NETAPP UNIVERSITY

NCDA Boot Camp Training Review

Certification Study Guide

ATTENTION

The information contained in this guide is intended for training use only. This guide contains information and activities that, while beneficial for the purposes of training in a closed, non-production environment, can result in downtime or other severe consequences and therefore are not intended as a reference guide. This guide is not a technical reference and should not, under any circumstances, be used in production environments. To obtain reference materials, please refer to the NetApp product documentation located at http://now.netapp.com/ for product information.

COPYRIGHT

RESTRICTED RIGHTS LEGEND

TRADEMARK INFORMATION

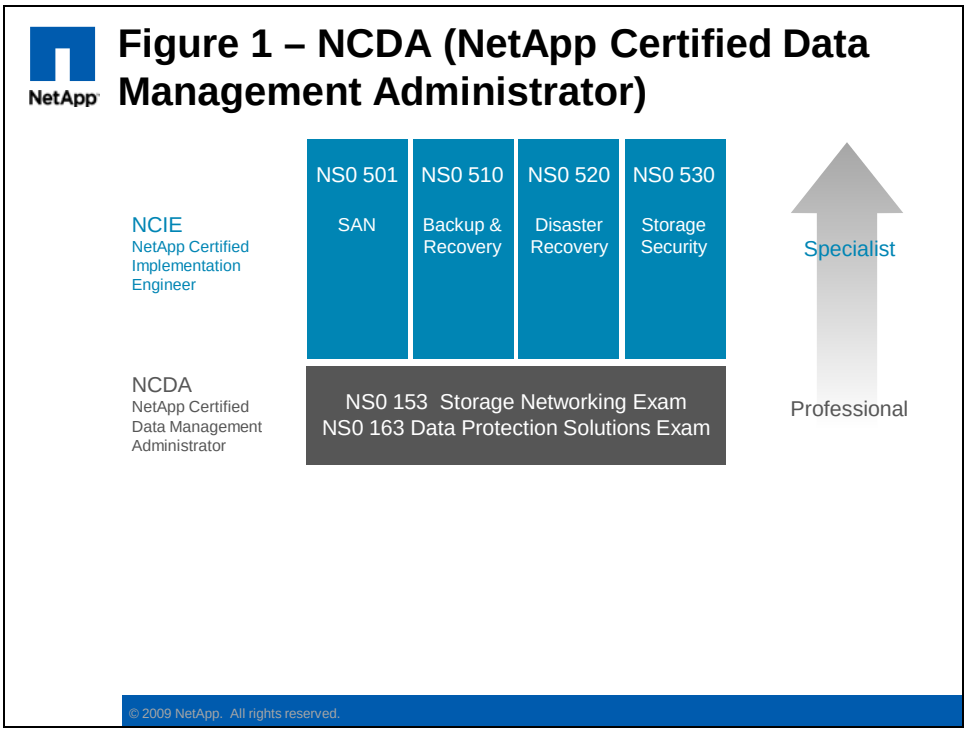
TABLE OF CONTENTS

INTRODUCTION

This guide was developed as an aid for those preparing to sit for the NCDA certification exams. It is intended to be a concise review of the material that is included in the **NAS Boot Camp** and **SAN Boot Camp** courses.

CERTIFICATION DESCRIPTION

As a NetApp Certified Data Management Administrator, you will have proven skills in performing in-depth support, administrative functions, and performance management for CIFS, NFS, and FCP for SCSI or iSCSI for TCP/IP protocols on a NetApp storage appliance running the Data ONTAP® operating system in NFS and Windows® (CIFS) multiprotocol environments. You will also be able to implement active-active controller configuration and SyncMirror® to ensure continuous data availability and rapid recovery of data in the event of a disaster, and use the SnapMirror®, SnapRestore®, and SnapVault® products to manage and protect mission-critical data.



EXAMS:

The following two sections include training review material that may assist you in studying for the relevant certification exams:

- Storage Networking (Exam 153)
- Data Protection Solutions (Exam 163)

Good luck!

EXAM NS0-153 – STORAGE NETWORKING

As a NetApp Certified Data Management Administrator, you will have proven skills in performing in-depth support, administrative functions, and performance management for CIFS, NFS, and FCP for SCSI or iSCSI for TCP/IP protocols on a NetApp storage appliance running the Data ONTAP operating system in NFS and Windows (CIFS) multiprotocol environments. You will also be able to implement active-active controller configuration and SyncMirror to ensure continuous data availability and rapid recovery of data in the event of a disaster, and use the SnapMirror, SnapRestore, and SnapVault products to manage and protect mission-critical data.

SKILLS TESTED:

- Describe the configuration requirements for NFS in a storage appliance environment
- Configure the storage appliance to support Kerberos™ security
- Explain and identify common components of the Common Internet File System (CIFS)
- Create, configure, and manage CIFS shares, groups, and permissions
- Review and identify potential performance issues given storage system statistics
- Analyze NFS performance using sysstat and nfsstat commands
- Investigate, identify, troubleshoot, and implement solutions in a CIFS or NFS environment
- Identify supported SAN configurations and necessary hardware and software components (including NetApp Support Console)
- Identify concepts, commands, and procedures for using the UNIX® file system especially in relationship to creating and mounting file systems using UFS on storage system-based LUNs
- Identify the different components and tools required to create a LUN
- Install or update HBA drivers and firmware for FCP for SCSI and iSCSI for TCP/IP protocols
- Use the sio_ntap utility as well as storage system commands to gather data for performance and problem analysis
- Collect data to assist with troubleshooting hardware, operating systems, and applications

RECOMMENDED COURSES:

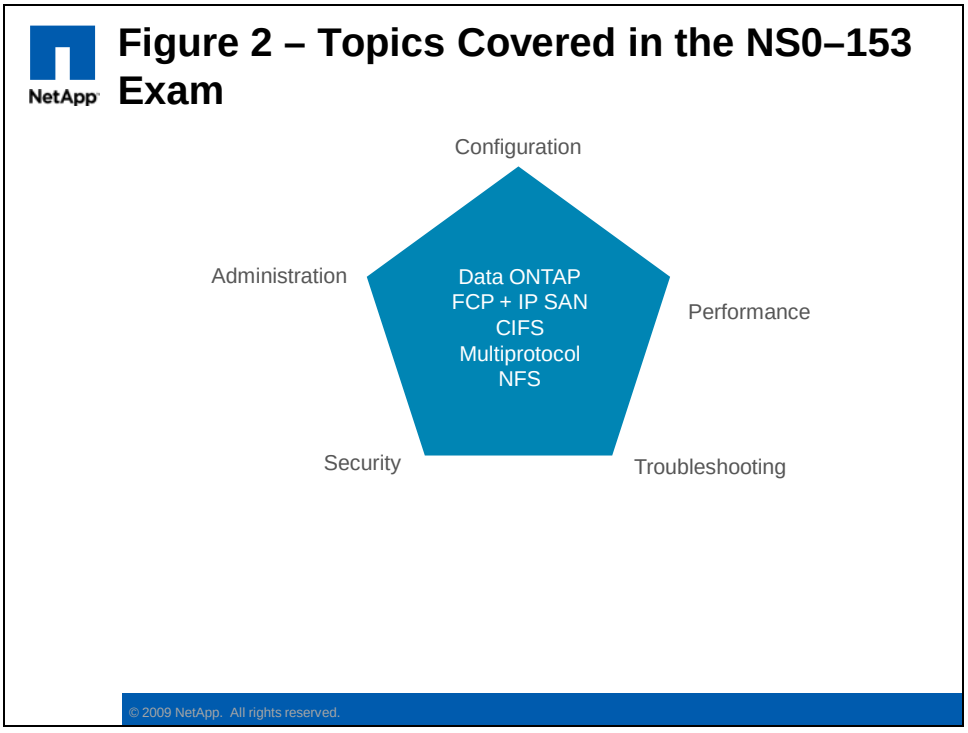
- ILT Course: Data ONTAP Fundamentals
- ILT Course: Data ONTAP SAN Administration
- ILT Course: Data ONTAP CIFS Administration
- ILT Course: Data ONTAP NFS Administration
- WBT Course: Data ONTAP Fundamentals

NOTE: ILT – Instructor-Led Training and WBT – Web-Based Training

EXAM PREPARATION

This section describes a number of NetApp FAS learning points that are relevant to the NS0-153 exam. However, it is not limited to just the exam topics and attempts to provide a brief summary of a range of NetApp technologies.

Figure 2 highlights the main subjects covered in the NS0-153 exam (white text) and the range of topics covered within each subject (black text).



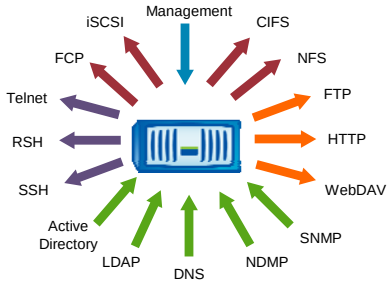
A brief overview of the relevant training and other material is provided in the following sections.

DATA ONTAP

The operating system of the NetApp FAS storage controller is known as Data ONTAP. It is the foundation of the NetApp Unified Storage architecture, supporting a wide range of storage protocols and storage management software features.

The NetApp FAS storage controllers are designed around a unified storage architecture, and support numerous storage protocols, software features and storage tiers in a simple appliance-like design. Refer to Figure 3 for an example of the supported protocols.

Figure 3 – Protocols Supported by the NetApp FAS Controllers



Legend

Red	– SAN and NAS protocols	Green	– External protocols (for example, Authentication)
Orange	– FTP and Web protocols	Purple	– Management protocols
		Blue	– Single point of administration

NOTE: This figure only shows the major storage, management, and authentication protocols. For brevity, some protocols such as the NetApp API (ZAPI) and NIS are not shown.

CONFIGURATION

When the storage controller is powered on for the first time it will automatically run the **`setup`** script. This script configures such fundamental parameters as the hostname, network IP addresses, and CIFS authentication (if licensed). You can rerun the **`setup`** script manually at any time so as to reconfigure these parameters.

To manually configure the FAS controller's Ethernet network interfaces you need to use the following commands:

- Use the **`ifconfig`** command
  - `ifconfig ns0 192.168.1.1 netmask 255.255.255.0`
- Or, use the **FilerView®** Web interface

NOTE: Most changes made from the CLI are transient and need to be entered into the `/etc/rc` file so as to be persistent after a system reboot.

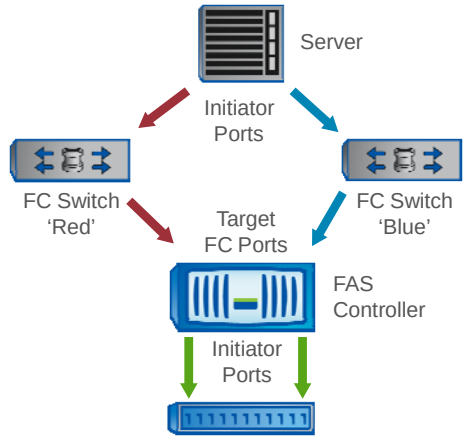
You may also need to configure the inbuilt FC ports (located on the motherboard itself). The inbuilt FC ports can function in one of two modes:

- **initiator** mode (the default)
  - For connection to disk expansion drawers
  - Use the `fcadmin config <adapter_name> -t initiator` to set an adapter to initiator mode
- **target** mode
  - For connection to host systems (that is, Windows or UNIX servers)
  - Use the `fcadmin config <adapter_name> -t target` to set an adapter to target mode

NOTE: This only applies to the inbuilt FC ports. Any FC HBA adapters that you purchase will come predefined as either initiator or target adapters.

The **`fcp show adapter`** command will only display those FC ports that are configured to function in *target* mode. The **`fcp show initiator`** command will display all host FC *initiator* ports that are visible via the controller's FC *target* ports.

Figure 4 – Fibre Channel Initiator/Target Port Relationships



NOTE: Similar initiator/target relationships also exist in iSCSI connections.

ADMINISTRATION

The administration of the NetApp FAS controller can be performed via a number of administrative interfaces. For example:

- **FilerView**, the Web-based GUI that is included in the FAS controller itself
  - Access via **http://<fasname>/na_admin**
- **CLI**, accessed via telnet, rsh, or ssh
  - The `aggr status` command displays the existing aggregates
  - The `cifs shares –add …` command defines a new CIFS share
  - The `options dns.enable on` command would enable DNS name resolution (which would then require further configuration)
- **Operations Manager**, a licensed tool that is installed on a host, provides sophisticated management tools for one or more FAS controllers. This product includes:
  - **Performance Advisor**
  - **Protection Manager**
  - **Provisioning Manager**

NOTE: Management of the FAS storage infrastructure can be promoted up to the host OS and application layer with the optional (licensed) **SnapDrive** (for Windows or UNIX) and **SnapManager** (for Application Integration) tools respectively.

The various SAN and NAS protocols differ in many technical details. SAN provides block-level access and NAS provides file-level access; some use fibre channel connections and others Ethernet. However, the end result is to provide remote systems (hosts) with access to centralized, shared storage.

Perhaps the simplest way to define the difference between SAN and NAS is to look at who manages the file system, and sharing of access. For a SAN device, the controller provides block-level access to the host/s, the host/s then create and manage their own local file system (for example, Ext3), and there is generally no sharing of this local file system with other hosts. In comparison, NAS storage utilizes the file system on the controller itself (WAFL) and the controller provides shared file-level access to multiple remote systems (hosts or desktops).

Both FC and iSCSI SANs support **Multipathing**, where two or more redundant physical paths exist between a host and the controller. Refer to Figure 4 for a simple example. This is critical to ensure storage availability in the case of a SAN hardware failure. NetApp supports various multipathing options for both FCP and iSCSI. For example:

- FCP and iSCSI multipathing
  - With the host platform's **native MPIO** driver (multipath I/O)
  - With the **NetApp DSM** for Windows (Device Specific Module)
  - With the **Veritas dynamic multipath** software (VxDMP)
- **iSCSI** only multipathing
  - With iSCSI's inherent **MCS** (multiple connections per session)

Which particular multipath solution that you choose to implement will depend on the needs of your particular SAN environment.

The **NetApp On the Web (NOW)** website is a resource available to all customers and business partners. You need to develop some familiarity with the NOW site, as it is a primary resource if you have any questions regarding the configuration or performance of a FAS storage controller. Some of the resources available on the NOW site include:

- A searchable **knowledgebase** of NetApp product information
- **Online manuals**, for all NetApp products
- **Software downloads**, including updates and evaluation versions
- The **RMA** process, used to request replacement for any failed hardware
- **Bugs online**, used to review all known software bugs
- **Release comparison**, used to see in which version of DOT that a particular bug was fixed

PERFORMANCE

There are a number of commands on the FAS storage controller to collect system performance data. This can be either a broad summary of system performance, or can drill down to very specific performance parameters. Some common commands for collecting performance data are:

- **`sysstat`**
  - The default output includes cifs, nfs, http, and also CPU, NVRAM, NICs, and Disk performance values. The `-b` parameter will add block-level (that is, FCP and iSCSI) performance to the output
  - The `sysstat -s 5` command will run every five seconds and print a summary on termination (the default interval is 15 seconds).
  - The `sysstat -u` command will display extended utilization data, including Consistency Point (CP) time and type. This can be used to identify when a controller is busy (cp_from_log_full ('F') and cp_from_cp ('B') are good indications of this).
- **`statit`**
  - This command can output performance data on various object types, specific instances of an object type, or other performance counters.
  - The `statit` command will display many performance items, including per disk performance data
- **`stats`**
  - An advanced mode command that can provide low level performance data
  - The `stats show cifs:cifs:cifs_latency` command will display the average latency value for the CIFS protocol
  - These performance counters can also be accessed via the Windows **PerfMon** GUI (but only if CIFS is also enabled)

There are also several commands to collect Ethernet network interface performance statistics and other data. For example:

- **`netstat`**
  - The `netstat -i` command will display the network interfaces and the number of in/out network packets, as well as errors and collision values
- **`ifstat`**
  - The `ifstat -a` command will display a low level view of interface performance data, including Tx/Rx bytes per second

If you cannot resolve a performance problem using the above tools, then you may need to download the **`perfstat`** command from the NOW website. This command runs on a host

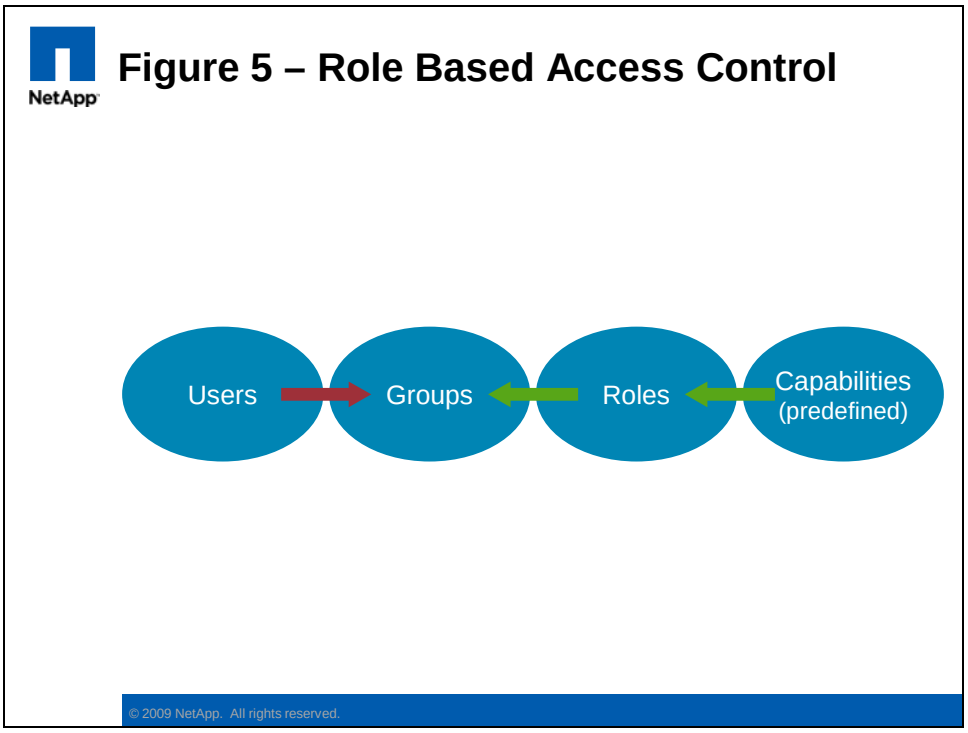
(Windows or UNIX) and simultaneously collects both host and controller data for correlated analysis. For example:

- **`perfstat`**
  - Captures all needed performance information with one command
  - Captures information from host(s) and filer(s)
  - Captures all information simultaneously for cross correlation
  - Operates on all host platforms and all filer platforms
  - All captured data is recorded in a single plain-text output file

SECURITY

The security of the FAS controller (that is, the DOT operating system) is a different topic than the security of the data stored on the controller. This section deals with the DOT security only.

