NETAPP TECHNICAL REPORT

SnapMirror Async Overview and Best Practices Guide

Srinath Alapati, Darrin Chapman, NetApp

TR 3446

Updated for Data ONTAP 7.3

ABSTRACT

This document is a deployment guide for designing and deploying NetApp® SnapMirror® Async in a customer environment. It describes replicating data to a NetApp destination system by using NetApp SnapMirror technology. As always, please refer to the latest technical publications on the NOW™ (NetApp on the Web) site for updates on processes, Data ONTAP® command syntax, and the latest requirements, issues, and limitations. This document is intended to assist field personnel in designing and deploying a SnapMirror solution.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 INTENDED AUDIENCE

This technical report is designed for storage administrators and architects who are already familiar with SnapMirror software and are considering deployments for production environments.

1.2 PURPOSE

This paper presents an overview of implementing SnapMirror Async technology, with step-by-step configuration examples and recommendations to assist the reader in designing an optimal SnapMirror solution.

1.3 PREREQUISITES AND ASSUMPTIONS

For the information and procedures described in this document to be useful to the reader, the following assumptions are made. The reader has:

- Minimal knowledge of NetApp platforms and products, particularly in the area of data protection
- General knowledge of disaster recovery (DR) solutions
- Working knowledge of the NetApp SnapMirror solution
- Reviewed the *Data Protection Online Backup and Recovery Guide* on NOW

This report is based on features that are available in Data ONTAP 7.3.

1.4 BUSINESS APPLICATIONS

There are several approaches to increasing data availability in the face of hardware, software, or even site failures. Backups provide a way to recover lost data from an archival medium (tape or disk). Redundant hardware technologies also help mitigate the damage caused by hardware issues or failures. Mirroring provides a third mechanism to ensure data availability and minimize downtime. NetApp SnapMirror offers a fast and flexible enterprise solution for mirroring or replicating data over local area, wide area, and Fibre Channel (FC) networks. SnapMirror can be a key component in implementing enterprise data protection strategies. If a disaster occurs at a source site, businesses can access mission-critical data from a replica on a remote NetApp storage system for uninterrupted operation.

By providing a simple solution for replicating data across local, wide area, and FC networks, SnapMirror addresses six critical application areas.

DISASTER RECOVERY

If critical data is replicated to a different physical location, a serious disaster does not necessarily mean extended periods of data unavailability. The client can access replicated data across the network until the damage caused by the disaster is repaired. Recovery might include recovery from corruption, natural disaster at the production site, accidental deletion, sabotage, and so on. SnapMirror is often an integral part of disaster recovery plans. Data could be replicated to a destination system at a DR facility. Preferably, application servers would be replicated to this facility as well. If the DR facility needs to be made operational, applications can be switched over to the servers at the DR site and all application traffic directed to these servers for as long as necessary to recover the production site. When the source site is back online, SnapMirror can be used to transfer the data efficiently back to the production storage systems. After the production site takes over normal application operations again, SnapMirror transfers to the DR facility can resume without requiring a second complete data transfer.

REMOTE DATA ACCESS

The data replication capability of SnapMirror allows the distribution of large amounts of data throughout the enterprise, enabling read-only access to data at DR and remote locations. Remote data access not only provides faster access to data by local clients, it also results in a more efficient and predictable use of

expensive network and server resources. Storage administrators can replicate production data at a chosen time to minimize overall network utilization.

APPLICATION TESTING

Examples include test beds, database environments used for testing or simulating production environments, performance testing and monitoring, and development testing.

DISASTER RECOVERY TESTING

When NetApp FlexClone® is used along with SnapMirror, the remote site can be used for DR testing without interrupting production operations and DR replication.

REMOTE TAPE ARCHIVING

Some environments require off-site storage or off-site archiving. When a tape device is attached to a NetApp SnapMirror destination, data can be moved to tape periodically. SnapMirror can also be used for backup consolidation and for offloading tape backup overhead from production servers. This facilitates centralized backup operations, reducing backup administrative requirements at remote locations. It can also dramatically reduce overhead from stressful backup operations caused by small backup windows on production storage systems. Because backup operations are not occurring on the production systems, small backup windows are not as important.

LOAD SHARING

Load sharing is similar to the remote data access example described earlier in both implementation and benefit. The difference in a load-sharing implementation lies in the distance between the source and target volumes of the replication as well as in the performance goals associated with the replication implementation. In load sharing, the goal is to minimize the contention for critical application or database server resources by moving all read-only activities off the critical "transaction" server to a "mirror" or read-only server. The benefit can be twofold: (1) Optimize and partition network access to the data set, and (2) reduce CPU contention on the source application server by providing read-only and reporting access to the mirrored data. NetApp FlexCache™ can also be used for the purpose of load sharing.

1.5 BENEFITS OF SNAPMIRROR

- Block-level updates reduce bandwidth and time requirements.
- Data consistency can be maintained at a DR site.
- A DR plan can be tested without affecting production and replication.
- A DR location can keep many Snapshot™ copies at once; data can be restored to a point in time before data corruption occurred.
- Data can be replicated between dissimilar NetApp storage systems.
- A standard IP or FC network can be used for replication.
- SnapMirror Async supports one-to-one, one-to-many, many-to-one, or many-to-many replication, referred to as *cascading* and *multihop.*

2 OVERVIEW

2.1 THE BASICS

When mirroring asynchronously, SnapMirror replicates Snapshot copy images from a source volume or qtree to a partner destination volume or qtree, thus replicating source object data to destination objects at regular intervals. SnapMirror source volumes and qtrees are writable data objects whose data is to be replicated. The source volumes and qtrees are the objects that are normally visible, accessible, and writable by the storage system’s clients.

The SnapMirror destination volumes and qtrees are read-only objects, usually on a separate storage system, to which the source volumes and qtrees are replicated. Customers might want to use these read-only objects for auditing purposes before the objects are converted to writable objects. In addition, the read-only objects can be used for data verification. The more obvious use for the destination volumes and qtrees is to use them as true replicas for recovery from a disaster. In this case, a disaster takes down the source volumes or qtrees and the administrator uses SnapMirror commands to make the replicated data at the destination accessible and writable.

SnapMirror uses information in control files to maintain relationships and schedules. One of these control files, the `snapmirror.conf` file located on the destination system, allows scheduling to be maintained. This file, along with information entered by using the `snapmirror.access` option or the `snapmirror.allow` file is used to establish a relationship between a specified *source* volume, or qtree for replication, and the *destination* volume, or qtree where the mirror is kept.

Note: The `snapmirror.conf` file is not required to establish relationships.

The SnapMirror update process performs the following tasks:

1. Creates a Snapshot copy of the data on the source volume.
2. Copies the data to the destination, a read-only volume or qtree on the same system or on a remote destination system.
3. Updates the destination file system to reflect incremental changes occurring to the source.

The result of this process is an online, read-only dataset that is a point-in-time view of the data on the source at the time of the most recent update.

When using `snapmirror.conf`, the SnapMirror Snapshot copy creation and updates are controlled by a schedule that is local to the destination NetApp system. In a SAN environment, Snapshot copy creation involving logical unit numbers (LUNs) must be controlled by host systems. Scripts are set up to create Snapshot copies and to initiate the SnapMirror update to the remote storage system. For more information, refer to “Server Suite” in section 8.1, “NetApp Manageability Suite.”

2.2 SNAPSHOT COPY BEHAVIOR IN SNAPMIRROR

SnapMirror uses a Snapshot copy as a marker for a point in time for the replication process. A copy is kept on the source volume as the current point in time that both mirrors are in sync. When an update occurs, a new Snapshot copy is created and is compared against the previous Snapshot copy to determine the changes since the last update. SnapMirror marks the copies it needs to keep for a particular destination mirror in such a way that the `snap list` command displays the keyword `snapmirror` next to the necessary Snapshot copies. For more information, refer to “Snapshot Copy Behavior and Status in Volume SnapMirror” and “Snapshot Copy Behavior and Status in Qtree SnapMirror” in sections 2.4 and 2.5 respectively.

The `snapmirror destinations` command can be used to see which replica of a particular copy is marked as required at any time. On the source volume, SnapMirror creates the Snapshot copy for a particular destination and immediately marks it for that destination. At this point, both the previous copy and the new copy are marked for this destination. After a transfer is successfully completed, the mark for the previous copy is removed and deleted. Snapshot copies left for *cascade* mirrors from the destination also have the `snapmirror` tag in the `snap list` command output. (*Cascade mirrors* are a variation on the basic SnapMirror deployment, involving a writable source volume replicated to multiple read-only destinations, either one-to-one or one-to-many.)

Use the `snapmirror destinations -s` command to find out why a particular Snapshot copy is marked. This mark is kept as a reminder for SnapMirror to not delete a copy. This mark does not stop a user from deleting a copy marked for a destination that will no longer be a mirror; use the `snapmirror release` command to force a source to forget about a particular destination. This is a safe way to have SnapMirror remove its marks and clean up Snapshot copies that are no longer needed. Deleting a Snapshot copy that is marked as needed by SnapMirror is not advisable and must be done with caution in order not to disallow a mirror from updating. While a transfer is in progress, SnapMirror uses the busy lock on a Snapshot copy. This can be seen in the `snap list` command output. These locks do prevent users from deleting the Snapshot copy. The busy locks are removed when the transfer is complete.

For volume replication, SnapMirror creates a Snapshot copy of the whole source volume that is copied to the destination volume. For qtree replication, SnapMirror creates Snapshot copies of one or more source volumes that contain qtrees identified for replication. This data is copied to a qtree on the destination volume and a Snapshot copy of that destination volume is created.

A volume SnapMirror Snapshot copy name has the following format:

dest_name(sysid)_name.number

Example: `fasA(0050409813)_vol1.6 (snapmirror)`

dest_name is the host name of the destination storage system.

sysid is the destination system ID number.

name is the name of the destination volume.

number is the number of successful transfers for the Snapshot copy, starting at 1. Data ONTAP increments this number for each transfer.

A qtree SnapMirror Snapshot copy name has the following format:

dest_name(sysid)_name-src|dst.number

Example: `fasA(0050409813)_vol1_qtree3-dst.15 (snapmirror)`

dest_name is the host name of the destination storage system.

sysid is the destination system ID number.

name is the name of the destination volume or qtree path.

src|dst identifies the Snapshot copy location.

number is an arbitrary start point number for the Snapshot copy. Data ONTAP increments this number for each transfer.

In the output of the `snap list` command, Snapshot copies needed by SnapMirror are followed by the SnapMirror name in parentheses.

Caution: Deleting Snapshot copies marked `snapmirror` can cause SnapMirror updates to fail.

2.3 VOLUME SNAPMIRROR AND QTREE SNAPMIRROR

SnapMirror software provides the ability to replicate individual qtrees as well as whole volumes. The two types of replication are physical and logical. There are tradeoffs, including performance, manageability, configuration, and infrastructure resources. A comparison of the two is necessary to understand their implications.

2.4 SNAPMIRROR VOLUME REPLICATION

Volume SnapMirror has the following characteristics:

- SnapMirror volume replication can be synchronous or asynchronous.
- SnapMirror volume replication can occur only with volumes of the same type; that is, both volumes are traditional volumes or both are flexible volumes.
- SnapMirror volume replication copies a volume and *all* of its Snapshot copies to a destination volume.
- A destination volume that is set up for SnapMirror volume replication must first be set to restricted, read-only status.
- The destination volume (entire volume) is read-only unless it is made writable.

- SnapMirror volume replication is block-for-block replication; it transfers the file system verbatim. Therefore earlier major releases of Data ONTAP cannot understand file system transfers from a later major release. Data ONTAP 7.2 and 7.3 are examples of two different major release versions. Data ONTAP 7.2.2 and 7.2.3 are examples of same major release but different minor releases. Table 1 shows version restrictions for volume SnapMirror.

  Table 1) Volume SnapMirror version restrictions.

Volume SnapMirror Source	Volume SnapMirror Destination	Replication Possible?
Data ONTAP 7.2	Data ONTAP 7.3	Yes
Data ONTAP 7.3	Data ONTAP 7.2	No
Data ONTAP 7.2.x	Data ONTAP 7.2.y	Yes

- You can use volume-based SnapMirror to replicate data to a newer major release to assist in migrating to a newer Data ONTAP version. However you cannot do this in the reverse direction.
- Everything contained in a volume is replicated from one system or location to another, including metadata about the volume itself, such as language translation settings and other volume options stored as part of the volume, as well as all Snapshot copies of the volume.
- With flexible volumes, volume replication can be as granular as traditional deployments of qtree-based replication. The entire volume is replicated and can be very large, but it can also be very small and used in the same way that a qtree is used in traditional deployments.
- SnapMirror creates a Snapshot copy before performing the initial transfer. This copy is referred to as the baseline Snapshot copy. After performing an initial transfer of all data in the volume, volume SnapMirror sends to the destination only the blocks that have changed since the last successful replication. When SnapMirror performs an update transfer, it creates another new Snapshot copy and compares the changed blocks. These changed blocks are sent as part of the update transfer.

SNAPSHOT COPY BEHAVIOR AND STATUS IN VOLUME SNAPMIRROR

Table 2 shows how Snapshot copies are replicated between the volume SnapMirror source and destination systems and also the state of the Snapshot copies on both source and destination systems. The example assumes that fas1 is the source storage system and vol1 is the source volume; and that fas2 is the destination storage system and vol2 is the destination volume.

Table 2) Snapshot copies on source and destination for volume SnapMirror

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2
After volume initialization	`fas2(0099909262)_vol2.1 (snapmirror)`	`fas2(0099909262)_vol2.1`
After first update of vol2	`fas2(0099909262)_vol2.2 (snapmirror)`	`fas2(0099909262)_vol2.2` `fas2(0099909262)_vol2.1`
Create a Snapshot called *demo* on fas1:vol1	`demo` `fas2(0099909262)_vol2.2 (snapmirror)`	`fas2(0099909262)_vol2.2` `fas2(0099909262)_vol2.1`

Second update of vol2	`fas2(0099909262)_vol2.3 (snapmirror)`
	`demo`
		`fas2(0099909262)_vol2.3`
		`demo`
		`fas2(0099909262)_vol2.2`

Delete the Snapshot called *demo* on fas1:vol1	`fas2(0099909262)_vol2.3 (snapmirror)`	`fas2(0099909262)_vol2.3`
		`demo`
		`fas2(0099909262)_vol2.2`

Third update of vol2	`fas2(0099909262)_vol2.4 (snapmirror)`	`fas2(0099909262)_vol2.4` `fas2(0099909262)_vol2.3`

The `snapmirror` tag next to the Snapshot copy indicates a soft lock created by SnapMirror. Data ONTAP does not delete the Snapshot copies with soft locks but a user is able to delete these types of Snapshot copies. As seen above, in case of volume SnapMirror, if a Snapshot copy is deleted on the volume SnapMirror source, it is deleted on the destination at the next update. If a Snapshot copy is created on the volume SnapMirror source, it is created on the destination at the next update. This is not the case for qtree SnapMirror. See the next section for more information.

