Best Practices for running VMware vSphere™ on Network Attached Storage

WHITE PAPER

vmware®

Table of Contents

Introduction . . . 3
Background . . . 3
Overview of the Steps to Provision NFS Datastores . . . 4
Issues to Consider for High Availability . . . 5
Security Considerations . . . 7
Additional Attributes of NFS Storage . . . 7
- Thin Provisioning . . . 7
- De-duplication . . . 8
- Backup and Restore Granularity . . . 8

Summary of Best Practices . . . 8
- Networking Settings . . . 8
- Datastore Settings . . . 8
- Filer Settings . . . 9
- ESX Server Advanced Settings and Timeout settings . . . 9

Previously thought to be Best Practices . . . 9
Conclusion . . . 10
Appendix 1: Troubleshooting Tips . . . 11
Appendix 2: NFS Advanced Options . . . 12
Appendix 3: Solutions to common problems . . . 13
About the Author . . . 14

Introduction

The significant presence of Network Filesystem Storage (NFS) in the datacenter today, as well as the lower cost-per-port for IP based Storage, has lead to many people wanting to deploy virtualization environments with Network Attached Storage (NAS) shared storage resources.

As virtualization increases adoption, so does the deployment of VMware ESX servers that leverage NAS. For the purpose of clarity, both NFS and NAS refer to the same type of storage protocol and will be used as terms for the same thing throughout this paper.

The capabilities of VMware vSphere 4 on NFS are very similar to the VMware vSphere™ on block-based storage. VMware offers support for almost all features and functions on NFS—as it does for vSphere on SAN. Running vSphere on NFS is a very viable option for many virtualization deployments as it offers strong performance and stability if configured correctly.

This paper provides an overview of the considerations and best practices for deployment of VMware vSphere on NFS based storage. It also examines the myths that exist and will attempt to dispel confusion as to when NFS should and should not be used with vSphere. It will also provide some troubleshooting tips and tricks.

Background

VMware introduced the support of IP based storage in release 3 of the ESX server. Prior to that release, the only option for shared storage pools was Fibre Channel (FC). With VI3, both iSCSI and NFS storage were introduced as storage resources that could be shared across a cluster of ESX servers. The addition of new choices has led to a number of people asking "What is the best storage protocol choice for one to deploy a virtualization project on?" The answer to that question has been the subject of much debate, and there seems to be no single correct answer.

The considerations for this choice tend to hinge on the issue of cost, performance, availability, and ease of manageability. However, an additional factor should also be the legacy environment and the storage administrator familiarity with one protocol vs. the other based on what is already installed.

The bottom line is, rather than ask "which storage protocol to deploy virtualization on," the question should be, "Which virtualization solution enables one to leverage multiple storage protocols for their virtualization environment?" And, "Which will give them the best ability to move virtual machines from one storage pool to another, regardless of what storage protocol it uses, without downtime, or application disruption?" Once those questions are considered, the clear answer is VMware vSphere.

However, to investigate the options a bit further, performance of FC is perceived as being a bit more industrial strength than IP based storage. However, for most virtualization environments, NFS and iSCSI provide suitable I/O performance. The comparison has been the subject of many papers and projects. One posted on VMTN is located at: http://www.vmware.com/resources/techresources/10034.

The general conclusion reached by the above paper is that for most workloads, the performance is similar with a slight increase in ESX Server CPU overhead per transaction for NFS and a bit more for software iSCSI. For most virtualization environments, the end user might not even be able to detect the performance delta from one virtual machine running on IP based storage vs. another on FC storage.

The more important consideration that often leads people to choose NFS storage for their virtualization environment is the ease of provisioning and maintaining NFS shared storage pools. NFS storage is often less costly than FC storage to set up and maintain. For this reason, NFS tends to be the choice taken by small to medium businesses that are deploying virtualization—as well as the choice for deployment of virtual desktop infrastructures. This paper will investigate the trade offs and considerations in more detail.

Overview of the Steps to Provision NFS Datastores

Before NFS storage can be addressed by an ESX server, the following issues need to be addressed:

1. Have a virtual switch configured for IP based storage.
2. The ESX hosts needs to be configured to enable its NFS client.
3. The NFS storage server needs to have been configured to export a mount point that is accessible to the ESX server on a trusted network.

