EMC²
where information lives®

White Paper

MirrorView Knowledgebook: FLARE 26

Applied Technology

Abstract

This white paper is a comprehensive guide of MirrorView™ functionality, operations, and best practices. The specifics of both synchronous (MirrorView/S) and asynchronous (MirrorView/A) products are discussed. Comparisons of both operational models are made to help users determine how each is best deployed.

July 2007

Part number H2417.2

Table of Contents

Executive summary

EMC® CLARiiON® MirrorView™ software offers two complementary, but separately licensed, storage system-based remote mirroring products: MirrorView/Synchronous (MirrorView/S) and MirrorView/Asynchronous (MirrorView/A). These products are designed as disaster recovery solutions for mirroring local production data to a remote/disaster recovery site. MirrorView/S is a synchronous product that mirrors data in real time between local and remote storage systems. MirrorView/A is an asynchronous product that offers extended-distance replication based on a periodic incremental-update model. It periodically updates the remote copy of the data with all the changes that occurred on the primary copy since the last update.

Features such as consistency groups allow applications that are write-order dependent across volumes to manage several volumes as one volume. Lastly, data replication for both products is supported over storage area networks (SANs) as well as IP extended SAN environments.

Introduction

MirrorView is storage system-based disaster recovery (DR) software that provides end-to-end data protection by replicating the contents of a primary volume to a secondary volume that resides on a different CLARiiON storage system. It provides end-to-end data protection because, in addition to performing replication, it protects the secondary volume from tampering or corruption by only making the volume available for server access when initiated through MirrorView.

MirrorView offers consistency groups, a unique consistency technology for the midrange market that replicates write-order dependent volumes. Using this technology, MirrorView maintains write ordering across secondary volumes in the event of an interruption of service to one, some, or all of the write-order dependent volumes.

Business requirements determine the structure of a disaster recovery solution. The business will decide how much data loss is tolerable, if any, and how soon the data must be accessible again in the event of a disaster. It is common for businesses to classify their applications and employ a disaster recovery strategy for each application, balancing service levels with cost.

MirrorView/S offers a zero data loss option, while MirrorView/A offers an alternative when minutes of data loss may be tolerable. Consistency groups enable both MirrorView products to offer application *restart*. That is, applications with write-ordered dependent volumes, like databases, can resume operating on the secondary image with little to no maintenance required. This is compared to application *recovery*, which typically entails restoring from an older copy, such as a backup, and applying changes to bring the application to its state before the disaster.

SAN Copy™ is another storage-system-based remote replication product. It is designed as a multipurpose replication product for data mobility, migrations, content distribution and DR. However, SAN Copy does *not* provide the complete end-to-end protection that MirrorView provides; instead it provides an efficient replication mechanism that can be used as part of a DR strategy.

SAN Copy typically plays a complementary role with other software to form a robust DR solution. Applications such as Replication Manager, RepliStor®, or native application replication can be used in conjunction with SAN Copy to manage application integration and consistency across volumes. Table 1 is a brief side-by-side comparison of MirrorView and SAN Copy.

Table 1. MirrorView and SAN Copy use case comparison

Product	MirrorView	SAN Copy
Storage system support	Replication between CLARiiON primary and secondary systems	Replication between CLARiiON, Symmetrix®, and non-EMC systems
Content distribution	1 primary to 1 secondary async; 2 secondaries sync (1:1 with con groups)	1 source copied to up to 100 targets
Data protection	Continuous data protection of secondary volumes	Remote copy is available for server access
Consistency across volumes	Native consistency group support	Consistency managed by the user (Ex: hot backup mode) or another application (Ex: Replication Manager)

Audience

This white paper is intended for systems integrators, systems administrators, customers, and members of the EMC and partners' professional services community. It is assumed that the audience is familiar with CLARiiON storage systems and replication applications such as SAN Copy and SnapView™. This paper provides an overview for MirrorView software and discusses key software features, functions, and best practices. Also refer to the "References" section of this white paper for related information, including help files, administrator guides, and white papers.

What's new

This white paper is updated to describe enhancements to MirrorView, including:

- The "Interoperability" section for the FLARE® operating environment is updated to include information about FLARE release 26.
- A "MirrorView over iSCSI (new for release 26)" section was added to the "CLARiiON MirrorView family" section.
- The "MirrorView ports and connections" section is updated to include MirrorView replication over iSCSI between CX3 series FC/iSCSI storage systems that have both FC and iSCSI front-end ports.
- The "MirrorView failure recovery scenarios" section discusses the CLARiiON Asymmetric Active/Active feature, which is based on the Asymmetric Logical Unit Access (ALUA) standard. It also discusses added back-end fault tolerance capabilities.
- An "iSCSI connections" section has been added to discuss creating iSCSI connections between storage systems

Please also note that as of release 24, MirrorView documentation is available in the *Navisphere Manager Help* file and individual administrator guides are no longer available. The help file is available separately in the Documentation/White Paper Library section of Powerlink (EMC's customer- and partner-only extranet) at **Navisphere Management Suite > Maintenance/Administration** > **Navisphere Manager Help**. Command line interface administrator's guides are still available for MirrorView/A and MirrorView/S.

Terminology

Terminology, operations, and object statuses are the same for both MirrorView products. The following is a list of frequently used terms and conditions. A more comprehensive list is available in the product documentation.

- **Primary image –** The LUN containing production data and the contents of which are replicated to the secondary image. Also referred to as the primary.
- **Secondary image –** A LUN that contains a mirror of the primary image LUN. Also referred to as the secondary. This LUN must reside on a different CLARiiON storage system than the primary image.