The DOT operating system supports **Role Based Administration Control (RBAC)**, where defined roles, with specific capabilities, can be bound to groups, to which individual users are assigned. Those users can then only perform tasks for which their groups/roles/capabilities give them sufficient privilege.



For administrative purposes one or more accounts are usually defined on the FAS storage controller. The **`useradmin`** command is used to create and modify local admin accounts. For example:

- To create a new admin user
  - `useradmin` **`user`** `add <username> -g <groupname>`

- To list the local groups
  - `useradmin **group** list`
- To delete an existing role
  - `useradmin **role** delete <rolename>`
- To assign a capability to an existing role
  - `useradmin role modify <rolename> -a <**capability**>`

These commands, with the obvious variations, can be used to add, list, delete, or modify either users, groups, or roles. The capabilities are predefined and cannot be changed.

NOTE: Actually, there is always at least one administrative account, **`root`**, which cannot be deleted.

TROUBLESHOOTING

If you need to capture a network packet trace so as to analyze a protocol level problem, then you could use the **`pktt`** command on the FAS controller. For example:

- **`pktt`**
  - The `pktt` command is normally run for a short period, capturing network traffic on a particular interface, potentially filtered so as to target a particular client.
  - The `pktt` command produces output in standard tcpdump format, which can be analyzed in several third-party network monitoring tools. For example:
    - **`ethereal`** (download from wireshark.org)
    - **`netmon`** (download from microsoft.com)
  - You will need to convert the tcpdump output to netmon format first using the **`capconv.exe`** command that you can download from the NOW website.

SAN

The NetApp FAS storage controllers support both the Fibre Channel Protocol (FCP) and iSCSI SAN standards.

One thing (among many) that distinguishes the NetApp SAN architecture from competitive solutions is that the LUNs are defined below the SAN protocol layer. This means that a NetApp LUN is not a FCP LUN or an iSCSI LUN, but it can be exposed to a host via either or both protocols, even simultaneously. This also allows for easy migration between the SAN access protocols if required.

CONFIGURATION

Refer tothe section titled '*Managing systems with onboard Fibre Channel adapters'* in the '*Data ONTAP 7.3 Block Access Management Guide for iSCSI and FC'* for a description of configuring the initiator/target status of the controller's inbuilt FC ports.

In an Active/Active configuration, a LUN that is being accessed via FCP is visible from every FC target port on both FAS storage controllers (even though the LUN is actually "owned" by only one controller). This FC access mode is called **single image** and has been the new default FC clustering mode since DOT 7.2. You can change (with caution!) between FC clustering modes with the **`fcp cfmode`** command. You will need to reboot the controller after such a change.

The full list of FC clustering modes are:

- **single_image** (the default since DOT 7.2)
  - LUNs are visible on all FC target ports (WWPN) on both controllers
  - A common WWNN is shared by both controllers
- **partner**
- **mixed**
- **dual_fabric**
- **standby**

Refer to the product documentation if you require a description of the earlier FC cluster modes

Although all these modes are supported (for existing systems), only single image mode may be selected for new storage controllers. The following table details some of the features and limitations of the various FC clustering modes:

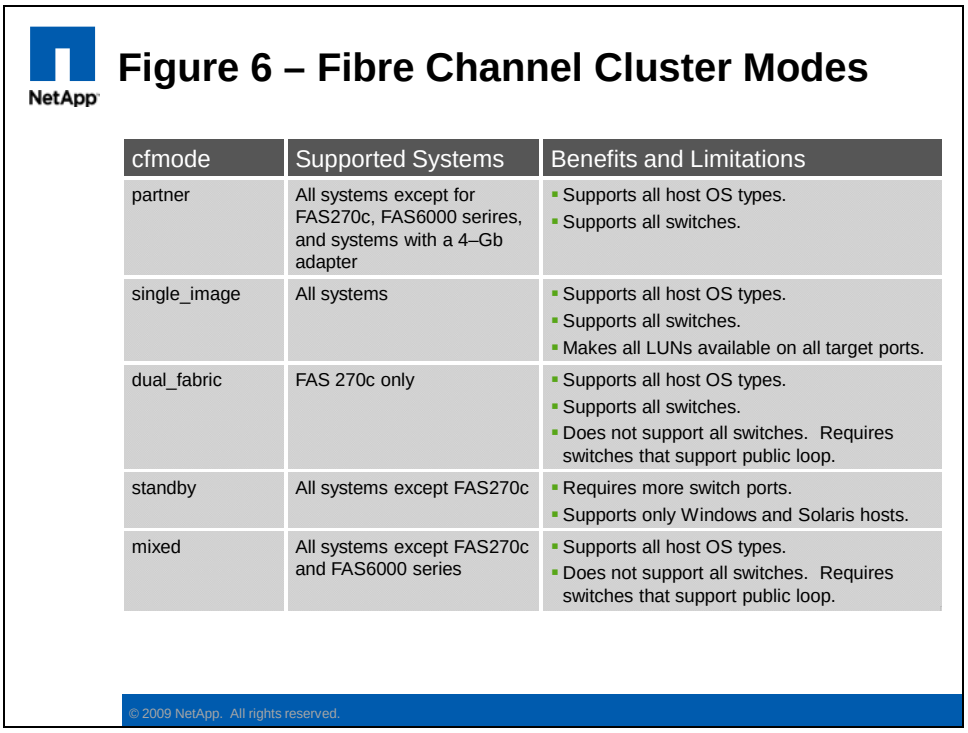


Figure 6 – Fibre Channel Cluster Modes

cfmode	Supported Systems	Benefits and Limitations
partner	All systems except for FAS270c, FAS6000 serires, and systems with a 4–Gb adapter	▪ Supports all host OS types. ▪ Supports all switches.
single_image	All systems	▪ Supports all host OS types. ▪ Supports all switches. ▪ Makes all LUNs available on all target ports.
dual_fabric	FAS 270c only	▪ Supports all host OS types. ▪ Supports all switches. ▪ Does not support all switches. Requires switches that support public loop.
standby	All systems except FAS270c	▪ Requires more switch ports. ▪ Supports only Windows and Solaris hosts.
mixed	All systems except FAS270c and FAS6000 series	▪ Supports all host OS types. ▪ Does not support all switches. Requires switches that support public loop.

NOTE: The FC clustering mode is an FCP-specific concept. The iSCSI protocol handles cluster failover in a completely different manner.

Conceptually, FCP and iSCSI are very similar, although they vary greatly in detail and implementation (for example, **FCP** is a *wire-level* protocol, whereas **iSCSI** travels on top of a *TCP* connection). The end result is the same though; they both provide block-level services to a host so that it can access a LUN.

Both FCP and iSCSI are licensed protocols, the only difference being that the iSCSI license is provided for no charge with the every FAS controller. You need to use the **`license add <licnum>`** command to license a feature (such as FCP or iSCSI) on the controller.

Once the SAN protocols have been licensed they need to be started before any further configuration or host access can occur. Use the following commands to start each of the SAN protocols:

- **FCP**
  - `fcp start`, to start the protocol
  - `fcp status`, to check the status of the protocol
- **iSCSI**
  - `iscsi start`, to start the protocol
  - `iscsi status`, to check the status of the protocol

Another difference between the FCP and iSCSI protocols is that the FCP protocol relies on having dedicated target mode FC ports on specialized FC HBA adapters. In comparison, the iSCSI protocol can use either a specialized iSCSI HBA adapter or any standard Ethernet port. If using a standard Ethernet port then the controller is using the **iSCSI software target (known as ISWT)**.

If you are using the ISWT support in an Active/Active controller configuration, then you may notice two such devices:

- **ISWTa** = for the local controller
- **ISWTb** = for the partner controller, used for cluster failover

The SAN protocols provide block-level access to LUNs on the storage controller. You can create these LUNs using various tools, as follows:

- **FilerView**, the storage controller's web-based GUI
- **CLI**, using either of the following two methods
  - To create each item manually, run the **`lun create + igroup create + lun map`** commands
  - Or, to create everything from a wizard, run the **`lun setup`** script and follow the prompts. When using the wizard there is no need to manually create the igroups or do the LUN mappings.
- **SnapDrive**, which allows you to manage storage from the host

NOTE: When you create a LUN you will need to know certain facts, such as the location (path) of the LUN, the OS of the respective host, the capacity required, and the LUN id.

The LUN id is the means by which a host to identifies a LUN, and also how it distinguishes between multiple LUNs. As such all LUNs presented to one host must be unique for that host, but of course each host has its own LUNs and LUN ids, which are completely independent. In short:

- Multiple LUNs to the same host must have unique LUN ids
- Each host can have its own ids (they don't conflict between hosts)

When creating a LUN with the `lun setup` wizard a number of separate commands are rolled up into the script. However, if you choose to create a LUN manually, then you will next need to run two more commands before it is potentially accessible from a host. To filter which LUNs are visible to which hosts (sometimes called **"LUN masking"**) you need to use the **`igroup create`** and **`lun map`** commands. For example:

- **`igroup create`**
  - This defines a new relationship between a host (WWPN), the OS type, and whether this is an FCP or iSCSI connection
  - `igroup create -f -t windows <ig_name> <wwpn>`
- **`lun map`**
  - This defines a new relationship between a LUN, the igroup and set the LUN id
  - `lun map </path/lun> <ig_name> <lun_id>`

NOTE: Assuming that the SAN zoning is correct, the host should now be able to rescan for, and connect to, the new LUN.

You need to be aware of the **minimum and maximum LUN sizes** supported by each of the host operating systems. Refer to the table below:

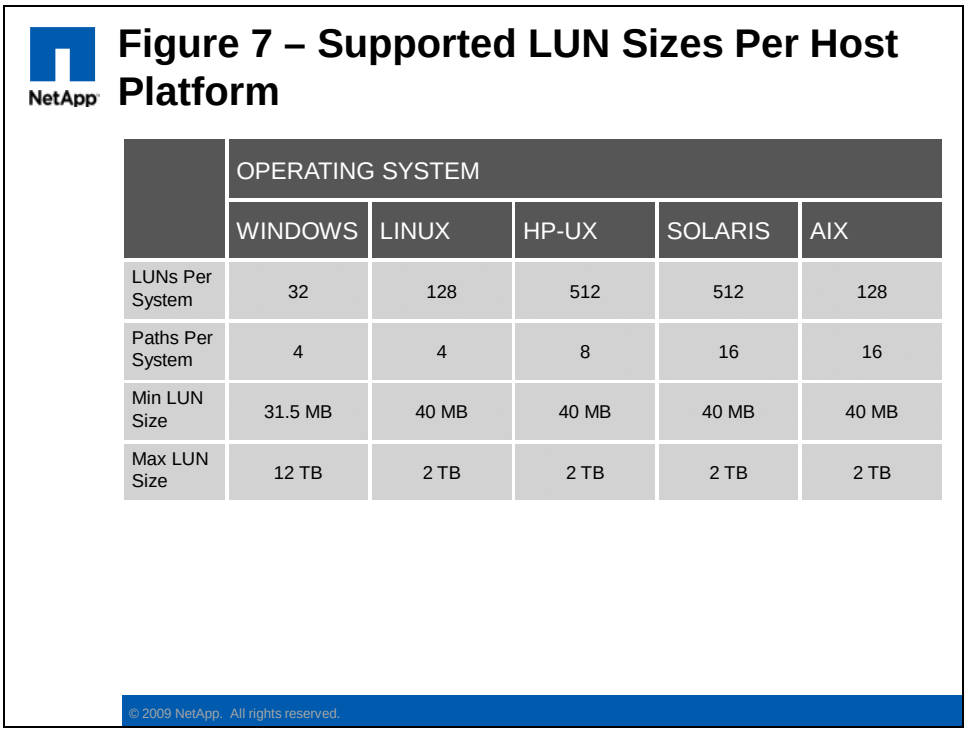


Figure 7 – Supported LUN Sizes Per Host Platform

	OPERATING SYSTEM				
	WINDOWS	LINUX	HP-UX	SOLARIS	AIX
LUNs Per System	32	128	512	512	128
Paths Per System	4	4	8	16	16
Min LUN Size	31.5 MB	40 MB	40 MB	40 MB	40 MB
Max LUN Size	12 TB	2 TB	2 TB	2 TB	2 TB

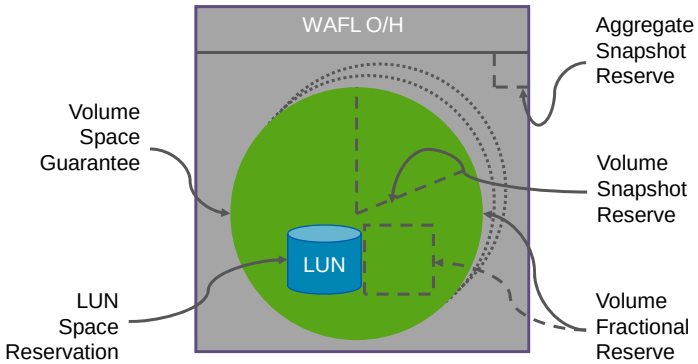
NOTE: Not all supported OS platforms are shown here as this is an extract from the SAN Boot Camp training material (and may not be up to date). Refer to the NetApp and host platform documentation if you require further detail.

How a LUN consumes capacity on its parent volume, and how this is affected by creating snapshots is a complex topic. If configured incorrectly, it is possible for the host to think it has free capacity for writing data, while in reality the volume containing the LUN has been

totally consumed with snapshot data. At this point the LUN would go offline and manual rectification is required. This is obviously a situation we want to avoid, and there are several ways to prevent this problem from occurring, for example:

- **Space Reservation**
  - Until recently this was the recommended method to guarantee space for writes to a LUN (regardless of the number of snapshots)
  - Some of the parameters to be configured for this method are **Volume Fractional Reserve** to **100%** and **LUN Space Reservation** to **Yes**
    - `vol options <vol> fractional_reserve 100` (100% by default)
    - `lun create` (LSR is on by default)
    - `lun set reservation` (to change an existing LUN)
  - This caused an amount of space equal to the size of the LUN (100%) to be excluded from potential snapshots, thus guaranteeing that writable capacity would always be available for the host
  - This is sometimes referred to the "Two times plus Delta" (2X+Δ) overhead

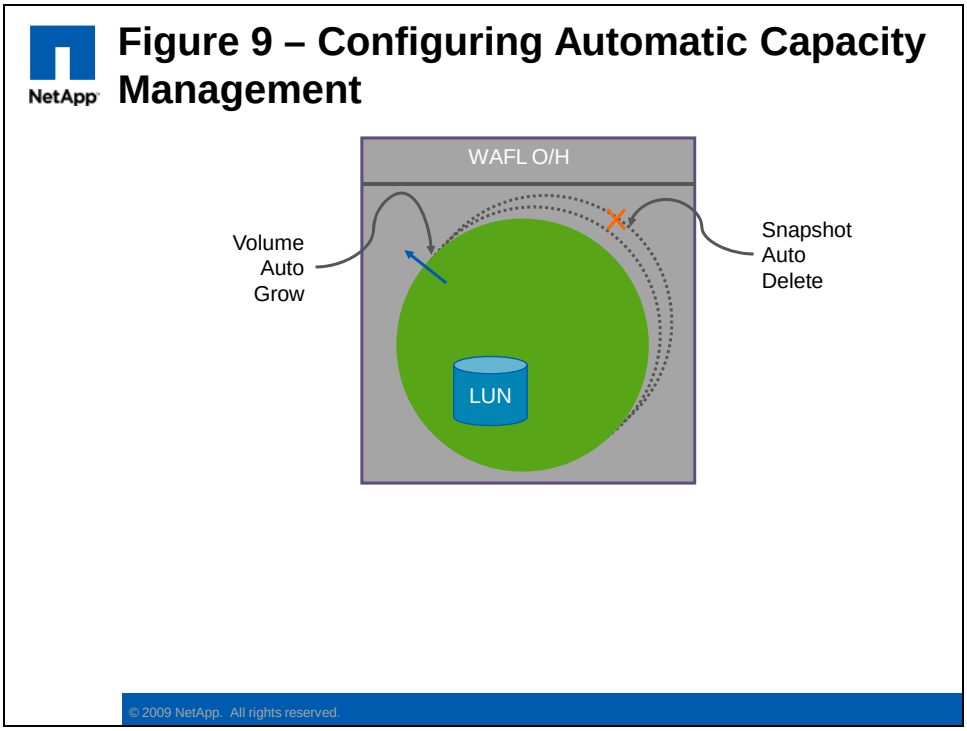
Figure 8 – Configuring Volume and LUN Provisioning



NOTE: Some documentation may still refer to these superseded best practices.

