Technical Report

7-Mode SnapMirror Sync and SnapMirror Semi-Sync Overview and Design Considerations

Amit Prakash Sawant, NetApp

April 2013 | TR-3326

Abstract

This guide provides an overview of NetApp® SnapMirror® Sync and SnapMirror Semi-Sync technology and describes in detail how to implement and utilize it. It should prove useful for customers requiring assistance in understanding and deploying solutions with SnapMirror Sync and NetApp storage systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

NetApp SnapMirror software combines disaster recovery and data distribution in a streamlined solution that supports today's global enterprises. SnapMirror is a very cost-effective solution with efficient storage and network bandwidth utilization and provides additional value by enabling you to put the DR site to active business use.

SnapMirror Sync is not a separate product and is more accurately described as an operational mode of SnapMirror. SnapMirror allows a dataset to be replicated between NetApp storage systems over an IP or Fibre Channel network for backup or disaster recovery purposes.

The destination file system can be made available for read-only access or can be made writable either by using NetApp FlexClone® technology or by “breaking” the mirror relationship. When using FlexClone technology to make the destination copies writable, the replication is uninterrupted. In case the replication relationship is broken, it can be reestablished by synchronizing the changes made to the destination back to the source file system.

In the asynchronous mode of operation, the storage administrator defines the schedule for updates of new and changed data from the source to the destination. The update frequency could be from minutes to as infrequent as once per week, depending on user needs.

In case of synchronous mode, updates are sent to the destination as they occur, rather than on a schedule. If configured correctly, this can guarantee that data written on the source system is protected on the destination even if the entire source system fails due to natural or human-caused disaster. A semi-synchronous mode is also provided to minimize the loss of data in a disaster. This mode also minimizes the performance impact of replication on the source/production system. In order to maintain consistency and ease of use, the asynchronous and synchronous interfaces are identical, with the exception of a few additional parameters in the configuration file. This results in a very simple and easy-to-use single user interface for all three SnapMirror modes: asynchronous, synchronous, and semi-synchronous.

The rest of this report describes the synchronous replication modes in more detail, discusses performance impact of each mode, provides guidelines for capacity planning and configuration, and aids in planning which replication mode and options are best for your environment.

1.1 Intended Audience

This guide is designed for storage administrators or architects who are already familiar with SnapMirror software as used in asynchronous mode and are considering synchronous deployments for environments or datasets with special needs.

1.2 Purpose

The purpose of this paper is to present an overview for implementing SnapMirror Sync software and address design considerations and recommendations to assist the reader in designing an optimal synchronous solution.

1.3 Prerequisites and Assumptions

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

- A minimal knowledge of NetApp platforms and products, particularly in the area of data protection
- A general knowledge of disaster recovery (DR) solutions
- Sufficient working knowledge about the NetApp SnapMirror solution

This report is based on features available in Data ONTAP® 7.3 7-Mode, Data ONTAP 8.0 7-Mode, and Data ONTAP 8.1 7-Mode. Readers should also review the technical report *SnapMirror Async Overview and Best Practices Guide* (TR-3446).

1.4 Business Applications

There are several approaches to increasing data availability in the event 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 assure data availability and minimize downtime. NetApp SnapMirror Sync and SnapMirror Semi-Sync provide a fast and flexible enterprise solution for replicating data over local area, metro 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 mirror on a remote NetApp system, assuring uninterrupted operation.

By providing a simple solution for replicating data across local, metro, and FC networks, SnapMirror Sync and SnapMirror Semi-Sync address problems in the following critical application areas:

Disaster Recovery

If critical data is mirrored to a different physical location, a serious disaster does not necessarily mean extended periods of data unavailability. The mirrored data can be made available to clients across the network until the damage caused by the disaster is repaired. Recovery can include recovery from corruption, natural disaster at the source site, accidental deletion, sabotage, and so on. SnapMirror is often used for DR. Data can be mirrored to a destination system at a DR facility. Preferably, application servers can be mirrored 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 production 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 operation 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, allowing local read-only access to data. Remote data access not only provides faster access to data by local clients, but also results in a more efficient and predictable use of expensive network and server resources. This allows the storage administrators to replicate source data at a chosen time to minimize overall network usage.

Application Dev/Test Environments

With the use of FlexClone on SnapMirror destination volumes, the read-only replicas can be made writable without consuming space and without interrupting replication operations from the primary to DR site. The clone creation typically takes only a few seconds. The writable clones can then be used to develop applications or for testing (functional or performance) before production deployment.

Disaster RecoveryTesting

An actual DR test involves downtime for production environments. As a result, many customers choose not to perform frequent DR testing even though a DR plan exists. When FlexClone is used with SnapMirror DR volumes, the remote site can be used for DR testing without interruption to production operations and DR replication. Applications can be brought up at the DR site to assure data consistency. The clones can be destroyed after the DR testing.

2 SnapMirror Sync and SnapMirror Semi-Sync Requirements

2.1 Business Applications

As of Data ONTAP 7.3, source and destination storage controllers must be identical. This applies to both SnapMirror Sync and SnapMirror Semi-Sync.

Best Practice

SnapMirror Sync and SnapMirror Semi-Sync are supported on any FAS or V-Series platform. However, low-NVRAM platforms such as those having 256MB or less NVRAM per controller are not recommended. Recommended platforms are those having 512MB or more NVRAM per controller.

2.2 Storage

Disk Types Before Data ONTAP 7.3.3

Before Data ONTAP 7.3.3, disk types (FC or ATA) must match across the entire storage system between the storage systems in the synchronous and semi-synchronous modes of SnapMirror. In other words, SnapMirror Sync and SnapMirror Semi-Sync only support these two configurations:

- ATA-only SnapMirror source to ATA-only SnapMirror destination
- FC-only SnapMirror source to FC-only SnapMirror destination

In case of flexible volumes, there are no disk geometry or disk speed restrictions for source and destination systems.

Mixed-disk systems are not supported even if synchronous replication occurs between volumes of same disk types. This restriction applies to both SnapMirror Sync and SnapMirror Semi-Sync.

For V-series, the back-end array must be presenting storage to the V-Series system using only one type of disk (FC or SATA), and the source/destination must use the same type of disks. To replicate between a V-Series system and a FAS system, they must be like controllers (for example, FAS3170 <->V3170), and the disks on the FAS must be the same type as the disks on the back-end array supporting the V-Series system's storage.

Disk Types Starting with Data ONTAP 7.3.3

Starting with Data ONTAP 7.3.3, SnapMirror Sync and SnapMirror Semi-Sync are supported between systems containing mixed disk types. However, the synchronous and semi-synchronous replication must occur between supported disk types. This applies to both FAS and V-series systems.