- **Image Condition**[1] **–** The condition of a secondary image provides additional information about the status of updates for the image. Values include **normal**, **administratively fractured**, **system fractured**, **queued to be synchronized**, or **synchronizing**.
- **State**[1] **–** Remote mirror states and image states. The remote mirror states are: **Active** and **Attention**. The image states are: **Synchronized**, **Consistent**, **Synchronizing**, and **Out-of-Sync**.
- **Consistency group –** A set of mirrors that are managed as a single entity and whose secondary images always remain in a consistent and recoverable state with respect to their primary image and each other.
- **Consistency Group State –** Indicates the current state of the consistency group: **Synchronized**, **Consistent**, **Synchronizing**, **Out-of-Sync**, **Scrambled**, **Empty**, **Incomplete**, or **Local**.
- **Consistency Group Condition –** Displays more detailed information about the consistency group, including whether the group is **active**, **inactive**, **admin fractured**, **system fractured**, **waiting on admin**, or **invalid.**
- **Fracture –** A condition in which I/O is not mirrored to the secondary image; this can be caused when you initiate the fracture (**Admin Fracture**) or when the system determines that the secondary image is unreachable (**System Fracture**).
- **Promote –** The operation by which the administrator changes an image's role from secondary to primary. As part of this operation, the previous primary image becomes a secondary image. If the previous primary image is unavailable when you promote the secondary image (perhaps because the primary site suffered a disaster), the software does not include it as a secondary image in the new mirror. A secondary image can be promoted if it is in either the **Synchronized** state or the **Consistent** state. An image cannot be promoted if it is **out-of-sync** or **synchronizing**.

Remote replication overview

It is a requirement that critical business information always be available. To protect this information, it is necessary for a disaster recovery plan to be in place to safeguard against any disaster that could make the data at the primary site unavailable.

Recovery objectives

Recovery objectives are service levels that must be met to minimize the loss of information and revenue in the event of a disaster. The criticality of business applications and information defines the recovery objectives. The terms commonly used to define the recovery objectives are: *recovery point objective (RPO)* and *recovery time objective (RTO)*.

Recovery point objective (RPO)

Recovery point objective (RPO) defines the amount of acceptable data loss in the event of a disaster. The business requirement for RPO is typically expressed as duration of time. For instance, application A may have zero tolerance for loss of data in the event of a disaster. This is typical for financial applications where all completed transactions must be recovered. Application B may be able to sustain the loss of minutes or hours worth of data. RPO determines the required update frequency of the remote site. The rate of change of the information determines how much data needs to be transferred. This, combined with the RPO, has a significant impact on the distance, protocol, and bandwidth of the link between remote sites.

1 A complete list of mirror and image states and conditions are listed in "Appendix A: Remote mirror conditions and image states."

Recovery time objective (RTO)

RTO is defined as the amount of time required to bring the business application back online after a disaster occurs. Mission-critical applications may be required to be back online in seconds, without any noticeable impact to the end users. For other applications, a delay of a few minutes or hours may be tolerable.

Figure 1 shows an RPO and RTO timeline. Stringent RPO and RTO requirements can add cost to a DR solution. Therefore, it is important to distinguish between absolutely critical business applications and all other applications. Every business application may have different values for RPO and RTO, based on the criticality of the application.

Figure 1. RPO and RTO timeline

Replication models

There are a number of replication solutions available to replicate data from the primary (production) to the secondary (remote) site. These replication solutions can be broadly categorized as synchronous and asynchronous.

Synchronous replication model

In a synchronous replication model, each server *write* on the primary side is written concurrently to the secondary site. The primary benefit of this model is that its RPO is zero, since the transfer of each I/O to the secondary occurs before acknowledgement is sent to the server. Figure 2 depicts the data flow of synchronous replication. In case of a disaster at the primary site, data at the secondary site is exactly the same as data at the primary site at the time of disaster.

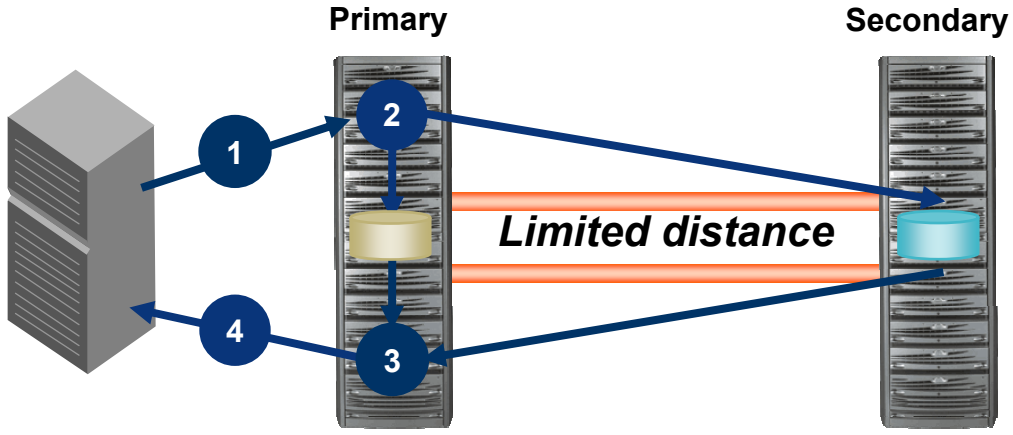

Figure 2. Synchronous replication data flow

The main consideration of the synchronous model is that each server I/O is affected by the time it takes to replicate the data to the remote storage system. Therefore, the distance of the replication may be limited by the application's tolerance for latency. Secondly, the link between the primary and remote systems must be able to handle peak workload bandwidths, which can add cost to the link.

Asynchronous replication model(s)

There are a few general approaches to asynchronous replication and each implementation may vary to some degree. At the highest level, all asynchronous replication models decouple the remote replication of the I/O from the acknowledgement to the server. The primary benefit of this is that it allows longer distance replication because application write response time is not dependent on the latency of the link. The trade off is that RPO will be greater than zero.

In classic asynchronous models, writes are sent to the remote system as they are received from the server. Acknowledgement to the server is not held for a response from the secondary, so if writes are coming into the primary faster than they can be sent to the remote system, multiple I/Os may be queued up on the source system awaiting transfer.