For more details on NFS storage options and setup, consult the best practices for VMware provided by the storage vendor.

Regarding item one above, to configure the vSwitch for IP storage access you will need to create a new vSwitch under ESX server configuration, networking tab in vCenter. Indicating it is a vmkernel type connection will automatically add to the vSwitch. You will need to populate the network access information.

Regarding item two above, to configure the ESX host for running its NFS client, you'll need to open a firewall port for the NFS client. To do this, select the configuration tab for the ESX Server in Virtual Center and click on Security Profile (listed under software options) and then check the box for NFS Client listed under the remote access choices in the Firewall Properties screen.

With these items addressed, an NFS datastore can now be added to the ESX server following the same process used to configure a data store for block based (FC or iSCSI) datastores.

- On the ESX Server configuration tab in VMware VirtualCenter, select storage (listed under hardware options) and then click the add button.
- On the screen for select storage type, select Network File System and in the next screen enter the IP address of the NFS server, mount point for the specific destination on that server and the desired name for that new datastore.
- If everything is completed correctly, the new NFS datastore will show up in the refreshed list of datastores available for that ESX server.

The main differences in provisioning an NFS datastores compared to block base storage datastores are:

- For NFS there are fewer screens to navigate through but more data entry required than block based storage.
- The NFS device needs to be specified via an IP address and folder (mount point) on that filer, rather than a pick list of options to choose from.

Issues to Consider for High Availability

To achieve high availability, the LAN on which the NFS traffic will run needs to be designed with availability, downtime-avoidance, isolation, and no single-fail-point of failure in mind.

Multiple administrators need to be involved in designing for high-availability: Both the virtual administrator and the network administrator. If done correctly the failover capabilities of an IP based storage network can be as robust as that of a FC storage network.

This section outlines these steps and investigates three levels of high availability available in a NFS storage configuration for vSphere.

Terminology

First, it is important to define a few terms that often cause confusion in the discussion of IP based storage networking. Some common terms and there definitions are as follows:

NIC/adapter/port/link – End points of a network connection.

Teamed/trunked/bonded/bundled ports – Pairing of two connections that are treated as one connection by a network switch or server. The result of this pairing are also referred to as an ether-channel. This is pairing of connections is defined as Link Aggregation in the 802.3 networking specification.

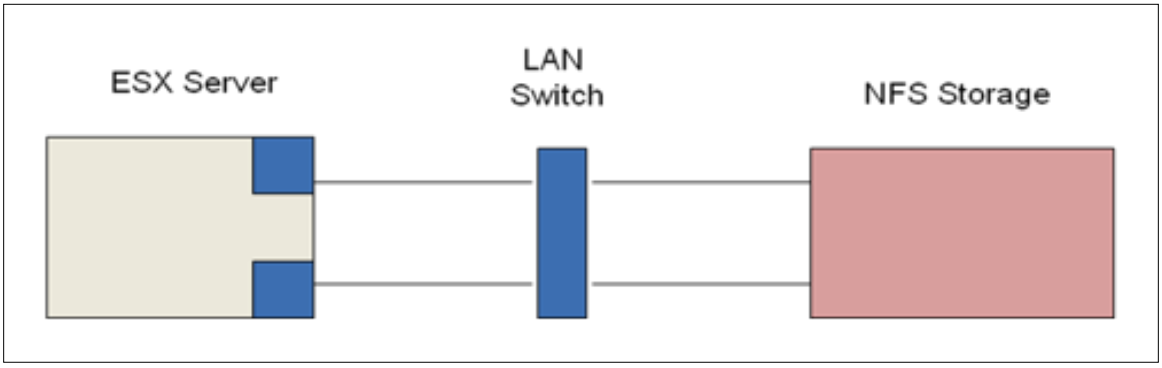
Cross stack ether channel – A pairing of ports that can span across two physical LAN switches managed as one logical switch. This is only an option with a limited number of switches that are available today.

IP hash – Method of switching to an alternate path based on a hash of the IP address of both end points for multiple connections.