- **Volume AutoGrow** and **SnapShot AutoDelete**
  - In the last few years two new automatic utilization thresholds have been introduced, allowing the best practices to be changed away from 100% fractional reserve
  - Some of the parameters to be configured for this method are **Volume AutoGrow** and **SnapShot AutoDelete**. The **Volume Fractional Reserve** can then safely be set to **0%**
    - `vol autosize <vol-name> on` (off by default)
    - `snap autodelete <vol-name> on` (off by default)
    - `vol options <vol> fractional_reserve 0`

- This sets a utilization threshold at which the containing volume can automatically grow, and/or at which certain existing snapshots will be deleted. This ensures that space is always available for the host to write to the LUN
- This changes the capacity required to "One times plus Delta" (1X+Δ) and potentially even better with thin provisioning



NOTE: Refer to the product documentation if you require more information about Volume and LUN space management.

ADMINISTRATION

As well as creating a LUN, the **`lun`** command has many other capabilities. For example:

- **`lun offline`** makes a LUN unavailable for host access
- **`lun move`** relocates a LUN to a different path in the same volume
- **`lun show`** displays various types of information about the LUNs
  - To show the LUN to igroup mappings: `lun show -m`
  - To show the LUNs OS platform types: `lun show -v`
- **`lun clone`** instantly creates a new r/w LUN as a clone of an existing LUN.
  - The new LUN is thin provisioned (no space is consumed until new data is written) and the two LUNs share blocks with a snapshot of the original LUN (known as the "backing" snapshot)
  - `lun create -b /vol/vol1/.snapshot/lun1 /vol/vol1/lun2`
- **`lun clone split`**, split a LUN clone from its backing snapshot
  - Since a LUN clone locks the backing snapshot (that is, it cannot be deleted), you should split the relationship for long term use

- `lun clone split start /vol/vol1/lun2`

NOTE: Refer to the product documentation if you require a detailed description of the many `lun` command options.

When it comes to creating a Snapshot of a LUN, it is important to remember that the Snapshot is created at the volume level. So all data in the volume is captured in the Snapshot, which could contain multiple LUNs or NAS data too. It is best practice to include only a single (or related) LUNs in each volume, so as to simplify the Snapshot process.

Another complexity is that LUNs contain a file system that is managed by the host, and probably also contain a database that is managed by an application on the host as well. Only the host and the application can ensure the consistency of the file system and database at Snapshot creation time.

Therefore it is usually necessary to coordinate with the relevant host during the Snapshot backup of a LUN. Usually this process can occur with no disruption to service. For example:

- **On the host**
  - Quiesce the application/database
  - Flush the host's file system buffers to disk (LUN)
- **On the controller**
  - Create the Snapshot of the LUN
  - `snap create /vol/vol1 <snapshot_name>`
    - This now contains a consistent image of the host's file system and application database in an idle or offline state
- **On the host**
  - Unquiesce the application/database

NOTE: The NetApp host attachment kit includes a utility to flush the host's file system buffers to disk.

You will also need to consider the **free capacity** in the containing volume when creating Snapshot backups of LUNs. For example, assume that you have configured a 400GB volume, and then created a 300GB LUN inside that volume. If you completely fill the LUN with data, then any subsequent attempt to create a Snapshot will fail. This happens because there would be insufficient free capacity in the volume to allow the host to continue to write to the LUN if the Snapshot had succeeded.

Refer to the section titled *'How space management works'* in the *'Data ONTAP 7.3 Storage Management Guide'* for more detail on the various capacity management methods that are relevant to Snapshot backup of volumes/LUNs.

If it is necessary to restore from a Snapshot backup of a LUN, then you have two methods available. For example:

- **Volume SnapRestore**
  - This will restore the entire volume and LUN
  - The LUN must be offline or unmapped before proceeding

- **LUN clone**
  - In this method you create a clone of the LUN, and then mount the clone; you may then copy individual files back to the original LUN

NOTE: If this was a Snapshot of NAS data (that is, not a LUN) then you could simply browse to the *.snapshot* directory and copy the individual files back to the active file system (or do a *Single File SnapRestore* for a very large file).

Perhaps the best way to manage (and Snapshot) SAN storage is to use NetApp's host and application integration tools. For each supported host OS platform there is a **SnapDrive** package, and for a number of popular business applications there is a corresponding **SnapManager** package. These tools provide a easy to use (GUI and CLI) and highly functional interface for managing the storage controller. For example:

- **SnapDrive**
  - Create a new LUN
  - Connect to an existing LUN
  - Trigger a new consistent file system Snapshot
  - Restore a Snapshot backup
  - Clone an existing LUN
  - Manage iSCSI connections
- **SnapManager**
  - Trigger a new consistent application Snapshot
  - Manage the retention of Snapshot backups
  - Restore an application Snapshot backup
  - Clone an existing application/database (only for certain apps)
  - Migrate existing application data onto LUNs

NOTE: SnapShot application integration is also possible without SnapManager support, though you will need to create any required automation scripts yourself. Many examples of these are available on the NOW site for download.

PERFORMANCE

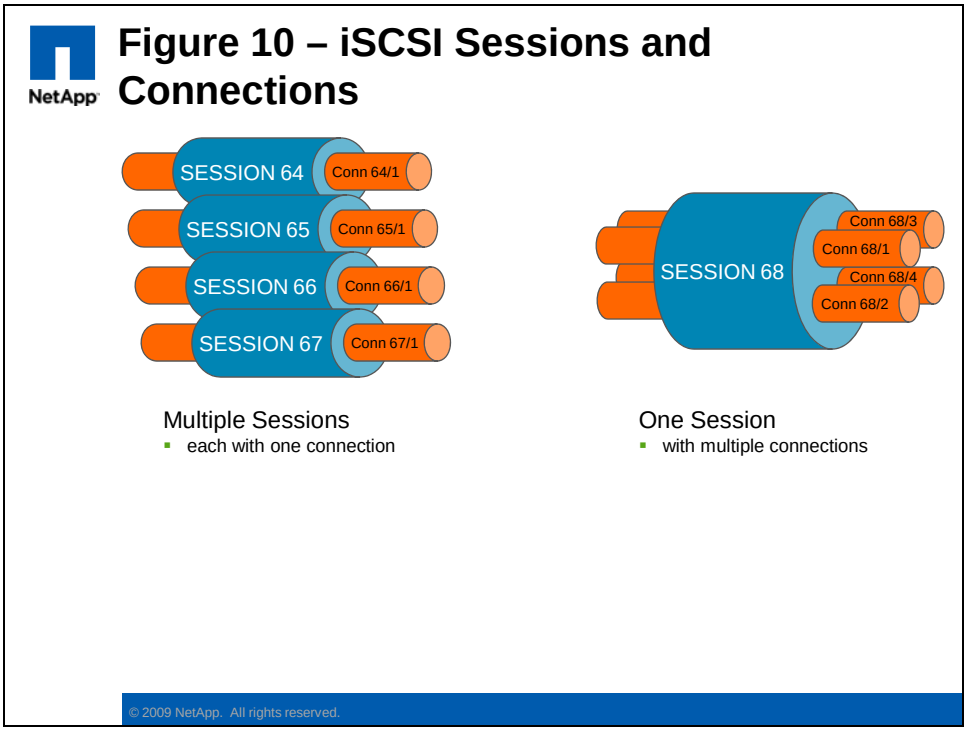
The performance of the SAN is dependent on the available physical resources and the host, switch, and controller configurations. For example:

- **Controller**
  - Model (versus expected performance level)
  - Number of spindles supporting the LUNs
  - Network speed
  - Number of ports/paths
  - Balanced workload between active/active controllers
  - Background tasks (Replication, RAID scrub, RAID rebuild, and so on)
- **Switch**
  - Network speed
  - Port oversubscription
  - Switch workload
- **Host**
  - Workload type (random versus sequential, read versus write)
  - Network speed

- Multipathing configuration (active/active or failover only)
- Number of paths
- HBA tuning
- Correct partition alignment

As all iSCSI traffic is carried over the TCP protocol, you can list these **connections** (an iSCSI initiator-to-target relationship) and **sessions** (TCP) separately. The connections-to-sessions relationship can be configured in two modes.

- **Single connection per session (1:1)**
  - This mode creates one iSCSI connection per TCP session
  - Supported by the Microsoft MPIO and NetApp DSM for Windows
- **Multiple connections per session (MCS)**
  - This mode creates multiple iSCSI connections per TCP session
  - This provides a "native" iSCSI multipathing capability



The commands to list the iSCSI connections and sessions are:

- `iscsi session show`
- `iscsi connection show -v`

NOTE: The advantages and disadvantages of either multipathing method are beyond the scope of this document.

SECURITY

The security of the SAN is enforced at several levels, some on the controller, some in the fabric, and some in conjunction with the hosts. For example:

- **FCP**
  - LUN masking (igroups)
  - Port masking (portsets)
  - SAN zoning
- **iSCSI**
  - LUN masking (igroups)
  - Port masking (iscsi accesslist)
  - Ethernet VLANs
  - Passwords (iscsi security)
  - Encryption (ipsec)

In a Fibre Channel SAN the ability of the various devices to discover each other and communicate across the fabric is controlled by the zone definitions. Only devices that reside in the same zone can communicate. There are two main types of SAN zones that we are concerned with, and they are:

- **Soft zones**
  - Usually defined using the WWPN of the host's and controller's FC ports; allows communication between the device's FC ports
  - Blocks inter-zone traffic by filtering device discovery at the fabric name service, but does NOT *explicitly* block traffic
- **Hard zones**
  - Usually defined using the physical port numbers of the FC switch; allows communication between those physical switch ports
  - Blocks inter-zone traffic by filtering device discovery at the fabric name service, and by *preventing* traffic between switch ports

Some good recommendations for designing FC SAN zones are:

- Decide on a naming convention, and stick to it
- Keep disk and tape devices in separate zones
- Define separate zones for every required initiator/target pairing

An additional concern with an iSCSI SAN is that the block-level traffic might be traveling over the same physical network as other less sensitive data. This potentially exposes the iSCSI traffic to network snooping or other attacks.

The NetApp FAS controllers support port masking, bidirectional password access, and network encryption of the iSCSI traffic. For example:

- **iSCSI port masking**
  - This allows you to restrict which network adapters expose an iSCSI initiator group to the hosts
  - `iscsi interface accesslist remove ig0 e0a`
- **iSCSI password**

  - This requires the iSCSI host to respond to a password challenge before gaining access to the LUN. It can also require the controller to answer a password challenge from the host.
  - `iscsi security add -i <igroup_name> -s CHAP -p <inpassword> -n <inname> [ -o <outpassword> -m <outname> ]`

- **IPSec encryption**
  - This enables encryption of the ethernet network traffic. It is not an iSCSI specific setting and will encrypt any Ethernet traffic (but the client will need to know how to decrypt the traffic).
  - `options ip.ipsec.enable on` and `ipsec policy add`

NOTE: Network encryption, although effective, may impose a significant performance impact on (what should be) a high performance iSCSI solution. Therefore many sites choose not to encrypt iSCSI traffic and instead deploy a separate network infrastructure or VLANs to isolate the iSCSI traffic.

TROUBLESHOOTING

Troubleshooting a SAN problem requires knowledge of the host, the SAN switches and network infrastructure, and the storage controller itself. Much of this is outside the scope of this document. Here are some example steps for troubleshooting a simple LUN access problem:

- **On the controller**
  - Can the storage controller see the host's FC adapters?
    - `fcp show initiator`
  - Is the LUN being presented to the host?
    - `lun map –v`
    - `igroup show`
  - Is there a problem with iSCSI password access?
    - `iscsi security show`

- **On the SAN switches**
  - Are the host's FC initiator ports and the controller's FC target ports in the same zone?
    - Run the `zoneshow` command (Brocade)
    - Run the `show zone` command (Cisco MDS)

- **On the host** (for example, Solaris 9)
  - Is the host configured to detect the LUN?
    - Is the LUN id in the `/kernel/drv/sd.conf` file?
  - Has the host rescanned to detect the LUN?
    - Run the `devfsadm` command

NOTE: Refer to each product's relevant documentation for further troubleshooting recommendations.

CIFS

The Common Internet File System (CIFS) is the default NAS protocol included with Microsoft Windows. The NetApp storage controller can participate in an Active Directory domain and present files via the CIFS protocol.

CONFIGURATION

The CIFS protocol is a licensed feature and needs to be enabled before it can be configured and used to present files for client access. For example:

- **`license add <licnum>`**

NOTE: If the controller was ordered new with the CIFS license then it will already be installed, and the CIFS setup wizard will start automatically at the for system boot.

The easiest way to configure the CIFS protocol is to run the **CIFS setup wizard**. The wizard prompt you for all aspects of the CIFS configuration, such as the NetBIOS hostname, authentication mode (for example, AD), and local administrator password. To start the wizard, run the following command:

- **`cifs setup`**
  - The choices for CIFS user authentication are:
    - Active Directory
    - NT Domain
    - Windows Workgroup
    - Non-Windows workgroup

NOTE: If you wish to configure Active Directory mode for user authentication, then the system time on the storage controller must be within five minutes of the AD server. This requirement is inherited from the Kerberos protocol, which AD uses for improved security. You should configure both systems to use the same network time server, if one is available.

As well as using an external authentication system such as Active Directory, you can also define local users and groups. You should define a local administrator for use in case the AD server is unavailable.

It is also possible to add Domain users (such as a Domain Admin) into local groups. This can be useful when managing the storage controller as part of a larger enterprise. For example, to add the Domain user "Steven" into the local "administrators" group, run the following command:

- **`Useradmin domainuser add steven -g Administrators`**

Refer to the section titled *'Managing local users and groups'* in the *'Data ONTAP 7.3 File Access and Protocols Management Guide'* for more information on how to manage local users and groups.

ADMINISTRATION

To create and manage CIFS shares you need to use the appropriately named **`cifs`** command, followed by the **`shares`** parameter. Here are some examples of its use:

- **List the existing shares**
  - `cifs shares`
- **Create a new share**
  - `cifs shares –add <sharename> /vol/vol1`
- **Change an existing share**
  - `cifs shares –change <sharename> -comment`

With the CIFS protocol you can create **shares** to expose the following object types for user access:

- **Volume**
  - `/vol/vol1`
- **Qtree**
  - `/vol/vol1/qtree1`
- **Directory**
  - `/vol/vol1/dir1`

It is sometimes desirable to set a **quota** on the amount of disk space that an individual user or group can consume via the CIFS (or NFS) protocol. There are several types of quotas and options on how to enforce them. For example:

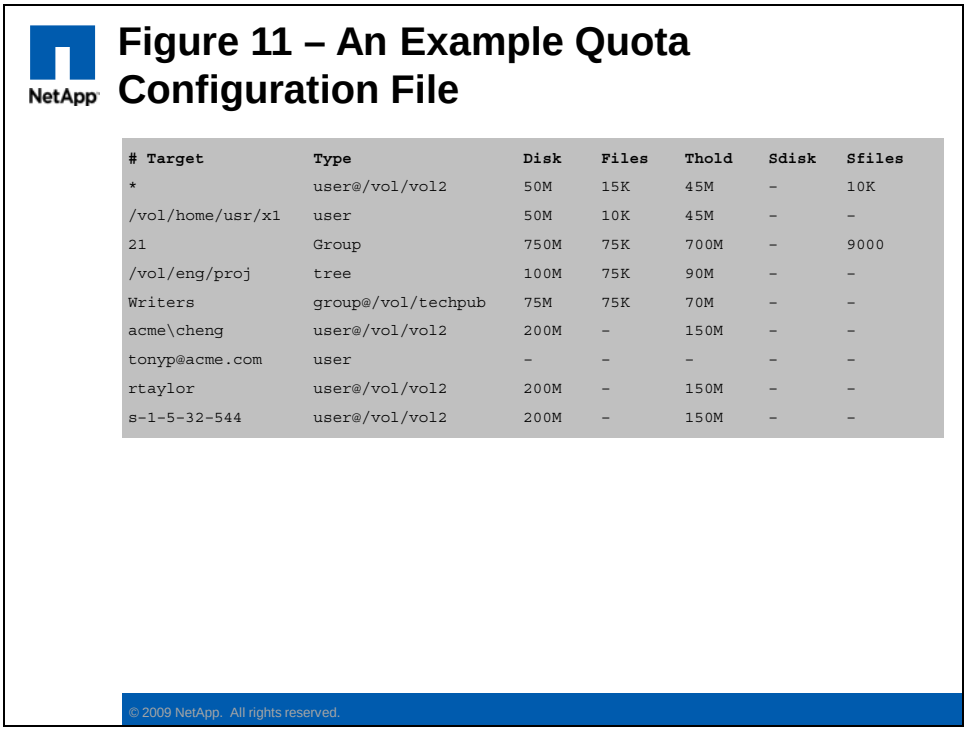
- **Quota targets**
  - User
    - Controls how much data a specific user can store
  - Group
    - Controls how much data a specific group of users can store
  - Qtree
    - Controls how much data that can be stored in a specific qtree (similar to a directory)
    - **NOTE:** The *Administrator* and *root* users are exempt from the other quota types, but not from the qtree quota type
  - Default
    - This is a special quota that applies only if a specific quota has not been defined for a user or group.
- **Quota objects**
  - Capacity
    - Sets a limit on the maximum disk space used
  - Files
    - Sets a limit on the maximum number of files stored
- **Quota thresholds**
  - Soft

        - When this limit is breached the controller will only send an SNMP trap;the user's write will succeed
    - Hard
        - When this limit is reached the controller will prevent the users attempt to write more data; the client will display a *file system full* error

The quotas are configured and managed through either the **`quota`** command or the **FilerView > Volumes > Quotas** web interface. Both methods store the quota configuration in the **`/etc/quotas`** file on the controller's root volume. For example:

- **List the active quotas**
    - `quota status`
- **Enable quotas**
    - `quota on <volume_name>`
    - Setting a new quota requires a file system scan. This can take some time for a large file system.
- **Resizing a quota**
    - `quota resize <volume_name>`
    - If you have modified the limits on an existing quota there is no need to do a file system scan. Note this is only valid for existing quotas.
- **List usage**
    - `quota report`
    - Prints the current file and space consumption for each user or group with a quota and for each qtree.