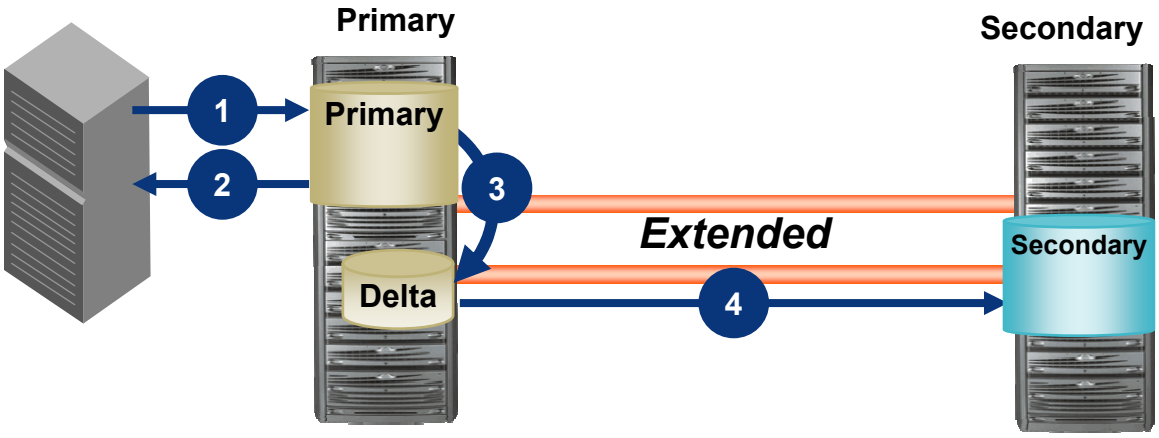

Figure 3. Asynchronous (periodic update) data replication

Periodic update models, as shown in Figure 3, track changes on the primary side and then apply those changes to the secondary at a user-determined frequency. By tracking data areas that change, it is possible to send less data if writes occur to the same areas of the volume between updates. Updates are smoothed out over the entire update period, regardless of bursts of writes to the source LUN allowing for a low bandwidth and low-cost link between sites.

CLARiiON MirrorView family

Since it is storage-system-based software, MirrorView does not consume server resources to perform replication. It can take advantage of Fibre Channel SANs, IP extended SANs, and TCP/IP networks for the transfer medium. Two complementary and separately licensed products are offered in the MirrorView family. Both products run on the CX3 series and CX400 to CX700 storage system models.

- MirrorView/Synchronous (MirrorView/S) for synchronous replication
- MirrorView/Asynchronous (MirrorView/A) for asynchronous replication and uses the periodic update model

For maximum protection of information, both products have advanced features, including:

- Bidirectional mirroring. Any one storage system can host both primary and secondary images as long as the primary and secondary images within any one mirror reside on different storage systems.
- The ability to have both synchronous and asynchronous mirrors on the same storage system
- Consistency groups for maintaining data consistency across write-order dependent volumes

- Mirroring between supported CLARiiON storage system models[2]
- The ability to mirror over Fibre Channel and iSCSI. CX3 Series storage systems with FC and iSCSI front-end ports can mirror over Fibre Channel ports to some storage systems, while mirroring over iSCSI front-end ports to other storage systems.

MirrorView configurations

MirrorView allows for a large variety of topologies and configurations. In all configurations, the primary and secondary images must be the same size, because they are allowed to reverse role for failover and failback. The following sections describe configuration options between primary and secondary storage systems and primary and secondary images.

Bidirectional mirroring

Storage systems running MirrorView can simultaneously house primary volumes for some applications and secondary volumes for others. Each mirrored relationship can run in either **Synchronous** or **Asynchronous** mode. In such instances, system sizing considerations should be well thought out in case either site has to run both primary and secondary applications during an outage.

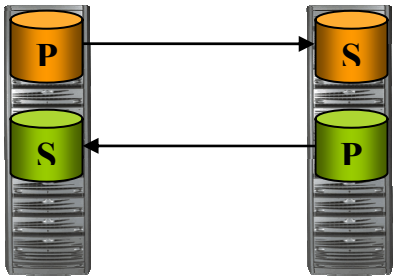

Figure 4. Bidirectional mirroring

System level fan-in/fan-out

CLARiiON systems can have MirrorView paths (MirrorView connections) configured to four other CLARiiONs. Each CLARiiON can have primary images and/or secondary images with any of the systems with which it has a MirrorView connection. Each one of the primary/secondary pairs can be synchronously or asynchronously mirrored allowing for specific levels of protection at the application level.

2 Generally, N-1 generations can replicate between one another. For example, CX3 series systems are qualified to replicate between CX series systems, but not the FC4700. Check the MirrorView/S and MirrorView/A release notes on EMC Powerlink for supported configurations.

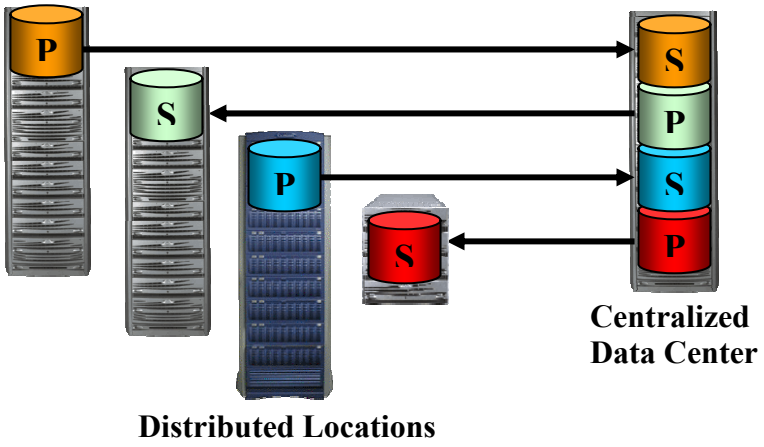

Figure 5. MirrorView 4:1 fan-out/fan-in ratio

LUN level fan-out

MirrorView/S allows for a 1:2 fan-out ratio. Each primary volume mirrored synchronously can have one or two secondary images. The secondary images are managed independently, but must reside on separate storage systems. No secondary image can reside on the same storage system as its primary image. When using consistency groups, MirrorView/S's LUN fan out ratio is 1:1.

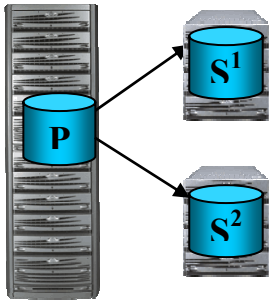

Figure 6. MirrorView/S 1:2 fan-out

MirrorView/A allows for one secondary image per primary image as depicted in Figure 7.