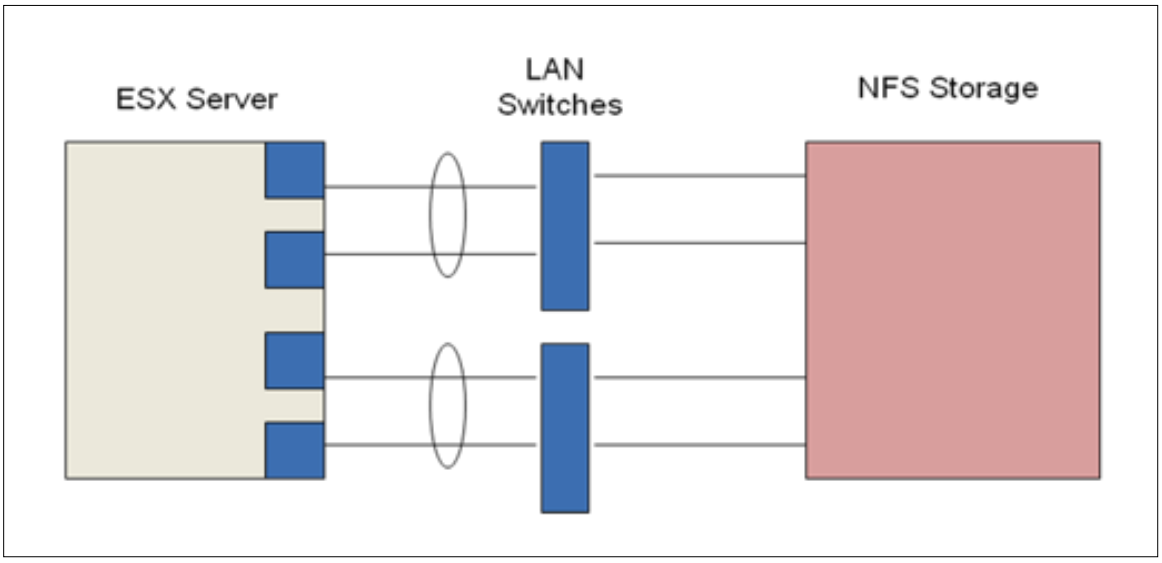
Virtual IP (VIF) – An interface used by the NAS device to present the same IP address out of two ports from that single array.

Avoiding single points of failure a the NIC, switch, filer levels

The first level of High Availability (HA) is to avoid a single point of failure being a NIC card in an ESX server, or the cable between the NIC card and the switch. With the solution having two NICs connected to the same LAN switch and configured as teamed at the switch and having IP hash failover enabled at the ESX server.

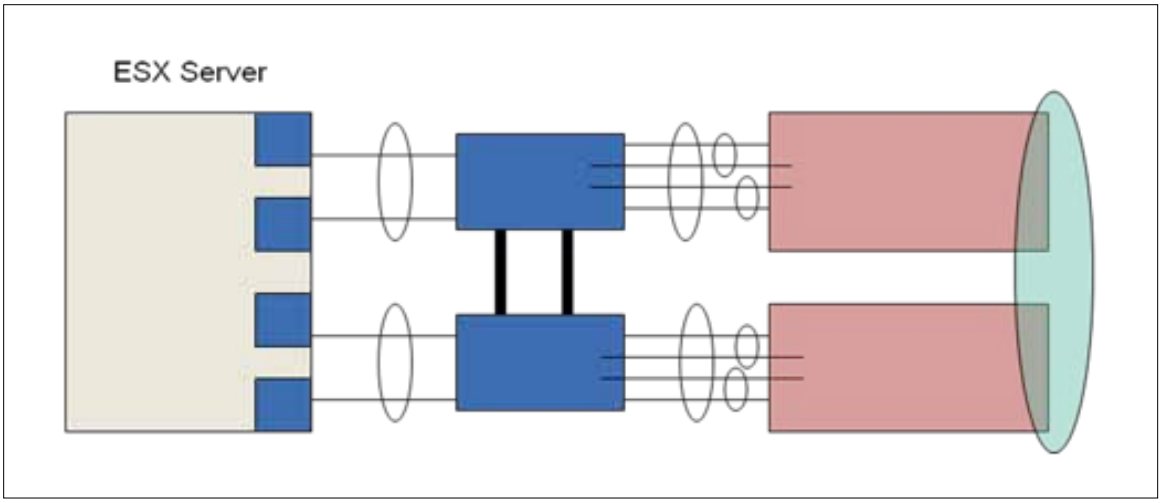


The second level of HA is to avoid a single point of failure being a loss of the switch to which the ESX connects. With this solution, one has four potential NIC cards in the ESX server configured with IP hash failover and two pairs going to separate LAN switches – with each pair configured as teamed at the respective LAN switches.



