interface card, this comes with a boot agent, Broadcom's Multi-Boot Agent (MBA) software utility. This agent enables a host to execute a boot process using images from remote servers, including iSCSI targets. In the MBA configuration menu, iSCSI must be selected as the boot protocol. If iSCSI isn't available as a boot protocol, it might mean that the iSCSI firmware has not been installed or that iSCSI has not been enabled on the network interface card. Refer to your network interface card vendor documentation for further information.

There are a number of different parameters to configure. When doing the initial installation, **Boot to iSCSI Target** must also be left **Disabled**. You must change it to Enabled for subsequent boots.

In the **initiator parameters** section, one would enter values for the IP address, subnet mask, default gateway, primary DNS and secondary DNS parameters of the initiator as needed. If authentication is required, then the CHAP ID (Challenge Handshake Authentication Protocol ID) and CHAP secret parameters should be entered.

In the **target parameters** section, one would enter values for the target IP address, target name and login information. If authentication is required, then once again the CHAP ID and CHAP secret parameters must be entered.

NOTE: The **Boot LUN ID** *(the LUN on the target that is used for the vSphere host installation and subsequent boots) should also be configured.*

Exit and save to complete the BIOS configuration. Whereas this sequence of steps is for the Broadcom MBA software utility, a similar set of steps must be taken for other network interface cards. We are now ready to install a vSphere host onto an iSCSI LUN via the software iSCSI initiator.

9.4 Installing a vSphere Host on an iSCSI LUN

After configuring the MBA parameters in the Broadcom network interface card, you can now go ahead with the installation of the vSphere host. The installation media is placed in the CD-ROM or made available via some other method in the BIOS on the host (for example, virtual media). The next step is to ensure that boot controller/device order is set correctly in the BIOS. For network interface cards, the network adaptor should appear before the CD-ROM in the boot order.

To simplify the procedure of booting from iSCSI, ensure that the boot LUN is initially mapped on one path only to the vSphere host. Another advisable step is to map LUN id 0 to the host. Although this requirement changes from storage array to storage array, it is easier to simply follow this instruction rather than to consult the documentation of the storage-array vendor to determine if it is required or not.

When the host is powered on, the system BIOS loads the OptionROM code of the network interface card and starts executing it. The OptionROM contains boot code and iSCSI initiator firmware. The iSCSI initiator firmware establishes an iSCSI session with the target.

On boot, a successful login to the target should be observed before installation starts. If you get a failure at this point, you must revisit the configuration steps done previously. The installation now begins.

As part of the installation process, a memory-only stateless VMkernel is first loaded. This must discover suitable LUNs for installation, one of which is the iSCSI LUN. However, for the VMkernel's iSCSI driver to communicate with the target, it requires that the TCP/IP protocol be set up. This is all done as part of the startup **init** script. The OptionROM of the network interface card is also responsible for handing off the initiator and target configuration data to the VMkernel. The handoff protocol is the **iBFT**. After the required networking is set up, an iSCSI session is established to the target configured in the iBFT. LUNs beneath the targets are discovered and registered with VMkernel SCSI stack (PSA).

If everything is successful during the initial installation, the iSCSI LUN is offered as a destination for the vSphere host image. You can now complete the vSphere host installation normally.

9.5 Booting from an iSCSI LUN

After the installation is complete, a single iSCSI configuration change is required in the iSCSI configuration. Again, using Broadcom NetXtreme as an example, the **Boot to iSCSI target** is set to **Enabled**. When the host is rebooted, it will boot the vSphere host from the iSCSI LUN via the software iSCSI initiator.

9.6 Troubleshooting Checklist When Booting from iSCSI

This paragraph contains a list of items that should be checked in the event that issues are encountered when booting from iSCSI. Ensure that the network interface card is on the HCL for iSCSI boot. Remember to check the footnotes. You should also make sure that your device has a firmware version that supports iSCSI boot and that the iSCSI configuration settings for initiator and target are valid. Check the login screen to make sure your initiator can log in to the target. If you make changes to the physical network, these must be reflected in the iBFT. Finally, the CLI command, **esxcfg-swiscsi -b -q**, displays the iBFT settings in the VMkernel and can be used to check the settings.