Table 1) SnapMirror Sync and SnapMirror Semi-Sync supported replication between mixed disk types.

Relationship Disk Type	Supported
FC to FC	Yes
SATA to SATA	Yes
FC to or from SAS	Yes
SAS to or from SATA	No
FC to or from SATA	No

Aggregates

Starting with Data ONTAP 7.2.2, the NVLOG files are written to the parent aggregate of the flexible volume that is being synchronously replicated. Therefore, the SnapMirror Sync destination aggregates need a minimum free space of approximately 10 times the source NVRAM size per volume. For example, if 10 volumes in an aggregate are being synchronously replicated to a single FAS3070 storage system, the amount of required free space in that aggregate is approximately 50GB (10 x 512MB x 10). Since NVLOG files are used only in case of SnapMirror Sync, this free space requirement does not apply to SnapMirror Semi-Sync.

Note: In case of traditional volumes, SnapMirror Sync continues to write the NVLOG files in the destination system's root volume.

Best Practice

For performance, single large aggregates are recommended. When growing an aggregate, add a minimum of four nonparity disks.

Data ONTAP 8.0 7-Mode introduces a new aggregate type, 64-bit aggregate, with a larger maximum aggregate and flexible volume sizes. The maximum size for legacy aggregates, now called 32-bit aggregates, remains 16TB. There are some restrictions when replicating between volumes in 32-bit and 64-bit aggregates using SnapMirror Sync and SnapMirror Semi-Sync in Data ONTAP 8.0. Starting in Data ONTAP 8.1, it is now possible to replicate between volumes in 32-bit and 64-bit aggregates using SnapMirror Sync and SnapMirror Semi-Sync. The following table provides an overview of possible replication between 32-bit and 64-bit aggregates for SnapMirror Sync and SnapMirror Semi-Sync.

Table 2) SnapMirror Sync and Semi-Sync interoperability matrix.

	Destination Volume							
	Data ONTAP		8.0.x		8.1		8.2	
		Aggregate Type	32-Bit	64-Bit	32-Bit	64-Bit	32-Bit	64-Bit
Source Volume	8.0.x	32-bit	Yes	No	No	No	No	No
		64-bit	No	Yes	No	No	No	No
	8.1	32-bit	No	No	Yes	Yes	No	No
		64-bit	No	No	Yes [1]	Yes	No	No
	8.2	32-bit	No	No	No	No	Yes	Yes
		64-bit	No	No	No	No	Yes [1]	Yes

[1] Do not grow source volume beyond 16TB until 64-bit expansion is complete on the destination aggregate and the destination volume is grown beyond 16TB.

Note: The Data ONTAP version of the source and destination systems must be identical when using SnapMirror Sync and SnapMirror Semi-Sync.

Volumes

SnapMirror Sync and SnapMirror Semi-Sync volumes must be at least 10GB in size. Starting with Data ONTAP 8.2, when autosize increases the size of the source volume of a SnapMirror relationship, the destination volume also automatically increases in size. Mirroring of root volumes is not supported using SnapMirror Sync or SnapMirror Semi-Sync. Both traditional and flexible volumes are supported. Refer to section 4.1 for concurrent replication operations limits.

Qtrees

Qtrees can be replicated using SnapMirror Sync and SnapMirror Semi-Sync if they are present in a volume. Synchronous and semi-synchronous replication does not support replicating at the qtree level.

Replication Link

SnapMirror supports TCP/IP or Fibre Channel as interconnects between the source and destination systems. When Fibre Channel interconnects are used, the interface cards are dedicated for SnapMirror communications, switches must be used, and communication is done using TCP/IP through a virtual interface (VI) over the Fibre Channel connections. Therefore, FC-VI cards are required to use SnapMirror over Fibre Channel. Refer to the Data Protection Online Backup and Recovery Guide on the NetApp Support (formerly NOW®) site for specific product requirements and the Requirements for SnapMirror over Fibre Channel document on the NetApp Support site.

2.3 Software

Licensing

No additional license fees need to be paid to use synchronous or semi-synchronous SnapMirror. Up to and including Data ONTAP 8.0, a license key (`snapmirror_sync`) in addition to the standard SnapMirror license needs to be installed on each NetApp system where synchronous or semi-synchronous mode is desired. From Data ONTAP 8.1 onward, just the SnapMirror license works for both synchronous and semi-synchronous modes of replication. See the Data Protection Online Backup and Recovery Guide for more information about the special license key on the NetApp Support site.

Data ONTAP

In case of flexible volumes, any Data ONTAP version that supports flexible volumes also supports SnapMirror Sync and SnapMirror Semi-Sync. The source and destination systems must be of the same Data ONTAP version.

Table 3) SnapMirror Sync and SnapMirror Semi-Sync version restrictions.

Source	Destination	Support
Data ONTAP 7.2	Data ONTAP 7.3	No
Data ONTAP 7.3	Data ONTAP 7.2	No
Data ONTAP 7.2.x	Data ONTAP 7.2.x	Yes

Best Practice

Significant improvements for SnapMirror Sync and SnapMirror Semi-Sync have been made in Data ONTAP 7.3. Therefore, use Data ONTAP 7.3 or a later version when deploying SnapMirror Sync or SnapMirror Semi-Sync. Obtain NetApp approval if you intend to deploy SnapMirror Sync or SnapMirror Semi-Sync on versions earlier than Data ONTAP 7.3 with flexible volumes.

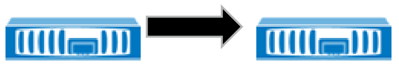
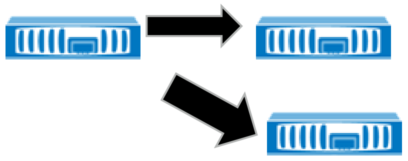
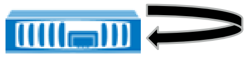
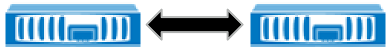
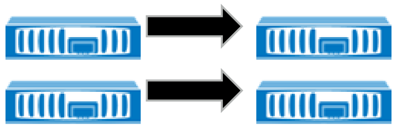
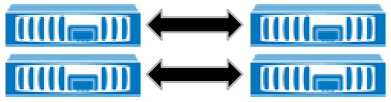
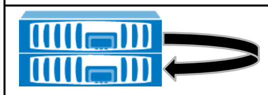
Bidirectional SnapMirror Sync

Sometimes customers use both storage systems for primary workload and replicate data to each other to protect the primary data. Starting with Data ONTAP 7.2.2, bidirectional synchronous replication is supported in the following configurations for flexible volumes:

- Between single controllers that are not part of an HA pair
- Between HA pairs

In other words, bidirectional synchronous replication controllers that belong to the same active-active pair are *not supported*. The following table summarizes supported and unsupported configurations for SnapMirror Sync and SnapMirror Semi-Sync for traditional and flexible volumes. For cascade configuration support, refer to section 3.10, “Cascading Support and Other Configurations.”