2.5 SNAPMIRROR QTREE REPLICATION

SnapMirror qtree replication has the following characteristics:

- SnapMirror qtree replication is available only in asynchronous mode.
- SnapMirror qtree replication occurs between qtrees regardless of the type of volume (traditional or flexible) in which the qtree resides.
- A destination qtree is read-only, but the volume on which it is located must be online and writable.
- SnapMirror qtree replication is logical replication; all of the files and directories in the source file system are created in the destination file system. Therefore replication can occur between any Data ONTAP releases.
- To replicate qtrees, qtree SnapMirror can either first create a Snapshot copy on the source volume that contains the qtree to be replicated or can also use an existing Snapshot copy on the source volume by specifying the –s flag. Note that NetApp Snapshot technology always operates on volumes, not on qtrees. This Snapshot copy contains a point-in-time copy of all of the data on the source volume, including both the data in the qtree to be replicated and also (presumably) other data that is not to be replicated.
- Qtree SnapMirror determines changed data by first looking through the inode file for inodes that have changed and changed inodes of the interesting qtree for changed data blocks. The SnapMirror software then transfers only the new or changed data blocks from this Snapshot copy that is associated with the designated qtree. On the destination volume, a new Snapshot copy is then created that contains a complete point-in-time copy of the entire destination volume, but that is associated specifically with the particular qtree that has been replicated.

Note: If the source file system contains a file type that cannot be represented on the destination file system, the replication fails. For example, Data ONTAP 7.0 supports files up to 16TB in size, whereas earlier Data ONTAP versions support files up to 4TB. If the source storage system is running Data ONTAP 7.0, the qtree that you want to replicate contains a file greater than 4TB, and the destination storage system is running an earlier version of Data ONTAP, the replication fails.

SNAPSHOT COPY BEHAVIOR AND STATUS IN QTREE SNAPMIROR

Table 3 shows how Snapshot copies are replicated between qtree SnapMirror source and destination systems and also the state of the Snapshot copies on both source and destination systems. The example assumes that fas1 is the source storage system and qt1 is the source qtree in volume vol1; and that fas2 is the destination storage system and qt2 is the destination qtree in volume vol2.

Table 3) Snapshot copies for source and destination for qtree SnapMirror

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2
After qtree initialization	`fas2(0099909262)_vol2_qt2-src.0 (snapmirror)`	`fas2(0099909262)_vol2_qt2-dst.2 (busy,snapmirror)`
After first update of qt2	`fas2(0099909262)_vol2_qt2-src.1 (snapmirror)`	`fas2(0099909262)_vol2_qt2-dst.4 (busy,snapmirror)`
Create a Snapshot called *demo* on fas1:vol1	`demo` `fas2(0099909262)_vol2_qt2-src.1 (snapmirror)`	`fas2(0099909262)_vol2_qt2-dst.4 (busy,snapmirror)`
Second update of vol2	`fas2(0099909262)_vol2_qt2-src.2 (snapmirror)` `demo`	`fas2(0099909262)_vol2_qt2-dst.6 (busy,snapmirror)`
Delete the Snapshot called *demo* on fas1:vol1	`fas2(0099909262)_vol2_qt2-src.2 (snapmirror)`	`fas2(0099909262)_vol2_qt2-dst.6 (busy,snapmirror)`
Third update of vol2	`fas2(0099909262)_vol2_qt2-src.3 (snapmirror)`	`fas2(0099909262)_vol2_qt2-dst.8 (busy,snapmirror)`

The `snapmirror` tag next to the Snapshot copy indicates a soft lock created by SnapMirror. The `busy,snapmirror` tag indicates a hard lock created by SnapMirror. A user cannot delete a hard lock. As seen above, in case of qtree SnapMirror, the same Snapshot copies do not exist on both source and destination systems.

2.6 KEY DIFFERENCES BETWEEN VOLUME AND QTREE SNAPMIRROR

The following differences between volume and qtree SnapMirror are not listed in order of importance.

Note: Both volume and qtree SnapMirror can operate over Ethernet and Fiber Channel or a combination of both. For more information, refer to section 2.10, “Multipath Support.”

Qtree SnapMirror	Volume SnapMirror
Unaffected by disk size or disk checksum differences between the source and destination irrespective of type of volumes used (traditional or flexible)	Unaffected by disk size or disk checksum differences between the source and destination if flexible volumes are used Affected by disk size or disk checksum differences between the source and destination if traditional volumes are used
Destination volume must have free space available equal to approximately 105% of the data being replicated	Destination volume must be equal or larger than the source volume
Sensitive to the number of files in a qtree due to the nature of the qtree replication process. The initial phase of scanning the inode map may be longer with larger number (tens of millions) of files	Not sensitive to the number of files in a volume
Qtree SnapMirror destinations can be placed on the root volume of the destination storage system	The root volume cannot be used as a destination for volume SnapMirror
Replicates only one Snapshot copy of the source volume where the qtree resides (the copy created by the SnapMirror software at the time of the transfer) to the destination qtree. Therefore, qtree SnapMirror	Replicates all Snapshot copies on the source volume to the destination volume. Similarly, if a Snapshot copy is deleted on the source system, volume SnapMirror deletes the Snapshot copy at the next update. Therefore volume

allows independent Snapshot copies on the source and destination	SnapMirror is typically recommended for disaster recovery scenarios, because the same data exists on both source and destination. Note that the volume SnapMirror destination always keeps an extra SnapMirror Snapshot copy
A qtree SnapMirror destination volume might contain replicated qtrees from multiple source volumes on one or more systems and might also contain qtrees or non-qtree data not managed by SnapMirror software	A volume SnapMirror destination volume is always a replica of a single source volume
Multiple relationships would have to be created to replicate all qtrees in a given volume by using qtree-based replication	Volume-based replication can take care of this in one relationship (as long as the one volume contains all relevant qtrees)
For low-bandwidth wide area networks, qtree SnapMirror can be initialized using the LREP tool available on NOW. See section 5, Network-Free Seeding for more information	Volume SnapMirror can be initialized using a tape device (SnapMirror to Tape) by using the `snapmirror store` and `snapmirror retrieve` commands. See section 5, Network-Free Seeding for more information
Qtree SnapMirror can only occur in a single hop. Cascading of mirrors (replicating from a qtree SnapMirror destination to another qtree SnapMirror source) is not supported	Cascading of mirrors is supported for volume SnapMirror
Qtree SnapMirror updates are not affected by backup operations. This allows a strategy called continuous backup, in which traditional backup windows are eliminated and tape library investments are fully used. SnapVault® software, discussed later in this report, is optimized for continuous backup applications	Volume SnapMirror updates can occur concurrently with a dump operation of the destination volume to tape by using the `dump` command or NDMP-based backup tools. However, if the volume SnapMirror update involves a deletion of the Snapshot copy that the dump operation is currently writing to tape, the SnapMirror update will be delayed until the dump operation is complete
The latest Snapshot copy is used by qtree SnapMirror for future updates if the –s flag is not used	Volume SnapMirror can use any common Snapshot copy for future updates
Qtrees in source deduplicated volumes that are replicated with qtree SnapMirror are full size at the destination	Source deduplicated volumes that are replicated with volume SnapMirror remain deduplicated at the destination
Even though the source volume is deduplicated, qtree SnapMirror will expand the data and send the entire data to the destination	Deduplication savings also extend to the bandwidth savings because volume SnapMirror only transfers unique blocks
Source and destination volumes can be independently deduplicated	Destination volume is read-only and therefore destination volume cannot be independently deduplicated. If deduplication savings are desired on the destination volume, then the source volume must be deduplicated
The files in the file system gain new identity (inode numbers etc.) in the destination system. Therefore, file handles cannot be migrated to the destination system	The files in the file system have the same identity on both source and destination system
LUN clones can be created on the destination volume, but not in the destination qtree	LUN clones cannot be created on the destination volume because the volume is read-only. However, LUN clones can be created on a FlexClone volume because the FlexClone volume is writable

The decision of which to use depends on individual site requirements. Volume SnapMirror and qtree SnapMirror can be freely mixed on both source and destination systems, although any individual destination volume can be a destination for only one or the other.

2.7 SUPPORT FOR VOLUME TYPES

Table 4 shows support for SnapMirror replication between the two volumes types.

Table 4) Volume replication support

Replication	Volume SnapMirror	Qtree SnapMirror
TradVol ←→ TradVol	Yes*	Yes
TradVol ←→ FlexVol	No	Yes
FlexVol ←→ FlexVol	Yes*	Yes

* Volume SnapMirror requires the destination system's Data ONTAP version to be same as or higher than that of the source system.

2.8 MODES OF SNAPMIRROR

SnapMirror can be used in three different modes: SnapMirror Async, SnapMirror Sync, and SnapMirror Semi-Sync.

SNAPMIRROR ASYNC

SnapMirror Async can operate on both qtrees and volumes. In this mode, SnapMirror performs incremental, block-based replication as frequently as once per minute.

The first and most important step in this mode involves the creation of a one-time, baseline transfer of the entire data set. This is required before incremental updates can be performed. This operation proceeds as follows:

1. The source storage system creates a Snapshot copy (a read-only, point-in-time image of the file system). This copy is called the baseline copy.
2. All data blocks referenced by this Snapshot copy and any previous copies are transferred in case of volume SnapMirror and written to the destination file system. Qtree SnapMirror only copies the latest Snapshot copy.
3. After the initialization is complete, the source and destination file systems have at least one Snapshot copy in common.

After the initialization is complete, scheduled or manually triggered updates can occur. Each update transfers only the new and changed blocks from the source to the destination file system. This operation proceeds as follows:

1. The source storage system creates a Snapshot copy.
2. The new copy is compared to the baseline copy to determine which blocks have changed.
3. The changed blocks are sent to the destination and written to the file system.
4. After the update is complete, both file systems have the new Snapshot copy, which becomes the baseline copy for the next update.

Because asynchronous replication is periodic, SnapMirror Async is able to consolidate the changed blocks and conserve network bandwidth. There is minimal impact on write throughput and write latency.

SNAPMIRROR SYNC

Certain environments have very strict uptime requirements. All data that is written to one site must be mirrored to a remote site or system synchronously. SnapMirror Sync mode is a mode of replication that

sends updates from the source to the destination as they occur, rather than according to a predetermined schedule. This guarantees that data written on the source system is protected on the destination even if the entire source system fails. SnapMirror Semi-Sync mode, which minimizes data loss in a disaster while also minimizing the extent to which replication affects the performance of the source system, is also provided.

No additional license fees need to be paid to use this feature, although a free special license `snapmirror_sync` must be installed; the only requirements are appropriate hardware, the correct version of Data ONTAP, and a SnapMirror license for each storage system. Unlike SnapMirror Async mode, which can replicate volumes or qtrees, SnapMirror Sync and Semi-Sync modes work only with volumes. SnapMirror Sync can have a significant performance impact and is not necessary or appropriate for all applications.

The first step in synchronous replication is a one-time, baseline transfer of the entire data set. After the baseline transfer is completed, SnapMirror will transition into synchronous mode with the help of NVLOG and CP forwarding. Once SnapMirror has transitioned into synchronous mode, the output of a SnapMirror status query shows that the relationship is "In-Sync."

For complete information about NVLOG forwarding and CP synchronization, refer to TR 3326, *SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations,* available on NOW.

SNAPMIRROR SEMI-SYNC

SnapMirror provides a semisynchronous mode, also called SnapMirror Semi-Sync. This mode differs from the synchronous mode in two key ways:

1. User writes don't need to wait for the secondary or destination storage to acknowledge the write before continuing with the transaction. User writes are acknowledged immediately after they are committed to the primary or source system's memory.
2. NVLOG forwarding is not used in semisynchronous mode. Therefore SnapMirror Semi-Sync might offer faster application response times. This mode makes a reasonable compromise between performance and RPO for many applications.

Note: Before Data ONTAP 7.3, SnapMirror Semi-Sync was tunable, so that the destination system could be configured to lag behind the source system by a user-defined number of write operations or seconds. This was configurable by specifying a variable called `outstanding` in the SnapMirror configuration file. Starting in Data ONTAP 7.3, the `outstanding` parameter functionality is removed and there is a new mode called `semi-sync`. When using `semi-sync` mode, only the consistency points are synchronized. Therefore this mode is also referred to as CP Sync mode.

Configuration of semisynchronous mode is very similar to that of synchronous mode; simply replace `sync` with `semi-sync`, as in the following example:

```
fas1:vol1   fas2:vol1   -   semi-sync
```

For more information about SnapMirror Semi-Sync, refer to TR 3326, *SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations,* available on NOW.

VISIBILITY INTERVAL

The visibility interval specifies how often the source system takes a Snapshot copy in SnapMirror Sync and Semi-Sync modes. The default interval is 3 minutes. Because the same Snapshot copies exist on both source and destination system, this means that updated data in the file system is visible on the SnapMirror destination system in 3-minute increments. This generates more Snapshot creations and deletions, so if this value is small, a performance impact might be seen on the source volume. NetApp recommends using the default value, unless a different value is completely necessary. This value is set with an option in the `/etc/snapmirror.conf` file and can be set on an individual volume basis.

For more information about visibility interval's and its impact, refer to TR 3326, *SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations,* available on NOW.

2.9 CONTROL FILES

ACCESS AND SECURITY

snapmirror.access

The `snapmirror.access` option specifies which SnapMirror destination storage systems can initiate transfers and which network interfaces they can use. This is the preferred method for controlling SnapMirror access on a SnapMirror source storage system.

On the source storage system console, use the `options snapmirror.access` command to specify the host names of storage systems that are allowed to copy data directly from the source storage system. For example:

```
options snapmirror.access host=fas2
```

The syntax for specifying which storage systems are allowed access to the server is the same for SNMP, telnet, and rsh and is described in the Data ONTAP man pages and in the product documentation available on NOW.

Note: If you set the `snapmirror.access` option to legacy, the `snapmirror.allow` file is used instead.

/etc/snapmirror.allow

You can generate a `snapmirror.allow` file in the `/etc` directory on the source storage system. The `/etc/snapmirror.allow` file specifies the host names of storage systems that are allowed to copy data directly from the source storage system. For more information about the `options` command, see the Data ONTAP man pages or the product documentation available on NOW.

/etc/snapmirror.conf

This is the core configuration file for all SnapMirror operations. The `/etc/snapmirror.conf` file defines the relationship between the source and the destination, the schedule used by the destination to copy data, and the arguments that control SnapMirror when copying data. This file resides on the SnapMirror destination system.

DISTRIBUTION

You can create a single `/etc/snapmirror.conf` file for your site and copy it to all the storage systems that use SnapMirror. This file can contain entries pertaining to other storage systems. For example, the `/etc/snapmirror.conf` file on `fas2` can contain an entry for copying a volume from `fas3` to `fas4`. When `fas2` reads the `/etc/snapmirror.conf` file, it ignores the entries for other storage systems. This relationship between `fas3` and `fas4` is considered invalid and therefore ignored. However, each time the file is read, a warning message is displayed on the system console for each line that is ignored.

There is no limit on the total number of *entries* in the `/etc/snapmirror.conf` file; however, there is a limit of 712 *valid* relationships in the file. Entries beyond the entry limit for each storage system are ignored, and a warning message is displayed on the system console.

In an active-active configuration, the limit on the maximum number of entries applies to the storage system pair combination. If one controller in an active-active configuration fails, the limit stays at 712 entries.