10. Advanced Settings

There are a number of tunable parameters available when using iSCSI datastores. Before drilling into these advanced settings in more detail, you should understand that the recommended values for some of these settings might (and probably will) vary from storage-array vendor to storage-array vendor. The objective here is to give you a clear and concise explanation of the tunable parameters and enable you to make your own decisions when tuning the values.

10.1 LoginTimeout

When iSCSI establishes a session between initiator and target, it must log in to the target. It will try to log in for a period of **LoginTimeout** seconds. If that is exceeded, the login fails.

10.2 LogoutTimeout

When iSCSI finishes a session between initiator and target, it must log out of the target. It will try to log out for a period of **LogoutTimeout** seconds. If that is exceeded, the logout fails.

10.3 RecoveryTimeout

The other options relate to how a dead path is determined. **RecoveryTimeout** is used to determine how long we should wait, in seconds, after PDUs are no longer being sent or received before placing a once-active path into a dead state. Realistically it's a bit longer than that, because other considerations are taken into account as well.

10.4 NoopInterval and NoopTimeout

The **noop** settings are used to determine if a path is dead when it is not the active path. iSCSI will passively discover if this path is dead by using the noop timeout. This test is carried out on nonactive paths every **NoopInterval** seconds. If a response isn't received by **NoopTimeout**, measured in seconds, the path is marked as dead.

Unless faster failover times are desirable, it is not required to change these parameters from their default settings. Use caution when modifying these parameters, because if paths fail too quickly and then recover, you might have LUNs/devices moving ownership unnecessarily between targets, and that can lead to path thrashing.

10.5 QFullSampleSize and QFullThreshold

Some of our storage partners require the use of the parameters **QFullSampleSize** and **QFullThreshold** to enable the adaptive queue-depth algorithm of VMware. With the algorithm enabled, no additional I/O throttling is required on the vSphere hosts. Refer to your storage-array vendor's documentation to see if this is applicable to your storage.

10.6 Disk.DiskMaxIOSize

To improve the performance of virtual machines that generate large I/O sizes, administrators can consider setting the advanced parameter **Disk.DiskMaxIOSize**. Some of our partners suggest setting this to 128KB to enhance storage performance. However, it would be best to understand the I/O size that the virtual machine is generating before setting this parameter. A different size might be more suitable to your application.

10.7 DelayedAck

A host receiving a stream of TCP data segments, as in the case of iSCSI, can increase efficiency in both the network and the hosts by sending less than one ack acknowledgment segment per data segment received. This is known as a delayed ack. The common practice is to send an ack for every other full-sized data segment and not to delay the ack for a segment by more than a specified threshold. This threshold varies between 100 and 500 milliseconds. vSphere hosts, as do most other servers, use a delayed ack because of its benefits.

Some arrays, however, take the very conservative approach of retransmitting only one lost data segment at a time and waiting for the host's ack before retransmitting the next one. This approach slows read performance to a halt in a congested network and might require the delayed ack feature to be disabled on the vSphere host. More details can be found in KB article 1002598.

11. Additional Considerations

11.1 Disk Alignment

This is not a recommendation specific to iSCSI, because it also can have an adverse effect on the performance of all block storage. Nevertheless, to account for every contingency, it should be considered a best practice to have the partitions of the guest OS running with the virtual machine aligned to the storage. Detailed descriptions of the way to do this alignment are beyond the scope of this white paper. Refer to the documentation of your individual storage-array vendor for further details.

11.2 Microsoft Clustering Support

With the release of vSphere 5.1, VMware supports as many as five nodes in a Microsoft Cluster. However, at the time of this writing, VMware does not support the cluster quorum disk over the iSCSI protocol. Additional quality testing of the iSCSI protocol with Microsoft Cluster Services is required to support such a configuration.

NOTE: With Microsoft Clustering, you cannot use the Round Robin path selection policy on the quorum disk.