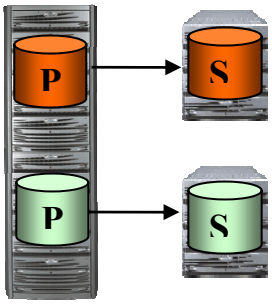

Figure 7. MirrorView/A 1:1 fan-out

MirrorView over iSCSI (new for release 26)

MirrorView over iSCSI is supported between CX3 FC/iSCSI systems (CX3-10, CX3-20, and CX3-40). The storage systems must also be running release 26 FLARE code. It is possible to have iSCSI connections between CX3 FC/iSCSI systems and CX300i and CX500i storage systems, but those connections are only valid for SAN Copy use.

The connection type (Fibre Channel or iSCSI) is determined per storage system pair. It's possible for one system to mirror over iSCSI to one storage system(s), while mirroring over Fibre Channel to others. Any combination of iSCSI and Fibre Channel MirrorView connections is possible within the 4:1 storage system fan-out ratio discussed in the previous section.

The MirrorView Wizard has been enhanced to support MirrorView over iSCSI. If both Fibre Channel and iSCSI connectivity exists, the user can simply choose which connection type to use. As long as connectivity exists over the preferred link, the wizard will perform all setup and configuration steps. The MirrorView Wizard is discussed in detail in the "MirrorView Management" section of this paper. MirrorView port changes are discussed in the "MirrorView ports and connections" section. Additional information for advanced users is available in the "iSCSI connections" section.

MirrorView ports and connections

MirrorView operates over discrete paths between SPs of the primary system and secondary system. A path must exist between the MirrorView ports of SPA of the primary and SPA of the secondary system. The same relationship must be established for SP B. MirrorView operates over one predetermined Fibre Channel port and/or one predetermined iSCSI port per storage processor (SP). The port used by MirrorView depends on the storage system type, because various models have different numbers of Fibre Channel and iSCSI front-end ports. Table 2 lists the SP front-end port numbering scheme for different models and the port that MirrorView uses.

Table 2. Front-end port numbering and MirrorView port by system type

Model	FC MV port	iSCSI MV Port
CX3-80, CX700, CX600 FC	3	--
CX3-40*, CX3-20*, CX500, CX400 FC	1	--
CX3-20, CX3-40 FC/iSCSI	5	3
CX3-10 FC/iSCSI	3	1

*Note: The MirrorView port (port 1) is the same across all CX3-20 and CX3-40 Fibre Channel-only models.

Host attach protocols are unrelated to the protocol used for MirrorView replication. Therefore, the protocol used to attach the hosts to the primary/secondary volumes can be either Fibre Channel or iSCSI, regardless of the protocol used by MirrorView.

Ports used by MirrorView can be shared with server traffic; adding traffic over this port, however, may impact the performance of MirrorView and vice versa. Therefore, when possible, EMC recommends you dedicate a port to MirrorView. SAN Copy cannot share MirrorView ports. When a system has MirrorView and SAN Copy, SAN Copy can run on any front-end port(s) other than the MirrorView Ports. SAN Copy may also impact host traffic sharing the same front-end port, so front-end port usage is important to plan carefully.

Connections should exist between SPA MirrorView ports and SPB MirrorView ports, but they should not be cross connected. For Fibre Channel SANs, place the SP A MirrorView port and SP B MirrorView port in separate zones. If there are servers zoned across SPs and storage systems, create multiple zones for each server initiator so that SP A and SP B MirrorView ports remain separated. Furthermore, iSCSI connections should be made separately between SPA MirrorView ports and SPB MirrorView ports. However, this is handled by the MirrorView Wizard.

EMC also recommends that for high-availability configurations, connectivity for SP A and SP B reside in separate physical networks, whether it is a Fibre Channel SAN or a TCP/IP network. This is illustrated Figure 8. Figure 8 depicts two CX3-40 FC/iSCSI storage systems, "Boston" and "Los Angeles". The Boston and Los Angeles systems have both Fibre Channel and iSCSI connectivity between them. These systems are used in several figures and examples throughout this paper.

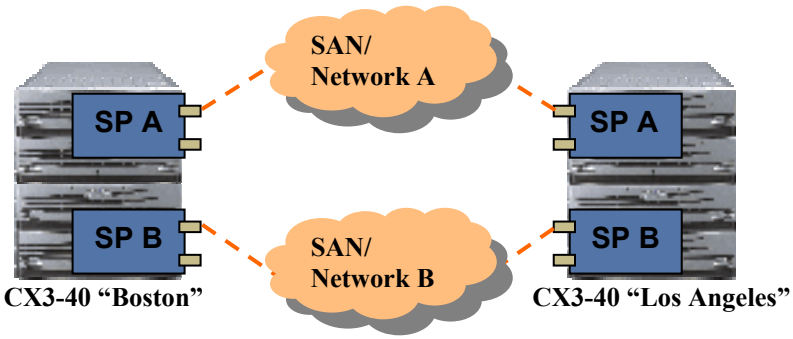

Figure 8. Example of MirrorView Port zoning

SP ownership of the mirror is determined by the ownership of the primary image. Secondary images have the same SP ownership on the secondary storage system as the primary image has on the primary storage system. Therefore, connections for both SPs must be defined to allow failover from one SP to the other. Failover is discussed in detail in the section "MirrorView failure recovery scenarios."

Individual SP port connectivity can be verified in the **Connectivity Status** dialog box in Navisphere Manager. The **Connectivity Status** dialog box shows all initiators logged in to the system's front-end ports. Click **Show Storage System Initiators** at the bottom of the dialog box to display SP to SP connections (FC and/or iSCSI). The **Connectivity Status** dialog box for the Boston storage system (using the system names shown in the above figure) is illustrated in more detail in Figure 9.

The first two entries in the dialog box show that Fibre Channel ports SPA-5 and SPB-5 from the Los Angeles storage system are logged in and registered to the FC MirrorView ports (SPA-5, SPB-5) of the Boston system. The second two entries show that there are also iSCSI connections between the iSCSI MirrorView ports (SPA-3, SPB-3). In the dialog box, the storage system's **server name** is shown as the World Wide Name.