Note: This limitation is different from the maximum number of simultaneous (or concurrent) replications you can have on a storage system. For that information, refer to section 3.2, “Concurrent Replication Operations” or the *Data Protection Online Backup and Recovery Guide* on NOW.

CONFIGURATION CHANGES

If SnapMirror is enabled, changes to the `/etc/snapmirror.conf` file take effect within 2 minutes. If SnapMirror is not enabled, changes to the `/etc/snapmirror.conf` file take effect immediately after you enter the `snapmirror on` command to enable SnapMirror.

2.10 MULTIPATH SUPPORT

More than one physical path between a source and a destination system might be desired for a mirror relationship. SnapMirror Async (volume and qtree), SnapMirror Sync, and SnapMirror Semi-Sync support multiple paths for replication. Multipath support allows SnapMirror traffic to be load balanced between these paths and provides for failover in the event of a network outage. Specifically, SnapMirror supports up to two paths for a particular relationship. Therefore each replication relationship can be configured to use a distinct

multipath connection. These multipath connections can be Ethernet, Fibre Channel, or a combination of the two. There are two modes of multipath operation:

Multiplexing mode. Both paths are used simultaneously, load-balancing transfers across the two. When a failure occurs, the load from both transfers moves to the remaining path.

Failover mode. One path is specified as the primary path in the configuration file. This path is the desired path. In case of failure, the second path is used.

Best practice: NetApp recommends multipath to improve availability of the replication network.

3 OPERATIONAL BEHAVIORS

When evaluating your SnapMirror implementation, it is important to consider the following common SnapMirror behaviors and to understand when and why they might occur.

ACTIVE-ACTIVE CONFIGURATION

The SnapMirror product complements NetApp active-active configuration technology by providing an additional level of recoverability. If a catastrophe disables access to an active-active pair of storage systems, one or more SnapMirror volumes can be immediately accessed in read-only mode while recovery takes place. If read-write access is required, the mirrored volume can be converted to a writable volume while the recovery takes place. If SnapMirror is actively updating data when a takeover or giveback operation is run, the update aborts leaving the destination volume in the state of the last completed update. After the takeover or giveback operation is completed, SnapMirror transfer continues as before from a restart checkpoint. No specific additional steps are required for the implementation of SnapMirror in an active-active configuration environment. For more information on NetApp active-active configuration technology and takeover and giveback scenarios, refer to the *Data ONTAP System Administrator's Guide*, available on NOW.

DISK GEOMETRY

In case of traditional volumes, volume SnapMirror performance was affected due to disk geometry. If the source disks were not the same size as the destination disks, problems occurred that resulted in data not being properly distributed across some spindles. For example, data cleanly striped across three 5GB drives on the source that is replicated to a destination system with 15GB disks would result in the data being laid out on one of the destination system spindles. Qtree SnapMirror does not have this performance issue.

Flexible volumes in Data ONTAP 7G eliminated the performance impact due to geometry mismatch for volume SnapMirror as well. Destination volumes no longer have to contain the same number of disks or the same size disks as the source volumes, allowing more efficient deployment of resources. With flexible volumes, SnapMirror is no longer bound by the physical limitations of copying physical disks block for block. The physical nature of volume SnapMirror has been virtualized.

The size of a flexible volume can be changed dynamically. It can also act as a hard quota for a group or project assigned to it. In each volume, user- and group-level quotas as well as qtrees can be used to obtain finer granularity of quota management.

GROWING THE DESTINATION VOLUME

Volume SnapMirror requires the destination volume to be the same or larger than the source volume. If the source volume has been manually or automatically grown and is larger than the destination volume, SnapMirror updates will fail. The destination volume will need to be resized accordingly to successfully resume the updates. For more information, refer to the “Growing Destination Volume” in section 4.

CASCADING

A variation on the basic SnapMirror deployment and function involves mirroring from established mirrors to more SnapMirror destinations. An example cascade configuration with two hops is shown below.

The function of this deployment is to make a uniform set of data available on a read-only basis to users from various locations throughout a network and to allow updating that data uniformly at regular intervals. However, cascading a SnapMirror replication from A to B to C and so on is allowed only with volume SnapMirror. Qtree SnapMirror does not support replication for more than one hop. During qtree SnapMirror replication, mapping takes place between source inodes and destination inodes. For example, suppose that the `/vol/vol1/qt7/user/email.txt` file has the inode number 456. When this qtree is transferred to the destination by using qtree SnapMirror (such as vol1_rpl), the `/vol/vol1_rpl/qt7_rpl/user/email.txt` file might have the inode number 5987432.

To be able to apply a modification on number 456 to number 5987432, qtree SnapMirror needs to keep a map of the inodes. Mapping the inodes is necessary because qtree SnapMirror is taking qtrees from different volumes and mirroring them into one common volume. Files from those qtrees might have the same inode number (because they come from different volumes or storage systems). Therefore qtree SnapMirror reallocates the inodes, so that it doesn't have problems with conflicts in numbering. In addition, this inode mapping would cause problems because mapping the state can become confusing in a cascade, so this configuration is not allowed in a cascade configuration.

Note: SnapMirror Sync and Semi-Sync cannot be cascaded. This means that you cannot configure multiple hops of SnapMirror Sync and Semi-Sync. However SnapMirror Async (volume SnapMirror) can be cascaded from a SnapMirror Sync or Semi-Sync destination.

Table 5) Cascading support

Cascade Configuration	Support
Sync/Semi-Sync → Volume SnapMirror	Yes
Sync/Semi-Sync → Sync/Semi-Sync	No
Sync/Semi-Sync → Qtree SnapMirror	No
Volume SnapMirror → Volume SnapMirror	Yes
Volume SnapMirror → Sync/Semi-Sync	No
Qtree SnapMirror → Sync/Semi-Sync	No
Volume SnapMirror → Qtree SnapMirror	Yes
Qtree SnapMirror → Volume SnapMirror	Yes
Qtree SnapMirror → Qtree SnapMirror	No

SNAPSHOT COPY PROPAGATION IN CASCADE CONFIGURATION

This section demonstrates Snapshot copy propagation behavior in the following cascade configuration scenarios with examples.

Single hop volume SnapMirror:

This configuration involves volume SnapMirror replication between two systems, fas1 and fas2.

fas1:v2 → fas2:v2

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2
After volume initialization	fas2(0099909262)_v2.1 (snapmirror)	fas2(0099909262)_v2.1
After first update of v2	fas2(0099909262)_vol2.2 (snapmirror)	fas2(0099909262)_vol2.2 fas2(0099909262)_vol2.1

Snapshot copy behaviors to note:

1. SnapMirror creates a soft lock on the Snapshot copy of the source volume (`snapmirror` tag)
2. Destination system carries an extra Snapshot copy

Dual hop volume SnapMirror:

This configuration involves volume SnapMirror replication among three systems, fas1, fas2, and fas3.
fas1:v2 → fas2:v2 → fas3:v2

Note that in the above configuration, fas1:v2 to fas2:v2 and fas2:v2 to fas3:v2 transfers cannot occur at the same time.

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2	Snapshot Copies on fas3
1) After volume initialization on fas2	fas2(0099909262)_v2.1 (snapmirror)	fas2(0099909262)_v2.1	
2) Volume SnapMirror update on fas2	fas2(0099909262)_v2.2 (snapmirror)	fas2(0099909262)_v2.2 fas2(0099909262)_v2.1	
3) fas3:v2 initialization	fas2(0099909262)_v2.2 (snapmirror)	fas2(0099909262)_v2.2 (snapmirror) fas2(0099909262)_v2.1	fas2(0099909262)_v2.2 fas2(0099909262)_v2.1
4) Volume SnapMirror update on fas2	fas2(0099909262)_v2.3 (snapmirror) fas2(0099909262)_v2.2 (snapmirror)	fas2(0099909262)_v2.3 fas2(0099909262)_v2.2 (snapmirror)	fas2(0099909262)_v2.2 fas2(0099909262)_v2.1
5) Volume SnapMirror update on fas3	fas2(0099909262)_v2.3 (snapmirror) fas2(0099909262)_v2.2 (snapmirror)	fas2(0099909262)_v2.3 (snapmirror) fas2(0099909262)_v2.2	fas2(0099909262)_v2.3 fas2(0099909262)_v2.2

Snapshot copy behaviors to note:

1. There is an extra Snapshot copy on fas2 (destination) after the first SnapMirror update (step 2)
2. System fas3 also has the same number of Snapshot copies as fas2 after step 3 because there is a volume SnapMirror relationship between fas2 and fas3 systems
3. A new soft lock exists on fas2:v2 after step 3 because fas2:v2 is now the volume SnapMirror source for fas3:v2.
4. After step 4, the source system, fas1 contains two SnapMirror Snapshot copies. This is because the Snapshot copy `fas2(0099909262)_v2.2` is locked by fas2 system as it is required to continue to perform SnapMirror updates with fas3 system. This Snapshot copy on fas1 system is also used to perform direct SnapMirror updates with fas3 system in case fas2 system meets disaster
5. After an update is performed on fas3 system (step 5), the soft lock now exists on the latest SnapMirror Snapshot copy (`fas2(0099909262)_v2.3`) because this is the new baseline SnapMirror Snapshot copy between fas2 and fas3 systems.

Single hop qtree SnapMirror:

This configuration involves qtree SnapMirror replication between two systems, fas1 and fas2.
fas1:vol1/qt1 → fas2:vol2/qt2

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2
After qtree initialization and update on fas2	fas2(0099909262)_vol2_qt2-src.3 (snapmirror)	fas2(0099909262)_vol2_qt2-dst.8 (busy,snapmirror)

Snapshot copy behaviors to note:

1. Qtree SnapMirror Snapshot copy names are not identical. This is because the destination volume may contain other qtree or volume data besides qt2
2. The destination volume Snapshot copy has a hard lock created by SnapMirror (`busy,snapmirror`)
3. The source volume Snapshot copy has a soft lock created by SnapMirror (`snapmirror`)

Dual hop qtree SnapMirror and volume SnapMirror:
This configuration involves qtree SnapMirror replication in the first hop and volume SnapMirror replication in the second hop.
fas1:vol1/qt1 → fas2:vol2/qt2; fas2:vol2 → fas3:vol3

Timeline	Snapshot Copies on fas1	Snapshot Copies on fas2	Snapshot Copies on fas3
1) After qtree initialization and update on fas2	fas2(0099909262)_vol2_qt2-src.3 (snapmirror)	fas2(0099909262)_vol2_qt2-dst.8 (busy,snapmirror)	
2) After fas3:vol3 initialization	fas2(0099909262)_vol2_qt2-src.3 (snapmirror)	fas3(0099909261)_vol3.1 (snapmirror) fas2(0099909262)_vol2_qt2-dst.8 (busy,snapmirror)	fas3(0099909261)_vol3.1 fas2(0099909262)_vol2_qt2-dst.8
3) After a qtree SnapMirror update on fas2	fas2(0099909262)_vol2_qt2-src.4 (snapmirror) fas2(0099909262)_vol2_qt2-src.3 (snapmirror)	fas2(0099909262)_vol2_qt2-dst.10 (busy,snapmirror) fas3(0099909261)_vol3.1 (snapmirror) fas2(0099909262)_vol2_qt2-dst.8 (snapmirror)	fas3(0099909261)_vol3.1 fas2(0099909262)_vol2_qt2-dst.8
4) After a volume SnapMirror update on fas3	fas2(0099909262)_vol2_qt2-src.4 (snapmirror) fas2(0099909262)_vol2_qt2-src.3 (snapmirror)	fas3(0099909261)_vol3.2 (snapmirror) fas2(0099909262)_vol2_qt2-dst.10 (busy,snapmirror) fas2(0099909262)_vol2_qt2-dst.8	fas3(0099909261)_vol3.2 fas2(0099909262)_vol2_qt2-dst.10 fas3(0099909261)_vol3.1 fas2(0099909262)_vol2_qt2-dst.8

Important behaviors to note:

1. System fas3 also has the same number of Snapshot copies as fas2 after step 2 because there is a volume SnapMirror relationship between fas2 and fas3 systems
2. System fas1 retains the extra qtree SnapMirror Snapshot copy (`fas2(0099909262)_vol2_qt2-src.3`) in step 3 because fas3 will be able to resynchronize with fas1 in the event fas2 meets with disaster
3. System fas3 has an extra volume SnapMirror Snapshot copy after the first volume SnapMirror update on fas3 system.

LOGGING

The SnapMirror log file (located in `/etc/logs/snapmirror.log`) records the start and end of an update and other significant SnapMirror events. If problem exists with updates, review the log file to see what happened since the last successful update. Because the log file is kept on the source and destination storage systems, quite often the source or the destination system might log the failure, and the other partner knows only that there was a failure. For this reason, review logs at both the source and the destination systems to get the most information about a failure. The log file contains the start and end times of each

transfer, along with the amount of data transferred. It can be useful to look back and see the amount of data needed to make the update and the amount of time the updates take.

Note: The time versus data sent might not be an accurate measure of the achieved network throughput because the transfer is not constantly sending data.

DATA ONTAP VERSIONS AND RESYNC

Qtree SnapMirror is not affected by Data ONTAP versions of source and destination systems. Volume SnapMirror requires the destination to be at the same or higher major release of Data ONTAP as that of the source. If the destination (the DR site) is running a higher major release version of Data ONTAP than that at the source (production site), bear in mind that the production site system will need to be upgraded if the newly written data at the DR site needs to be resynchronized to the production site (reversing the SnapMirror relationship).

SnapMirror resynchronization does not always require a full level 0 transfer. If both the source and destination have at least one Snapshot copy in common, SnapMirror computes and transfers only the changed blocks since the common Snapshot copy. In the absence of the common Snapshot copy, resynchronization requires a full transfer.

DATA CHANGE RATE

Using the `snap delta` command, you can display the rate of change stored between two Snapshot copies as well as the rate of change between a Snapshot copy and the active file system. Data ONTAP displays the rates of change in two tables. The first table displays the rates of change between successive Snapshot copies. The second table displays a summary of the rate of change between the oldest Snapshot copy and the active file system.

SNAPMIRROR AND LUNS

If the volumes or qtrees contain LUNs, the LUNs on the SnapMirror destination system are read-only, online and unmapped starting in Data ONTAP 7.2. You can then map the LUNS and mount them read-only or use FlexClone to create a clone of the volume containing the LUNs and mount them read-write. This can be done without interrupting SnapMirror replication operations. Note that the use of FlexClone requires a license. In case of qtree SnapMirror, LUNS can be cloned in the volume using the `lun clone` command, which is available with Data ONTAP software.

SPACE GUARANTEES

When users require additional space, the administrator can increase the size of an aggregate volume by assigning additional disks to it. In a SnapMirror configuration, overcommitting the aggregate allows more efficient use of disk space on the destination. Only the data that is used on the SnapMirror source is used in the flexible volume on the SnapMirror destination. If that SnapMirror destination is broken, the disk usage is deducted from the overall aggregate. Unless mirrors are broken, you can have many source volumes of varying sizes all mapped to destination flexible volumes.

Overcommitting Aggregates on the Source System

To overcommit an aggregate volume, create flexible volumes with a guarantee of *none* or *file* so that the volume size is not limited by the aggregate size. The total size of the flexible volumes can be larger than the containing aggregate.