The third level of HA protects against loss of a filer (or NAS head) becoming unavailable. With storage vendors that provide clustered NAS heads that can take over for another in the event of a failure, one can configure the LAN such that downtime can be avoided in the event of losing a single filer, or NAS head.

An even higher level of performance and HA can build on the previous HA level with the addition of Cross Stack Ether-channel capable switches. With certain network switches, it is possible to team ports across two separate physical switches that are managed as one logical switch. This provides additional resilience as well as some performance optimization that one can get HA with fewer NICs, or have more paths available across which one can distribute load sharing.



Caveat: NIC teaming provides failover but not load-balanced performance (in the common case of a single NAS datastore)

It is also important to understand that there is only one active pipe for the connection between the ESX server and a single storage target (LUN or mountpoint). This means that although there may be alternate connections available for failover, the bandwidth for a single datastore and the underlying storage is limited to what a single connection can provide. To leverage more available bandwidth, an ESX server has multiple connections from server to storage targets. One would need to configure multiple datastores with each datastore using separate connections between the server and the storage. This is where one often runs into the distinction between load balancing and load sharing. The configuration of traffic spread across two or more datastores configured on separate connections between the ESX server and the storage array is load sharing.

Security Considerations

VMware vSphere implementation of NFS supports NFS version 3 in TCP. There is currently no support for NFS version 2, UDP, or CIFS/SMB. Kerberos is also not supported in the ESX Server 4, and as such traffic is not encrypted. Storage traffic is transmitted as clear text across the LAN. Therefore, it is considered best practice to use NFS storage on trusted networks only. And to isolate the traffic on separate physical switches or leverage a private VLAN.

Another security concern is that the ESX Server must mount the NFS server with root access. This raises some concerns about hackers getting access to the NFS server. To address the concern, it is best practice to use of either dedicated LAN or VLAN to provide protection and isolation.

Additional Attributes of NFS Storage

There are several additional options to consider when using NFS as a shared storage pool for virtualization. Some additional considerations are thin provisioning, de-duplication, and the ease-of-backup-and-restore of virtual machines, virtual disks, and even files on a virtual disk via array based snapshots. The next section focuses on the options available and the criterion might make one choice a better selection than the alternative.

Thin Provisioning

Virtual disks (VMDKs) created on NFS datastores are in thin provisioned format by default. This capability offers better disk utilization of the underlying storage capacity in that it removes what is often considered wasted disk space. For the purpose of this paper, VMware will define wasted disk space as allocated but not used. The thin-provisioning technology removes a significant amount of wasted disk space.

On NFS datastores, the default virtual disk format is thin. As such, less allocation of VMFS volume storage space than is needed for the same set of virtual disks provisioned as thick format. Although the thin format virtual disks can also be provisioned on block-based VMFS datastores. VMware vCenter™ now provides support for both the creation of thin virtual disks as well as the monitoring the datastores that may be over-committed (i.e. that can trigger alarms for various thresholds of either space used approaching capacity or provisioned pace out pacing capacity by varying degrees of over commit.)

De-duplication

Some NAS storage vendors offer data de-duplication features that can greatly reduce the amount of storage space required. It is important to distinguish between in-place de-duplication and de-duplication for backup streams. Both offer significant savings in space requirements, but in-place de-duplication seems to be far more significant for virtualization environments. Some customers have been able to reduce their storage needs by up to 75 percent of their previous storage footprint with the use of in place de-duplication technology.

Backup and Restore Granularity

The backup of a VMware virtual environment can be done in several ways. Some common methods are to use VMware Consolidated Backup (VCB), placing agents in each Guest OS, or leveraging array based snapshot technology. However, the process of restoring the virtual machine gets to be a bit more complicated as there are often many levels of granularity that are requested for restore. Some recovery solutions might require the entire datastore to be restored while others might require a specific virtual machine or even a virtual disk for a virtual machine to be recovered. The most granular request is to have only a specific file on a virtual disk for a given virtual machine restored.