Here is an example of an `/etc/quotas` configuration file:



NOTE: The second line (starting with "*") is a default quota.

PERFORMANCE

The FAS storage controller has a large number of performance counters, statistics, and other metrics. There are a number of CIFS specific performance analysis commands available. For example:

- **`cifs stat`**
  - Displays CIFS performance statistics
- **`cifs top`**
  - Displays the most active CIFS clients based on various criteria

NOTE: The CIFS statistics displayed are cumulative for all clients by default. You need to enable the **`cifs.per_client_stats.enable`** option in order to collect statistics for individual clients.

Refer to the relevant sections of the *'Data ONTAP® 7.3 Commands: Manual Page Reference'* for more detail on the non-CIFS specific controller performance commands (for example, **`sysstat, stats, statit`**).

Changing some CIFS performance parameters is **disruptive** and can only be done when no clients are connected. Here is an example process:

1. **`cifs terminate`**
   - This halts the CIFS service and disconnects all clients
2. **`options cifs.neg_buf_size <value>`**
   - This changes the buffer size

3. cifs restart

- This restarts the CIFS service (the clients can now reconnect)

One easy way to potentially improve CIFS performance is to correctly set the **unicode** settings on the volume containing the shared files. For example

- **vol options <volume_name> convert_ucode on**
- **vol options <volume_name> create_ucode on**

This aids performance because the CIFS protocol always uses unicode, and if the volume is not configured for unicode support, then the controller has to continuously convert between the two modes.

SECURITY

To manage user access to the CIFS shares you also need to use the **`cifs command`**, but this time followed by the **`access`** parameter. Here are some examples of its use:

- List the existing shares
  - `cifs access <share> <user> <access rights>`
  - The list of share-level access rights are:
    - No Access
    - Read
    - Change
    - Full Control

User access to a share (and the shared files) is evaluated against the user's Security Id (SID), which is usually provided at login by the AD server. This is a two-step process, for example:

- Evaluate the user's SID versus Share permissions
  - Access is either granted or denied (to the share)
- Evaluate the user's SID versus File or Directory permissions
  - Access is either granted or denied (on a file-by-file basis)

The CIFS protocol enforces mandatory file locking if requested by the client.

TROUBLESHOOTING

CIFS is a complex protocol, using several TCP ports, and interacting with various external systems for user authentication, Kerberos security (AD), and hostname resolution. As such it has numerous potential points of failure, but is normally very reliable in operation.

A full description of CIFS troubleshooting is outside the scope of this document, and probably unnecessary at this time. Refer to the NCDA NAS Boot Camp training material if you require more information.

If you suspect connectivity problems are the source of your CIFS problems, you should try the following commands:

- **`ping`**
  - A standard TCP connection test
- **`testdc`**
  - Test communication with the Active Directory server
- **`ifstat`**
  - Display a low level view of network interface performance data
- **`netdiag`**
  - Analyzes the network protocol statistics to ensure correct operation and displays suggested remedial actions if required

Another potential issue is file access permissions, where the access protocol is CIFS, but the containing volume/qtree has a UNIX security style. This is called *Multiprotocol* access, and is covered in the next section.

MULTIPROTOCOL

The NetApp storage controller can simultaneously present files through both the CIFS and NFS protocols. It includes sophisticated mappings between Windows and UNIX usernames and file system permissions to make this a seamless operation.

CONFIGURATION

The only requirement for multiprotocol access is to configure both CIFS and NFS access to the same file system. Of course, some complexity will arise due to the different security semantics between Windows and UNIX systems, but that is unavoidable.

Certain security settings and username mappings do need to be configured, but they are covered in the following Security section.

ADMINISTRATION

The CIFS and NFS protocols are managed separately, even when they refer to the same file system, and should not introduce any new administration concerns. Refer to the administration sections for CIFS and NFS for more information.

PERFORMANCE

Multiprotocol access should not introduce any new performance concerns. Refer to the performance sections for CIFS and NFS for more information.

SECURITY

The **default security style** for all new volumes is controlled by a WAFL option. For example:

- options **`wafl.default_security_style <value>`**
  - Where **`<value> = "unix"`**
    - Every file and directory will have UNIX (UGO) permissions
  - Where **`<value> = "ntfs"`**
    - Every file and directory will have NTFS (ACL) permissions
  - Where **`<value> = "mixed"`**
    - Each file and directory can have either UNIX (UGO) or NTFS (ACL) permissions (but not both at the same time)

NOTE: You should only use the mixed mode security style if you have a particular requirement. It can make troubleshooting file access problems very complicated.

To manually set or view the security style for a specific volume or qtree, you would use the following commands:

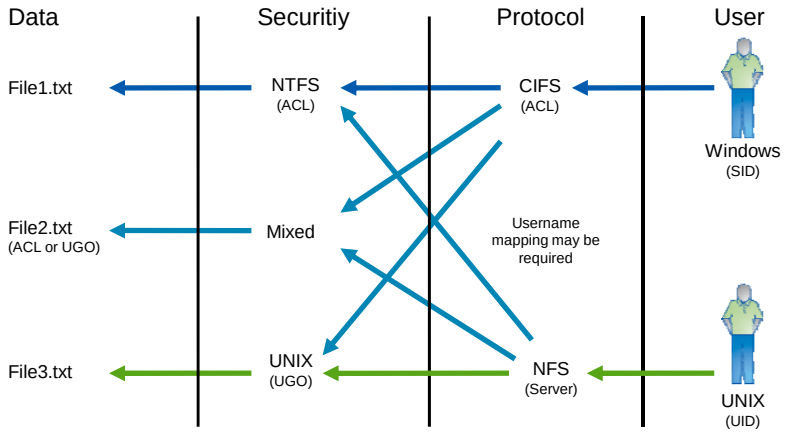
- **`qtree status`**
  - Display the list of volumes and qtrees and their security styles
- **`qtree security <path> [ unix | ntfs | mixed ]`**
  - Set the security mode for a nominated volume or qtree

Although the security style settings control the underlying file system security type, access via CIFS or NFS to the file data is still controlled by the normal user authorization process. The added complexity in a multiprotocol environment is due to having to map between the security semantics of the users, the shares/exports, and the file system. For example:

- Evaluate the user's SID/UID versus Share/Export permissions
  - Access is either granted or denied (to the share/export)
- Evaluate the user's SID/UID versus File or Directory permissions
  - If they are of a different type (for example, Windows/UNIX) then it may be necessary to map the username into the correct type
  - Access is either granted or denied (on a file-by-file basis)

The following diagram identifies where the **username mapping** may need to occur.

Figure 12 – Multiprotocol Access and Username Mapping



The username mapping is defined in the **`/etc/usermap.cfg`** file. This is a simple text file in the root volume of the storage controller. The process of mapping usernames between Windows and UNIX context is reasonably straightforward, as shown below:

- **Automatic**
  - If the Windows and UNIX usernames match
- **Specified** ("win_user = unix_user")
  - If defined in `/etc/username.cfg`
- **Default**
  - If the usernames are different, and there is no specific mapping, attempt to use the defined default UNIX or Windows username (if any)

- `options wafl.default_nt_user <username>`
- `options wafl.default_unix_user <username>`

TROUBLESHOOTING

The majority of problems with multiprotocol access are caused by incorrect security styles and/or incorrect username mappings.

To identify how the username mappings are being resolved, issue the following command:

- `wcc –u <unix_user>`
  - This will display the UNIX to Windows username mapping
- `wcc –s <windows_user>`
  - This will display the Windows to UNIX username mapping
  - `Domain\Administrator => root`

If you are connecting as the **Administrator** or **root** user to a volume with a foreign security style, sometimes the easiest way to overcome any access problems is to set the following options:

- `options wafl.nt_admin_priv_map_to_root on`
- `options cifs.nfs_root_ignore_acl on`

These two options grant the Administrator/root users equivalent privileges on the foreign volume.

In some cases, due to the different user populations, and different file locking abilities in CIFS and NFS, you might need to debug a file access problem (for example, an NFS client can't open the file because it is locked by a CIFS client). You can list which clients have an active CIFS connection with the following command:

- **`cifs sessions`**

Another possible issue with mixed NFS and CIFS access is that NFS clients support **symbolic links** in the file system, while CIFS clients generally do not. However if you set the **`cifs.symlinks.enable`** option to **`on`** (which is the default value) then the CIFS clients will successfully resolve any symbolic links that were created by NFS clients.

NOTE: Refer to the product documentation if you need more detail on the interaction of CIFS clients with the various types of symbolic links.

And for the simplest access problem of all: If you want multiprotocol access to work then you must remember to create both a **CIFS share** and an **NFS export.**

NFS

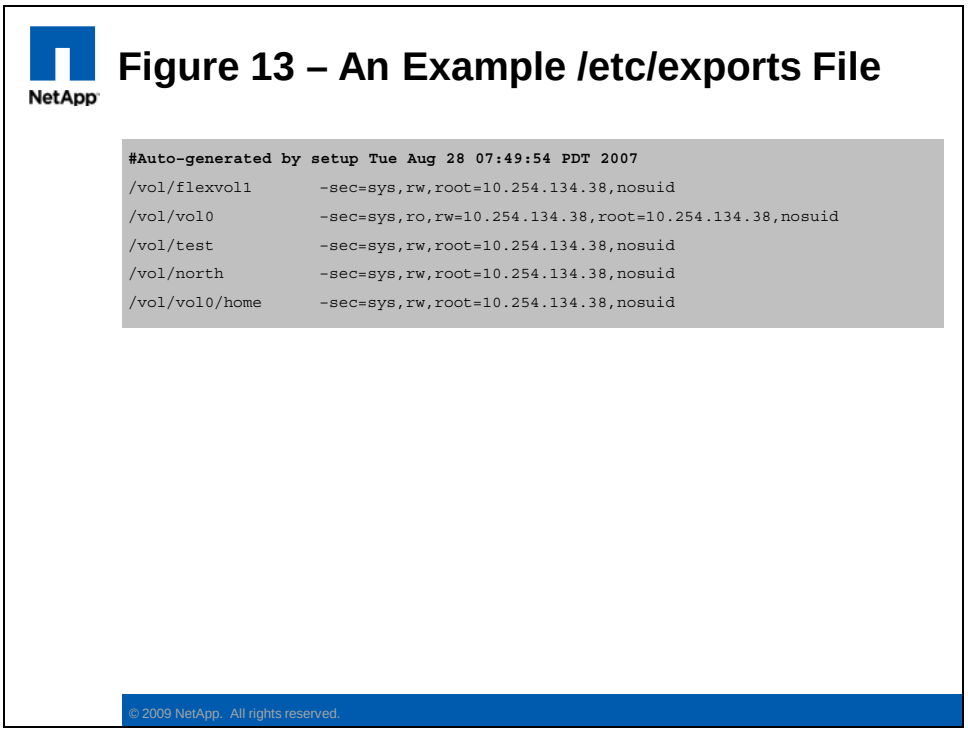
The Network File System (NFS) is the default NAS protocol included with all UNIX platforms. The NetApp storage controller can present files through the NFS protocol, and can also participate in a Kerberos domain.

CONFIGURATION

The NFS protocol is a licensed feature and needs to be enabled before it can be configured and used to present files for NFS client access. For example:

- **`license add <licnum>`**

The NFS server configuration is based around the **`/etc/exports`** file. This file defines all of the NFS exports, who can access them, and with what privilege level (for example, read-write or read-only). For example:



NOTE: Any volume (or qtree, and so on) that is to be exported will need an entry in this configuration file.

The `/etc/exports` file contains three types of information:

- **Resource List**
  - Exports define what resources are available to which NFS clients
  - `/vol/flexvol1` defines a resource to be exported
- **Identification**

  - Hostnames, DNS subdomains, IP addresses, IP subnets, and so on, can all be used to identify who the NFS clients are
    - `root=10.254.134.38` defines an NFS client

- **Authorization**
  - Exports define access permissions to the NFS clients
  - `rw` and `nosuid` define the access permissions

ADMINISTRATION

The NFS server administration is based around the appropriately named `exportfs` command, and the `/etc/exports` file. Here are some examples of their use:

- **List the current exports** (in memory)
  - `exportfs`
- **List the persistent exports** (available after a reboot)
  - `rdfile /etc/exports`
- **Create a new export** (in memory)
  - `exportfs –i –o rw=host1 /vol/vol1`
- **Create a new persistent export** (available after a reboot)
  - `wrfile –a /etc/exports /vol/vol1 –rw=host1`
  - `exportfs -a`

With the NFS protocol you can create "**exports**" to expose the following object types for user access:

- **Volume**
  - `/vol/vol1`
- **Qtree**
  - `/vol/vol1/qtree1`
- **Directory**
  - `/vol/vol1/dir1`
- **File**
  - `/vol/vol1/file1.iso`

NOTE: Unlike most UNIX variants, the FAS controller can successfully export nested directories (also known as **ancestors** and **descendants**). For example, `/vol/vol1` and `/vol/vol1/qtree1` could both be exported, and the NFS client would need to satisfy the access controls associated with the mount point it initially accessed.

Refer to the CIFS configuration section for a description of file system **quotas**.

PERFORMANCE

The FAS storage controller has a large number of performance counters, statistics, and other metrics. There are a number of NFS specific performance analysis commands available. For example:

- **`nfsstat`**
  - Displays NFS performance statistics
- **`nfs_hist`**
  - This is an Advanced mode command that displays NFS delay time distributions (that is, the number of I/O ops per millisecond grouping)
- **`netapp-top.pl`**
  - A Perl script, downloadable from the NOW website, that displays a list of the most active NFS clients. It is run on a client system.
    - http://now.netapp.com/NOW/download/tools/ntaptop/

NOTE: The NFS statistics displayed are cumulative for all clients by default. You need to enable the **`nfs.per_client_stats.`**enable option in order to collect statistics for individual clients.

Refer to the relevant sections of the '*Data ONTAP® 7.3 Commands: Manual Page Reference'* for more detail on the non-NFS specific controller performance commands (the example, `sysstat, stats, statit`).

Performance testing is a complex topic, and is certainly beyond the scope of this document. However, as a last resort you could run some very basic performance testing using the following proceedure (from the NFS client):

- **Write traffic**
  - Run the `time mkfile` command
- **Read traffic**
  - Run the `time dd` command
- **Read/Write traffic**
  - Run the `time cp` command

NOTE: If you are serious about repeatable performance testing then you should investigate tools such as *iometer, iozone, bonnie++*, and *Netapp's sio.*

SECURITY

Traditionally, security has been seen as a weak spot for the NFS protocol, but recent versions of NFS support very strong security. The traditional security model is called **AUTH_SYS**, while the newer model is called **Kerberos**. Here is a summary of the differences between the two security models:

- **AUTH_SYS**
  - User authentication is performed on the remote NFS client (which is typically a UNIX server). This implies that we trust the authentication process on the NFS client, and trust that the NFS client is not an impostor

  - No additional authentication, data integrity, or data encryption is performed

- **Kerberos**
  - User authentication is still performed on the remote NFS client, but Kerberos also authenticates that the NFS client is genuine. Kerberos security for NFS comes in several varieties, which are:
  - **krb5**
    - Authentication occurs with each NFS request and response
  - **krb5i**
    - Same as krb5, but adds integrity checking to verify that requests and responses have not been tampered with
  - **krb5p**
    - Same as krb5i, but adds data encryption to each request and response

NOTE: If you wish to configure Kerberos mode for user authentication, then the system time on the storage controller must be within five minutes of the Kerberos server. This is a requirement of the Kerberos protocol. You should configure both systems to use the same network time server, if one is available.