Overcommitting Aggregates on the Destination System

The disadvantage of overcommitting an aggregate is that SnapMirror updates will fail when the volume runs out of space. Another disadvantage is that not all volumes can be guaranteed if they all need to be made writable at once by breaking the SnapMirror relationship.

Prior to Data ONTAP 7.3, as long as the destination volume is a SnapMirror destination (replica), the guarantee is volume-disabled. Subsequently, when the destination is broken, the guarantee mode is the same as the volume mode.

Starting in Data ONTAP 7.3, it is possible to set guarantees on the SnapMirror destination volume so that the SnapMirror updates never fail on that volume. The default behavior is that the volume guarantees are turned off.

Therefore, for a 1TB SnapMirror source volume that is 75% full, the SnapMirror destination volume (or replica) needs 750GB with the guarantee disabled and the full 1TB with the guarantee enabled.

3.1 UPDATE FAILURES

If a manually issued update fails for any reason, the user is informed. The update is not tried automatically because the user is in a position to reissue the command. If a scheduled transfer fails, and if the error is "retriable," it is retried at the next minute. If it fails for nonretriable errors, such as user abort, or if the volume SnapMirror source denied the transfer for any reason, it is not retried at the next minute. Whether the error is retriable or nonretriable, an update is always attempted whenever the schedule in `snapmirror.conf` specifies that the update should be performed.

If an update is in progress when another is scheduled to occur, SnapMirror starts another transfer as soon as the transfer is complete. However, if three updates pass while the current transfer is in progress, SnapMirror does only one more update; it does not go back and run updates that have been made obsolete by those scheduled later.

If a transfer fails and has to be retried, it is not generally started from the beginning. SnapMirror makes a restart checkpoint every 5 minutes during a transfer. If a restart checkpoint exists and if the baseline and incremental SnapMirror Snapshot copies exist, SnapMirror restarts the previous transfer where it left off. If not, SnapMirror creates a new Snapshot copy and starts a new transfer. Some of the conditions that prevent SnapMirror from restarting from a checkpoint are hostname changes, volume name changes, source volume size changes, and Data ONTAP version upgrades and downgrades.

3.2 CONCURRENT REPLICATION OPERATIONS

A SnapMirror replication consists of two operations, one on the source side of the transfer and the other on the destination side. Therefore, if a storage system is the source of one replication and the destination of another replication, it uses two replication operations. Similarly, if a storage system is the source and the destination of the same replication, it uses two replication operations. Migrating from traditional volumes to FlexVol volumes with qtree SnapMirror within the same controller is an example of the same storage system being both a source and a destination.

The number of concurrent replication operations in Data ONTAP 7.3 increased dramatically for asynchronous SnapMirror (both volume and qtree). The increase depends on the platform. For example, FAS3050 supported 16 concurrent operations in Data ONTAP 7.2. Data ONTAP 7.3 supports up to 50 concurrent operations for volume SnapMirror.

For more information on the maximum number of concurrent replication operations that each storage system model can support, refer to the *Data Protection Online Backup and Recovery Guide* on NOW for the appropriate Data ONTAP release.

A storage system might not reach the maximum number of concurrent replication operations for the following reasons:

- Storage system resources, such as CPU usage, memory, disk bandwidth, or network bandwidth, are taken away from SnapMirror or SnapVault operations.
- Each storage system in a high-availability (HA) configuration has the maximum number of concurrent replication operations. If a failover occurs, the surviving storage system cannot process more than the maximum number of concurrent replication operations specified for that storage system. These can be operations that were scheduled for the surviving storage system, the failed-over storage system, or both. For example, each FAS3050 in a cluster can run a maximum of 16 concurrent replication operations. If one FAS3050 fails over to the other, it still has a maximum of 16 operations, which can be operations that were scheduled by the surviving FAS3050, the failed FAS3050, or both.
- Before Data ONTAP 7.3, concurrent operations were reduced to half when ATA drives are present in the storage system unless a NearStore® option license is installed. Starting with Data ONTAP 7.3, this limitation due to ATA drives is removed.

3.3 NEARSTORE PERSONALITY

Starting with Data ONTAP 7.2, a software license option called the NearStore Personality option (`nearstore_option`) has been introduced to use the FAS storage systems as a secondary storage

system. Please check the product manuals on NOW for specific Data ONTAP version requirements for this license. The goal of this license option is to provide increased concurrent streams when FAS storage systems are used as destinations for SnapMirror/SnapVault transfers and to enable SnapVault for NetBackup™. This license option should not be installed on primary storage systems where performance is paramount.

Before Data ONTAP 7.3, the concurrent operations were halved when ATA drives are added to the system. When the NearStore option license is installed, the concurrent operations are doubled again. An example follows:

Without the license, the total concurrent volume SnapMirror operations limits for the FAS3020 in Data ONTAP 7.2 are 16 for FC drives and 8 for FC/ATA drives. With the NearStore option license, the total concurrent volume SnapMirror operations limits for the FAS3020 are 16 regardless of FC or ATA drives in the system.

Note: Because concurrent operations limits change with Data ONTAP versions and platforms, please refer to the appropriate release documentation for that particular version. The concurrent operation limits are total for any given system and not cumulative. If the FAS3050 system allows 50 maximum concurrent replication operations for volume SnapMirror or 16 maximum concurrent replication operations for SnapMirror Sync, the system will not allow both the 50 volume SnapMirror transfers and 16 SnapMirror Sync transfers at any given time.

3.4 SYSTEMWIDE THROTTLE

Starting with Data ONTAP 7.2, there is a system-wide option to limit the total bandwidth used by all transfers at any time. This can be either the transmit bandwidth on the source or the receive bandwidth on the destination or both. The per-transfer throttle from `snapmirror.conf` will still be applied. When both per-transfer and system-wide throttling are configured, throttling at system wide is applied only if the combined bandwidth used by all the relationships goes above the system-wide throttling value. System-wide throttling is enabled by using three new options using the `options` command.

`replication.throttle.enable`

This option enables global network throttling of SnapMirror and SnapVault transfers. The default value for this option is off.

`replication.throttle.incoming.max_kbs`

This option is set on the destination system. This option specifies the maximum total bandwidth used by all the incoming SnapMirror and SnapVault transfers, specified in kilobytes/sec. The default value for this option is Unlimited, which means that there is no limit on total bandwidth used. This option is valid only when the `replication.throttle.enable` option is on.

`replication.throttle.outgoing.max_kbs`

This option is set on the source system. This option specifies the maximum total bandwidth used by all the outgoing SnapMirror and SnapVault transfers specified in kilobytes/sec. The default value for this option is Unlimited, which means that there is no limit on total bandwidth used. This option is valid only when the `replication.throttle.enable` option is on.

3.5 DYNAMIC THROTTLE

Starting in Data ONTAP 7.1, an active SnapMirror relationship can be throttled to decrease the amount of bandwidth it uses. This dynamic relationship does not need to be stopped for the new throttle value to take effect.

The syntax and example follow:

`snapmirror throttle <n> <system>:<destination path>`

`<n>` is the new throttle value in kilobytes per second.

Example:

```
fas1> snapmirror throttle 2000 fas2:/vol/vol1/home
```

The new value is used only for current transfer. the next scheduled transfer will use the throttle value specified in the `/etc/snapmirror.conf` file. This command can be used *only* when the transfer is active

If the throttle value for the next scheduled transfer needs to be changed, then the value in the SnapMirror configuration file should be modified. The command can be run from either the source or the destination.

There is another way to change the throttle value for an active transfer, by changing the value in the `/etc/snapmirror.conf` file. This change takes effect in 2 minutes. The `snapmirror throttle` command does not change the throttle value specified in `/etc/snapmirror.conf`.

3.6 FIREWALL CONFIGURATION

SnapMirror uses the typical socket/bind/listen/accept sequence on a TCP socket.

SnapMirror Async:

SnapMirror source system binds on port 10566. A firewall configuration must allow requests to this port on the SnapMirror source system. When using multipath connection, the destination system listens on port 10565.

SnapMirror Sync and SnapMirror Semi-Sync:

SnapMirror requires additional TCP ports to be open. The source system listens on TCP ports 10566 and 10569. The destination system listens on TCP ports 10565, 10567, and 10568. Therefore, a range of TCP ports from 10565 to 10569 is recommended.

4 BEST PRACTICES AND RECOMMENDATIONS

The following best practices refer primarily to the asynchronous mode of SnapMirror. The best practices for synchronous mode of SnapMirror are covered in a separate technical report (TR 3326).

GROWING DESTINATION VOLUME

For volume SnapMirror, the destination volume must be same as or larger than the source volume. Volume SnapMirror updates fail if the destination volume is smaller than the source volume. The best practice is to keep the source and destination volumes the same size to avoid any potential issues in failover and failback scenarios, because the source and destination roles are reversed during these scenarios.

If the source volume size has been increased by `autogrow` or by manual process, the destination volume needs to be matched with the source volume. There are different ways to handle the size mismatch. The first way is to provision the destination volume larger than the potential size to which the source volume would ever grow. In this scenario, the SnapMirror update matches the source volume size because the underlying volume has sufficient space. If the destination volume is not provisioned to have enough space, then first turn the `fs_size_fixed` option off on the destination volume and then resize the destination volume to the same size as the source volume by using the `vol size` command. The next SnapMirror update command will then match the destination volume size to be the same size as the source volume.

TCP WINDOW SIZE

The TCP window is the amount of data that a source can send on a connection before it requires acknowledgment from the destination that the data was received. The default window size for SnapMirror operations is 1,994,752 bytes (2MB). There are a couple of issues with large window size. First, TCP does not respond well to packet loss when it has a very large window size. The classic adjustment is to halve the window size when packet loss is encountered and it takes a very long time to increase the window size again. Another issue is that if the window is larger than the amount of in-flight data, the data has queued somewhere. An optimal TCP window size is very difficult to calculate. Use the following simple formula to calculate the theoretical window size:

Window Size = Round Trip Time (RTT) X Desired Rate

So if you have a 10Mb/sec network and the average RTT is 100ms, the window size should be:

0.01 seconds X 10 Mb/second = 125,000 bytes.

```
options snapmirror.window_size 125000
```

The calculated window size is merely a theoretical window size and might not be the optimal window size for your network. For most scenarios, it is not required to change the TCP window size. However, when experiencing SnapMirror throughput issues, it is worth exploring the effects of SnapMirror throughput by changing the default TCP window size to the calculated TCP window size.

Finally, note that synchronous replication with SnapMirror is not feasible over large distances such as wide area networks, which typically have large round-trip time and packet loss, resulting in performance impact on the primary workload.

REPLICATION NETWORK CONFIGURATION

When possible, use a private network between source and destination for replication purpose. This isolates replication traffic from the client traffic. Otherwise, these two types of traffic compete for bandwidth. Sometimes, SnapMirror might need to be throttled for two primary reasons: decrease WAN bandwidth use by SnapMirror; and decrease the storage system's resource consumption by SnapMirror. To figure out the amount of bandwidth required, follow the simple steps:

1. Find out the peak data change rate (using `snap diff` command)
2. Find out the RPO (or SnapMirror update interval)

Once you have the above, you can now calculate the minimum theoretical required bandwidth. An example follows:

Assume you have a data set of 1TB with daily change rate of 5% (or 50GB per day or 2GB/hour). Assume your RPO requirement is 1 hour. This means that you will need to complete the SnapMirror transfer of 2GB in one hour or approx. 4.7Mbps. This is the minimum theoretical bandwidth required to complete the SnapMirror transfer. The actual throughput will depend on the storage system utilization, round trip latency of the network link, and the network pipe utilization by other applications.

Also keep in mind data change rate is not uniform through out the day even though the above example assumes same data change rate through out the 24-hour period. Use the peak data change rate in your scenario to calculate the minimum bandwidth requirement.

REPLICATION FREQUENCY

Although it is possible to do replication updates every minute, it is not recommended. The first step in a SnapMirror update involves computation of changed blocks. This can be a CPU-intensive process. The storage system can spend a lot of time in computing changed blocks when SnapMirror updates are set up to run every minute on multiple volumes. This in turn could affect the primary workloads. The other issue is that the entire SnapMirror update process must finish within the minute for the all the volumes before the next update starts again. There might not be sufficient time to calculate the block changes and transfer the data within this short period. Therefore, asynchronous SnapMirror updates at every minute are not recommended.

If the RPO requirements are very low (< 3 minutes or so), consider the use of SnapMirror Semi-Sync. For more information, refer to the design guidelines in TR 3326, *SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations* on NOW.

SNAPSHOT SCHEDULES

For optimal performance, make sure that the SnapMirror updates and Snapshot schedules (`snap sched`) do not occur at the same time.

DESTINATION QTREE NAMES

Destination qtree names cannot contain wildcards and special characters. There is also a 64-character limit on the length of names. A qtree name must be present and appended to the full volume path. The full qtree path must also be preceded with `/vol` before the volume name that the qtree resides in. For non-qtree data, "-" is specified in place of the qtree name. This replicates all the data in the volume that is not in a qtree to a qtree. This can have impact on mount points because the data paths have changed.

MANY-TO-ONE CONFIGURATION

When multiple systems are replicating data to a single system, make sure that the destination storage system can service the combined write throughput requirements of all source systems.

DATA ONTAP VERSIONS

Qtree SnapMirror does not have any Data ONTAP version restrictions between source and destination systems. For volume SnapMirror, the source and destination systems must have the same or higher Data ONTAP major release version, but they can have different minor release versions. Examples of major and minor releases follow. Data ONTAP 7.3 and Data ONTAP 7.2 are different major releases. Data ONTAP 7.2.2 and Data ONTAP 7.2.3 are different minor releases of the same major release.

UPGRADING TO FLEXIBLE VOLUMES

For volume SnapMirror, the source and destination volumes must be "like" volumes. That is, both source and destination must be either traditional or flexible volumes. In keeping with this requirement, the source and destination volumes need to be upgraded simultaneously. Furthermore, because the destination is a read-only volume, for migration purposes, the destination volume must be writable, so that a container file can be created. Therefore the SnapMirror relationship must be broken prior to starting the migration process. After starting the migration process on the source and destination volumes, the SnapMirror relationship can be resumed.

UNICODE

Directories on all SnapMirror source volumes that support CIFS clients must be in Unicode format before being replicated to a destination; otherwise, the directories in the read-only destination volume will not be in Unicode format and attempts through CIFS to access directories and open files on the destination volume might receive "access denied" errors. This is true in qtree SnapMirror deployments only.

FAS DEDUPLICATION

Refer to section 8.8 for details on FAS Deduplication and SnapMirror.

HIGH FILE COUNT ENVIRONMENTS AND QTREE SNAPMIRROR

A high file count (HFC) environment is defined as any single volume containing millions of files. The `filestats` command helps identify HFC environments. The command requires CPU time, so it should be run during low I/O periods. When using qtree SnapMirror in an HFC environment, follow these guidelines:

- Avoid HFC with numerous qtrees. Each qtree triggers an additional scan of changed inodes. NetApp recommends that users stay under two qtrees in HFC environments.
- Avoid HFC and large directories with applications that generate lots of activity in each of these directories.
- Avoid HFC and many small directories with applications that generate lots of activity in each of these directories.