Table 4) Supported and unsupported configurations for SnapMirror Sync and SnapMirror Semi-Sync.

Configuration	Description	FlexVol Support	Traditional Volume Support
	Unidirectional, single controller	Yes	Yes
	Unidirectional, one volume to many volumes	No	No
	Unidirectional, local replication	No	No
	Bidirectional, single controller	Data ONTAP 7.2.2+	Yes
	Unidirectional, inter-HA pair	Yes	Yes
	Bidirectional, inter-HA pair	Data ONTAP 7.2.2+	Yes
	Omnidirectional, intra-HA pair	No	No

3 SnapMirror Sync and SnapMirror Semi-Sync Fundamentals

3.1 SnapMirror Sync Overview

To avoid any potential confusion, it is appropriate to review exactly what is meant by the word synchronous in this context. The best way to do this is to examine scenarios where the primary data storage system fails completely to evaluate the disaster's impact on an application. In a typical application environment:

A user saves some information in the application.

The client software communicates with a server and transmits the information.

The server software processes the information and transmits it to the operating system on the server.

The operating system software sends the information to the storage.

The storage acknowledges receipt of the data.

The operating system tells the application server that the write is complete.

The application server tells the client that the write is complete.

The client software tells the user that the write is complete.

In most cases, these steps take only tiny fractions of a second to complete. If the storage system fails in such a way that all data on it is lost (such as a fire or flood that destroys all of the storage media), the impact to an individual transaction will vary based on when the failure occurs.

- If the failure occurs before step 5, the storage system will not acknowledge receipt of the data. This results in the user receiving an error message from the application, indicating it failed to save the transaction.
- If the failure occurs after step 5, the user sees client behavior that indicates correct operation (at least until the following transaction is attempted). Despite the indication by the client software (in step 8) that the write was successful, the data is lost.

The first case is obviously preferable to the second, because it provides the user or application with the knowledge of the failure and the opportunity to preserve the data until the transaction can be attempted again. In the second case, the data might be discarded based on the belief that the data is safely stored.

With asynchronous SnapMirror, data is replicated from the primary storage to a secondary or destination storage device on a schedule. If this schedule was configured to cause updates once per hour, for example, it is possible for a full hour of transactions to be written to the primary storage system and acknowledged by the application, only to be lost when a failure occurs before the next update. For this reason, many customers attempt to minimize the time between transfers. Some customers replicate as frequently as once every few minutes, which significantly reduces the amount of data that could be lost in a disaster.

This level of flexibility is good enough for the vast majority of applications and users. In most real-world environments, loss of few minutes of data is of trivial concern compared to the downtime incurred during such an event; any disaster that completely destroys the data on the storage system would most likely also destroy the relevant application servers, critical network infrastructure, and so on.

However, there are some customers and applications that have a zero data loss requirement, even in the event of a complete failure at the primary site. Synchronous mode is appropriate for these situations. It modifies the application environment described earlier such that replication of data to the secondary storage occurs with each transaction.

1. A user saves some information in the application.
2. The client software communicates with a server and transmits the information.
3. The server software processes the information and transmits it to the operating system on the server.

4. The operating system software sends the information to the storage.
5. The primary storage sends the information to the secondary storage.
6. The secondary storage acknowledges receipt of the data.
7. The primary storage acknowledges receipt of the data.
8. The operating system tells the application server that the write is complete.
9. The application server tells the client that the write is complete.
10. The client software tells the user that the write is complete.

The key difference, from the application's point of view, is that the storage does not acknowledge the write until the data has been written to both the primary and the secondary storage systems. This has some performance impact, as will be discussed later in the document, but modifies the failure scenario in beneficial ways.

- If the failure occurs before step 7, the storage will never acknowledge receipt of the data. This will result in the user receiving an error message from the application, indicating it failed to save the transaction. This causes inconvenience, but no data loss.
- If the failure occurs during or after step 7, the data is safely preserved on the secondary storage system despite the failure of the primary.

Note that regardless of what technology is used, it is always possible to lose data; the key point is that with synchronous mode, the loss of data that *has been acknowledged* is prevented.

3.2 How SnapMirror Sync Works

The first step involved in synchronous replication is a one-time, baseline transfer of the entire dataset. The baseline transfer occurs as follows:

1. The primary storage system creates a Snapshot™ copy, a read-only point-in-time image of the file system. This Snapshot copy is called the *baseline Snapshot copy*.

All data blocks referenced by this Snapshot copy and any previous Snapshot copies are transferred and written to the secondary file system.

After the initialization is complete, the primary and secondary file systems will have at least one Snapshot copy in common.

After the baseline transfer is complete, SnapMirror can transition into synchronous mode:

1. A first asynchronous SnapMirror update occurs, as described earlier.
2. Consistency point (CP) forwarding begins. This is a method to make sure that writes of data from memory to disk storage are transferred from the primary system to the secondary system. For more information, see the CP forwarding section later in this guide. A second asynchronous SnapMirror update is started.
3. After the second asynchronous SnapMirror update completes, NVLOG forwarding begins. This is a method to transfer updates as they occur. For more information, see the NVLOG forwarding section later in this guide.

New writes from clients or hosts to the primary file system begin to block until acknowledgment of those writes has been received from the secondary system.

After SnapMirror has determined that all data acknowledged by the primary system has been safely stored on the secondary, the system is in synchronous mode. At this point the output of a SnapMirror status query will display the relationship as "In-Sync."

3.3 Understanding NVLOG Forwarding

NVLOG forwarding is a critical component of how synchronous mode works. It is the method used to take write operations submitted from clients against the primary file systems to be replicated to the destination.

In order to completely understand NVLOG forwarding, a basic knowledge of how Data ONTAP performs local file system writes is required. The following description assumes a NetApp system running Data ONTAP, ignores interaction with SnapMirror (which will be covered next), and for simplicity describes a standalone controller rather than an active-active configuration. We also assume a functional operating environment where all writes succeed; description of how each possible error condition is handled is outside the scope of this document.