With NFS and array based snapshots, one has the greatest ease and flexibility on what level of granularity can be restored. With an array-based snapshot of a NFS datastore, one can quickly mount a point-in-time copy of the entire NFS datastore, and then selectively extract any level of granularity they want. Although this does open up a bit of a security risk, NFS does provide one of the most flexible and efficient restore from backup option available today. For this reason, NFS earns high marks for ease of backup and restore capability.

Summary of Best Practices

To address the security concerns and provide optimal performance, VMware provides the following best practices:

Networking Settings

To isolate storage traffic from other networking traffic, it is considered best practice to use either dedicated switches or VLANs for your NFS and iSCSI ESX server traffic. The minimum NIC speed should be 1 gig E. In VMware vSphere, use of 10gig E is supported. Best to look at the VMware HCL to confirm which models are supported.

It is important to not over-subscribe the network connection between the LAN switch and the storage array. The retransmitting of dropped packets can further degrade the performance of an already heavily congested network fabric.

Datastore Settings

The default setting for the maximum number of mount points/datastore an ESX server can concurrently mount is eight. Although the limit can be increased to 64 in the existing release. If you increase max NFS mounts above the default setting of eight, make sure to also increase Net.TcpipHeapSize as well. If 32 mount points are used, increase tcpip.Heapsize to 30MB.

TCP/IP Heap Size
The safest way to calculate the tcpip heap size given the number of NFS volumes configured is to linearly scale the default values. For 8 NFS volumes the default min/max sizes of the tcpip heap are respectively 16MB/64MB. This means the tcpip heap size for a host configured with 64 NFS volumes should have the min/max tcpip heap sizes set to 32MB/128MB (current maximum values).

Filer Settings

In a VMware cluster, it is important to make sure to mount datastores the same way on all ESX servers. Same host (hostname/FQDN/IP), export and datastore name. Also make sure NFS server settings are persistent on the NFS filer (use FilerView, exportfs –p, or edit /etc/exports).

ESX Server Advanced Settings and Timeout settings

When setting the NIC teaming settings, it is considered best practice to select "no" for NIC teaming failback option. If there is some intermittent behavior in the network, this will prevent the flip-flopping of NIC cards being used.

When setting up VMware HA, it is a good starting point to also set the following ESX server timeouts and settings under the ESX server advanced setting tab.

- NFS.HeartbeatFrequency = 12
- NFS.HeartbeatTimeout = 5
- NFS.HeartbeatMaxFailures = 10

NFS Heartbeats
NFS heartbeats are used to determine if an NFS volume is still available. NFS heartbeats are actually GETATTR requests on the root file handle of the NFS Volume. There is a system world that runs every NFS. HeartbeatFrequency seconds to check if it needs to issue heartbeat requests for any of the NFS volumes. If a volume is marked available, a heartbeat will only be issues if it has been > NFS.HeartBeatDelta seconds since a successful GETATTR (not necessarily a heartbeat GETATTR) for that volume was issued The NFS heartbeat world will always issue heartbeats for NFS volumes that are marked unavailable. Here is the formula to calculate how long it can take ESX to mark an NFS volume as unavailable:

RoundUp(NFS.HeartbeatDelta, NFS.HeartbeatFrequency) + (NFS.HeartbeatFrequency * (NFS.HeartbeatMaxFailures - 1)) + NFS.HeartbeatTimeout

Once a volume is back up it can take NFS.HeartbeatFrequency seconds before ESX marks the volume as available. See Appendix 2 for more details on these settings.

Previously thought to be Best Practices

Some early adopters of VI3 on NFS created some best practices that are no longer viewed favorably. They are:

1. Turn off the NFS locking within the ESX server
2. Not placing virtual machine swap space on NFS storage.

Both these have been debunked and the following section provides more details as to why these are not no longer considered best practices.