SNAPMIRROR OVER FIBRE CHANNEL

SnapMirror over Fibre Channel enables SnapMirror replication over a Fibre Channel SAN environment. SnapMirror over Fibre Channel includes all the features that are available with SnapMirror over Ethernet. For specific product requirements, refer to the *Data Protection Online Backup and Recovery Guide* on NOW. Also see *Requirements for SnapMirror over Fibre Channel* on NOW.

READ PERFORMANCE ON A FLEXVOL VOLUME SNAPMIRROR DESTINATION

When a volume SnapMirror update finishes in a FlexVol configuration, the destination storage system launches a process to recreate the fast-path metadata so that it is consistent with the FlexVol and aggregate layout on the destination storage system. During metadata creation, read performance on a flexible volume that is a volume SnapMirror destination might be significantly worse than that experienced from a flexible volume that is not a volume SnapMirror destination. This does not affect qtree SnapMirror, flexible volume SnapMirror source, or volume SnapMirror for traditional volumes.

When this process finishes, reads to the SnapMirror destination use the normal fast-path, and read performance on the destination flexible volume returns to normal.

This read performance issue might affect two classic scenarios: 1) Users who need immediate access to the volume SnapMirror FlexVol destination soon after an update is completed, and 2) Users who perform tape

backups from the destination volume soon after an update is completed. Slow performance is seen until the process completes recreation of the fast-path metadata.

Best Practice: To minimize impact due to read performance, NetApp recommends Data ONTAP 7.2.4 or later.

5 NETWORK-FREE SEEDING

Network-free seeding is defined as initial data transfer between SnapMirror source and destination without the use of a network. This is extremely useful in limited network bandwidth scenarios such as wide area networks. Since the initial SnapMirror transfer involves the entire data transfer in a given volume or qtree, this can take a very long time over a small network pipe. SnapMirror supports network-free seeding with the use of two technologies—SnapMirror to tape and LREP.

5.1 SNAPMIRROR TO TAPE

SnapMirror to tape (SM2T) gives users the ability to perform the initial full-volume transfer of a SnapMirror volume by using tapes. The user can transfer the contents of a source volume to tapes. When the tapes have been created, remote volumes can be "seeded" with the source volume data by shipping these tapes to the remote locations and restoring their contents to the target volumes. After the initial data has been transferred to the target volumes from tapes, subsequent incremental transfers are performed over the network. This feature can significantly reduce the data transfer over the WAN and the amount of time required to initialize a SnapMirror target volume. SnapMirror to tape does not support incremental transfers and is not intended for backups or data archiving. Symantec NetBackup 4.5 and later can also be used to control the SnapMirror to tape operations.

RESTRICTIONS

Only a baseline transfer to tape is currently supported, not incremental. There is no way to get just the changed blocks since the last backup, as you can with `snapmirror update` command. SM2T also transfers all Snapshot copies in the volume and the active file system. There is no way to select a single Snapshot copy where you know that an application or file system was consistent (unlike with an NDMP-based application).

SM2T works only with volumes and is based on volume SnapMirror. Therefore backup and restore works only between volumes of the same type (both flexible or both traditional volumes). Because SM2T is based on volume SnapMirror, SM2T has the same version restrictions as volume SnapMirror.

Note: When using SM2T, using the `-g` option allows you to specify the destination geometry at the time of writing the tape with `snapmirror store`. This is to mitigate any performance issues with disk geometry mismatch. This only applies to traditional volumes.

5.2 LREP

LREP is a logical replication tool that is useful for qtree SnapMirror or SnapVault initial transfers (also commonly referred to as seeding baselines). Like SnapMirror to tape, LREP is used to perform the initial transfer (the baseline) to portable media. The portable media is shipped to the remote site and is then written locally there. No network bandwidth is used, only a manual process of moving the media. Once the data is on the destination system, modify the SnapMirror relationship to reflect the actual source and destination relationship.

Two utilities are required for the entire LREP process: `lrep_writer` is used at the location of the destination system and `lrep_reader` is used at the source system.

While SnapMirror to tape runs on Data ONTAP, LREP is available for open systems clients such as Windows®, UNIX®, and Linux®. LREP can also use a disk drive or even a FAS system as the "portable" media.

The latest LREP tool and user guide can be downloaded from the NOW ToolChest.

6 SUMMARY OF SNAPMIRROR CHANGES IN DATA ONTAP 7.3

This section summarizes key changes in SnapMirror in Data ONTAP 7.3. For a complete list, see the release notes and the product documentation in the Data Protection Online Backup and Recovery Guide on NOW.

INCREASED CONCURRENT TRANSFERS

Data ONTAP 7.3 allows increase in concurrent SnapMirror transfers for asynchronous modes. This applies to both volume and qtree SnapMirror. The increase depends on the platform. For example, FAS3050 allows 16 concurrent volume SnapMirror transfers in Data ONTAP 7.2 versus 50 concurrent volume SnapMirror transfers in Data ONTAP 7.3. Both of these numbers are without the NearStore option license.

In Data ONTAP 7.2, the concurrent transfers are halved when ATA disks are present in the system unless the NearStore option license is installed. For example, in Data ONTAP 7.2, FAS3050 allows 16 concurrent volume SnapMirror transfers with FC only drives and 8 concurrent volume SnapMirror transfers when ATA drives are present. Data ONTAP 7.3 removes this limitation. Therefore a FAS3050 allows 50 volume SnapMirror concurrent streams regardless of the type of drive present in the system.

SnapMirror Sync and Semi-Sync do not have increased concurrent streams in Data ONTAP 7.3.

EFFICIENT USE OF MULTIPROCESSOR SYSTEMS

SnapMirror Async in Data ONTAP 7.3 efficiently uses multiprocessor systems. This mitigates or eliminates domain bottleneck issues. SnapMirror Sync and SnapMirror Semi-Sync does not have these improvements in Data ONTAP 7.3

EFFICIENT COMPUTATION OF CHANGED BLOCKS

A SnapMirror update involves computation of changed blocks prior to the transfer. This can take a long time for very large volumes. If the volume has minimal changes, the computation time can be longer than the transfer time.

SnapMirror in Data ONTAP 7.3 employs a more efficient comparison of the active map file to find the changed blocks. This results in significantly faster computation of changed blocks.

7 SNAPMIRROR MANAGEMENT

The most common methods to manage SnapMirror are CLI and FilerView®. These methods work very well in small environments with a handful of systems. In large environments, these two methods become tedious and cumbersome.

PROTECTION MANAGER

Protection Manager provides policy-based management and automated data protection configurations. Automation and policy-based management approaches reduce the possibility of user errors. Protection Manager also provides a holistic view of the NetApp disk-based data protection status. Protection Manager runs within the NetApp Management Console alongside Performance Advisor and Provisioning Manager. Protection Manager can detect, manage, and monitor SnapMirror, SnapVault, and Open Systems SnapVault relationships. Currently, Protection Manager cannot manage SnapMirror Sync and SnapMirror Semi-Sync.

CONCEPTS

Protection Manager uses three fundamental concepts—policies, resource pools, and data sets.

A policy is a rule that describes how to protect data. Protection Manager helps define policies in a graphical and intuitive manner. The policy can be applied to a volume, a LUN, or a user-defined group called a data set. An example of a data set is a group of LUNs that support the same application. The policies define the protection levels of the data sets to a group of resources called resource pools.

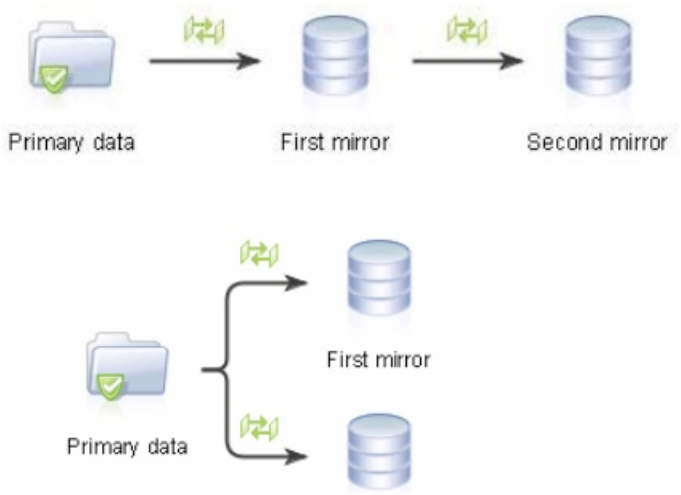

Figure 1) Examples of policies.

Note: Mirrors defined in Protection Manager use volume SnapMirror, and backup definition uses qtree SnapMirror.

Protection Manager also allows setting up throttle schedules to limit network bandwidth used by the mirror and backup operations. Once the schedule is created, the schedule can be applied to one or multiple policies.

DATA CONFORMANCE AND MONITORING

Protection Manager provides a conformance monitor that regularly checks for data set conformance to a policy. Data sets are marked as either in conformance or out of conformance. When the conformance change is detected, Protection Manager can attempt to perform corrective steps to bring the data set back into conformance or notify the administrator of the conformance changes.

Protection Manager also allows the administrator to monitor data protection status and to set alerts when certain conditions are met. These conditions can be failed data protection tasks or when SnapMirror lag times exceed certain thresholds.

DISASTER RECOVERY

Protection Manager 3.7 provides a new DR management feature that allows the user to apply a DR-capable policy to a data set. DR-capable data sets have attributes such as failover readiness and capabilities such as automated failover, and they export the DR volumes to the DR clients. DR failover readiness is shown on the dashboard with the status. Figures 2 and 3 show the failover readiness dashboard and DR-capable attribute of various data sets.

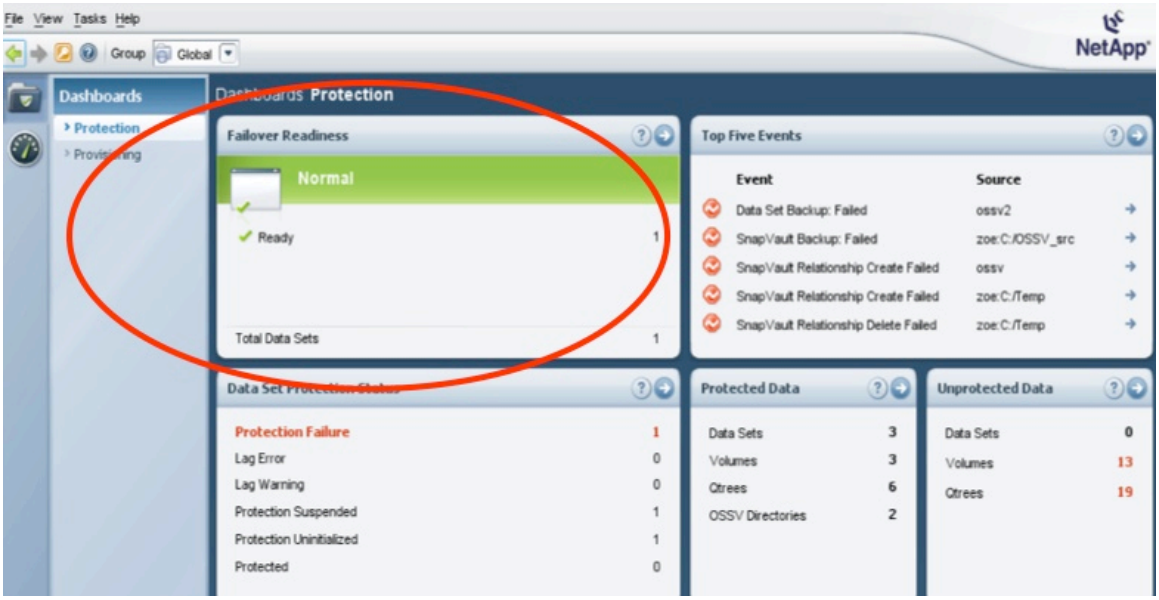

Figure 2) Failover readiness dashboard.

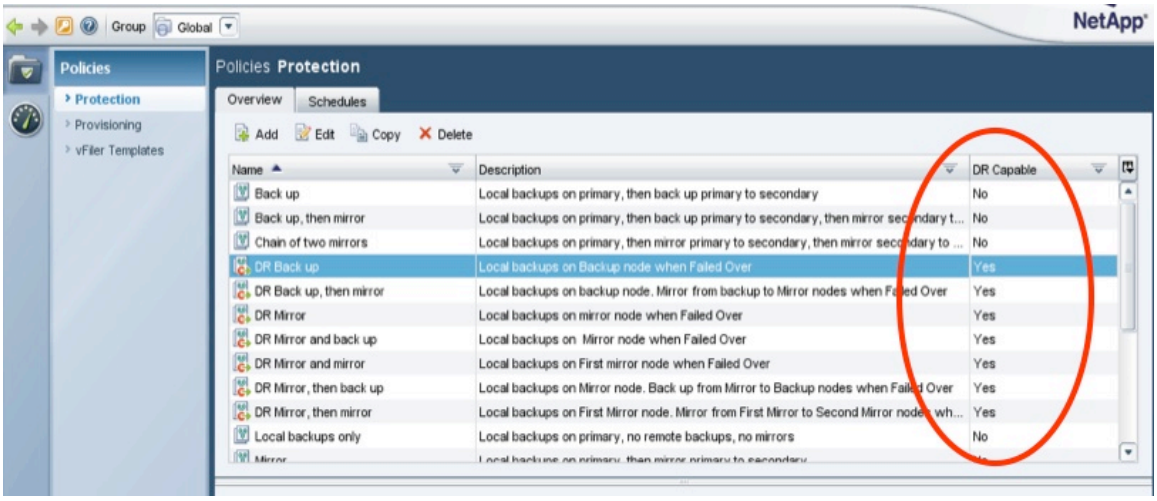

Figure 3) Disaster-recovery capable attribute of data sets.

There is also a new disaster recovery tab in Protection Manager 3.7. Only DR-capable data sets are listed under this tab. Four important action buttons are available. The Failover button makes the DR volumes writable by breaking the SnapMirror relationships. The Test button verifies that the failover scripts work properly without doing a failover. The Update button performs a SnapMirror update. The Cancel button cancels a failover task. Protection Manager 3.7 does not provide an automated failback. This can be done by using CLI.

For more information about Protection Manager, refer to the administration guide on NOW.

8 USE OF SNAPMIRROR WITH OTHER NETAPP PRODUCTS

8.1 NETAPP MANAGEABILITY SUITE

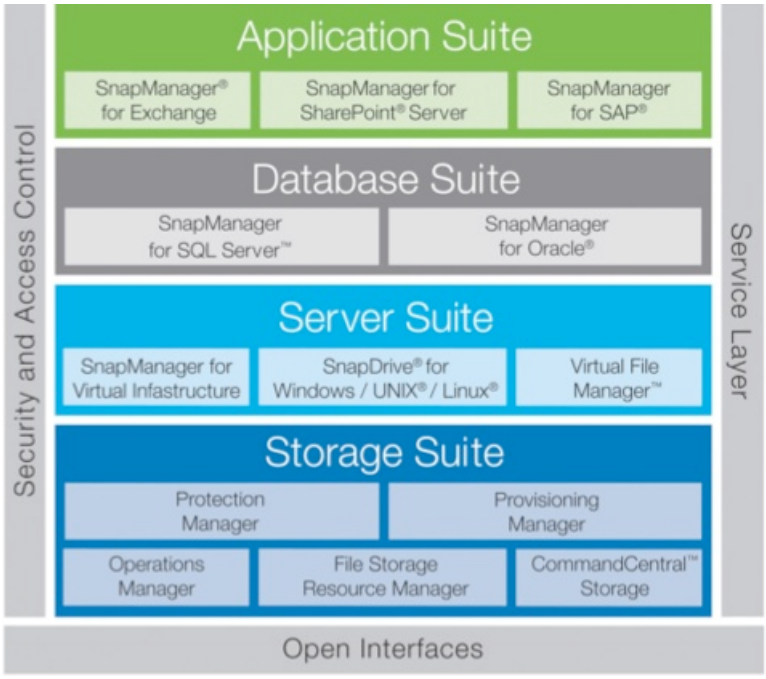

Figure 4) NetApp Manageability Suite.