Before a user can access an export, the NFS client (remote UNIX server) must be able to mount the export. After that, the user's access to the shared files in the export is evaluated against the user's UNIX user Id and group Id (UID/GID), which is usually provided at login time by the NFS client. This is a two-step process, for example:

- Evaluate the server's hostname versus Export permissions
  - Access is either granted or denied (to mount the export, r/w or r/o)
- Evaluate the user's UID/GID versus File or Directory permissions
  - Access is either granted or denied (on a file-by-file basis)

The NFSv2 and NFSv3 protocols use advisory file locking, while the NFSv4 protocol enforces mandatory file locking if requested by the client.

TROUBLESHOOTING

NFS is a complex protocol, using several TCP ports and interacting with various external systems for user authentication, Kerberos security, and hostname resolution. As such it has numerous potential points of failure, but is normally very reliable in operation.

A full description of NFS troubleshooting is outside the scope of this document. Refer to the NCDA NAS Boot Camp training material if you require more information.

If you suspect **RPC problems** are the source of your NFS problems, you could try the following process:

- Verify RCP is enabled
- Verify NFS daemons are running
- Verify that mount points exist

If you are having **problems mounting an NFS export**, you could try the following commands:

- `showmount –e`
  - Run from the NFS client, this command lists the available exports on the NFS server
- `nfsstat –d`
  - Run from the controller, this command displays low level statistics that are useful in debugging a mount problem

If you are experiencing "**stale NFS handle**" errors, you could try the following process:

- Check the `/etc/fstab` file on the host for errors
- Check connectivity between the two systems
- Use the `ping` command
- List the available exports on the NFS server
  - On the NFS client, run the `showmount –e` command
- Check the controller's `/etc/exports` file
- Check the controller's current exports in memory
  - Run the `exportfs` command

EXAM NS0-163 – DATA PROTECTION SOLUTIONS

As a NetApp Certified Data Management Administrator, you will have proven skills in performing in-depth support, administrative functions, and performance management for CIFS, NFS, and FCP for SCSI or iSCSI for TCP/IP protocols on a NetApp storage appliance running the Data ONTAP operating system in NFS and Windows (CIFS) multiprotocol environments. You will also be able to implement active-active controller configuration and SyncMirror to ensure continuous data availability and rapid recovery of data in the event of a disaster, and use the SnapMirror, SnapRestore, and SnapVault products to manage and protect mission-critical data.

SKILLS TESTED:

- Set up and maintain Snapshot™ copies
- Configure and administer SnapRestore
- Configure and administer Asynchronous SnapMirror
- Configure and administer Synchronous SnapMirror
- Configure and administer Open System SnapVault
- Configure and administer Operations Manager
- Configure and administer SnapLock®
- Analyze and resolve data protection problems
- Implement active-active controller configuration (including SyncMirror)

RECOMMENDED COURSES:

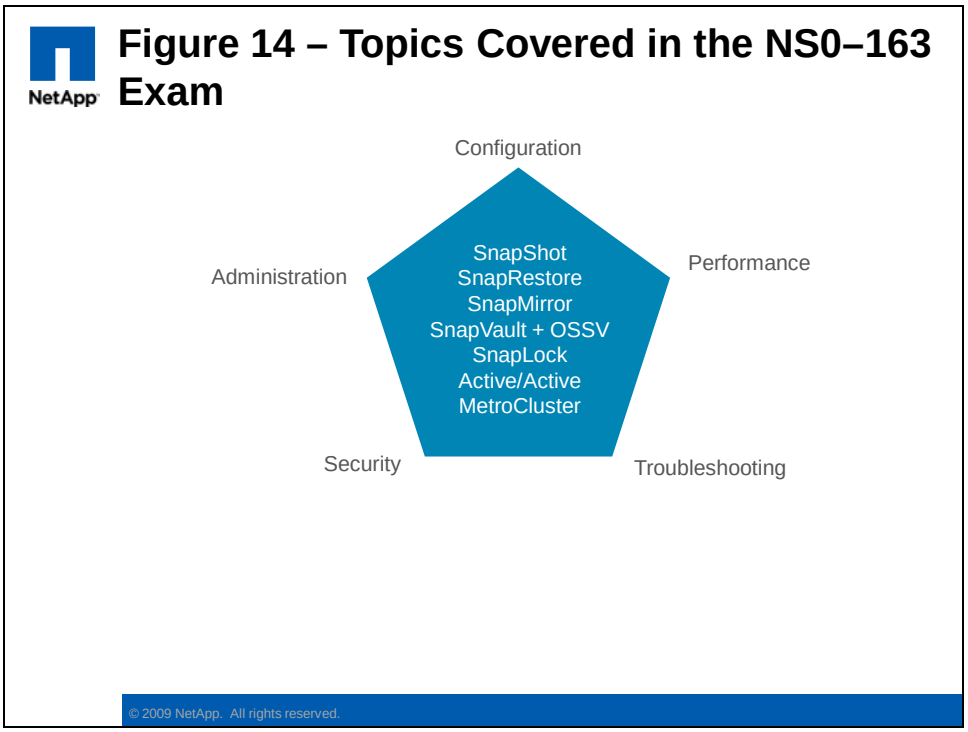
- ILT Course: Data ONTAP Fundamentals, Release 7.2
- ILT Course: Data ONTAP SAN Administration Basics, Release 7.2
- ILT Course: Data ONTAP CIFS Administration, Release 7.2
- ILT Course: Data ONTAP NFS Administration, Release 7.2
- WBT Course: Data ONTAP Fundamentals, Release 7.2

NOTE: ILT – Instructor-Led Training and WBT – Web-Based Training

EXAM PREPARATION

This section describes a number of NetApp FAS learning points that are relevant to the NS0-163 exam. However, it is not limited to just the exam topics and attempts to provide a brief summary of a range of NetApp technologies.

Figure 14 highlights the main subjects covered in this exam (white text) and the range of topics covered within each subject (black text).



A brief overview of the relevant training and other material is provided in the following sections.

SNAPSHOT

A snapshot copy is a read-only image of a volume, or an aggregate, that captures the state of the file system at a point in time. Many snapshot backups may be kept online, or vaulted to another system, and used for rapid data recovery if required.

CONFIGURATION

The snapshot capability of the FAS storage controller is a native capability provided by the WAFL file system layer. Both SAN and NAS data can be captured in a snapshot backup.

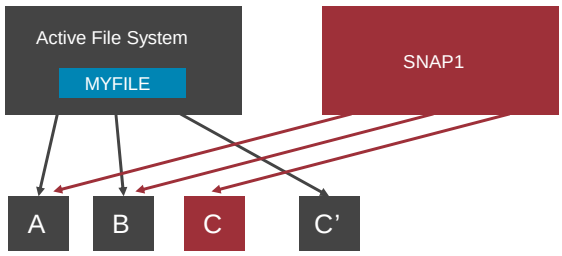
The NetApp SnapShot technology is particularly efficient, providing instant snapshot creation, and near-zero capacity overhead at snapshot creation time.

This is possible because, like most UNIX file systems, the WAFL file system uses **inodes** in the active file system to reference disk blocks, and a Snapshot backup is just a new **root**

inode that refers to the existing blocks of data on disk. The existing data blocks that are referenced by a Snapshot are then locked against overwriting, causing any updates to the active file system to be written to other locations on disk.

Refer to Figure 15 for an example of how the Snapshot process occurs.

Figure 15 – A Snapshot Locking Some Blocks in the AFS



NOTE: Each volume can retain up to 255 snapshots.

Whenever you create a new volume, a default snapshot schedule is also created. This schedule will create Snapshots according to its default settings, but these are usually modified or disabled to suit the local backup requirements. For example:

- **List the default Snapshot schedule**
  - The default schedule creates four "hourly snapshots" (at 8, 12, 16, and 20) and retains the six most recent, and two "daily snapshots" (taken at 24:00 on Mon-Sat) and zero weekly snapshots (taken at 24:00 on Sunday)
  - `snap sched <vol>`
  - Output is: `Volume <vol>: 0 2 6@8,12,16,20`
- **Change the Snapshot schedule**
  - You can modify the snapshot schedule with the following command:
  - `snap sched <vol> weekly nightly hourly@<time>`
  - `snap sched <vol> 2 7 6@6,9,12,15,18,21`
- **Disable the Snapshot schedule**
  - You can disable the scheduled snapshots with either of the following commands:
  - `snap sched <vol> 0 0 0`
  - `vol options <vol> nosnap on`

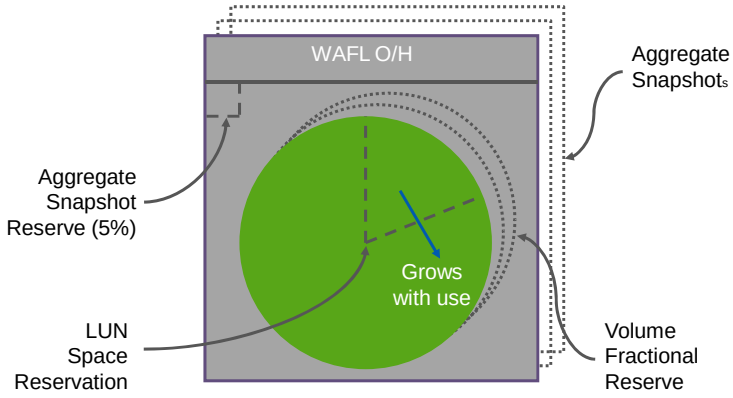
NOTE: You should normally disable the controller initiated snapshots on volumes that contains LUNs. This is because the consistency of the file system in the LUN can only be guaranteed by the host accessing that LUN. You should then use a tool such as SnapDrive to initiate the snapshots from the host.

A percentage of every new volume (and aggregate) is reserved for storing snapshot data. This is known as the **SnapShot Reserve**. The default reserve is 5% for aggregates and 20% for volumes. You can modify the default values with the **`snap reserve`** command. For example:

- **`snap reserve <vol> <percentage>`**

Refer to Figure 16 to identify where the snap reserve values apply.

Figure 16 – Aggregate and Volume Snapshot Reserve



NOTE: The volume snapshot reserve is the minimum amount of space reserved for snapshot data. As more snapshots are created they may consume more than the initial reserve value.

ADMINISTRATION

Almost all management of the snapshots is performed with the **`snap`** command. For example:

- **`snap list`**
  - Show the currently retained Snapshots
- **`snap create <vol_name> <snap_name>`**
  - Create a new volume snapshot

- If you want an Aggregate level Snapshot, then specify **-A**

- **snap delete <vol_name> <snap_name>**
  - Delete the specified snapshot, and free its disk space

NOTE: Some special types of snapshots are created and managed by the storage controller and should not be interfered with (for example, SnapMirror and SnapVault snapshots).

Another type of Snapshot that you should not manage from the storage controller itself are snapshots created by **SnapDrive** or **SnapManager**. These snapshots are created, retained, and deleted under the control of these host and application integration agents. They contain consistent backup images that are being retained by schedules and policies on the agents and should not be deleted manually.

PERFORMANCE

There is typically very little impact in the creation, retention, and deletion of snapshot backups. This is due to the very efficient "no copy" snapshot technology.

Refer to the SnapRestore performance section for more information.

SECURITY

By definition, the snapshot backups are read-only views of the file system state at the point in time of the snapshot creation. Therefore their contents cannot be modified by the end users.

User access to the data in a snapshot is controlled by the file system security settings (for example, NTFS ACLs) that were in place at the time the snapshot was created.

NOTE: If the security style of the volume has been changed since the snapshot was created, then the users might not be able to access the file system view in the snapshot directory (unless their username mapping is correctly configured to allow them to access the foreign security style). This is because the previous security settings are preserved in the snapshot view.

The storage administrator can configure the visibility of the snapshot directory (for NAS clients). The following commands either enable or disable client access to snapshot directory:

- **Per Volume**
  - The default volume settings DO allow the snapshot directory to potentially be seen by the NAS protocols. Use the following command to *disable* the snapshot directory per volume.
  - `vol options <volume_name> nosnapdir on`
- **CIFS access**
  - The default CIFS settings do NOT allow the snapshot directory to be seen by CIFS clients. Use the following command to *enable* the snapshot directory.
  - `options cifs.show_snapshot on`
- **NFS access**
  - The default NFS settings DOES allow the snapshot directory to be seen by NFS clients. Use the following command to *disable* the snapshot directory per volume.
  - `options nfs.hide_snapshot on`

TROUBLESHOOTING

Usually there are no problems with snapshot creation itself, but certain complications can arise due to their incorrect scheduling or low disk space conditions. For example:

- **Inconsistent LUN snapshots**
  - If you create a snapshot of a volume that contains a LUN, then there is no guarantee that the file system in the LUN will be in a consistent state. You may not be able to successfully recover from the LUN snapshot if the file system is corrupt.
  - The consistency of the file system in the LUN can only be guaranteed by the host accessing that LUN.
  - You should always use a tool such as SnapDrive to initiate the LUN snapshots from the host (so that it may flush the local file system buffers to disk).
- **Host LUN versus Controller volume free space**
  - The host accessing a LUN will always assume that it has exclusive control over the contents of the LUN and the available free space
  - As snapshots are created they will gradually consume space in the containing volume; if not managed correctly the snapshots can end up consuming all of the free space in the volume
  - This will cause an "out of space" error on the host when it next attempts to write to the LUN (which from its perspective might only be half full); the controller will then take the LUN offline in an attempt to prevent any data corruption
  - Refer to Figure 16 for a description of **Fractional Reserve**, and the **Volume AutoGrow** and **Snapshot Autodelete** options and how they ensure adequate free space for guaranteed LUN availability

SNAPRESTORE

The SnapRestore feature provides the ability to very rapidly recover from a snapshot backup. Entire volumes, individual files, or LUNs may be restored in a matter of seconds, irrespective of the size of the data.

CONFIGURATION

SnapRestore is a licensed feature and needs to be enabled before it can be configured and used. For example:

- `license add <licnum>`

NOTE: SnapRestore is a system-wide license. There is no ability to enable or disable it at a per volume level.

The only other prerequisite for using SnapRestore is that, since it restores data from snapshots, the snapshots must already exist. You can't recover from backups that you have not created or retained.

ADMINISTRATION

The SnapRestore function is an extension of the **`snap`** command, adding the **`restore`** keyword. It can either restore an entire volume or an individual file (or LUN) from a snapshot backup. For example:

- **Volume SnapRestore**

  - This will revert the *entire volume* back to exactly how it was at the time the snapshot backup was created
  - Be aware that all subsequent snapshots are also deleted
  - `snap restore -t vol -s <snap_name> <vol_name>`

- **Single File SnapRestore**
  - This will revert an *individual file* back to exactly how it was at the time the snapshot backup was created
  - `snap restore -t file -s <snap_name> <file_name>`
  - To recover to a new filename or a new directory location, add the **`-r <new_path_and_file_name>`** parameter. The new directory must already exist.

NOTE: Before running the SnapRestore command (volume or file) the volume to be processed must be online.

It is important to remember that SnapRestore recovers **volume and file content only**, and does not recover the following settings:

- Snapshot copies schedule
- Volume option settings
- RAID group size
- Maximum number of files per volume

Important NOTE: The `volume SnapRestore` command will revert the entire Active File System (AFS) back to the point at which the Snapshot backup was created. Any other snapshots between the old AFS and the restored snapshot are also deleted. Be very sure of your intention with SnapRestore because **there is no way to back out these changes!**

PERFORMANCE

If you do a single file SnapRestore, it might impact subsequent **snapshot delete** performance. This is because before any snapshots are deleted, the active maps across all snapshots need to be checked for active blocks related to the file you restored. This performance impact might be visible to the hosts accessing the controller, depending on the workload and scheduling.

SECURITY

After doing a SnapRestore (volume or file level), the file system metadata such as security settings and timestamps are all reverted to exactly as they were at the time the restored snapshot was created.

- **Security settings**
  - Any changes made to the file security settings will have reverted to their earlier values. You should review the file security if you suspect this is a problem.
- **File timestamps**
  - The file timestamps will be invalid for incremental backups (if using a third-party backup tool), so you should run a new full backup.
- **Virus scanning**

- If any virus-infected files (subsequently cleaned) had been captured in the snapshot, they will be recovered in their infected state. You should schedule a virus scan on any recovered file or volume.

TROUBLESHOOTING

Since the volume remains **online and writable** during the SnapRestore activity, there is always the possibility that users may be accessing files on the volume. This can potentially cause file corruption, or generate NFS errors such as "stale file handle." There are several methods of avoiding or correcting such issues:

- Disconnect the users prior to doing the SnapRestore
- Have the users reopen any files that present a problem

The volume to be SnapRestored cannot be a **snapmirror destination**. This is to protect the SnapMirror replication relationship. If you do want to "restore" a SnapMirror destination volume, then you should use the (licensed) FlexClone® feature to link the destination's Snapshot backup to a new writable volume.

SNAPMIRROR

The SnapMirror feature provides the ability to replicate data from one volume to another, or (typically) one controller to a remote controller. This provides a consistent, recoverable, offsite Disaster Recovery capability.

CONFIGURATION

SnapMirror is a licensed feature and needs to be enabled before it can be configured and used. The SnapMirror feature actually has **two licenses**; the first is a chargeable license and provides the Asynchronous replication capability, while the second is an additional no-charge license which provides the Synchronous and Semi-Sync capability (but requires the Async license as a prerequisite). For example:

- **First, license the SnapMirror Async function**
  - `license add <licnum>`
- **Then, license the SnapMirror Sync/Semi-sync function** (if required)
  - The no-charge SM-Sync license code is printed in the:
    - Data ONTAP Data Protection Online Backup and Recovery Guide
  - `license add <licnum>`
    - **NOTE:** Some older NetApp controllers (**FAS820 and prior**) are unable to support the SnapMirror Synchronous function

NOTE: The SnapMirror feature needs to be licensed on both the source and the destination systems (the example, Production and DR controllers).