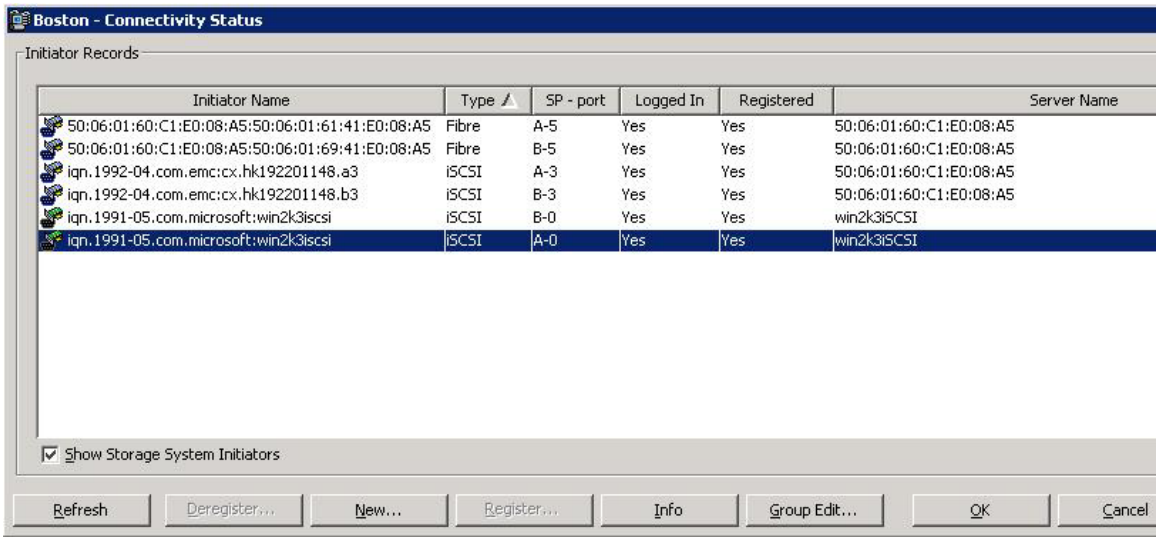

Figure 9. Connectivity Status dialog box

The **Initiator Name** column shows the 8-byte WWN and 8-byte WWPN combination for Fibre Channel initiators and the iqn number for iSCSI initiators logged in to the Boston system. You can compare these with the initiator ports of the Los Angeles system.

You can verify the WWPN and iqn numbers in Navisphere Manager by navigating through the storage system object tree to the **Physical** node and expanding the nodes under **Enclosure SPE**. Figure 10 shows the WWPNs for the front-end ports of the Los Angeles storage system.

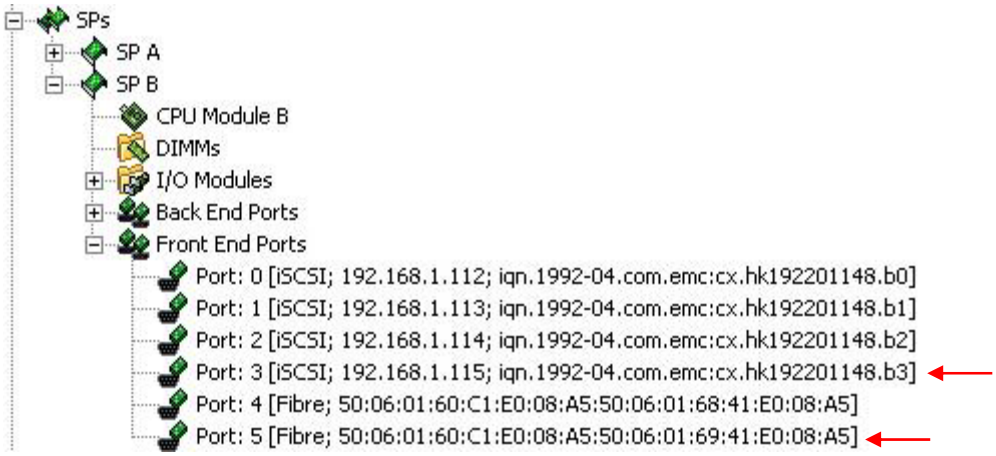

Figure 10. SP WWPNs shown in Navisphere Manager

After connectivity is established, *MirrorView connections* can be created to specify which connected arrays have mirrored relationships. MirrorView connections are automatically created when connectivity exists and you create the first mirror between storage systems using the MirrorView Wizard (which can be opened on the Navisphere Task Bar).

You can also create MirrorView connections in the Navisphere **MirrorView Connections** dialog box or in Navisphere CLI. You open the MirrorView Connections dialog box by right clicking the storage system name and selecting **MirrorView > Manage Mirror Connections.** The following Navisphere CLI commands enable and disable MirrorView connections:

```
Naviseccli mirror -enablepath -pathSPHost <-connectiontype fibre|iSCSI>

Naviseccli mirror -disablepath -pathSPHost <-connectiontype fibre|iSCSI>
```

(For simplicity, the required credentials and SP name/IP Address for Naviseccli not shown in these commands.)

Figure 11 shows the **MirrorView Connections** dialog box for the storage system named Boston, which has a MirrorView connection configured with the Los_Angeles storage system. The systems shown in Figure 11 are both zoned in a Fibre Channel SAN and have iSCSI connections defined. By looking at the **Type** column and the **Logged-In Status** columns, you can see that the MirrorView ports of SPA and SPB are logged in to one another.