APPLICATION AND DATABASE SUITE

The NetApp Manageability Application Suite consists of SnapManager® for Exchange, SnapManager for SharePoint Server, and SnapManager for SAP. The NetApp Manageability Database Suite consists of SnapManager for Oracle and SnapManager for SQL Server. The SnapManager Suite integrates with key applications, thereby reducing the risks associated with backups and provides more flexibility in streamlining IT operations.

The SnapManager Suite can be used to take consistent backups by leveraging the application integration, and with the use of NetApp Snapshot technology, SnapMirror can be used to extend the protection by replicating these consistent backups to a different storage system located either within the data center or to another data center located on the campus or at a remote DR site.

Within the SnapManager Suite, SnapManager for Exchange offers the most comprehensive integration with SnapMirror, providing the ability to *automatically* replicate the consistent Snapshot copies by using SnapMirror. SnapManager for Exchange 5.0 also provides *automated* failover for Exchange environments.

For more information, refer to section 11, “References.”

SERVER SUITE

The NetApp Manageability Server Suite includes SnapManager for Virtual Infrastructure, Virtual File Manager, and SnapDrive (for Windows, UNIX, and Linux).

SnapManager for Virtual Infrastructure provides storage and virtual infrastructure administrators with an automated solution for data protection and recovery of virtual machines in a VMware® ESX environment. This is achieved by integrating NetApp Snapshot, SnapRestore®, and SnapMirror for automated backup and recovery of data stores.

VFM® (Virtual File Manager) is a comprehensive and integrated solution for file services. It uniquely provides non-disruptive storage consolidation, remote office data management, disaster recovery, and data lifecycle management through policy-based management, leveraging a global namespace for file services environments. VFM is tightly integrated with NetApp SnapMirror to provide a rich set of data management capabilities.

SnapDrive automates storage provisioning tasks and simplifies the process of taking error-free, host-consistent data Snapshot copies. SnapDrive provides a server-aware alternative to maintaining manual host connections to underlying NetApp storage systems. It reduces the risk of data disruption and increases storage management flexibility, delivering higher productivity and utilization.

SnapDrive® for Windows® and SnapDrive for UNIX® can be used for managing (creating, deleting, and renaming) Snapshot copies on the source volume of SnapMirror. Any changes to Snapshot copies on the source system are immediately made visible on the destination system.

In general, SnapDrive is well integrated with volume SnapMirror. For example, SnapDrive for Windows can create a rolling Snapshot copy and then perform a volume SnapMirror update. SnapDrive for UNIX cannot perform any SnapMirror operations. Also, SnapDrive neither has integration with qtree SnapMirror nor does it support qtree-level SnapMirror operations.

For more information, refer to the SnapDrive for Windows and SnapDrive for Windows Installation and Administration Guide on NOW.

8.2 FLEXCLONE

Starting with Data ONTAP 7G, storage administrators have access to a powerful new feature that allows them to instantly create clones of a flexible volume. A FlexClone volume is a writable point-in-time image of a flexible volume or another FlexClone volume. They take only a few seconds to create and do not cause interruption to the parent flexible volume. FlexClone volumes use space very efficiently, leveraging the Data ONTAP architecture to store only data that changes between the parent and clone volumes. FlexClone offers substantial space savings for work environments that require multiple copies of the same data, such as source trees, chip simulations, and weather simulations, without causing any performance bottlenecks.

FlexClone also makes it possible to create a writable volume from a read-only SnapMirror destination without interrupting the SnapMirror replication process and the production operations. A FlexClone volume can also be split from its parent to create a new standalone volume. Cloning is available only with flexible volumes, not with traditional volumes. Cloning does not require any special hardware. However, the storage system must have the `flex_clone` license installed.

VOLUME SNAPMIRROR, SNAPDRIVE, AND FLEXCLONE

When a clone is created on a volume SnapMirror destination volume, Data ONTAP locks the Snapshot copy that the clone is based on. This means that users cannot delete this Snapshot copy. This is done to protect the clone. There is also a soft lock on the corresponding Snapshot copy on the SnapMirror source system. Data ONTAP will not delete this Snapshot copy; however, the user can delete this Snapshot copy on the SnapMirror source volume. If the user deletes the Snapshot copy on the source volume, the next SnapMirror update fails because it attempts and fails to delete the corresponding Snapshot on the destination volume. All SnapMirror updates to the destination volume continue to fail until the clone is destroyed or split. Use caution when deleting Snapshot copies when SnapMirror and FlexClone are involved.

Also, if a FlexClone volume is created from a Snapshot copy in the destination volume that is not the most recent copy, and therefore has locked down the Snapshot copy, if that Snapshot copy no longer exists on the source volume, every update attempts to delete the copy on the destination. In this case, all SnapMirror updates to the destination volume will fail until the clone is destroyed or split. This does not occur if the clone is created from the most recent Snapshot copy in the SnapMirror destination, because that copy still exists in the source volume.

SnapDrive for Windows creates rolling Snapshot copies on the source volume. When SnapDrive creates a new rolling Snapshot copy, it deletes the old rolling Snapshot copy. Therefore if a FlexClone volume is created on the SnapMirror destination using the SnapDrive rolling Snapshot copy, the next SnapDrive update will delete the corresponding Snapshot copy on the source volume and SnapMirror updates will fail from that point onwards. There are two ways around this issue. See the best practices below.

Best Practices:

- Do not create FlexClone volumes on the destination from scheduled Snapshot copies
- Create FlexClone volumes on the destination from manually created Snapshot copies
- If you wish to create FlexClone volumes on the destination from SnapDrive created rolling Snapshot copies, do one of the following:

    - Perform a manual SnapMirror update following the creation of the FlexClone volume. This update process will propagate a soft lock on the corresponding Snapshot copy on the source system. The next SnapDrive update will then create another rolling Snapshot copy without deleting the Snapshot copy which has a soft lock associated with it.
    - Rename the rolling Snapshot copy created by SnapDrive before creating the FlexClone volume. This step will ensure that SnapDrive will not delete this renamed Snapshot.
- Do not create FlexClone volumes on the destination using the volume SnapMirror Snapshot copies. If you have to create FlexClone volumes from volume SnapMirror Snapshot copies, use the latest SnapMirror Snapshot copy
- Do not delete the Snapshot copies on the source if a FlexClone volume exists on the destination using the corresponding Snapshot copy

SPLITTING A CLONE

Splitting a FlexClone volume from its parent removes the connection between the clone and its parent. The administrator can split a FlexClone volume in a SnapMirror environment without affecting the SnapMirror transfer, because it becomes an independent entity after the FlexClone volume is split. In fact, it is a good idea to split clones that have been used for an extended period of time to avoid any impact on SnapMirror, especially if the source Snapshot copy could be deleted.

DESTROYING A CLONE

If a FlexClone volume is not required any more, it can be directly destroyed without splitting it.

PROHIBITED OPERATIONS

FlexClone data resides in the parent Snapshot copy, so operations that would destroy the parent volume are not allowed. The following operations are *not* allowed:

- Destroy parent volume (but it can be taken offline)
- Destroy parent and clone shared Snapshot copy
- Use `vol copy` over a parent volume
- Use `snapmirror initialize` over a parent volume
- Use `snap restore` with a parent volume
- Use `snapmirror resync` with a parent volume before Data ONTAP 7.3. SnapMirror `resync` is possible to a parent volume in Data ONTAP 7.3, as long as the resync procedure does not delete the clone snapshot.)

Note: A FlexClone volume cannot be used as a SnapMirror destination.

8.3 SNAPVAULT

While SnapMirror is typically used for disaster recovery purpose, SnapVault is used for long-term backup of production data or disaster recovery data. There are two types of SnapVault deployments with SnapMirror.

DR PROTECTION FOR LONG-TERM BACKUP DATA

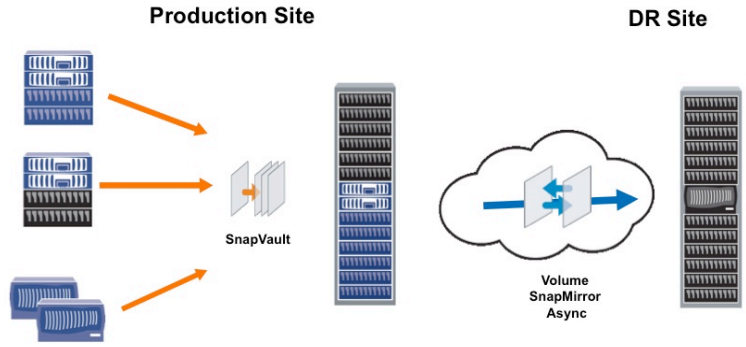

Figure 5) Example of DR protection for backup data

In the configuration shown in Figure 5, data on various production systems requires long-term backup protection. SnapVault is used to achieve this requirement. In case of disaster, the long-term backups are replicated to a DR site by using volume SnapMirror. Because volume SnapMirror copies all Snapshot copies, all the long-term backups at the SnapVault destination are available at the DR site. The data at the DR site can be used to restore or access the desired data in the event of partial data loss or an entire SnapVault system failure at the production site.

LONG-TERM BACKUP OF PRIMARY OR DR DATA

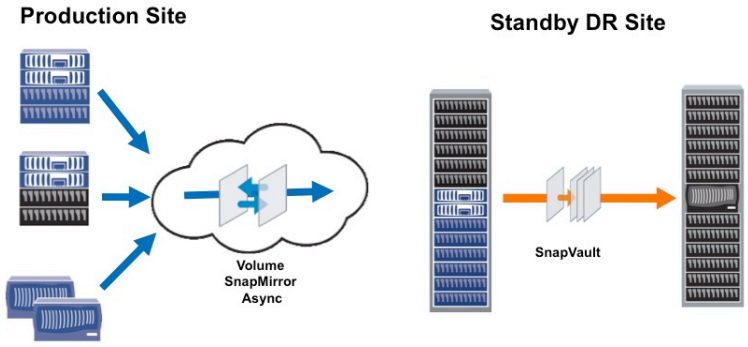

Figure 6) Example of backup protection for DR data

In the deployment shown in Figure 6, data from various primary production systems is replicated to a remote DR site. Volume SnapMirror ensures identical data at the source and destination, which means that if 7-day retention is set up at the production site, data is retained for 7 days on the SnapMirror destination at the DR site as well. If long-term retention (say 90 days) is required at the DR site, SnapVault can be used to back up from the SnapMirror destination. Note that SnapVault cannot create Snapshot copies on the volume SnapMirror destination because destination volume is read-only. However, SnapVault transfers the most recent volume SnapMirror-based Snapshot from the volume SnapMirror destination, as shown in. the following example:

`fas1` is the volume SnapMirror source at the production site.

`fas2` is the volume SnapMirror destination at the standby DR site.

`fas3` is the SnapVault destination at the standby DR site.

`fas1` has SnapMirror license installed. `fas2` has both SnapMirror and SnapVault Primary licenses installed. `fas3` has SnapVault Secondary license installed.

`fas1` and `fas2` have a volume SnapMirror relationship for vol1.

```
Source                Destination           State          Lag        Status
fas1:vol1             fas2:vol1             Snapmirrored   00:00:05   Idle
```

SnapVault is backing up `vol/vol1/qt1` to `fas2`. SnapVault will transfer the most recent base volume SnapMirror Snapshot. On `fas2`:

```
Source                  Destination             State          Lag        Status
fas2:/vol/vol1/qt1      fas3:/vol/vol1/qt1      Snapvaulted    01:48:46   Idle
```

```
  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  May 30 11:20  fas2(0099909261)_vol1.3
 28% (28%)    0% ( 0%)  May 30 09:30  fas2(0099909261)_vol1.2 (snapvault)
```

On `fas3`:

```
  %/used       %/total  date          name
----------  ----------  ------------  --------
 29% (29%)    0% ( 0%)  May 30 11:19  fas3(0099909261)_vol1-base.0 (busy,snapvault)
```

The next volume SnapMirror update between `fas1` and `fas2` will propagate the soft lock for this SnapVault baseline Snapshot.

On `fas1`:

```
  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  May 30 11:20  fas2(0099909261)_vol1.3 (snapmirror)
 28% (28%)    0% ( 0%)  May 30 09:30  fas2(0099909261)_vol1.2 (snapvault)
```

Restrictions and Best Practices

First, if a user on `fas1` deletes the Snapshot with the `snapvault` soft lock, all future volume SnapMirror updates between `fas1` and `fas2` will fail. This is because volume SnapMirror attempts to delete this Snapshot copy on the volume SnapMirror destination (`fas2`) and fails.

Second, as of Data ONTAP 7.3, it is not possible to run SnapVault and SnapMirror simultaneously in the scenario just described. If SnapVault updates are triggered while volume SnapMirror transfers are in progress, the volume SnapMirror transfers are aborted. Therefore volume SnapMirror transfers must be suspended in order for SnapVault updates to occur.

Third, SnapVault transfers the most recent volume SnapMirror Snapshot copy from the volume SnapMirror destination even if a user requests a specific Snapshot copy with the –s option.

8.4 SNAPLOCK

SnapLock® volumes are write-once, read-many (WORM) volumes intended for permanent data archiving. There are two types of SnapLock volumes:

SnapLock Compliance volume: For strict regulatory environments, such as SEC 17a-4 compliant environments

SnapLock Enterprise volume: For environments without regulatory restrictions

SnapMirror can be used to mirror SnapLock volumes with the following restrictions:

- Data ONTAP 6.4.1 supports only destination volumes or qtrees other than SnapLock. Data ONTAP 6.5 and 7.0 support both SnapLock and other than SnapLock destination volumes or qtrees.
- In all Data ONTAP releases, the SnapMirror resync feature cannot be used to reestablish a volume SnapMirror relationship to a SnapLock Compliance destination volume because this operation would result in the destruction of WORM data on the destination volume and would make SnapLock noncompliant with government regulations regarding non-erasability of records. This important consideration must be kept in mind while planning DR scenarios for SnapLock Compliance volumes.
- In the case of a qtree SnapMirror relationship to a SnapLock Compliance destination volume, the resync ability was available as a setflag prior to Data ONTAP 7.0. The resync option was generally available starting in Data ONTAP 7.0.