By default, SnapMirror uses a **TCP connection** between the two controllers to send the replication data. This would typically be over an Ethernet or TCP WAN link and is usually the most cost-effective transport. However, for customers with access to inter-site Fibre connections, there is also the option of installing the **model X1024 FC adapter** and replicating across the optical media.

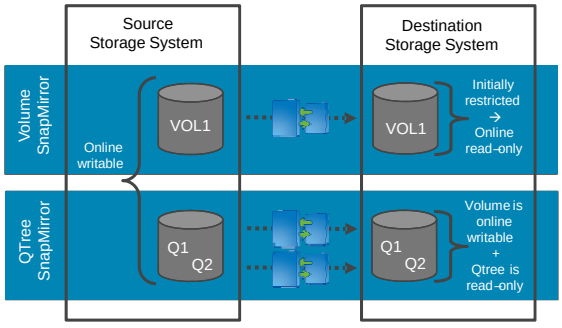
The second step (after licensing) in configuring a *Volume* SnapMirror relationship is to create the destination volume. This may be on the same controller (for example, for data migration) or on a remote controller (for example, for DR). For example:

- **Create a "restricted" mode destination volume**
  - Run this command on the destination system:
  - `vol create <vol_name>` (with parameters to suit)
  - `vol restrict <vol_name>`
- **Check the volume's status and size**
  - It must be *online* but in a *restricted* state
  - `vol status -b`

NOTE: For a qtree SnapMirror relationship the destination volume remains in an online and writable state (not restricted) and the destination qtrees are created automatically when the baseline transfer is performed.

It is important to understand the requirements and states of the source and destination volumes, and how they differ between **Volume** and **qtree** SnapMirror. The following diagram illustrates the requirements:

Figure 17 – SnapMirror Volume and Qtree Configuration



NOTE: In Volume SnapMirror, if the relationship is stopped (that is, *broken* or *released*) then the destination volume will change to a **writable** state, and the **`fs_size_fixed`** parameter is enabled on that volume. This prevents the inadvertent resizing of the destination volume, which could cause problems if/when the relationship is resynchronized.

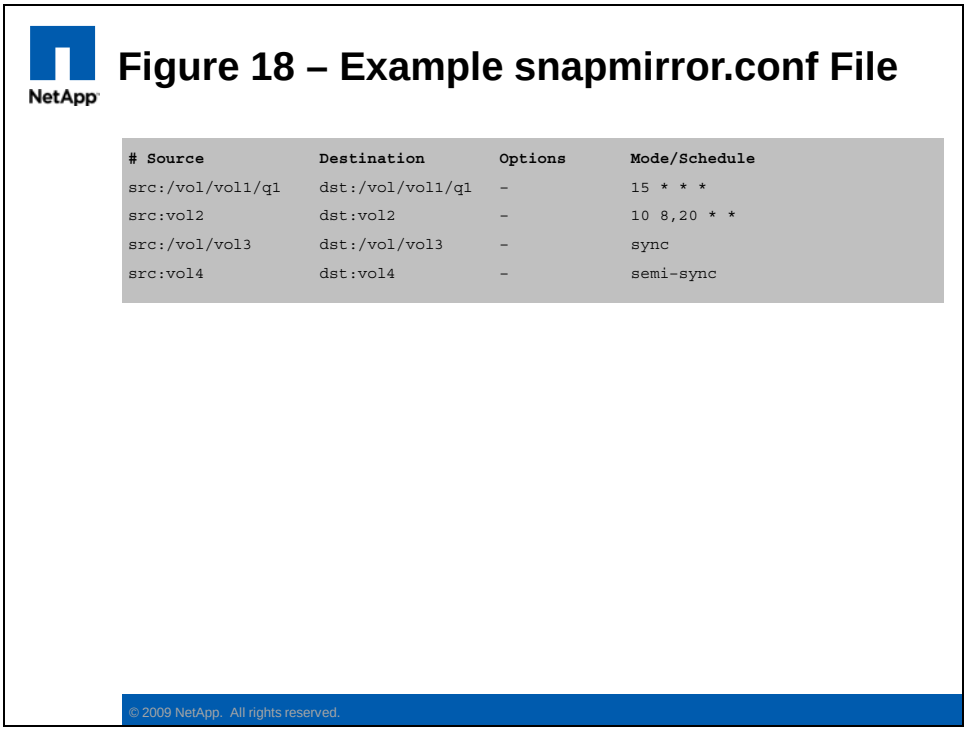
Before you can enable the SnapMirror relationship, you first need to configure the **SnapMirror access control** between the primary and secondary storage controllers. Refer to the Security section for a description of these settings.

After the source and destination volumes are defined you can configure the SnapMirror relationship. This will also perform the initial **baseline transfer**, copying all of the data from the source to the destination volume.

- **`snapmirror initialize -S src:vol1 dst:vol2`**

When the baseline transfer has completed, the destination volume will be an exact replica of the source volume (at that point in time).

Next you need to configure the **ongoing replication relationship**. This will control the mode and/or schedule for replication of the changed data from the source to the destination volume. The SnapMirror replication parameters are defined in the **`snapmirror.conf`** file, as shown in the example below:



NOTE: The `snapmirror.conf` file is configured on the **destination** controller.

As shown above, the SnapMirror relationship can operate in three different modes, performing either Async, Sync, or Semi-Sync replication. The three modes are described in more detail below:

- **Asynchronous**
  - Replicates Snapshot copies from a source volume or qtree to a destination volume or qtree
  - The host receives acknowledgment after the write has been committed to the source volume
  - Block-level, incremental updates to the destination volume are based on **schedules**
  - VSM example: `src:vol2 dst:vol2 - 10 8,20 * *`
  - QSM example: `src:/vol/vol1/q1 dst:/vol/vol1/q1 – 15 * * *`
- **Synchronous**
  - Replicates writes from the source volume to the destination volume *at the same time* that they are written to the source volume
  - The host only receives acknowledgment after the write has been committed to both the source and destination volumes
  - `src:/vol/vol1/q1 dst:/vol/vol1/q1 – sync`
- **Semi-Synchronous**
  - Replicates writes from a source volume or qtree to a destination volume or qtree with *minimal delay*
  - The host receives acknowledgment after the write has been committed to the source volume
  - This minimizes any performance impact on the host system
  - Old syntax, `src:vol1 dst:vol1 outstanding=5s sync`
  - New syntax, src`:vol1 dst:vol1 – semi-sync`

NOTE: For a description of the various replication options (such as schedule definitions or throughput throttling), refer to the product documentation.

It is also possible to configure SnapMirror to use two redundant data paths for the replication traffic. These can be either TCP or FC connections, or a mixture of the two. This is configured with the following keywords in the `snapmirror.conf` file:

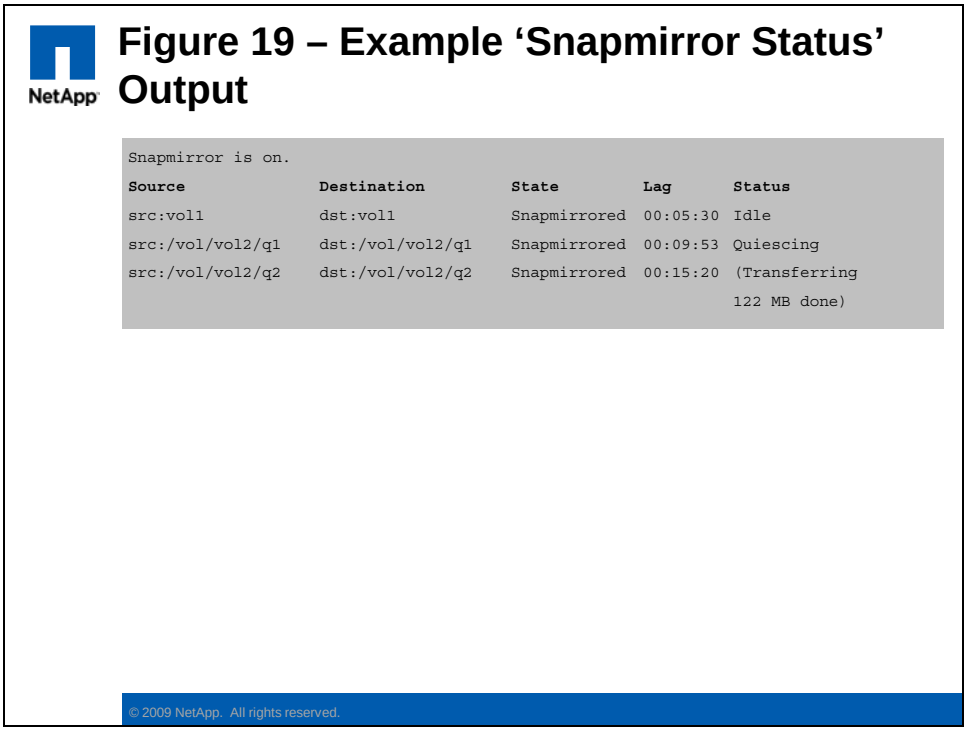
- **Multiplexing**
  - Both paths are used at the same time for load balancing
- **Failover**
  - The first path specified is active, the second path is in standby mode, and only becomes active if the first path fails

NOTE: Editing the `/etc/snapmirror.conf` on the destination will cause an in-sync relationship to temporarily fall out-of-sync.

ADMINISTRATION

The administration of a SnapMirror relationship can be performed from either the source or the destination system, although certain functions are only available on their respective systems.

The **`snapmirror status`** command is used to display the state of the currently defined SnapMirror relationships, as shown in the example below:



The **Lag** column displays the time since the last successful replication of the SnapMirror managed Snapshot from the source volume.

The same **`snapmirror`** command is also used to manage all aspects of the SnapMirror relationship, such as suspending and restarting the replication, or destroying the relationship entirely. The following are some examples of common `snapmirror` functions:

- **`snapmirror quiesce <dst_vol>`**
  - Executed on the *destination* system, this command will temporarily pause the replication. The destination volume remains read-only.
  - The relationship is still defined, and may be resumed if required.
- **`snapmirror resume <dst_vol>`**
  - Executed on the *destination* system, this command will resume the volume replication.
- **`snapmirror break <dst_vol>`**
  - Executed on the *destination* system, this command will stop the replication and convert the destination volume to a writable state.
  - The relationship is still defined, and may be resynched if required.
- **`snapmirror resync <hostname:vol>`**
  - This command will identify the latest common (*SnapMirror managed*) snapshot between the source and destination volumes and re-synchronizes the data between the two volumes
  - The *direction* of synchronization is dependent on which system the command was executed on; this will overwrite any new data on the controller on which the command was executed (bringing it back into sync with the opposite volume)

    - If executed on the *destination* system, then the relationship continues in its original manner (src→dst)
    - However, if executed on the *source* system, then the relationship reverses its original direction (dst→src)

- **`snapmirror release <src_vol> <dst_hostname:dst_vol>`**
  - Executed on the *source* system, this command will stop the replication and convert the destination volume to a writable state
  - The relationship is deleted, and cannot be restarted
- **`snapmirror update <dst_vol>`**
  - Executed on the *destination* system, this command performs an immediate update from the source volume to the destination volume

The process of capturing consistent SnapShot backups on the source volume, and then transferring them to the destination system, will vary depending on your application's capabilities, use of SnapDrive and SnapManager, the replication mode, and the intended result. The following is one example of a process to create a consistent snapshot at the destination of a qtree SnapMirror relationship:

- Make the source volume consistent on disk
  - Halt the application
  - Flush the file system buffers
- *Quiesce* the SnapMirror relationship
- Create a snapshot of the destination volume
- *Resume* the SnapMirror relationship

NOTE: In environments using SnapManager, the snapshot and replication process is usually automated through the SnapManager utility and can be performed with no disruption to the application.

PERFORMANCE

One of the challenges in a new SnapMirror configuration is transferring the baseline copy from the source to the destination system. Although the WAN connection may be adequate to handle the incremental synchronization traffic, it may not be able to complete the baseline transfer in a timely manner. In this case you might consider using the *SnapMirror to Tape* function. This method can perform the initial baseline transfer by using physical tape media.

After the initial baseline transfer is complete (which is usually constrained by the **bandwidth** of the connection), then the incremental synchronization needs to occur (which is usually constrained by the **latency** of the connection).

The appropriate choice of the SnapMirror mode (sync, semi-sync, or async) will often be driven by the latency of the WAN connection. Since latency increases over distance, this effectively limits the **sync** mode to less than 100 kM range. If the customer requires a "sync-like" replication feature past that distance, or simply to reduce the performance impact on the source system, then you should consider using the **semi-sync** mode.

In contrast, the **async** mode uses scheduled replication and is not affected by connection latency. One way to effectively improve async performance is to increase the interval between the replication times. This allows for "file system churn,"which means that, as data

is rewritten throughout the day, only the latest version will be included in the less frequent replication schedules.

In contrast to flexible volumes, the physical characteristics of **traditional volumes** will have an effect on SnapMirror performance. For best SnapMirror performance when using traditional volumes, you should configure the source and destination volumes with the same *RAID size*, *RAID group size*, and *number of RAID groups*.

Another parameter that controls the *apparent* performance of the SnapMirror synchronization is the **visibility_interval**. This parameter controls the view of the data on the destination system. Even after the data has been received, the destination file system view will not be updated until the visibility interval has elapsed. The default visibility time is **three minutes**, with a minimum setting of 30 seconds. Do not reduce the default value because this can have a detrimental impact on controller performance.

NOTE: It is also possible to throttle the SnapMirror traffic so as to reduce its impact on other applications that also use the WAN connection.

SECURITY

Before you can enable the replication relationship you first need to configure the **SnapMirror access control** between the source and destination storage controllers.

The source controller needs to grant access for the destination controller so that the destination controller can pull updates from the source. The destination controller should also grant access to the source controller, so that the replication relationship may be reversed after a disaster event has been resolved (for example, synchronizing back from the DR site to the Production site).

There are two ways to configure SnapMirror access control on the storage controllers, as shown below:

- **Method #1**
  - `options snapmirror.access host=<other controller>`
    - By default this option is set to the keyword "`legacy`". This causes the system to refer to the snapmirror.allow file for access control
    - Alternatively, you can set this option to "`host=<hostname>`" to enable SnapMirror access from the remote controller
- **Method #2**
  - `options snapmirror.access host=legacy`
    - Edit the `/etc/snapmirror.allow` file and add the other storage controller's hostname

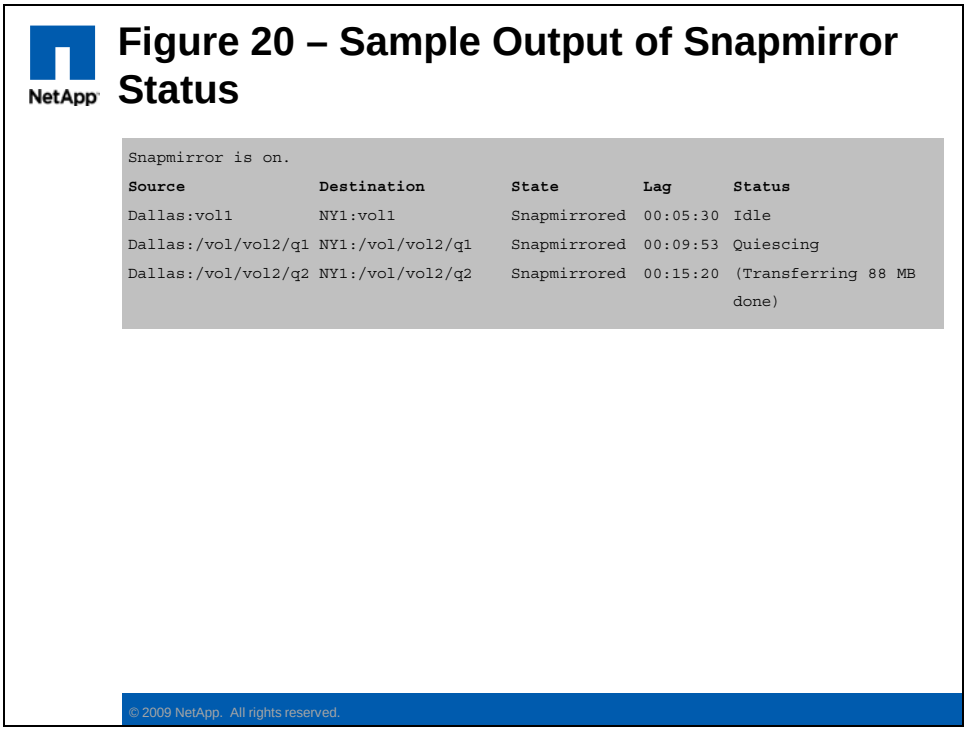
NOTE: Method #1 is the preferred way to enable the remote access.

The traffic between the source and destination controllers is not encrypted. In a security-conscious environment it may be necessary to implement some type of network-level encryption for the replication traffic; such as the NetApp "DataFort" encryption devices.