11.3 In-Guest iSCSI Support

A number of in-guest iSCSI software solutions are available. The iSCSI driver of Microsoft is one commonly seen running in a virtual machine when the guest OS is a version of Microsoft Windows. The support statement for this driver can be found in KB article 1010547, which states that "if you encounter connectivity issues using a third-party software iSCSI initiator to the third-party storage device, engage the third-party vendors for assistance. If the third-party vendors determine that the issue is due to a lack of network connectivity to the virtual machine, contact VMware for troubleshooting assistance."

11.4 All Paths Down and Permanent Device Loss

All Paths Down (APD) can occur on a vSphere host when a storage device is removed in an uncontrolled manner or if the device fails and the VMkernel core storage stack cannot detect how long the loss of device access will last. One possible scenario for an APD condition is an FC switch failure that brings down all the storage paths, or, in the case of an iSCSI array, a network connectivity issue that similarly brings down all the storage paths.

A new condition known as Permanent Device Loss (PDL) was introduced in vSphere 5.0. The PDL condition enabled the vSphere host to take specific actions when it detected that the device loss was permanent. The vSphere host can be informed of a PDL situation by specific SCSI sense codes sent by the target array.

In vSphere 5.1, VMware introduced a PDL detection method for those iSCSI arrays that present only one LUN for each target. These arrays were problematic, because after LUN access was lost, the target also was lost. Therefore, the vSphere host had no way of reclaiming any SCSI sense codes.

vSphere 5.1 extends PDL detection to those arrays that have only a single LUN per target. With vSphere 5.1, for those iSCSI arrays that have a single LUN per target, an attempt is made to log in again to the target after a dropped session. If there is a PDL condition, the storage system rejects the effort to access the device. Depending on how the array rejects the efforts to access the LUN, the vSphere host can determine whether the device has been lost permanently (PDL) or is temporarily unreachable.

11.5 Read-Only File Systems on Linux Guest OS

VMware has identified a problem with certain Linux guest operating systems. RHEL 5 (GA), RHEL 4 U4, RHEL 4 U3, SLES 10 (GA) and SLES 9 SP3 might have their file systems become read-only in the event of a busy I/O retry or path failover of the ESX server's SAN or iSCSI storage. See VMware KB article 51306 for further details.

11.6 Round Robin Path Policy Setting IOPS=1

A number of our partners have documented that if using the Round Robin path policy, best results can be achieved with an IOPS=1 setting. This might well be true in very small environments where there are a small number of virtual machines and a small number of datastores. However, because the environment scales with a greater number of virtual machines and a greater number of datastores, VMware considers that the default settings associated with the Round Robin path policy to be sufficient. Consult your storage array vendor for advice on this setting.

11.7 Data Center Bridging (DCB) Support

Our storage partner Dell now supports iSCSI over DCB under the PVSP (Partner Verified and Supported Products) program of VMware. This is for the Dell EqualLogic (EQL) array only with certain Converged Network Adapters (CNAs) and only on vSphere version 5.1. See KB article 2044431 for further details.

11.8 Tape Device Support

vSphere hosts do not support iSCSI-connected tape devices.

Conclusion

iSCSI is now an extremely popular storage protocol found in vSphere infrastructures. This white paper brings together various disparate pieces of knowledge and documentation for VMware customers who are considering implementing iSCSI or have already implemented iSCSI and are looking for some best practices. This document should be used in conjunction with documentation available from our storage-array partners.

Acknowledgments

I would like to thank the following VMware partners for contributing to this paper: Jason Boche, Justin Braun and Darren W. Miller of Dell; and Christopher Viamon, Janice Henderson and Eric Siebert of HP. I would also like to thank Paudie O'Riordan, Rahul Dev, and Duncan Epping of VMware for their technical review of this white paper.

About the Author

Cormac Hogan is a senior technical marketing architect within the Cloud Infrastructure Product Marketing group at VMware. He is responsible for storage in general, with a focus on core VMware vSphere storage technologies and virtual storage, including the VMware vSphere Storage Appliance. Cormac has written a number of storage-related white papers and has given numerous presentations on storage best practices and new features. He has been with VMware since 2005.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2013 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW-WP-RNING-vSPHR-iSCSI-USLET-101 Docsource: OIC - 13VM008.14