1. The storage system receives a write request. This request could be a file-oriented request from an NFS, CIFS, or DAFS client, or it could be a block-oriented request using FCP or iSCSI.
2. The request is journaled in battery backed-up, nonvolatile memory (NVRAM). It is also recorded in cache memory, which has faster access times than NVRAM.
3. After the request is safely stored in NVRAM and cache memory, Data ONTAP acknowledges the write request to the client system, and the application that requested the write is free to continue processing. At this point the data has not been written to disk, but is protected from power failure and most types of hardware problems by the NVRAM.
4. Under certain conditions, a consistency point (CP) is triggered. Typically this scenario will occur when the NVRAM journal is one-half full, or when 10 seconds have passed since the most recent CP, whichever comes first.
5. When a CP is triggered, Data ONTAP uses the transaction data in cache memory to build a list of data block changes that need to be written to disk. It also computes parity information at this time. After the required disk modifications have been determined, WAFL® sends a list of data blocks to be written to the RAID software.

The RAID software writes the data blocks out to disk. When it is complete, it returns an acknowledgment to the WAFL software, and Data ONTAP considers the CP complete.

When NVLOG forwarding is active in synchronous mode, steps 2 and 3 are modified as follows:

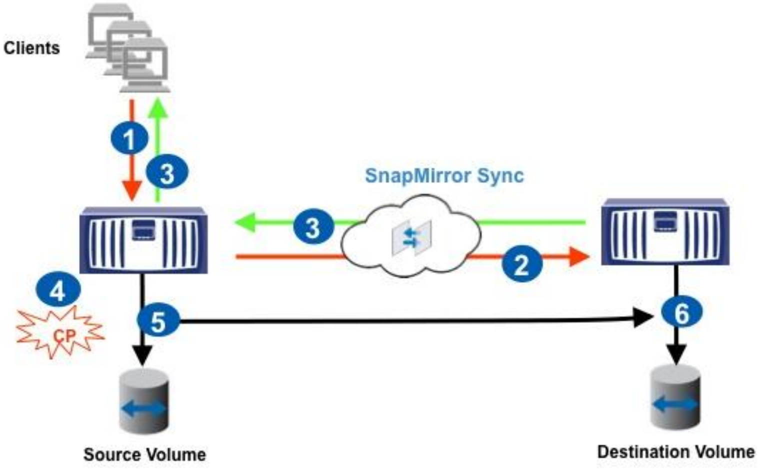
2. The request is journaled in NVRAM. It is also recorded in the cache memory and transferred over the network to the SnapMirror destination system, where it is journaled in NVRAM.
3. After the request is safely stored in NVRAM and cache memory on the primary system and in the NVRAM of the secondary system, Data ONTAP acknowledges the write to the client system. The application that requested the write request can continue to process user requests.

Therefore, to synchronously protect data, NVLOG forwarding is the primary mechanism.

It is important to know how the NVLOG data is stored on the secondary storage system. Since this system has its own storage as well as the mirrored data from the primary system (at the very least, every system has its own root volume), and because CPs need to be kept synchronized on any mirrored volumes, the NVLOG data cannot be stored in NVRAM on the secondary system in the same way as normal file system writes. Instead, the NVLOG data is treated as a stream of writes to a set of special files (replication log files) in the aggregate of the replicated volume on the secondary system. The write requests being sent to these files are logged in the secondary system's NVRAM just like any other write request.

Because the NVLOG data needs to be written to these files in the parent aggregate, there needs to be some free space in the aggregate to accommodate the log files.

Figure 1) SnapMirror Sync functional diagram.



Note: *Flexible* volumes only: Before Data ONTAP 7.2.2, these log files were stored in the root volume of the secondary system, and therefore the performance of the root volume on the secondary system had a direct impact on the overall performance of synchronous or semi-synchronous SnapMirror.

NVLOG data is still stored in the root volume on the secondary system in case of traditional volumes for all versions of Data ONTAP.

3.4 Understanding CP Forwarding

While NVLOG forwarding protects the data by synchronously replicating it from primary to secondary storage, it functions independently from file system writes to disk on each storage system. CP forwarding is the underlying mechanism used to make sure that the actual on-disk file system images, independent of NVRAM or NVLOG forwarding, are kept synchronized. CP forwarding (sometimes referred to as CP sync) modifies steps 5 and 6 earlier as follows:

5. When a CP is triggered, Data ONTAP uses the transaction data in the cache memory to build a list of data block changes that need to be written to disk. It also computes parity information at this time. After the required disk modifications are ready, the local RAID software writes the data blocks to the disk, and the data blocks are also sent across the network to the secondary system, which initiates its own write to disk.

The secondary storage also writes the data blocks out to disk. When each system completes this process, it returns an acknowledgment to the WAFL software on the primary storage system, and Data ONTAP considers the CP complete.

The summary of high-level steps and a diagram depicting both NVLOG and CP forwarding are as follows:

1. The storage system receives a write request. This request could be a file-oriented request from an NFS, CIFS, or DAFS client, or it could be a block-oriented request using FCP or iSCSI.
2. The request is journaled in NVRAM. It is also recorded in cache memory and forwarded over the network to the SnapMirror destination system, where it is journaled in NVRAM.
3. After the request is safely stored in NVRAM and cache memory on the primary system and in NVRAM of the secondary systems, Data ONTAP acknowledges the write to the client system, and the application that requested the write is free to continue processing.
4. Under certain conditions, a consistency point (CP) is triggered. Typically this occurs when the NVRAM journal is one-half full, or when 10 seconds have passed since the most recent CP, whichever comes first.

When a CP is triggered, Data ONTAP uses the transaction data in cache memory to build a list of data block changes that need to be written to disk. It also computes parity information at this time. After the required disk modifications are ready, the local RAID software writes the data blocks to the disk, and the data blocks are also sent across the network to the secondary system, which initiates its own write to disk.

The secondary storage also writes the data blocks out to disk. When each system completes this process, it returns an acknowledgment to the WAFL software on the primary storage system, and Data ONTAP considers the CP complete.

3.5 SnapMirror Semi-Sync Overview

SnapMirror provides a semi-synchronous mode, also called SnapMirror Semi-Sync. In this mode, the application does not need to wait for the secondary storage system to acknowledge the write request before continuing with the transaction. Therefore, it is possible to lose acknowledged data. This mode is similar to synchronous mode in that updates from the primary storage to the secondary storage occur when a CP is triggered, rather than waiting for scheduled transfers. Therefore, the amount of data loss in a disaster is very small.

Thus, SnapMirror Semi-Sync makes a reasonable compromise between performance and data protection for many applications. The preceding scenario resembles the following when using semi-synchronous mode:

1. A user saves some information in the application.
2. The client software communicates with a server and transmits the information.
3. The server software processes the information and transmits it to the operating system on the server.
4. The operating system software sends the information to the primary storage.
5. The operating system informs the application server that the write is complete.
6. The application server informs the client that the write is complete.
7. The client software informs the user that the write is complete.