TROUBLESHOOTING

Comprehensive logging of all SnapMirror activity is enabled by default. The log file is saved to the **`/etc/log/snapmirror.[0-5]`** file(s). If required, the log can be disabled by executing the following command:
`options snapmirror.log.enable [on|off].`

The **`snapmirror status`** command will display the current status of all SnapMirror relationships. Note that some status information is only available on the destination controller.



A SnapMirror relationship passes through several defined stages as it first initializes (**level-0**), then synchronizes (**level-1**), and possibly reestablishes a broken mirror. The exact process of troubleshooting and any rectification actions will depend on what stage was in progress at the time of failure. For example, if communications failed during the initial baseline transfer, then the destination would be incomplete, and you would need to rerun the initialization rather than trying to reestablish synchronization to the incomplete mirror.

SNAPVAULT

The SnapVault feature provides the ability to create and archive Snapshots from one volume to another, or (typically) from one controller to a remote controller. This provides a consistent, recoverable, offsite, long-term backup and archive capability.

CONFIGURATION

SnapVault is a licensed feature and needs to be enabled before it can be configured and used. The SnapVault feature actually has **two licenses**; one license is for the **primary** controller (backup source), while a second, different, license is required for the **secondary** controller (archive destination).

For example:

- **License the Primary controller** (for example, sv_ontap_pri)
  - `license add <licnum>`
- **License the Secondary controller** (for example, sv_ontap_sec)
  - `license add <licnum>`

NOTE: The two licenses enable different functionality, and the correct license needs to be enabled on the appropriate controller (for example, Production and DR).

The second step (after licensing) in configuring a SnapVault relationship is to create the destination volume. This is typically on a remote controller with lower cost storage (for example, SATA disks).

For example:

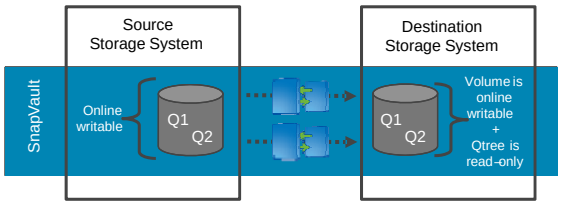
- **Create a normal destination volume**
  - Run this command on the destination system:
  - `vol create <vol_name>` (with parameters to suit)
- **Check the volume's status and size**
  - It must be *online* but in a *writable* state
  - `vol status -b`
- **DO NOT create the destination qtrees**
  - They will be **created automatically** when the SnapVault relationship is initialized.

NOTE: Although the destination volume remains writable, the individual destination qtrees are in a **read-only** state.

It is important to understand the requirements and states of the source and destination volumes, and how they differ to the SnapMirror requirements.

The following diagram illustrates the requirements:

Figure 21 – SnapVault Volume and qtree Configuration



The underlying technology behind SnapVault is based on the **qtree SnapMirror** function. This determines many of the features and limitations of SnapVault; such as the basic unit of SnapVault backup being the **qtree**, and all SnapVault transfers are based on schedules (that is, asynchronous).

Before you can enable the SnapVault relationship, you first need to configure the **SnapVault access control** between the source and destination storage controllers. Refer to the Security section for a description of these settings.

After the source and destination volumes are defined, you can configure the **SnapVault schedules** on the primary and secondary controllers and start the ongoing incremental backups. At this point we will also perform the initial **baseline transfer**, copying all of the data from the source qtree to the destination qtree.

- **Configure the primary controller**
  - Define a SnapVault schedule:
  - `snapvault snap sched vol1 sv_hourly 5@mon-fri@9-19`
- **Configure the secondary controller**
  - Perform the baseline transfer:
  - `snapvault start -S pri:/vol/vol1/q1 sec:/vol/vol1/q1`
    - When the baseline transfer has completed, the destination volume will be an exact replica of the source volume
  - Define a SnapVault schedule:
  - `snapvault snap sched -x vol1 sv_hourly 5@mon-fri@9-19`

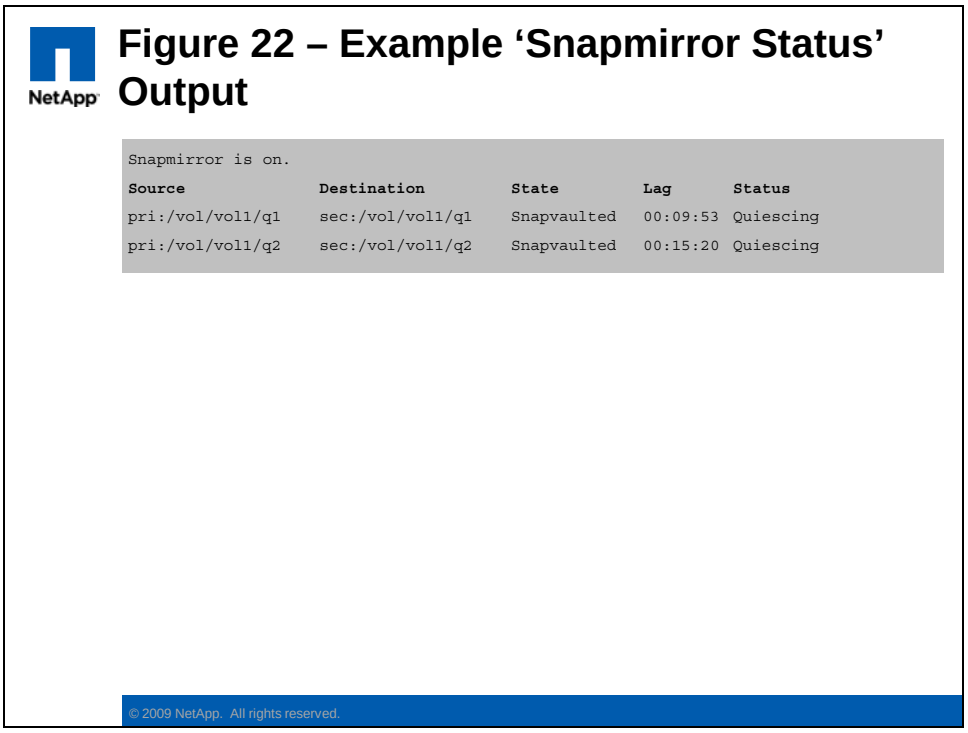
- The **-x** parameter instructs the secondary controller to request a resynchronization with the primary controller. This fetches the current file system state and then creates a Snapshot to retain the data.

NOTE: The SnapVault schedule definition is in the following format: **<snapshots_to_retain>@<day_of_the_week><@hour_of_the_day>**

ADMINISTRATION

The administration of a SnapVault relationship can be performed from either the primary or the secondary system, although certain functions are only available on their respective systems.

The **snapvault status** command is used to display the state of the currently defined SnapVault relationships, as shown in the example below:



NOTE: You can use the **snapvault status -c** option to display the SnapVault qtree configuration parameters.

The same **snapvault** command is also used to manage all aspect of the SnapVault relationship, such as updating the secondary, or restoring the backup. The following are some examples of common `snapvault` functions:

- **snapvault update sec_hostname:/vol/vol_name/qtree**
  - Executed on the *secondary* system, this command will trigger a manual (unscheduled) update of the specified qtree destination

- **`snapvault release <path> <other_hostname>:<path>`**
  - Executed on the *either* system, this command will delete the SnapVault relationship
- **`snapvault restore -S sec_hostname:/vol/vol_name/qtree pri_hostname:/vol/vol_name/qtree`**
  - Executed on the *primary* system, this command will restore the qtree contents from the backup
  - To restore to the original qtree location on the primary system you will first need to break the SnapVault relationship, or simply restore to a new qtree (and rename later)
  - To restore a small amount of data (for example, a single file) simply copy the files from a CIFS share on the secondary qtree
- **`snapvault start -r <path>`**
  - Executed on the *secondary* system, this command will resynchronize the relationship and resume backup operations after a SnapVault restore

NOTE: Refer to the product manual if you require information on the other SnapVault commands.

In addition to working directly with the SnapVault commands on the controller, a number of third-party backup applications have SnapVault integration. In order for these applications to communicate with the controller you will need to enable the **NDMP** protocol and define the **username** and **password** for the application to use.

Some of these applications provide additional SnapVault related functionality, such as **NetBackup**, which allows restores to be performed through either the **NBU admin console** or through **drag-and-drop** from a CIFS share. NetBackup also deduplicates backup data on NBU secondary volumes, and can leverage the **A-SIS deduplication** feature to deduplicate blocks for any regular backup volume.

NOTE: The SnapVault/NBU integration mentioned above is an example only, and such functionality was recently discontinued. Other third-party products continue to provide similar capabilities.

PERFORMANCE

One of the challenges in a new SnapVault configuration is transferring the baseline copy from the primary to the secondary system. Although the WAN connection may be adequate to handle the incremental backup traffic, it may not be able to complete the baseline transfer in a timely manner. In this case you might consider using the *Logical Replication (LREP)* function. This method can perform the initial baseline transfer by using external disk media, such as a USB drive connected to a laptop computer.

Since the SnapVault backups are always a scheduled activity (that is, asynchronous) they are only constrained by the **bandwidth** of the connection, and are not significantly affected by the link **latency**.

In common with qtree SnapMirror, the SnapVault process accesses the primary qtree at the file system level, and therefore sees (and backs up) the original (fat) version of any deduplicated data. This may cause more data to be sent across the WAN than otherwise

expected, and the secondary qtree will be written in the original (fat) capacity. Further deduplication may then be scheduled on the secondary system if required.

SECURITY

By default, no access is granted for SnapVault traffic, and specific access must be granted for any remote controller in a backup relationship.

The primary controller needs to grant access for the secondary controller so that the secondary controller can pull backups from the source. And the secondary controller should also grant access to the primary controller, so that the primary controller can request restores of the backups.

SnapVault access can be configured as shown below:

- **On the Primary (and Secondary) controller/s**
  - `options snapvault.access host=<other controller>`

TROUBLESHOOTING

Comprehensive logging of all SnapVault activity is enabled by default. Since the SnapVault function is based on the qtree SnapMirror function, all log information goes to the same file.

The log information is saved to the /**`etc/log/snapmirror.[0-5]`** file(s). If required, the log can be disabled by executing the following command:
`options snapmirror.log.enable [on|off].`

OSSV

The OSSV agent is a software application that allows SnapVault-like backups (block-level, incremental forever) to be performed by a non-NetApp platform (for example, Windows, UNIX, or Linux host). The OSSV software is installed on the host (primary) and sends the SnapVault backup data to a NetApp controller (secondary) using the SnapVault protocol. The backup data is then retained using normal SnapVault schedules on the NetApp controller.

CONFIGURATION

OSSV is a licensed feature and needs to be enabled before it can be configured and used. The OSSV/SnapVault feature actually requires **two licenses**; one license is for the **OSSV primary** server type (Windows or UNIX backup source), while a second, different, license is required for the **NetApp OnTap secondary** controller (archive destination).

For example:

- **License the primary host** (for example, sv_windows_pri)
  - `license add <licnum>`
  - **NOTE:** The OSSV primary license key needs to be enabled on the **secondary** controller and not on the Windows or UNIX host.
- **License the secondary controller** (for example, sv_ontap_sec)

- `license add <licnum>`

The installation details for the OSSV agent are operating system dependant. For example, a setup EXE for the Windows platform, and installation scripts for the various UNIX platforms.

Some of the OSSV utilities are listed below:

- **`svconfigpackager`**
  - Unattended installation utility
- **`svinstallcheck`**
  - Automatic post-installation check
- **`svconfigurator`** (GUI)
  - This is the primary configuration tool, used to start/stop the OSSV service, set the NDMP password, enable debugging, enable trace files, and so on.
- **`svsetstanza`**
  - Command line alternative to the `svconfigurator` GUI tool

NOTE: You will need to restart the OSSV service to read any configuration changes.

In the OSSV configuration tool (on the primary host) you need to configure the access controls, NDMP username and password, and which directory(s) to include in the SnapVault backup.

In general, the configuration of the secondary controller is identical to that performed for a normal SnapVault relationship.

ADMINISTRATION

Although the mechanism by which the OSSV agent determines the changed blocks to backup differs greatly from that used by SnapVault on a primary NetApp controller, the administration of the two is very similar.

For example:

- **To backup the OSSV client**
  - Perform an initial baseline backup to the destination
  - Schedule regular block-level incremental backups
- **To restore the OSSV client**
  - Use the `snapvault restore` command, similar to how you would on a normal SnapVault client

One difference between the OSSV client and a normal SnapVault client is that, because the (Windows or UNIX) clients lack native file system snapshots, a different mechanism is used to identify the changed blocks to backup. This **block-level incremental (BLI)** mechanism requires some free space on the OSSV client to store a database of previously backed-up files and their block checksum values. Subsequent backups are compared to this database to determine the changed data to back up.

You can use the **Free Space Estimator Utility** to determine if there is sufficient disk space (on the OSSV primary) to house the database and to perform a BLI backup.

PERFORMANCE

Many of the same considerations for SnapVault performance also apply to OSSV performance, as the two features perform almost identical functions.

One way in which OSSV may differ from a controller-based SnapVault primary is the scale of the backup environment. Even a modest OSSV implementation may see tens (if not hundreds) of OSSV agents installed on a customers Windows and/or UNIX hosts. All of this OSSV backup traffic will concentrate on the SnapVault primary controller, and some thought needs to be given to the number of **concurrent backup streams**.

Each model of NetApp controller supports a given number of concurrent backup streams, which varies according to the model type (and sometimes, software version). The maximum number of concurrent backup streams can be increased by enabling the **NearStore Personality License (NPL)** on the secondary controller. The NPL feature is available for all NetApp controller models. Previously, NetApp marketed a purpose-built **NearStore appliance**, which combined **SATA-only storage** and the NPL function.

Given that the backup traffic is a non-latency sensitive, sequential workload, you should store the secondary backup data on **SATA** disk.

NOTE: The NearStore feature was originally only available on the dedicated NearStore controller, but is now available as a licensed feature on all NetApp FAS controllers. Some documentation may still refer to the earlier NearStore hardware requirement. One difference between the hardware and software implementation of NearStore is that the NearStore hardware could be either a primary OR a secondary system. But with the software option the controller can provide both functions (in different relationships).

SECURITY

By default, no access is granted for OSSV traffic, and specific access must be granted for any remote controller in a backup relationship.

The OSSV primary host (Windows or UNIX) needs to grant access for the secondary controller so that the secondary controller can pull backups from the source. And the secondary controller should also grant access to the OSSV primary host, so that the host can request restores of the backups.

OSSV and SnapVault access can be configured as shown below:

- **On the Primary (Windows or UNIX) host**
  - Use the client configuration tool to edit the "QSM Access List" field, and enter the secondary controller's hostname
- **On the Secondary (NetApp) controller**
  - `options snapvault.access host=<other controller>`

You will also need to set the NDMP **username and password** on the OSSV client. This can be done with either the client configuration GUI or the **`svpasswd`** command.

TROUBLESHOOTING

The log file locations for the OSSV agents are operating system dependant.

For example:

- **Primary Windows OSSV client**
  - `c:\Program Files\netapp\snapvault\etc\snapvault.yyyymmdd`
- **Primary UNIX and Linux OSSV client**
  - `/usr/snapvault/snapvault.yyyymmdd`
- **Secondary NetApp SnapVault controller**
  - `/etc/log/snapmirror.[0-5]`

SNAPLOCK

SnapLock is a licensed feature that provides **data retention** capabilities to any NetApp storage controller. A SnapLock volume can be configured to make immutable any data stored within it, and is then accessed through either the NFS or CIFS protocols (or a specialized Archival application). Sometimes this is referred to as **"Write Once Read Many" (WORM)** storage.

CONFIGURATION

Snaplock is a licensed feature and needs to be enabled before it can be configured and used. The SnapLock feature actually has **two licenses**; one that provides the **SnapLock Compliance (SLC)** capability, and a second that provides the **SnapLock Enterprise (SLE)** capability.

For example:

- **To license the *Snaplock* (Compliance) function**
  - `license add <licnum>`
    - **NOTE:** The V-Series family of controllers is unable to support the SnapLock Compliance function. This is because the physical storage is provided by a third-party storage controller, and we cannot guarantee the immutability of the storage configuration.

To license the *Snaplock_Enterprise* (Enterprise) function (if required)

- `license add <licnum>`

The second step (after licensing) is to enable the **Compliance Clock**. This is an additional hardware clock in the storage controller. The Compliance Clock takes its initial settings from the controller's current time, but from that point on it can never be modified or disabled.

For example:

- **To enable the compliance clock**

- `date -c initialize`

NOTE: Great care should be taken to ensure that the controller's system time is correct before enabling the Compliance Clock.

The second step (after licensing) is to create the **SnapLock volume**. This may be either a Traditional volume or an Aggregate/Flexible volume(s).

For example:

- **To create a Traditional volume (SnapLock)**
  - `vol create -L <vol_name>` (with other parameters to suit)
- **To create an Aggregate/Flexible volume (SnapLock)**
  - `aggr create -L <aggr_name>` (with other parameters to suit)
  - **NOTE:** Any volume created in a SnapLock aggregate will also be a SnaLlock volume
  - `vol create <vol_name> <aggr_name> <size>` (and so on)

NOTE: The type of SnapLock volume created (SLC or SLE) is determined by the installed license, or can also be specified after the –L parameter.