- A SnapLock Compliance volume cannot be reinitialized because data on the volume cannot be changed. If the SnapMirror relationship is broken by using the `snapmirror break` command, the SnapLock Compliance destination volume can never be reinitialized. A new empty SnapLock Compliance destination volume can of course be reinitialized.
- There is no such restriction on resync in the case of SnapLock Enterprise volumes because the administrator is trusted.
- For SnapLock Compliance volumes, additional Data ONTAP version restrictions exist for source and destination systems for volume and qtree SnapMirror operations. Review the release notes and the product documentation for specific restriction details about the desired release.

Table 6) SnapMirror resync support

	SnapLock Compliance	SnapLock Enterprise
Qtree SnapMirror resync	Yes	Yes
Volume SnapMirror resync	No	Yes

REPLICATION RESTRICTIONS

Table 7 shows the restrictions for replication between non-SnapLock, SnapLock Enterprise, and SnapLock Compliance volumes.

Table 7) Replication restrictions between various types of volumes

	SnapMirror Destination		
SnapMirror Source	Non-SnapLock volume	SnapLock Enterprise volume	SnapLock Compliance volume
Non-SnapLock volume	Yes	Yes	No
SnapLock Enteprise volume	Yes	Yes	No
SnapLock Compliance volume	Yes	Yes	Yes

END-TO-END SNAPLOCK COMPLIANCE VOLUME SNAPMIRROR RELATIONSHIP

In Data ONTAP version 7.0 and later, in order to create an end-to-end SnapLock Compliance volume SnapMirror relationship, you simply create both the source and destination volumes as SnapLock Compliance volumes (by using the `-L` options) and then initialize the mirror by using `snapmirror initialize`, just as you would with regular volumes. No special steps are required.

Prior to Data ONTAP 7.0, to create an end-to-end SnapLock Compliance volume SnapMirror relationship, the destination volume must initially be a non-SnapLock volume. When initializing the mirror by using the `snapmirror initialize` command, the new `-L` option must be specified. The `-L` option instructs SnapMirror to convert the destination volume to a SnapLock Compliance volume at the completion of the initial level 0 transfer when the mirror relationship is established.

This does not apply to SnapLock Compliance qtree SnapMirror relationships, which are initialized just as would be done with regular volumes.

SYNCHRONOUS REPLICATION WITH SNAPLOCK COMPLIANCE VOLUMES

SnapLock Compliance does not yet support SnapMirror Sync and SnapMirror Semi-Sync. SnapLock Enterprise volumes do not have this restriction.

8.5 MULTISTORE

A storage system's hardware is made up of CPUs, network cards, power supplies, disk drives, and so on.

Using MultiStore®, the resources can be logically partitioned and dynamically assigned. The result is a virtual storage controller, also referred to as a vFiler™ controller. MultiStore technology provides an efficient architecture for consolidating multiple physical storage systems into a smaller number of systems. From the end user's perspective, each virtual storage controller appears as a separate physical storage system with a unique IP address. Security is a key concern when storage is consolidated either within an organization or by an application service provider. A virtual storage controller provides a confined environment. The data owned by a virtual storage controller cannot be accessed by any other virtual storage controllers, even though they are hosted on the same physical storage system. All requests for data access owned by a virtual storage controller are tagged with its context, making unauthorized access to data impossible.

VIRTUAL STORAGE CONTROLLER DR

Through integration with SnapMirror, virtual storage controllers can be created and automatically mirrored to other storage systems over a LAN or WAN for the purposes of data migration and DR. Integrated mirror relationships can be established to automate the migration of virtual storage controllers to other storage systems or to create a DR virtual storage controller on a destination storage system, which can be quickly activated in the event of an outage.

In a DR configuration, the source system remains active, serving data to its clients, and the destination system remains inactive but ready to be activated in case of a disaster. NetApp recommends having the disaster recovery site geographically farther from the source to recover from any site-wide disaster. The activation process must be performed manually.

VIRTUAL STORAGE CONTROLLER MIGRATION

Migration moves the specified virtual storage controller from the remote storage system to the local storage system. Migration is initiated on the destination storage system that will host the virtual storage controller after the migration. Migration across storage systems enables workload management. Migration automatically destroys the source virtual storage controller and activates the destination, which starts serving data to its clients automatically. Only the configuration is destroyed on the source, not the data. The migration process takes more time than activating the DR destination site, because it has to perform a SnapMirror baseline copy of the data. Migration can also be used to perform hardware maintenance on the storage systems with minimum downtime. Depending on the amount of data, the baseline copy can take a long time. However, clients still have access to the source storage system during this copy phase. When the baseline copy has been completed, an incremental SnapMirror update is performed to make sure that all the new data since the baseline copy has been transferred to the new system.

The ability to replicate and move virtual storage controllers from one storage system to another provides the following benefits:

- SnapMirror can be used to replicate virtual storage systems to one or more target storage systems, where the mirrored virtual filers can be quickly activated within minutes for DR purposes.
- Migrating or moving virtual filers to less busy or more powerful systems allows administrators to easily load-balance user activity and to relocate data for optimum performance and efficiency.
- Data management is simplified because changes to network storage services can be quickly and transparently redeployed to suit business needs.

For more information, refer to TR 3462, *Storage Virtualization and DR Using MultiStore (vFiler)*.

8.6 METROCLUSTER

MetroCluster is a cost-effective, integrated, high-availability and disaster recovery solution that protects against site failures resulting from human error, HVAC failures, power failures, building fire, architectural failures, and planned maintenance downtime. MetroCluster provides site protection within a metro, and supports replication up to 100 KM. In some instances, campus DR might not be sufficient. In these scenarios, it is feasible to use SnapMirror in conjunction with MetroCluster to extend the protection over a long distance (see Figure 7).

Note: The SnapMirror replication can be performed only from the controller that has the data/disk ownership.

Using FlexClone technology, DR testing can be performed at the DR site without interrupting production operations and SnapMirror replication. When the DR testing has been completed, these clones can be either split or destroyed. For more information on how MetroCluster and SnapMirror can be used to achieve

extended data protection, refer to TR 3606, *High Availability and Disaster Recovery for VMware Using NetApp SnapMirror and MetroCluster*.

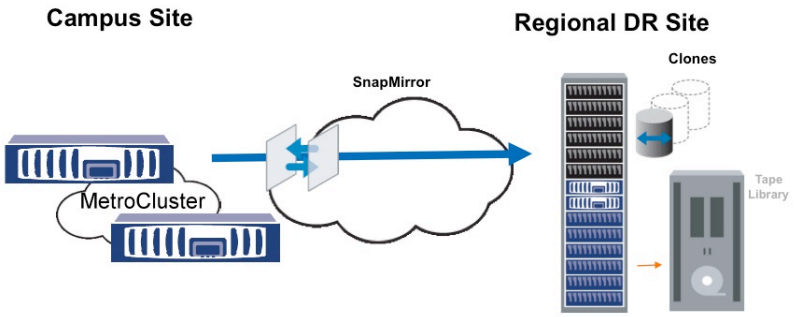

Figure 7) Example overview of regional protection of campus data

8.7 FLEXSHARE

FlexShare™ is a Data ONTAP software feature that prioritizes workload for a storage system. It prioritizes processing resources for key services when the system is under heavy load. FlexShare works on NetApp storage systems running Data ONTAP version 7.2 or later. The key features are relative priority of different volumes, per-volume user versus system priority, and per-volume cache policies. User operations are any data access operations that use NFS, CIFS, iSCSI, FCP, HTTP, or FTP. Examples of system operations are SnapMirror, SnapVault, WAFL scanners, SnapRestore, and NDMP.

The per-volume *system* setting affects all system activities, including SnapMirror operations. FlexShare treats all SnapMirror operations pertaining to a volume as a group, not as individual entities. For example, if a volume has many qtree SnapMirror relationships, the group of relationships is prioritized by FlexShare, not each individual relationship. In other words, FlexShare does not prioritize individual SnapMirror transfers; all SnapMirror transfers for a volume are prioritized together.

During high system load, storage administrators might want to prioritize user activity compared to system activity by minimizing SnapMirror operation impact. This can be accomplished by setting the *system* priority to be lower. Keep in mind that when the system priority is reduced, the amount of time that SnapMirror transfers take can increase.

For more information about FlexShare, refer to TR 3459, *FlexShare Design and Implementation Guide.*

8.8 DEDUPLICATION FOR FAS

Deduplication for FAS provides block-level deduplication within the entire flexible volume on NetApp storage systems. Deduplication only stores unique blocks in the flexible volume and creates a small amount of metadata in the process. This section discusses some best practices when deduplication is used along with SnapMirror.

SnapMirror takes a Snapshot copy before performing an update transfer. Any blocks in the Snapshot copy are locked and cannot be deduplicated. Therefore, for maximum space savings, NetApp strongly recommends running the deduplication process (by using the `sis start` command) *before* performing a SnapMirror update.

Volume SnapMirror: When the source volume is deduplicated, the destination volume automatically attains the deduplication savings. The deduplication benefits also extend to the bandwidth savings because volume SnapMirror only transfers unique blocks. The destination volume can only be deduplicated by replication from a deduplicated source volume; it cannot be deduplicated independently.

For example, a 100GB source volume is deduplicated and now consumes 50 GB achieving 50% storage savings. Volume SnapMirror replication transfers only 50 GB of data, and the destination volume consumes only 50 GB.

Qtree SnapMirror: The source and destination volumes can be deduplicated independently. If the source volume is deduplicated, a qtree being replicated does not automatically result in space savings on the destination, and the replication does not result in bandwidth savings due to deduplication on the source. If deduplication savings are desired on the destination qtree, the deduplication process must be run independently on the destination volume.

For example, a 100 GB source volume with one source qtree of 100 GB data is deduplicated and now consumes 50 GB. Qtree SnapMirror replication still sends 100 GB of data, and the destination qtree consumes 100 GB of data. If deduplication is then run independently on the destination, its consumption will be reduced to 50 GB.

Volume Sizes: When deduplication is not enabled, the maximum volume size is 16 TB as of Data ONTAP 7.3. Deduplication for FAS further restricts the volume sizes. The maximum volume size depends on the storage system model. For example, as of Data ONTAP 7.3, maximum deduplication volume size is 6 TB in a FAS3070 system, and 16 TB in a FAS6070 system.

Best Practice: When volume SnapMirror is used with deduplication, ensure the maximum deduplication volume size on both source and destination is lower of the two maximum volume sizes. Therefore, in the above example, make sure a deduplication volume is no larger than 6 TB on both SnapMirror source and destination systems. This best practice ensures that SnapMirror can be successfully resynchronized (by using the `snapmirror resync` command) in either direction (source to destination and destination to source in case of fail back).

DATA ONTAP 7.3 AND VOLUME SNAPMIRROR

A FAS deduplication volume contains associated metadata such as the finger print database and change log files. Before Data ONTAP 7.3, this metadata resided in the deduplicated volume. This resulted in two disadvantages:

1. The metadata was replicated each time with the volume SnapMirror update
2. Every Snapshot copy contained the metadata and thus minimizing the space savings.

To overcome the above stated disadvantages, starting with Data ONTAP 7.3, the deduplication metadata has been moved out of the deduplicated volume and into the aggregate. Even though the metadata is not part of the volume, the destination volume is still in a deduplicated state and the data in the destination volume can be accessed just as in a regular volume. However, due to the absence of the metadata in the volume, there are some side effects to keep in mind in failover and failback scenarios described below.

Failover:

In this scenario the destination (DR) volume is made writable for DR purposes. The following are the deduplication characteristics for the DR volume:

1. The DR volume is already in a deduplicated state because it is a volume SnapMirror destination volume for a deduplicated primary volume.
2. Any *new* data written to the DR volume since the failover enjoys space savings within the new data itself but does not enjoy additional space savings with the old data. In other words, deduplication savings are not shared between the new and the old data.
3. If deduplication savings are desired across the entire volume (new and old data), the deduplication metadata must be rebuilt by running the deduplication operation on the entire volume (by using the `sis start -s` command). The volume is available for read and write operations during the deduplication operation phase. For information on free space requirements in the aggregate and completion time for `sis start -s` command, refer to TR 3505, *NetApp Deduplication for FAS Deployment and Implementation Guide*.

Best practices:

1. If you plan to use the DR volume for production purpose for a short period of time (say < 1 month), deduplication operation is not necessary on the DR volume to establish the metadata because the amount of new data might not be large and therefore potentially minimal additional space savings.
2. Some customers choose to run production operations six months at the primary and six months at the DR site. In these configurations, it is recommended to run the deduplicaton operation on the entire volume (by using the `sis start -s` command) upon failover to the DR site because the amount of new data would be significant and this could result in considerable space savings. The volume is available for read and write operations during the deduplication operation phase.

Failback:

In this scenario the primary volume is made writable at the production site after a SnapMirror `resync` operation. It is also assumed that all the new data written at the DR site since failover is replicated from the DR volume to the primary volume. The following are the deduplication characteristics for the primary volume:

1. The primary volume is already in deduplicated state because it is a volume SnapMirror destination volume for a deduplicated DR volume.
2. Any *new* data written to the primary volume since the failback enjoys space savings with the old data that exists at the time of failover but not with the data written from the DR volume.
3. If deduplication savings are desired across the entire volume, the deduplication metadata must be rebuilt by running the deduplication operation on the entire volume (by using the `sis start -s` command). The volume is still available for read and write operations during the deduplication operation phase.

Best practices:

1. If the amount of data written at the DR site is not significant, it is not necessary to run the deduplication operation on the entire volume upon failback to the primary site.
2. If the amount of data written to the DR site is significant such as in the scenario where primary and DR sites are used six months at a time, it is recommended to run the deduplication operation on the entire volume upon failback to the primary site.

Migration:

In this scenario volume SnapMirror is used to migrate a volume between aggregates within a system or between two different systems. Upon migration, it is recommended to run the deduplication operation on the entire volume to rebuild the deduplication metadata (by using the `sis start -s` command) for maximum space savings.

For more information on FAS Deduplication and how it works, refer to TR 3505, *NetApp Deduplication for FAS Deployment and Implementation Guide*.

9 TIPS FOR TROUBLESHOOTING

The following is a brief list of things to remember when SnapMirror issues are encountered.

- SnapMirror log files are located in the `/etc/log` directory. The log files are `snapmirror`, `snapmirror.0`, `snapmirror.1`, and so forth.
- Make sure that the SnapMirror license is installed on both source and destination systems.
- Make sure that SnapMirror is enabled by using the `snapmirror on` command.
- If you are using names instead of IP addresses, make sure that the host names can be resolved..
- During initialization, the destination volume must be online and in a restricted state.
- The storage systems must be given permission in order for SnapMirror to work. Access is given by using the `options snapmirror.access` command.
- The source volume must be online and writable.
- Volume SnapMirror requires the destination system's Data ONTAP version be same or higher than that of the source system.
- The destination volume must be same as or larger than the source volume for volume SnapMirror.
- `snap diff` can be used to calculate the changed data rate or amount without transferring the data.
- Throttling can be used to limit the network bandwidth being used by SnapMirror.
- SnapMirror Sync and SnapMirror Semi-Sync cannot be used to replicate qtrees.
- SnapMirror Sync and SnapMirror Semi-Sync require an additional free license (`snapmirror_sync`).