If the secondary storage system is slow or unavailable, it is possible that a large number of transactions could be acknowledged by the primary storage system and yet not be protected on the secondary. These transactions represent a window of vulnerability to loss of acknowledged data. For a window of zero size, customers might of course use synchronous mode rather than semi-synchronous mode.

SnapMirror Semi-Sync Changes Starting with Data ONTAP 7.3

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 with the use of an option called `outstanding` in the SnapMirror configuration file. Starting with Data ONTAP 7.3, the `outstanding` option functionality is removed, and there is a new mode called `semi-sync`. For more information, refer to section 3.6, "How SnapMirror Semi-Sync Works."

3.6 How SnapMirror Semi-Sync Works

Semi-synchronous mode provides a middle ground that keeps the primary and secondary file systems more closely synchronized than in asynchronous mode, but with less application performance impact than in synchronous mode. Configuration of semi-synchronous mode is very similar to that of synchronous mode by simply replacing "`sync`" with "`semi-sync.`" An example follows:

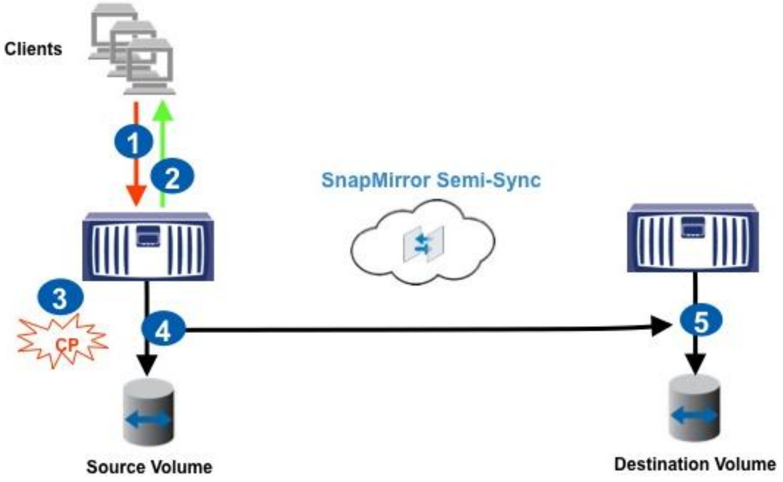
```
fas1:vol1   fas2:vol1   -   semi-sync
```

The key difference between synchronous and semi-synchronous modes is that NVLOG forwarding is turned off in semi-synchronous mode. This means that client writes are acknowledged soon after the primary system writes the data to its NVRAM. Since NVLOG forwarding is turned off, the data is now protected at the secondary site only through CP forwarding. Since the maximum window of time between any two CPs is 10 seconds, a disaster at the primary site could result in a data loss of approximately 10 seconds.

The summary of high-level steps and a diagram depicting SnapMirror Semi-Sync are as follows:

1. The storage system receives a write request. This request could be a file-oriented request from an NFS, CIFS, or DAFS client, or it could be a block-oriented request using FCP or iSCSI.
2. After the request is safely stored in NVRAM and cache memory on the primary system, Data ONTAP acknowledges the write to the client system, and the application that requested the write is free to continue processing.
3. Under certain conditions, a consistency point (CP) is triggered. Typically, this occurs when the NVRAM journal is one-half full, or when 10 seconds have passed since the most recent CP, whichever comes first.
4. When a CP is triggered, Data ONTAP uses the transaction data in cache memory to build a list of data block changes that need to be written to disk. It also computes parity information at this time. After the required disk modifications are ready, the local RAID software writes the data blocks to the disk, and the data blocks are also sent across the network to the secondary system, which initiates its own write to disk.
5. When the secondary storage system receives the CP data blocks, it sends an acknowledgement to the primary storage system, and Data ONTAP considers the CP complete.

Figure 2) SnapMirror Semi-Sync functional diagram.



3.7 Visibility Interval

With SnapMirror Sync and SnapMirror Semi-Sync, changes to the source volume do not show up immediately on the destination volume, even though the changes have been replicated. The changes are first shown after the source system creates an automatic Snapshot copy of the source volume. Snapshot copies created on the source volume are immediately propagated to the destination volume. The automatic Snapshot copies are created every three minutes by default. To change the interval for automatic Snapshot copies, change the `visibility_interval` in the `snapmirror.conf` file; however, performance can degrade if set to a smaller interval because frequent Snapshot copies cause additional processing such as cache consistency checking on the destination system. There is also a disadvantage in setting this to a large value. When the connection between the systems has an outage and SnapMirror goes into asynchronous mode, SnapMirror uses the last common Snapshot copy to transition into synchronous mode. This means that all data from the last valid common Snapshot copy will need to be replicated from the source to the destination storage system. If the `visibility_interval` is set to a large value, a large amount of data might have to be transferred and it might take longer time for SnapMirror to transition into synchronous mode. For these reasons, keep the default value of three minutes.

Best Practice
Use the default value of three minutes for the visibility interval.

3.8 LUN Visibility on the Destination System

There are a few subtleties with regard to LUN visibility on the destination storage system that are important to understand in order to architect or program host access during a failover situation. When the mirrored volume is in a *replicated* status, the LUNs contained in the volume on the destination storage system are online, read-only, unmapped, and therefore not visible to any host systems. After using the `snapmirror quiesce` and `snapmirror break` commands to split or break the mirror and make it writable, the LUNs can be made visible by mapping them to an igroup. The igroup itself is not copied or replicated from the original system to the destination system, so it most likely will have to be created on the destination system, and the desired hosts will need to be added. The following table displays the LUN status during a mirrored and broken mirror situation. NetApp LUNs are independent of the protocol they are accessed with, meaning that the LUNs on the destination system do not need to be accessed with the same protocol as they were accessed with on the primary system. For example, it is possible to access the primary storage system using Fibre Channel and the secondary system using iSCSI.

Using FlexClone, LUNs on the SnapMirror destination can be made writable *without* interrupting the SnapMirror process.

Note: In Data ONTAP 7.0, LUN commands might incorrectly show LUNs being online when the destination volume is in the replicated state.

Table 5) LUN status on SnapMirror destination systems.