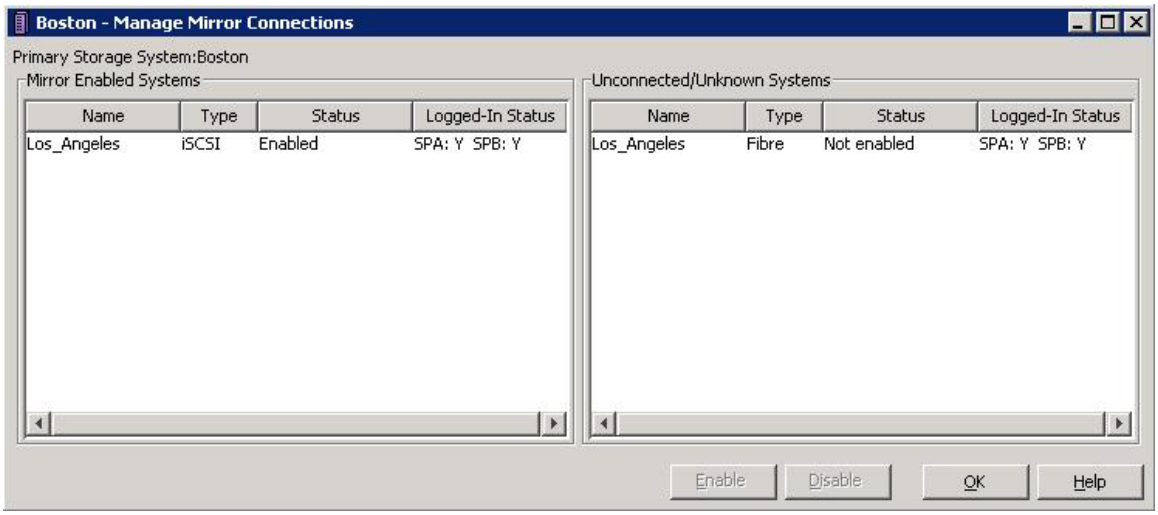

Figure 11. Manage Mirror Connections dialog box

If both connectivity types exist, users can switch between Fibre Channel and iSCSI mirror connections with relatively little impact. Mirrors will be able to incrementally synchronize on the new link type after the change. A full synchronization is *not* required.

At the time the change is made, MirrorView/A mirrors must be "Admin Fractured", while MirrorView/S mirrors will fracture automatically. An explicit Synchronize command is needed to make MirrorView/A resume replication. MirrorView/S will synchronize on the new link if the **Automatic Recovery Policy** is set. If the mirror is set to **Manual Recovery**, an explicit Synchronize command is required to resync to mirrors.

It should be noted that if the MirrorView connection is over iSCSI, iSCSI connections are created prior to the enabling of MirrorView connection. iSCSI connections establish the initiator/target relationship and login behavior between the iSCSI MirrorView ports. However, iSCSI connections are automatically created for the user when the first MirrorView connection is created by the wizard and/or the MirrorView Connections dialog box. Therefore, there are a very limited number of instances where the user will explicitly configure iSCSI connections for MirrorView. Explicit iSCSI connection management is discussed in the "iSCSI connections" section of this paper.

Common operations and settings

The following sections outline aspects of common operations for MirrorView/S and MirrorView/A. More specific information for either one of the products is included in the product-specific sections if necessary.

Synchronization

Synchronization is a copy operation MirrorView performs to copy the contents of the primary image to the secondary image. This is necessary to establish newly created mirrors or to reestablish existing mirrors after an interruption.

Initial synchronization is used for new mirrors to create a baseline copy of the primary image on the secondary image. In almost all cases, when a secondary image is added to a mirror, *initial synchronization* is a requirement. In the special case where the primary and secondary mirror LUNs are newly created and have not been assigned to a server, the user has the option of deselecting the **initial sync required** option, which is selected by default. In this case, MirrorView software considers the primary and secondary mirror to be synchronized. Use this option with caution and under restricted circumstances. If there is any doubt that the primary LUN is modified, use the default setting of **initial sync required**.

Primary images remain online during the initial synchronization process. During the synchronization, secondary images are in the **synchronizing** state. Until the initial synchronization is complete, the secondary image is not in a usable state. If the synchronization were interrupted, the secondary image would be in the **out-of-sync** state indicating the secondary data is not in a consistent state.

After the initial synchronization, MirrorView/S and MirrorView/A have different synchronization behaviors due to the nature of their replication models. MirrorView/S synchronizes only after an interruption, because under normal conditions all writes are sent to the secondary as they occur. After the initial synchronization, MirrorView/A images do not return to the synchronizing state under normal operating conditions. Secondary images are in the **updating** condition while incremental updates are in progress.

Synchronization rate

Synchronization rate is a mirror property that sets the relative value (Low, Medium, or High) for the priority of completing updates. The setting affects initial synchronizations, MirrorView/S resynchronizations, and MirrorView/A updates. Medium is the default setting, but can be changed at any time.

Low and Medium settings have low impact on storage system resources such as CPU and disk drives, but may take longer to complete a transfer. Synchronizations at the high setting will complete faster, but may affect performance of other operations such as server I/O or other storage system software.

Promoting a secondary image

A secondary image is promoted to the role of primary, when it is necessary to run production applications at the disaster recovery site. This may be in response to an actual disaster at the primary site, part of a migration strategy, or simply for testing purposes. A secondary image can be promoted if it is in the **consistent** or **synchronized** state. The definitions for consistent and synchronized image states are:

- **Synchronized --** The secondary image is identical to the primary image. This state persists only until the next write to the primary image, at which time the image state becomes **consistent.**

- **Consistent –** The secondary image is identical to either the current primary image or to some previous instance of the primary image. This means that the secondary image is available for recovery when promoted.

When a promote occurs, the secondary image is placed in a new mirror as a primary image. The new mirror has the same mirror name, but a different mirror ID. The existing mirror is either maintained or destroyed depending on the nature of the failure and whether in-band communication remains between the primary and secondary images.

Table 3. Mirror attributes before and after promote

Mirror state	Before promote	After promote
Mirror name	XYZ	XYZ
Mirror ID	aaa	bbb
Primary image	LUN xxx	LUN yyy
Secondary image	LUN yyy	LUN xxx

The promote options based on these conditions are:

Normal promote – The storage system executes a promote when the promote command is issued to a secondary image in the **synchronized** state and an active link exists between the primary and secondary images. Images in the synchronized state are exact, block-for-block copies of the primary image. When promoted, the secondary image becomes the primary image for the new mirror and the original primary becomes the secondary image. The old mirror is destroyed, so the user is left with a mirror of the same

name, whose images have opposite roles than the original mirror. I/O can then be directed to the new primary image, and no synchronization is required.

Force promote – Force promote is used when a normal promote fails. A normal promote may fail for a variety of reasons, including that the secondary image is in the **consistent** state, the mirror is fractured, and/or there is no active link between images. Active primary images often have corresponding secondary images in the consistent state. Images in the **consistent** state are in a recoverable/usable state but are not exact copies of the primary. Therefore, if the original primary is placed in the new mirror as a secondary, a full synchronization is necessary.

If there is an active link, force promote places the original primary in the new mirror as a secondary image. The original mirror is destroyed, so there is only one mirror with the original mirror name. The user would then issue a synchronize command to start the full synchronization.