NFS Locking

NFS locking on ESX does not use the NLM protocol. VMware has established its own locking protocol. These NFS locks are implemented by creating lock files on the NFS server. Lock files are named ".lck-<fileid>", where <fileid> is the value of the "fileid" field returned from a GETATTR request for the file being locking. Once a lock file is created, VMware periodically (every NFS.DiskFileLockUpdateFreq seconds) send updates to the lock file to let other ESX hosts know that the lock is still active. The lock file updates generate small (84 byte) WRITE requests to the NFS server. Changing any of the NFS locking parameters will change how long it takes to recover stale locks. The following formula can be used to calculate how long it takes to recover a stale NFS lock:

(NFS.DiskFileLockUpdateFreq * NFS.LockRenewMaxFailureNumber) + NFS.LockUpdateTimeout

If any of these parameters are modified, it's very important that all ESX hosts in the cluster use identical settings. Having inconsistent NFS lock settings across ESX hosts can result in data corruption!

In vSphere the option to change the NFS.Lockdisable setting has been removed. This was done to remove the temptation to disable the VMware locking mechanism for NFS. So it is no longer an option to turn it off in vSphere.

Virtual Machine Swap Space Location

The suggestion to not place virtual machine swap space on the NFS datastore was something that appeared in early training materials for VI3 and in some Knowledge Base articles. However, this was determined to be a bit too conservative and has been removed from the majority of the materials. The current position states it is far better to address the performance concern by not over committing the memory of the virtual machines that are on a NFS datastore than to move it to other storage and loose the benefit of encapsulation. The performance impact of having the virtual machine swap space on separate storage was not as big a difference as originally thought. So keeping the virtual machine swap space on the NFS datastore is now considered to be the best practice.

Conclusion

In short, Network Attached Storage has matured significantly in recent years and it offers a solid availability and high performance foundation for deployment with virtualization environments. Following the best practices outlined in this paper will ensure successful deployments of vSphere on NFS. Many people have deployed both vSphere and VI3 on NFS and are very pleased with the results they are experiencing. NFS offers a very solid storage platform for virtualization.

Both performance and availability are holding up to expectations and as best practices are being further defined, the experience of running VMware technology on NFS is proving to be a solid choice of storage protocols. Several storage technology partners are working closely with VMware to further define the best practices and extend the benefits for customers choosing to deployment VMware vSphere on this storage protocol option.

Appendix 1: Troubleshooting Tips

There are many possible causes for Network Attached Storage not to be available on an ESX server. It is important to troubleshoot the symptoms and isolate the probable cause in a systematic manner. Ruling out issues at the NFS storage server first and then making sure the list of items that must be set up correctly prior to adding a NFS datastore. The following items and the table are intended to help troubleshoot common areas of disconnect.

If you do not have a connection from the Service Console to the NAS device, you will get the following error message:

The mount request was denied by the NFS server. Check that the export exists and that the client can access it.

If you are trying to mount a share over UDP rather than TCP, you will get a descriptive message telling you that this is not allowed. Similarly, if you are trying to mount a share using a version of NFS which is not version 3, you will also get a very descriptive message.

If you do not have the **no_root_squash** option set, you will get the following error when you try to create a virtual machine on the NAS datastore:

Figure 1. no_root_squash permissions pop-up

This can be confusing since you can still create the datastore, but you will not be able to create any virtual machines on it.

*Note: the NetApp equivalent of no_root_squash is **anon=0** in the **/etc/exports** file.*

SYMPTOM	POSSIBLE CAUSE CATEGORY
Typically, failure to mount	The portmap or nfs daemons are not running
Typically, failure to mount or failure to write enable. A space between the mount point and the (rw) causes the share to be read-only—a frustrating and hard to diagnose problem	Syntax error on client mount command or server's/etc/exports
Mounts OK, but access to the data is impossible or not as specified	Problems with permissions, uid's and gid's
Mount failures, timeouts, excessively slow mounts, or intermittent mounts	Firewalls filtering packets necessary for NFS
Mount failures, timeouts, excessively slow mounts, or intermittent mounts	Bad DNS on server