Replication State	DR LUN Status on the SnapMirror Destination
Mirror is being synchronized	Online, read-only, unmapped *Note*: DR LUN can be made writable using FlexClone
Mirrors is broken or split	Online, read-write, unmapped

3.9 Destination System Unreachable

If the destination system is not accessible, regardless whether the problem lies with the destination system, networking, or elsewhere, synchronous and semi-synchronous modes automatically drop to asynchronous mode. In this scenario, the application continues to be up and available processing user transactions. SnapMirror periodically checks for network connectivity. When the replication network is available, SnapMirror first performs asynchronous updates and then transitions into synchronous mode. No action is necessary on the part of the user. To prevent SnapMirror from checking whether the network connectivity is available, use the `snapmirror quiesce` command.

The write requests taking place at the time of failure can have a response time of up to 25 seconds as the source system determines that the destination system is not available and switches to asynchronous mode. To allow monitoring of this important state change, an SNMP trap is sent.

3.10 Cascading Support and Other Configurations

Cascading is defined as replicating from established replicas. Suppose there are three storage systems, A, B, and C. Replicating from A to B and from B to C is considered a cascade configuration. While

SnapMirror Async has more flexibility in cascading, SnapMirror Sync and SnapMirror Semi-Sync configurations are slightly restricted. For details, see the following cascading configuration support table.

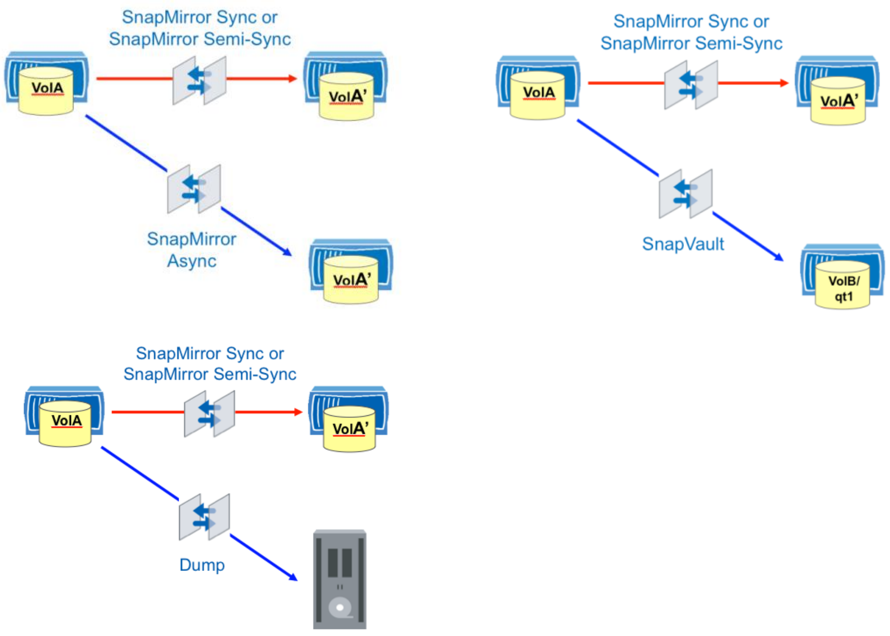
Table 6) SnapMirror Sync and SnapMirror Semi-Sync cascade support.

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

Other Supported Configurations

It is possible to replicate a SnapMirror Sync or SnapMirror Semi-Sync primary data by other means such as SnapVault, qtree SnapMirror or using dump. These supported configurations are shown in Figure 3. Note that qtree SnapMirror and SnapVault will only replicate qtrees from the source volume.

Figure 3) Other supported configurations.



4 Design and Performance Considerations

4.1 Concurrent Replication Operations

SnapMirror imposes limitations on how many volumes can be concurrently replicated between source and destination systems. This becomes a key factor for data layout since SnapMirror Sync and SnapMirror Semi-Sync are continuously mirroring the data, unlike SnapMirror Async, where the volume replication schedule can be staggered to prevent reaching the concurrency limits. The concurrent limits are dependent on the platform and are well documented in the Data Protection Online Backup and

Recovery Guide on the NetApp Support site. Refer to this document for the appropriate Data ONTAP release.

Therefore, if all the data on a storage system needs to be replicated, SnapMirror concurrent transfer limits must be kept in mind so that the number of volumes the data needs for synchronous replication is not greater than the concurrent transfer limits for a given platform and Data ONTAP release.

HA configuration: This applies to the scenario where both nodes in an HA configuration are replicating volumes. When a failover occurs, the surviving node cannot process more than the maximum number of concurrent replication operations specified for that node. For example, each FAS3050 node in an HA configuration can run a maximum of 16 concurrent replication operations. In an active-active configuration, the FAS3050HA can replicate 32 concurrent operations. If one FAS3050 fails over to the other, the surviving FAS3050 can now only replicate a maximum of 16 concurrent operations. Therefore, you will encounter replication failure for the excess 16 volumes until the HA node is back up and available.

4.2 Data ONTAP Upgrade and Revert Considerations

When upgrading and reverting between *major* Data ONTAP versions, exercise caution when deleting Snapshot copies that exist in the older Data ONTAP version. Consider the following scenario. The system is upgraded from Data ONTAP 7.2 to Data ONTAP 7.3. All the Snapshot copies are recycled since the upgrade, meaning all the Snapshot copies that currently exist have been created after the upgrade. Since SnapMirror Sync and SnapMirror Semi-Sync maintains the same set of Snapshot copies on source and destination, the same holds true for destination system as well. If the system needs to be reverted back to Data ONTAP 7.2 for whatever reason, the relationship needs to be initialized because one of the steps during revert is to delete the Snapshot copies created in Data ONTAP 7.3.

Best Practice

This situation can be avoided by manually creating a Snapshot copy on the older Data ONTAP version before the upgrade. This manually created Snapshot copy can be deleted when you recognize there is no need for revert procedure.

4.3 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 volume 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 on the source, 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 will fail when the volume runs out of space.

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

See the following example to demonstrate space usage with and without volume guarantees on the SnapMirror destination volume.

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.

4.4 Space Requirements

As discussed earlier, SnapMirror uses NetApp Snapshot copies to make changes visible on the destination storage system. Therefore, these Snapshot copies are the only extra space required by synchronous SnapMirror. Rolling Snapshot copies are used, and therefore the space required is typically minimal, because these Snapshot copies only exist for a short period of time, which is defined by the `visibility_interval` in the `snapmirror.conf` file. However, care needs to be taken, because if the connection between the primary site and the secondary site goes down, this Snapshot copy will exist until the link is reestablished and will grow during that time. Reserve space for these Snapshot copies and, depending on the amount of space reserved, processes needs to be established for dealing with the Snapshot copies if the connection between sites is lost.