If the secondary image state is synchronized or consistent when the promote is issued, but the link between them is not active, force promote does not place the original primary as a secondary of the new mirror. Instead, the original secondary is placed in a new mirror as a primary, but with no secondary image. The original primary remains in the original mirror, but there is no secondary image. The mirrors have the same name, but different mirror IDs. Both primaries remain available for host access, and when the link is back up, secondary images can be added to one or both images.

Local only promote (MirrorView/A only) - Like force promote, this option promotes the secondary without adding the original primary as a secondary image. The original mirror and the new mirror exist with primary images. However, local only promote *also* allows this type of promote with an active link. Use this option to recover changes made since the last update to the original primary.

Promoting with MirrorView/S

There is also a "promote" option in MirrorView/S. If the image state and link conditions are met for a normal promote, a normal promote is executed. If the conditions for normal promote are not met, MirrorView/S will execute a force promote.

MirrorView/Synchronous uses the **Quiesce Threshold** setting to determine when an image moves from the **consistent** state to the **synchronized** state. The **Quiesce Threshold** setting defines the amount of time, in seconds, MirrorView waits before transitioning the secondary image from the consistent state to the synchronized state when there is no write activity. It should be set to the amount of time it takes a server to flush its buffers to disk when write activity is halted. The Quiesce Threshold is a property of the mirror and can be changed at any time. Its default value is 60 seconds.

To perform a controlled failover to the secondary site and fail back *without* conducting a full synchronization, you must:

1. Quiesce I/O to the primary image.
2. Wait for the secondary image to become "Synchronized."
3. Promote the secondary image.
4. Resume I/O to the new primary image (the previous secondary).

Promoting with MirrorView/A

A secondary image will be in the **synchronized** state if I/O is stopped and a subsequent update is started and completed before I/O resumes to the primary image. This occurs in a controlled failover case, such as a data migration or a DR site test.

MirrorView/A secondaries are usually in the **consistent** state when the primary is active. A delta between primary and secondary images is expected in an asynchronous model. Therefore, users should expect to promote a secondary in the consistent state in the event of a disaster.

In a "normal" promote, the following steps outline how to fail over to the secondary site and fail back to the primary without requiring a full synchronization:

1. Quiesce I/O to the primary image (an unmount recommended).
2. Perform an update, so that the secondary becomes "synchronized."
3. Promote the secondary image.
4. Resume I/O to the new primary (previous secondary) image.

In cases where an MirrorView/A update is in progress at the time of a site failure, the secondary can still be recovered. This is possible because MirrorView/A maintains a protective snapshot (gold copy) of the secondary image. When the promote command is issued to the secondary, MirrorView/A rolls the contents of the gold copy back to the secondary image, and makes the secondary image available for access. The rollback only occurs if the image is promoted. In cases where the primary is recovered and the secondary is not promoted, the update will be allowed to continue. The gold copy process is discussed in detail in the "Data protection mechanisms" section of this paper.

Fracture

A fracture stops MirrorView replication from the primary image to the secondary mirror image. Administrative (Admin) fractures are initiated by the user when they wish to suspend replication as opposed to a system fracture, which is initiated by MirrorView software when there is a communication failure between the primary and secondary systems.

For MirrorView/S, this means that writes continue to the primary image, but are not replicated to the secondary. Replication can resume when the user issues the **synchronize** command. For MirrorView/A, the current update (if any) stops, and no further updates start until a synchronize request is issued. The last consistent copy remains in place on the secondary image if the mirror was updating.

Although not required, Admin fracture may be used to suspend replication while conducting preventive maintenance, such as software updates to connectivity elements and/or storage arrays. It is common to have components restart and/or run in degraded mode for some portion of the maintenance. During this time, users may choose to minimize the number of concurrent operations in the environment. After the maintenance is complete, a synchronization can be conducted to resume steady state operations.

Recovery Policy

The Recovery Policy determines MirrorView's behavior in recovering from link failure(s). The policy is designated when the mirror is created but can be changed at any time. There are two settings.

- **Auto Recovery –** Option to have synchronization start as soon as a system-fractured secondary image is reachable. This setting does not require human intervention to resume replication and is the default setting.
- **Manual Recovery –** Option to have MirrorView wait for a synchronization request from the user when a system-fractured secondary image is reachable. Intervention may be desired for several reasons. Users may choose to control link usage in a degraded link situation and/or to coordinate the synchronization with other operations like starting snapshots, fracturing clones, and so on.

MirrorView will automatically set the recover policy to manual after a promote is executed. The policy is set to manual to give users control over when the subsequent full synchronization will be started.

Minimum required images

Minimum required images specifies the number of secondary images that must be active for the mirror not to enter the **Attention State**. Possible values are 0, 1, or 2 for MirrorView/S and 0 or 1 for MirrorView/A. The default value is 0. Mirroring continues while in the attention state if there are any active secondaries. Users can set up alerts through Navisphere's Event Monitor feature (Event Code 0x71050129) to alert them if a mirror enters the **Attention State.**

Interoperability

MirrorView can be used with a wide variety of operating systems, software, and other replication applications. The following sections address common interoperability questions at a high level. For each case, there are documents that provide more specific and detailed implementation guidelines. Those documents should always be consulted when designing a solution.

CLARiiON replication software

MirrorView can be used with other CLARiiON replication software. For example, SnapView snapshots and clones are often used for local replication of MirrorView images. The tables below summarize CLARiiON storage-system-based replication software interoperability with MirrorView. For a comprehensive description of interoperability, see the latest version of the *EMC CLARiiON Open Systems Configuration Guide for CX3-Series and CX-Series Storage Systems* on EMC Powerlink.

Table 4. CLARiiON replication software interoperability

LUN Type	Potential usages				
	Make a snapshot of it?	**Make a clone of it?**	**Use as source for MirrorView?**	**Use as source for full SAN Copy?**	**Use as source for incremental SAN Copy?**
LUN or metaLUN not yet replicated [1]	Yes	Yes	Yes MV/A or MV/S, not both	Yes	Yes
Clone	Yes	No	No	Yes	Yes
Snapshot	No	No	No	Yes	No
MV primary	Yes	Yes	No	Yes	Yes
MV secondary	Yes	Yes	No	No	Yes

Notes:
1. Includes any LUN used as a destination for either a full or incremental SAN Copy session.