Table 1. NFS Configuration issues

Appendix 2: NFS Advanced Options

- **NFS.DiskFileLockUpdateFreq**
  Time between updates to the NFS lock file on the NFS server. Increasing this value will increase the time it takes to recover stale NFS locks. (See NFS Locking)
- **NFS.LockUpdateTimeout**
  Amount of time VMWare waits before we stop a lock update request. (See NFS Locking)
- **NFS.LockRenewMaxFailureNumber**
  Number of lock update failures that must occur before VMare marks the lock as stale. (See NFS Locking)
- **NFS.HeartbeatFrequency**
  How often the NFS heartbeat world runs to see if any NFS volumes need a heartbeat request. (See NFS Heartbeats)
- **NFS.HeartbeatTimeout**
  Amount of time VMware waits before stopping a heartbeat request. (See NFS Heartbeats)
- **NFS.HeartbeatDelta**
  Amount of time after a successful GETATTR request before the heartbeat world will issue a heartbeat request for a volume. If an NFS volume is in an unavailable state, an update will be sent every time the heartbeat world runs (NFS.HeartbeatFrequency seconds). (See NFS Heartbeats)
- **NFS.HeartbeatMaxFailures**
  Number of consecutive heartbeat requests that must fail before VMwares mark a server as unavailable. (See NFS Heartbeats)
- **NFS.MaxVolumes**
  Maximum number of NFS volume that can be mounted. The TCP/IP heap must be increased to accommo-date the number of NFS volumes configured (See TCP/IP Heap Size)
- **NFS.SendBufferSize**
  This is the size of the send buffer for NFS sockets. This value was chosen based on internal performance testing. Customers should not need to adjust this value.
- **NFS.ReceiveBufferSize**
  This is the size of the receive buffer for NFS sockets. This value was chosen based on internal performance testing. Customers should not need to adjust this value.
- **NFS.VolumeRemountFrequency**
  This determines how often VMWare would try to mount an NFS volume that was initially unmountable. Once a volume is mounted, it never needs to be remounted. The volume may be marked unavailable if VMWare loses connectivity to the NFS server—but it will still remain mounted.

Appendix 3: Solutions to common problems

When I mount a volume, I do not see an entry in /vmfs/volumes.
This generally indicates the mount request failed. The vmkernel log file (`/proc/vmware/log` or `/var/log/vmkernel`) should contain some warnings to indicate what went wrong.

The server does not support NFS v3 over TCP.
From the console run: `rpcinfo -p [server name]`

Verify you see entries for:

100000	2	tcp	111	portmapper
100003	2	tcp	2049	nfs
100005	3	tcp	896	mountd

The server name or volume is entered incorrectly
From the console run: `showmount -e [server name]`

Verify you see the volume listed correctly.

The client is not authorized by the server to mount the volume.
NFS server can impose restrictions on which clients are allowed to access a given NFS volume.

From the console run: `showmount -e`

In general, this will show you the IP addresses that are allowed to mount the volume. If you are trying to mount a volume from a Linux NFS Server the server's /etc/exports should look something like:

```
*(rw,no_root_squash,sync)
```

Note: Work with root squashed volumes by rc1.

There is no default gateway configured
From the console run: `esxcfg-route`

If the gateway is not configured, go to the step above for configuring your default route.

The portgroup for the vmknic has no uplink

For some some unknown reason an upgrade from build to build can result in a situation where a portgroup loses its uplink. It might look like the following:

Switch Name vswitch0 vmnic0	Num Ports 32	Used Ports	Configured Ports 4	Uplinks 32
PortGroup Name nfs	Internal ["ID"]	VLAN ["ID"]	Used Ports 0	Uplinks 0

Note: In this case the portgroup nfs has no uplink. If this occurs:

1. Disable any vmknic using this portgroup: `esxcfg-vmknic -D [portgroup name]`
2. Remove the portgroup: `esxcfg-vswitch -D [portgroup name]`
3. Remove the vswitch: `esxcfg-vswitch -d [vswitch name]` 1 Go through the process above of configuring a new vswitch and portgroup. Verify the portgroup has an uplink. Do not forget to re-enable your vmknic and double check your default route is still set correctly.

About the Author

Paul Manning is a Storage Architect in the technical marketing group at VMware and is focused on virtual storage management. Previously, he worked at EMC and Oracle, where he had more than ten years' experience designing and developing storage infrastructure and deployment best practices. He has also developed and delivered training courses on best practices for highly available storage infrastructure to a variety of customers and partners in the United States and abroad. He is the author of numerous publications and presented many talks on the topic of best practices for storage deployments and performance optimization.

Check out the following multi vendor blog posts on NFS:

The Virtual Storage Guy

Virtual Geek

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com