4.5 SnapDrive

SnapDrive® for Windows® and SnapDrive for UNIX® can be used for managing (creating, deleting, 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. Starting with SnapDrive for Windows 3.2, SnapDrive for Windows is capable of breaking the SnapMirror relationship and connecting to the LUNs on the destination system with synchronous and semi-synchronous modes of SnapMirror. For more information, refer to the SnapDrive for Windows Installation and Administration Guide on the NetApp Support site.

SnapDrive for UNIX does not have any integration with SnapMirror Sync and SnapMirror Semi-Sync. However, you can still create Snapshot copies using SnapDrive for UNIX on the SnapMirror source. SnapMirror will then replicate those Snapshot copies to the destination. SnapDrive for UNIX will not break the SnapMirror relationship when connecting to the LUNs on the destination volume, unlike SnapDrive for Windows. For more information, refer to the SnapDrive for UNIX Installation and Administration Guide on the NetApp Support site.

4.6 Use of SnapMirror Sync and SnapMirror Semi-Sync with Other NetApp Products

The following table summarizes the support of other NetApp products with SnapMirror Sync and SnapMirror Semi-Sync.

Table 7) Use of SnapMirror Semi-Sync with other products.

Configuration	Support	Comments
Deduplication	No	A SnapMirror Sync and SnapMirror Semi-Sync volume cannot be deduplicated. Furthermore, a FAS deduplication volume and SnapMirror Sync and SnapMirror Semi-Sync volume cannot belong to the same aggregate.
FlexClone	Yes	FlexClone can be used on both SnapMirror source and destination systems.

MultiStore®	Yes	SnapMirror Sync and SnapMirror Semi-Sync is supported from a default vFiler® instance (vfiler0) context. The replication relationship must be configured in the default vFiler /etc/snapmirror.conf with default vFiler hostname. For more information, refer to the MultiStore administration guide on the NetApp Support site.
MetroCluster™	Yes	There are no restrictions using MetroCluster and SnapMirror Sync and SnapMirror Semi-Sync together.
Protection Manager	No	Protection Manager does not support configuring, managing, or monitoring SnapMirror Sync and SnapMirror Semi-Sync relationships.
SnapDrive for Windows	Yes	SnapMirror *source*: SnapDrive for Windows is capable of creating Snapshot copies on the source system regardless of the mode of SnapMirror being used. SnapMirror *destination*: SnapDrive for Windows is capable of making the SnapMirror LUNs writable when connecting to the destination SnapMirror LUNs.
SnapDrive for UNIX	Yes	SnapMirror *source*: SnapDrive for UNIX is capable of creating Snapshot copies on the source system regardless of the mode of SnapMirror being used. SnapMirror *destination*: SnapDrive for UNIX has no integration with SnapMirror destination LUNs. The SnapMirror destination LUNs can be made writable by manually breaking the SnapMirror relationship.
SnapManager® suite	Yes	The SnapManager suite of products can be used with SnapMirror Sync or SnapMirror Semi-Sync. However, there is ***no*** integration or automation with SnapMirror Sync or SnapMirror Semi-Sync similar to SnapManager for Exchange integration with volume SnapMirror. Refer to the respective SnapManager application documentation for support and best practices information.
SnapLock® Compliance	No	A SnapMirror Sync and SnapMirror Semi-Sync volume cannot be a SnapLock Compliance volume.
SnapLock Enterprise	Yes	There are no special restrictions when using SnapLock Enterprise volumes with SnapMirror Sync and SnapMirror Semi-Sync.

4.7 Combined SnapMirror Async and SnapMirror Sync Configuration

For many database environments a combination of synchronous and asynchronous is a very good solution in terms of performance and recovery time tradeoffs. A typical combined configuration is a separate volume containing the data log files, and this volume is replicated with either SnapMirror Sync or SnapMirror Semi-Sync to the secondary site. The other volume containing the database data files is then asynchronously replicated to the secondary site. When an event renders the primary site unusable, the database is then started on the secondary site. The log files holding the updates since the last asynchronous SnapMirror update of the database will then have to be played back into the database. For many database environments this configuration lengthens the recovery time minimally while reducing the performance impact of running full synchronous or semi-synchronous operations on all volumes. When using this methodology of combining synchronous and asynchronous modes for log and database volumes respectively, refer to the appropriate documentation on best practices for the database in the NetApp technical library.

4.8 Replication Network

Bandwidth

Since all of the data written to the primary storage must be replicated to the secondary storage as it is written, write throughput to the primary storage cannot generally exceed the bandwidth available between the primary and secondary storage devices. It is important not to look at average write performance, but in order to avoid undesired performance impact, it is necessary to plan for the peaks in write performance.

In general, the configuration guideline is to configure the network between the primary and secondary storage with two times the peak write throughput at the primary system.

Multiple Paths for Replication

More than one physical path between a source system and a destination system might be desired for a replication relationship. SnapMirror supports 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. SnapMirror supports up to two replication paths for each 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 on one path, the load from both transfers will move to the remaining path.
- Failover mode: One path is specified as the primary path in the configuration file. This path is the desired path and will be used until a failure occurs. The second path would then be used.

Best Practice
Multipath is recommended to improve availability of the replication network.

4.9 Write Latency Due to Replication Distances

Network communications between the primary and secondary systems are limited to the speed of light over the transmission media. While it is best to measure the actual latency on an existing network, rather than make calculations based on theory, the speed of light across fiber can be estimated to produce approximately one millisecond of latency per 100 miles of distance. So if the primary and secondary storage systems are placed 100 miles apart, one could expect a round trip latency of two milliseconds. Use caution when using any synchronous replication over long distances (> 200 kilometers) for latency-sensitive applications. Know the latency requirements (peak and average) for the application before implementing any synchronous replication solution.

- SnapMirror Sync: The client receives an acknowledgement only after each write operation is written to both primary and secondary storage systems. Therefore, the round trip time needs to be added to the latency of the application write operations.
- SnapMirror Semi-Sync: Even though the writes from the client are immediately acknowledged, the primary and the secondary systems are still synchronized by consistency points (CPs). Higher network latencies could mean back-to-back CP scenarios, which impacts client writes on the primary. Back-to-back CP is where CP follows a CP before the primary system receives the CP

acknowledgement from the secondary system. This typically occurs in heavy write workload environments.

Best Practice

To minimize write impact of the primary workloads do not exceed two milliseconds of round trip time (RTT) for SnapMirror Sync and five to 10 milliseconds of RTT for SnapMirror Semi-Sync.

4.10 Firewall Configurations

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.

4.11 Choosing the Appropriate Mode

There are a few factors involved to help decide which mode of replication is appropriate for a given dataset. These factors are arranged in a matrix to help pick the most appropriate mode for the business requirement.