If you are configuring a SnapLock volume to be a **SnapMirror destination**, then you should instead create the volume as a non-SnapLock volume. Later, when you run the **`snapmirror initialize`** command, you need to add the **–L parameter** to the command line. This will perform the baseline transfer, and only enable the SnapLock function once the transfer is successful.

NOTE: Using SnapLock in a SnapMirror or SnapVault relationship entails a number of restrictions and retention features. Refer to the product manual if you need more information on this configuration.

The default data retention parameters are configured on a per volume basis.

For example:

- `vol options <vol_name> snaplock_minimum_period 1d`
- `vol options <vol_name> snaplock_maximum_period 90d`
- `vol options <vol_name> snaplock_default_period max`

ADMINISTRATION

SnapLock volumes are managed as per any other volume on a NetApp controller. The only difference being that the SnapLock retention parameters prevent the modification or deletion of retained data (and volumes).

Accessing and storing data on a SnapLock volume is identical to any other volume that is accessed through the NAS protocols (CIFS and/or NFS).

The process of committing the stored data to WORM retention consists of changing the file attributes from writable to read-only while the file is in the SnapLock volume.

For example:

- **From a Windows client**

  - Right-click on the file and select *Properties*
  - Check *Read Only*, and click *OK*
  - **NOTE:** The file's retention period is inherited from the volume's default retention period

- **From a UNIX client**
  - Change the access time (*atime*) of the file with the `touch` command
  - Make the file *read-only* using the `chmod` command
  - **NOTE:** The file's retention period is inherited from the volume's default retention period, unless set via the *atime* parameter

Also, since SnapLock volumes tend to retain, and not delete, data, be sure to provision sufficient capacity to hold the required data and retention period. It is possible to resize a SnapLock volume as you would any other volume (Flexible or Traditional).

PERFORMANCE

SnapLock volumes are typically accessed either directly through the NAS protocols (CIFS or NFS) or through a Content Management or Archival application (such as FileNet, Tivoli, or Enterprise Vault).

The performance through each of these methods should be no different than when accessing a non-SnapLock volume.

SECURITY

The SnapLock Compliance and Enterprise features impose the following restrictions (while retention managed data exists):

- **Enterprise** (SLE)
  - Users cannot change or modify any SnapLock managed data
  - The administrator can delete the entire volume (and all of the data it contains)
- **Compliance** (SLC)
  - Users cannot change or modify any SnapLock managed data
  - The administrator can NOT delete the volume (short of physically removing and destroying the disks)

It is not possible to alter the Compliance Clock time settings once it has been enabled. If the controller is powered off then the Compliance Clock will effectively lose time during the outage, but will gradually resynchronize with the system time over the next year (at a rate of no more that seven days per year). Even swapping the controller head, or moving the array to another controller, will not allow the deletion of SnapLock managed data.

TROUBLESHOOTING

In a SnapLock + SnapVault (AKA **LockVault**) configuration, the SnapVault log files reside in the LockVault log volume's `/etc/log` directory. They inherit the retention periods set at the volume level.

ACTIVE/ACTIVE CONFIGURATION (HA CLUSTER)

NetApp use the term "Active/Active" to describe the high availability cluster failover configuration. Two controller heads are configured as a cluster, with each node providing fail-over support for its partner.

CONFIGURATION

Active/Active is a licensed feature and needs to be enabled before it can be configured and used. The Active/Active feature needs to be licensed on both cluster nodes.

For example:

- **License the first node**
  - `license add <licnum>`
- **License the second node**
  - `license add <licnum>`

NOTE: You will then need to reboot and start the cluster feature.

Once the Active/Active feature is enabled, you can only **unlicense** the cluster feature when the cluster is in a normal state, and the cluster services have been manually disabled.

There are a number of **prerequisites** that you need to configure prior to enabling Active/Active clustering. For example, the cluster interconnect cables need to be attached, both controllers require fibre connection to all expansion drawers, and both controllers require access to the same IP subnets.

The Active/Active feature activates a number of high availability capabilities, such as NVRAM mirroring, enabling both controllers to provide failover support for each other.

For example, in the case of a controller failure:

- The surviving node spawns a virtual instance of the failed node
- The virtual node accesses its mirrored NVRAM to complete any interrupted writes
- The local network interface(s) assumes the IP address of both the local and partner interfaces (for Ethernet traffic)
- The local fibre interfaces keep their original WWPN addresses, and the host-based MPIO drivers direct all FCP traffic through them (assuming single-image cfmode is being used)

The process of **removing** an Active/Active configuration is described below:

- Disable the cluster (`cf disable`)
- Delete the cluster license (`license delete`)
- Remove the partner's network entries from the `/etc/rc` file
- Halt, and make sure the `partner-sysid` is blank
- Power down and remove/relocate the cluster interconnect card
  - (depending on model, or relocate it to the single-node slot location)
- Repeat on the other controller

ADMINISTRATION

In an Active/Active configuration both controllers have visibility to all of the disks. The disks then need to be assigned to one or the other controller before they can be used in an aggregate or as a spare. This is known as **software disk assignment**. If a disk has not been assigned to a controller (that is, it lists as "not owned"), then it cannot be used by either controller for any purpose.

Use the following commands to manage disk ownership:

- **Assign disk ownership**
  - `disk assign`
- **List disk ownership** (several methods)
  - `disk show -v`
  - `storage show disk`
  - `sysconfig -r`

NOTE: Earlier versions of Data ONTAP also supported *hardware disk assignment*, where ownership was determined by the fibre cabling topology. This mode is not supported on any of the current generation of controllers.

Generally, administration of an Active/Active configuration is identical to managing two non-clustered controllers. The clustered controllers are still managed separately, although some configuration settings need to be synchronized between the two. One of the few differences that you will need to master is the process of cluster fail-over and fail-back.

For example, **after a failed controller has rebooted** and is ready to assume its old identify and workload, it will display "*waiting for giveback*" or "*waiting for mb giveback*". At this point the administrator enters **`cf giveback`** on the operational controller to failback and return the cluster to its normal state.

NOTE: Refer to the section titled *'Management of an active/active configuration'* in the *'Data ONTAP 7.3 Active/Active Configuration Guide'* for more information on cluster management.

PERFORMANCE

Optimum performance is usually achieved when both controllers in an Active/Active configuration share the client workload evenly. This is partially the result of good solution planning, though some benefit comes from the automatic load balancing in the host-based MPIO drivers (for FCP traffic).

In an FCP SAN environment, always ensure that the host-based multipathing support is correctly configured (for example, use ALUA support where appropriate).

In most other ways, the performance concerns with an Active/Active configuration are identical to a non-clustered configuration.

SECURITY

In almost all aspects of security, an Active/Active configuration is identical to a non-clustered configuration.

TROUBLESHOOTING

In an Active/Active configuration both controllers require connectivity to all of the disk expansion shelves. It is not possible to have a shelf connected to one controller and not to the other. If a controller **loses access** to one of the disk shelves it will trigger a negotiated (that is, clean, but automatic) failover.

It is for this reason that we always recommend using MultiPath HA cabling for the disk expansion shelf connections. This prevents unnecessary cluster failover for non-critical reasons such as SFP failure, loop breaks, ESH module failure, and so on.

METROCLUSTER / SYNCMIRROR

The MetroCluster feature provides the ability to split the two controllers in an Active/Active cluster into separate physical locations. This also involves using a SyncMirror configuration to mirror the data between the two sites. Then, in the case of a site disaster, it is possible to fail-over the cluster and restore operations at the remote site with the minimum disruption possible.

CONFIGURATION

MetroCluster is a combination of a two specific hardware configurations and several licensed features. The two MetroCluster modes, **Stretch mode** or **Fabric mode**, require different hardware configurations. All of these components needs to be connected and enabled before they can be configured and used.

For example:

- **Hardware requirements**
  - Active/Active controller heads
  - In fabric mode only, the following additional hardware is required:
    - MetroCluster FC/VI adapters
    - FC switches
- **Software license requirements**
  - Cluster license
  - Cluster_Remote license
  - SyncMirror license

There are a number of prerequisites and limitations that you should be aware of before attempting to configure a MetroCluster environment.

For example:

- **Distance / Latency limitations**
  - Stretch mode
    - The maximum cable distance between the two controllers is 500M
    - This is primarily a limitation of the MultiMode Fibre cable, and may be less if cable type and patches are not adequate
  - Fabric mode
    - The maximum supported distance between the two controllers is 100kM

        - This is primarily a limitation of the latency across this distance, and may be less if additional latency is present
        - Other factors, such as the type of laser SFP used, or whether fibre repeaters or WDWDM devices are used, will also affect the maximum supported distance

- **Disk types and expansion shelves**
  - Both controllers require connectivity to all expansion drawers
  - Stretch mode supports both FC and SATA expansion shelves
  - Fabric mode only supports FC type expansion shelves (although SATA shelves can be present if they are not mirrored)
  - In fabric mode, disk ownership is determined by where the controller's HBAs connect to the switch, and where the disk shelves connect to the switch
    - For example, if the local controller's HBA is connected to switch bank #1, it owns the disks connected to switch bank #2

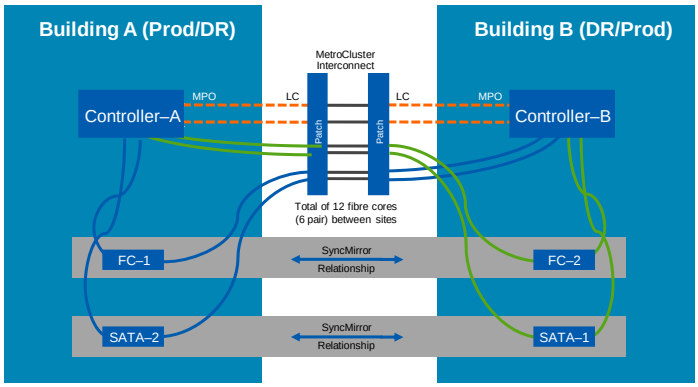
- **Networking**
  - Both controllers (at different sites) require connectivity to the same IP network subnet ranges and/or VLANs

- **SyncMirror configuration**
  - Requires an even number of disks, evenly divided between both plexes
  - All disks should be of the same size (or they will be right sized)
  - All disks must be of the same type (that is, you cannot mirror FC/SATA)

NOTE: This is only a summary of the configuration requirements and capabilities. Always refer to the product manuals before attempting to design or install a MetroCluster environment.

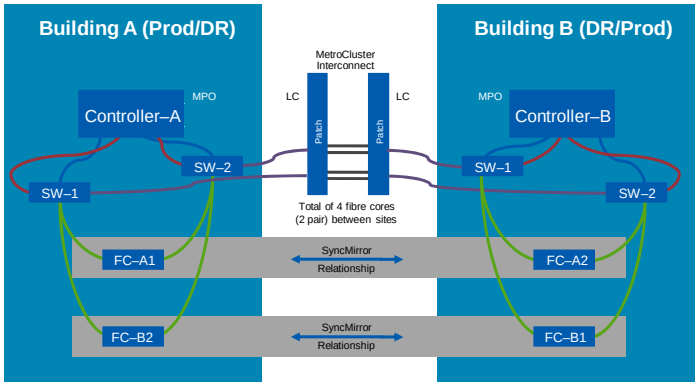
The following diagrams illustrate the main components of MetroCluster, and the differences between stretch mode and fabric mode.

Figure 23 – Stretch Mode MetroCluster Diagram



NOTE: The number and ownership of the disk expansion shelves is indicative only. Refer to the product documentation if you require more information on the supported expansion drawer configurations in a MetroCluster environment.

Figure 24 – Fabric Mode MetroCluster Diagram



NOTE: Each Blue/Green line indicates a dual fibre link (LC-LC duplex cable). The (dashed) Red lines indicate MPO-LC duplex fan-out cable.

ADMINISTRATION

After the MetroCluster-specific configuration has been completed (for example, disk pool assignment, SyncMirror configuration, and so on), then the storage administration is generally the same as for a normal Active/Active controller.

One exception to this is in the cluster failover management. Normally cluster failover happens automatically, but with MetroCluster it requires administrative intervention. This is necessary to protect against the **split brain** scenario, where, due to a communications outage, both controllers assume that the other controller has failed, and both attempt to assume their partner's role.

The specific command used to take over in a MetroCluster environment will depend on the state of the partner controller. In extreme cases, where a site disaster has made the partner unresponsive (that is, offline or destroyed), then it may be necessary to use the **`cf forcetakeover -d`** command. This will allow cluster failover to proceed even when some configuration problems would otherwise prevent a takeover. It also splits the SyncMirror relationship.

The use of the `forcetakeover` option is **very dangerous** and can potentially cause data corruption if the partner controller is still operational and able to access its own storage. Use this command ONLY if the remote MetroCluster partner node is powered off and inaccessible.

NOTE: Refer to the product documentation for more information on MetroCluster cluster takeover and giveback procedures.

If you are using **SyncMirror** to provide local mirroring (that is, not in a MetroCluster configuration) then at some point you may wish to split and disable the mirror. An example of this might be where you have been using SyncMirror to mirror/migrate data between disks on the controller.

The following is an example process to split and disable SyncMirror:

- **Check the current status**
  - Before splitting a SyncMirror volume, both plexes should be online and operational
- **Split the mirror**
  - Running the `vol split vol0/plex0 vol0` new command
  - You should now have two unmirrored volumes
- **Disable the SyncMirror license**
  - Run the `license delete XXYYZZ` command

NOTE: You cannot disable the SyncMirror license if one or more mirrored volumes still exist.

PERFORMANCE

Apart from the potential for additional latency in the SyncMirror configuration, the performance of MetroCluster should be identical to a standard Active/Active controller.

The 100kM maximum distance limitation for fabric MetroCluster is intended to limit the additional latency to a maximum of 1ms. This is usually seen as the practical limit for synchronous replication.

If the customer needs to replicate across a greater distance, or where there may be higher latency, then a standard **SnapMirror** (Semi-Sync or Async) configuration should be considered.

SECURITY

In almost all aspects of security a MetroCluster configuration is identical to a standard Active/Active controller.

Usually both controllers in the Active/Active MetroCluster configuration are enabled. This provides storage service to both sites, and provides for the simplest and fastest site failover in the case of a disaster. However, in some environments it may be desirable to **limit or prevent access to the remote controller (**that is, **DR).** This may be achieved in several ways, for example simply by configuring appropriate SAN and NAS security (**manual NFS fencing**). Or, for the ultimate isolation of the remote controller it is also possible to **power off the DR node** (which would then need to be manually powered on to perform a cluster takeover in the case of a disaster).

TROUBLESHOOTING

A MetroCluster configuration, typically spanning between two sites, is more vulnerable to **communications disruptions** than a normal Active/Active controller. The two controller heads need to maintain communication through (at least one of) the two cluster interconnect ports (over fibre), and through IP (over the WAN). Both controllers also need connectivity to all expansion shelves (over fibre), which can use either single-path or multi-path HA cabling.

It is good design to ensure that the various redundant inter-site connections (assuming they exist) are located in physically separate conduits. Always remember the maxim: Men with backhoes are irresistibly drawn to network cabling.

Figure 25 – The Natural Enemy of WAN Cabling

A prolonged disruption in communications between the two controllers is classified as a disaster, and will require administrative action in order to resolve the MetroCluster site failover.

If you have specified **Multipath HA cabling** to the disk expansion shelves, then you should double check the connectivity with the following command:

- `storage show disk -p`

In a Multipath HA configuration the output should list dual paths to each disk.

In a MetroCluster configuration each site has its own assigned disks, including **spare disks** (known as a **pool**). This ensures that adequate spare disks are located at each site.

If disk fails in a RAID-4 SyncMirror volume (or two disks in RAID-DP), and there are no spares in the same pool (that is, site), then the system only generates a **warning** and continues normal operation. Even though there is no disk redundancy in that volume, it does not go into degraded mode, or shut down after the RAID timeout. This is because the volume's data integrity is still guaranteed by the mirror plex at the other site (that is, pool).

ADDITIONAL MATERIAL

The following resources are recommended as preparation for attempting the NCDA certification exams.

Practice exams:

- NS0-153 – Storage Networking
  - http://now.netapp.com/NOW/products/education/public/certification/153exam/index.html
- NS0-163 – Data Protection solutions
  - http://now.netapp.com/NOW/products/education/public/certification/163exam/index.html

Further reading:

- Product documentation
  - http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml
    - For example:
      - System Administration Guide
      - Storage Management Guide
      - Active/Active Configuration guide
      - Network Management Guide
      - File Access and Protocols Management Guide
      - Block Access Management Guide
      - Data Protection Online Backup and Recovery Guide
      - Data Protection Tape Backup and Recovery Guide
      - MultiStore Management Guide
      - Archive and Compliance Management guide
      - Commands: Manual Page Reference, Volume 1
      - Commands: Manual Page Reference, Volume 2
      - Core Commands Quick Reference
- Technical reports
  - http://www.netapp.com/us/library/technical-reports.html
- Best Practice guides
  - http://now.netapp.com/NOW/knowledge/docs/docs.cgi
    - For example:
      - Data ONTAP, Platforms and Storage
      - SAN/IP SAN
      - Enterprise File Services
      - Data Migration
      - Data Protection
      - Database
      - Operations Manager
      - SnapDrive for Windows
      - SnapDrive for UNIX
      - SnapMirror Async Overview and Best Practices Guide
      - SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations
      - V-Series
      - V-Series FAQ Covering Product and Technical Topics