- SnapMirror Sync and SnapMirror Semi-Sync cannot be used to replicate within the same system or between systems within the same HA configuration.
- Performance data can be viewed by observing `systat`, `statit` and `perfstat`.

10 APPENDIX

10.1 FAILOVER AND FAILBACK WITH SNAPMIRROR

This appendix outlines the high-level steps required to perform planned and unplanned failover to the DR site. The steps also include a planned failback to the original production site. The steps assume that SnapMirror Async is being used for failover and failback. For the following scenarios, fas1 is the primary storage system and vol1 is the primary volume; fas2 is the DR storage system and vol2 is the DR volume.

10.2 PLANNED FAILOVER (NO DISASTER)

FAILOVER

This scenario assumes that there are ongoing SnapMirror updates between the production site and the DR site.

This is how the SnapMirror configuration file would look on fas2 (asynchronous volume SnapMirror updates every 30 minutes)

```
fas1:vol1 fas2:vol2 - 0,30 * * *
```

1. Shut down all applications at the production site.
2. Perform a final SnapMirror update to transfer all the changes to the DR site. Make the DR volumes writable by breaking the SnapMirror relationships.
   a. On fas2: `snapmirror update -w vol2`
   b. On fas2: `snapmirror break vol2`
3. Bring up the applications at the DR site. This assumes that all DNS changes, NFS and CIFS exports, and LUN mapping are completed.
4. Failover is now complete.

REPLICATE TO THE PRIMARY SITE

5. Because this is a planned failover, it is assumed that the data at the production site is intact at the time of failover.
6. Now that there is new data at the DR site, this data needs to be replicated back to the production site to prevent data loss in case of a disaster at the DR site. This is achieved by using the `snapmirror resync` command. This is always done at the desired destination site; in this step, the production site. The resynchronization step sends *only* the changes since the last common Snapshot copy between the production and the DR sites.
   a. On fas1: `snapmirror resync -S fas2:vol2 fas1:vol1`
7. Set up the primary site (now standby) for replication. The SnapMirror configuration file can be edited to add replication entries to perform this. After the configuration file is set up for asynchronous replication, SnapMirror performs asynchronous updates from the DR site to the primary site per the schedule specified in the configuration file. The SnapMirror configuration file on fas1 looks like this:
   a. `fas2:vol2 fas1:vol1 - 0,30 * * *`

FAILBACK TO THE PRIMARY SITE

8. Shut down all applications at the DR site.

9. Perform a final SnapMirror update to transfer all the changes to the primary site. Make the primary volumes writable by breaking the SnapMirror relationship. This is always done at the destination; in this step, at the primary site.
    a. On fas1: `snapmirror update -w vol1`
    b. On fas1: `snapmirror break vol1`
10. Bring up the applications at the primary site. This assumes that all DNS changes, NFS and CIFS exports, and LUN mapping are completed.
11. Failback is now complete.

REPLICATE TO THE DR SITE

12. Now that the primary site is active; there is new data at this site that needs to be replicated to the DR site.
13. This is achieved by using the `snapmirror resync` command. This is always done at the desired destination site; in this step, the DR site. The resynchronization step sends *only* the changes since the last common Snapshot copy between the primary and DR sites.
    a. On fas2: `snapmirror resync -S fas1:vol1 fas2:vol2`
14. Set up the DR site (now standby) for replication by restoring the original SnapMirror configuration file. After the original configuration file is in place, the DR site (standby site) receives asynchronous updates from the DR site as per the specified schedule in the configuration file. The SnapMirror configuration file on fas2 looks like this:
    a. `fas1:vol1 fas2:vol2 - 0,30 * * *`

10.3 FAILOVER IN THE EVENT OF A REAL DISASTER

This scenario assumes that the production site is lost and is not accessible.

FAILOVER

1. Because the primary site is inaccessible, applications cannot be shut down. Therefore, make the DR volumes writable by breaking the SnapMirror relationships.
    a. On fas2: `snapmirror break vol2`
2. Bring up the applications at the DR site. This assumes that all DNS changes, NFS and CIFS exports, and LUN mapping are completed.
3. Failover is now complete.

REPLICATE TO THE PRIMARY SITE

4. After the primary site is accessible, the first step is to determine whether the data is intact or lost.
5. If there is complete loss of data, the production site needs to be reinitialized (by using `snapmirror initialize`) from the DR site. The reinitialization is always performed at the destination site; in this case, the primary site. If there is no loss of data, only the changes can be transferred to the production site. This is achieved by using the `snapmirror resync` command. This is always done at the desired destination site; in this step, the primary site. The resynchronization step sends *only* the changes since the last common Snapshot copy between the production and the DR sites.
    a. Data loss case. On fas1: `snapmirror initialize -S fas2:vol2 fas1:vol1`
    b. Data intact case. On fas1: `snapmirror resync -S fas2:vol2 fas1:vol1`
6. Set up the primary site (now standby) for replication. The SnapMirror configuration file can be edited to add replication entries to perform this. After the configuration file is set up for asynchronous replication, SnapMirror performs asynchronous updates from the DR site to the primary site per the schedule specified in the configuration file. The SnapMirror configuration file on fas1 looks like this:
    a. `fas2:vol2 fas1:vol1 - 0,30 * * *`

FAILBACK TO THE PRIMARY SITE

7. Shut down all applications at the DR site.
8. Perform a final SnapMirror update to transfer all the changes to the production site. Make the production volumes writable by breaking the SnapMirror relationship. This is always done at the destination; in this step, at the production site.
    a. On fas1: `snapmirror update -w vol1`
    b. On fas1: `snapmirror break vol1`
9. Bring up the applications at the primary site. This assumes that all DNS changes, NFS and CIFS exports, and LUN mapping are completed.
10. Failback is now complete.

REPLICATE TO THE DR SITE

12. Now that the production site is active, there is new data at this site that needs to be replicated to the DR site.
13. This is achieved by using the `snapmirror resync` command. This is always done at the desired destination site; in this step, the DR site. The resynchronization step sends *only* the changes since the last common Snapshot copy between the production and the DR sites.
    a. On fas2: `snapmirror resync -S fas1:vol1 fas2:vol2`
14. Set up the DR site (now standby) for replication by restoring the original SnapMirror configuration file. After the original configuration file is in place, the DR site (standby site) receives asynchronous updates from the DR site as per the specified schedule in the configuration file. The SnapMirror configuration file on fas2 looks like this:
    a. `fas1:vol1 fas2:vol2 - 0,30 * * *`

HOUSEKEEPING

After the failback is completed, old SnapMirror relationships can be deleted by using the `snapmirror release` command. This command removes the relationships going from the DR storage system (fas2) to the production storage system (fas1). The `release` command is always run on the SnapMirror source system.

10.4 SNAPLOCK AND QTREE SNAPMIRROR RESYNC

This section presents a couple of disaster recovery scenarios that require SnapMirror resync. Depending on the scenario, the appropriate storage system is chosen as the source system for the resync operation.

Production failure: In this scenario, the production system (nodeA) failed and the users are failed over to the DR site.

- DR system (nodeB) is failed over for operations by breaking the SnapMirror relationship.
- Users are now actively writing to the DR node (nodeB).
- When the production site is back up, the data that has been written to the DR node during the outage needs to be transferred back to the production site. This requires the storage administrator to perform a `snapmirror resync` operation at the production site (nodeA).
- This resync operation transfers all the data that has been changed since the last qtree SnapMirror transfer from the production site to the DR site.
- When the data transfer has been completed, the SnapMirror relationship is broken to make the production site writable. The administrator can now redirect all users to the production site (nodeA), because it now has all the changes written at the DR site.

- There is one last step that needs to be performed to place the SnapMirror relationship back to its original state. To start replicating data from the production site (nodeA) to the DR site (nodeB), `snapmirror resync` needs to be performed at the DR site (nodeB). This brings any new changes written to the production site after the failback operation. From now on, SnapMirror updates can be performed at desired intervals to continue to protect the data at the production site.

DR testing: In this scenario, there is no failure at the production site; the administrator simply wants to perform DR testing. This scenario assumes that users are actively accessing the production site (nodeA) while the DR testing is being done at the DR site (nodeB).

- NodeA is a production system and nodeB is the DR system. There is a qtree SnapMirror relationship for a given qtree from nodeA to nodeB.
- The user breaks the qtree SnapMirror relationship to make the qtree on nodeB writable for testing.
- The user modifies data in the qtree on nodeB.
- The user has now completed testing and wants to reestablish the qtree SnapMirror relationship to the original state; that is, start replication from the production site to the DR site. Therefore the user issues a `snapmirror resync` on the DR node (nodeB).
- The `resync` command overwrites any new data that was written on the DR system (nodeB). For SnapLock Compliance volumes, files can never be deleted before their expiration date; therefore the resync operation saves all the changes made by the user to the qtree since the last successful qtree SnapMirror transfer.
- The dump image is stored on the WORM volume where the qtree exists on the DR node (nodeB) in `/etc/logs/snapmirror_resync_archive/volname_UUID_qtree`.

  Example: `/etc/log/snapmirror_resync_archive/slcsec_1374c60e-44ba-11d9-9991-0050560669_e7_qd`

To later extract the data from the dump file, perform the following steps:

1. Using a UNIX or Windows client, create a directory called `temp` on the SnapLock volume; or create a new directory called `temp` and copy the dump file into this directory, giving the file the new name `dump file`. Although this is not necessary, it makes running the `restore` command much easier, because the leading path information from the dump file's original location is long.
2. To view files contained in the dump file, run the following command:

   ```
   restore –tf /vol/<volume_name>/temp/dumpfile
   ```
3. To restore files contained in the dump file to their original location, run the following command:

   ```
   restore –rfQ /vol/<volume_name>/temp/dumpfile
   ```
4. To restore files contained in the dump file to a different location, such as the temp directory where the dump file resides, run the following command:

   ```
   restore –rfQD /vol/<volume_name>/temp/dumpfile /vol/<volume_name>/temp
   ```

   Extracted files are in their original SnapLock state, regardless of the approach used.
5. If it is desirable to migrate the dump file to a different appliance, use two UNIX or Windows clients to transfer the file with a utility such as ftp.

10.5 MAKING THE SNAPVAULT DESTINATION WRITABLE

Perform the following steps to convert an Open Systems SnapVault or SnapVault secondary backup destination to a usable/writable destination (typically for DR situations). All the commands are done on the SnapVault secondary (destination) system.

1. Secondary: Turn SnapMirror and SnapVault off.
2. Secondary: Switch to privileged mode (`priv set diag`).

3. Secondary: Convert SnapVault qtree to SnapMirror qtree (`snapmirror convert <sec_qtree_path>`).
4. Secondary: Turn SnapMirror on.
5. Secondary: Quiesce the qtree.
6. Secondary: Break the mirror, making it writable.
7. Secondary: Turn SnapVault on.

10.6 MIGRATING SNAPVAULT BY USING SNAPMIRROR

To migrate a volume that contains SnapVault destination qtrees from one secondary system to another secondary system without having to perform another baseline transfer, complete the following steps.

1. Identify the SnapVault baselines of the qtrees or directories that need migration.

   Example: In this procedure, assume that a baseline of the `bno:C:\500MB` directory was backed up to `r200-old:/vol/old_vol/bno_C_500MB`.

2. Using SnapMirror, replicate the volume from the present secondary system to a volume on the new secondary system. For details about creating SnapMirror relationships, see the SnapMirror chapter in the *Data Protection Online Backup and Recovery Guide* on NOW.

   Note: This is a SnapMirror replication of a *volume*, not a *qtree*.

   Example: To replicate the `old_vol` volume from the `r200-old` secondary system to the `new_vol` volume on the `r200-new` secondary system, complete the following steps on the new secondary system (`r200-new`):

   a. Create the `new_vol` volume.

   b. Restrict the new volume (`new_vol`).

   c. Transfer the `old_vol` volume to the `new_vol` volume by using SnapMirror initialization:
   `snapmirror initialize -S r200-old:old_vol new_vol`

3. Quiesce and break the SnapMirror relationship between the old secondary system and the new secondary system.

   Example: To quiesce and break the SnapMirror relationship between `r200-old` and `r200-new`, complete the following steps on `r200-new`:

   a. `snapmirror quiesce new_vol`

   b. `snapmirror break new_vol`

4. Check SnapMirror status and SnapVault status on the new secondary. The SnapMirror state should show as `Broken-off`. SnapVault state should show as `Snapvaulted`.

   Example: Perform the following steps from `r200-new`:

   a. `snapmirror status`
   ```
   Source        Destination          State
   r200-old:old_vol r200-new:new_vol Broken-off
   ```

   b. `snapvault status`
   ```
   Source Destination State
   bno:C:\500MB r200-new:/vol/new_vol/bno_C_500MB Snapvaulted
   ```

5. Confirm that SnapVault configuration information is not present on the new secondary system by using the `snapvault status -c` command.

Example: Perform the following step from `r200-new`:

```
snapvault status -c
Snapvault secondary is ON.
```

6. Add SnapVault configuration information to the registry on the new secondary system by using the `snapvault start` command. This does not start a new baseline; it updates the registry.

   Example: Perform the following step from `r200-new`

   ```
   snapvault start -S bno:C:\500MB r200-new:/vol/new_vol/bno_C_500MB
   Snapvault configuration for the qtree has been set.
   Qtree /vol/new_vol/bno_C_500MB is already a replica.
   ```

7. Confirm that SnapVault configuration information is present on the new secondary system by using the `snapvault status -c` command.

   Example: Perform the following step from `r200-new`:

   ```
   snapvault status -c
   Snapvault secondary is ON.
   /vol/new_vol/bno_C_500MB source=bno:C:\500MB
   ```

8. Test the new SnapVault relationship by manually updating `r200-new`.

   Example: Perform the following step from `r200-new`:

   ```
   snapvault update r200-new:/vol/new_vol/bno_C_500MB
   Transfer started.
   Monitor progress with 'snapvault status' or the snapmirror log.
   ```

9. Recreate any schedules used on the old secondary system to the new secondary system and make sure that access permissions are in place.

11 REFERENCES

TECHNICAL REPORTS

- TR 3347: A Thorough Introduction to FlexClone Volumes
- TR 3459: FlexShare Design and Implementation Guide
- TR 3606: High Availability and Disaster Recovery for VMware Using NetApp SnapMirror and MetroCluster
- TR 3548: MetroCluster Design and Implementation Guide
- TR 3584: Microsoft Exchange 2007 Disaster Recovery Model Using NetApp Solutions
- TR 3634: NetApp Virtual File Manager: Protecting Your Data: Business Continuity and Disaster Recovery
- TR 3598: Protecting Exchange Server 2007 with NetApp SnapManager for Exchange
- TR 3326: SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations Guide
- TR 3487: SnapVault Best Practices Guide
- TR 3462: Storage Virtualization and DR Using MultiStore (vFiler)

- TR 3483: Thin Provisioning in a NetApp SAN or IP SAN Enterprise Environment
- TR 3263: WORM Storage on Magnetic Disks Using SnapLock Compliance and SnapLock Enterprise
- TR 3598: Protecting Exchange Server 2007 with NetApp SnapManager for Exchange

RESOURCES ON NOW

- Data Protection Online Backup and Recovery Guide
- MultiStore Management Guide
- Data ONTAP System Administrator's Guide

Last updated on October 10, 2008