Table 8) Performance impact for each SnapMirror mode.

Mode of Replication	RPO Requirements	Round Trip Time Between Primary and Secondary Sites	Performance Impact
SnapMirror Sync	Zero or near-zero	2 ms	Medium to high
SnapMirror Semi-Sync	Near-zero to minutes	5 to 10 ms	Low to medium
SnapMirror Async	Minutes to hours	Any	Low

Appendix

Failover and Failback with SnapMirror

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

Planned Failover (No Disaster)

Failover

This assumes there is ongoing SnapMirror replication between the primary site and the DR site. This is how the SnapMirror configuration file would look on fas2:

```
fas1:vol1 fas2:vol2 - sync
```

1. Shut down all applications at the production site.

Perform a SnapMirror update to create a consistent Snapshot copy. Make the DR volumes writable by breaking the SnapMirror relationships.

a. On fas2: `snapmirror update –w vol2`

b. On fas2: `snapmirror quiesce vol2`

c. On fas2: `snapmirror break vol2`

Bring up the applications at the DR site. This assumes all DNS changes, NFS and CIFS exports, and LUN mapping are completed.

Failover is now complete.

Replicating to the Primary Site

1. Since this is a planned failover, it is assumed that the data at the primary site is intact at the time of failover.

Now that there is new data at the DR site, this data needs to be replicated back to the primary site to prevent data loss in case of a disaster at the DR site. This is achieved by the SnapMirror resync command. This is always done at the desired destination site, the primary site in this step. The resynchronization step will send *only* the changes since creation of the last common Snapshot copy between the primary and the DR sites.

a. On fas1: `snapmirror resync –S fas2:vol2 fas1:vol1`

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 synchronous replication, SnapMirror performs asynchronous updates from the DR site to the primary site and then transitions into synchronous mode. The SnapMirror configuration file on fas1 would look like this:

a. `fas2:vol2 fas1:vol1 - sync`

Failing Back to the Primary Site

1. Shut down all applications at the DR site.

Perform a SnapMirror update to create a consistent Snapshot copy. Make the primary volumes writable by breaking the SnapMirror relationships.

a. On fas1: `snapmirror update –w vol1`

b. On fas1: `snapmirror quiesce vol1`

c. On fas1: `snapmirror break vol1`

Bring up the applications at the production site. This assumes all DNS changes, NFS and CIFS exports, and LUN mapping are completed.

Failback is now complete.

Replicating to the DR Site

1. Now that the production site is active, there is new data at this site that needs to be replicated to the DR site.

This is achieved by the SnapMirror resync command. This is always done at the desired destination site, the DR site in this step. The resynchronization step will send *only* the changes since creation of the last common Snapshot copy between the primary and the DR sites.

   a. On fas2: `snapmirror resync –S fas1:vol1 fas2:vol2`

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) will receive synchronous updates from the DR site. SnapMirror performs asynchronous updates from the primary site to the DR site and then transitions into synchronous mode. The SnapMirror configuration file on fas2 would look like this:

   a. `fas1:vol1 fas2:vol2 - sync`

Failover in the Event of a Real Disaster

This assumes that primary 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. The NVLOG data on the DR system is replayed in case of synchronous SnapMirror.

   a. On fas2: `snapmirror break vol2`

Bring up the applications at the DR site. This assumes all DNS changes, NFS and CIFS exports, and LUN mapping are completed.

Failover is now complete.

Replicating to the Primary Site

1. After the primary site is accessible, the first step is to determine if the data is intact or lost.

If there is complete loss of data, the primary site needs to be reinitialized (using `snapmirror initialize`) from the DR site. The reinitialization is always performed at the destination site, the primary site in this case. If there is no loss of data, only the changes can be transferred to the primary site. This is achieved by the `snapmirror resync` command. This is always done at the desired destination site, the primary site in this step. The resynchronization step will send *only* the changes since creation of the last common Snapshot copy between the primary 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`

Set up the primary site (now standby) to be the SnapMirror destination by editing the SnapMirror configuration file. After the configuration file is set up, SnapMirror performs asynchronous updates from the DR site to the primary site and then transitions into synchronous mode. The SnapMirror configuration file on fas1 would look like this:

   a. `fas2:vol2 fas1:vol1 - sync`

Failing Back to the Primary Site

1. Shut down all applications at the DR site.

Perform a SnapMirror update to create a consistent Snapshot copy. Make the primary volumes writable by breaking the SnapMirror relationships.

   a. On fas1: `snapmirror update –w vol1`
   b. On fas1: `snapmirror quiesce vol1`
   c. On fas1: `snapmirror break vol1`

Bring up the applications at the primary site. This assumes all DNS changes, NFS and CIFS exports, and LUN mapping are completed.

Failback is now complete.

Replicating to the DR Site

1. Now that the primary site is active, there is new data at this site that needs to be replicated to the DR site.

This is achieved by the SnapMirror `resync` command. This is always done at the desired destination site, the DR site in this step. The resynchronization step will send *only* the changes since creation of the last common Snapshot copy between the primary and the DR sites.

  a. On fas2: `snapmirror resync –S fas1:vol1 fas2:vol2`

Set up the DR site (now standby) to be the SnapMirror destination by editing the SnapMirror configuration file. This step will restore the replication relationships to its original state. Once the configuration file is set up, SnapMirror performs asynchronous updates from the primary site to the DR site and then transitions into synchronous mode. The SnapMirror configuration file on fas2 would look like this:

  a. `fas1:vol1 fas2:vol2 – sync`

Housekeeping

After the failback is completed, old SnapMirror relationships can be deleted 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.

Version History

Version	Date	Document Version History
Version 2.8	February 2013	Author: Amit Prakash Sawant Updated the SnapMirror Sync and Semi-Sync Interoperability Matrix
Version 2.7	November 2012	Updated for licensing changes from Data ONTAP 8.0 to 8.1 7-Mode
Version 2.6	March 2012	Moving author names from the front page to the version history Authors: Srinath Alapati, Neil Shah
Version 2.5	November 2011	Updated for Data ONTAP 8.1 7-Mode Updated Storage Section for Aggregates (section 2.2)
Version 2.4	April 2011	Added table showing support between disk types of FC and SAS
Version 2.3	March 2011	Clarified section regarding hardware recommended with SnapMirror Sync
Version 2.2	July 2010	Updated for Data ONTAP 7.3.3 (mixed disk support)
Version 2.1	March 2009	Updated for V-series, added upgrade/revert considerations section
Version 2.0	July 2008	Reorganized the document Updated for Data ONTAP 7.3

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp®

www.netapp.com