Virtual LUNs

CLARiiON virtual LUNs allow users to grow LUN capacity or migrate LUNs to different physical disks, while remaining online for applications. The metaLUN feature is used to grow LUN capacity, while the LUN migration feature is used to move LUNs to different physical spindles.

MetaLUNs have two capacity attributes: total capacity and user capacity. Total capacity is the maximum capacity of the metaLUN in its current configuration. Additional capacity can be added to a metaLUN to increase its total capacity. User capacity is the amount of the total capacity that is presented to the server. Users have control of how much capacity is presented to the server. When the total capacity is increased, user capacity also can be increased.

When expanding a LUN that is using MirrorView, SnapView, or SAN Copy, the total capacity can be increased. However, the user capacity cannot be increased until the replication software is removed from the LUN. For example, assume that a user wants to stripe-expand a LUN. In the case of a striped expansion, data is restriped over added disks before the additional user capacity is available. The user can minimize the time the LUN is not being mirrored by allowing the striping operation to complete *before* removing the mirror from the LUN to increase user capacity.

Increasing user capacity is a nearly instantaneous process. Once user capacity is increased, the mirror can be re-created on the LUN. To use the same secondary image, user capacity has to be increased to the same user capacity as the primary image.

If I/O continues while the mirror is removed, a full synchronization is required when the mirror is re-created. A full synchronization can be avoided if I/O can be quiesced while the mirror is removed and user capacity is increased. We recommend that you take the following steps to avoid a full synchronization.

1. Quiesce I/O and remove either the host or LUNs from the storage group. You must make sure that there is no I/O or discovery of the new capacity while there is no mirror in place.

2. Ensure the secondary image is in the **synchronized** state as reported by the primary storage system. For MV/S, wait for the image to become synchronized once I/O is quiesced. For MirrorView/A, perform an update after I/O is quiesced.

3. Remove the secondary image and destroy the mirror.

4. Increase user capacity of the primary and secondary images to the exact same value.

5. Re-create the mirror and clear the **Initial Sync Required** option when adding the secondary image.

6. Add hosts or LUNs back into the storage group for discovery of new capacity and resume I/O.

For more information on LUN expansion with metaLUNs, see the white paper *EMC CLARiiON MetaLUNs: Concepts, Operations, and Management* on EMC.com and Powerlink.

LUN migration can be used with MirrorView images under certain conditions. Table 5 outlines the conditions under which MirrorView LUNs can and cannot be migrated. For more information on LUN migration, see *Navisphere Manager Help* and the white paper *EMC Virtual LUN Technology – A Detailed Review* on EMC.com and Powerlink.

Table 5. LUN migration rules for mirror images

LUN type	Migrate to same size LUN	Migrate to larger LUN
Primary image	Can be migrated[2]	Remove secondary image(s) and destroy mirror before migration
Secondary image	Can be migrated[2]	Remove secondary image from mirror
Write intent log LUN (MV/S)	Cannot be migrated	
Reserved LUN (MV/A)	Cannot be migrated	

Notes:
1. A mirror cannot be created on a LUN that is migrating.
2. In release 26, a migration can be started on a MV/S primary or secondary if the secondary is synchronizing. In prior versions, migration was only possible if the secondary image was in the **consistent** or **synchronized** state.

FLARE operating environment

EMC recommends that storage systems replicating LUNs between one another run the latest release of FLARE. This is designated as "Preferred" in Table 6. FLARE 26 is the latest release for all CX3 models, the CX700, and CX500. Release 19 is the latest release for the CX600 and CX400.

Depending on the model types used, it is not always possible to run the latest version of FLARE. For example, release 19 is the latest release for the CX600 and CX400. It is also possible for a new storage system to be added to an environment with older systems that cannot be upgraded immediately. Therefore, EMC supports running MirrorView between storage systems whose FLARE releases are one version apart. This relationship is designated as "Supported" in Table 6.

Releases can differ by two releases for the purposes of supporting upgrades. For example, assume a CX3-40 running release 24 is mirroring a CX700 running release 19. The CX3-40 can be upgraded to release 26 while keeping the mirror relationship. However, changing the MirrorView configuration is not supported in this case. The CX700 should be upgraded before performing any management tasks. Wherever possible, upgrade the older system first. This relationship is designated as "Upgrade" in Table 6.

Table 6. Software version interoperability

		Secondary		
		Release 26 (CX3 series, CX700, CX500)	**Release 24 (CX3 series, CX700, CX500)**	**Release 22* (CX3 series), Release 19 (CX400 – CX700)**
Primary	**Release 26 (CX3 series, CX700, CX500)**	Preferred	Supported	Upgrade
	Release 24 (CX3 series, CX700, CX500)	Supported	Preferred	Supported
	Release 22* (CX3 series) **Release 19 (CX400 – CX700)**	Upgrade	Supported	Preferred

*Note: Release 22 runs only on the CX3 series. For version relationships, it is considered equal to release 19 on the CX series (CX400-CX700).

VMware ESX Server

Both VMFS volumes and RDM volumes can be replicated with MirrorView. Some of the major implementation guidelines for MirrorView and other CLARiiON replication applications are listed below.

- RDM volumes must be in **Physical compatibility** mode when used with storage-system-based replication.
- If replicating an entire VMFS-3 or VMFS-2 volume that contains a number of virtual disks on a single CLARiiON LUN, the granularity of replication is the entire LUN with all of its virtual disks.
- Use the consistency technology available on SnapView/MirrorView when making copies of VMFS volumes that span multiple CLARiiON LUNs.
- For VMFS-3 and RDM volumes, the replica can be presented to the same ESX Server but *not* to the same virtual machine.

Please consult the E-Lab™ Navigator and/or the white paper *CLARiiON Integration with VMware ESX Server* for the comprehensive set of guidelines before using MirrorView with ESX Server.

Celerra

CLARiiON LUNs allocated to the Celerra® line of network-attached storage products can be replicated with MirrorView/S. The following are high-level guidelines for using MirrorView/S with Celerra. (More detailed information is included in the technical module *Using MirrorView/Synchronous with Celerra for Disaster Recovery* found on Powerlink, EMC's customer- and partner-only extranet.)

- All LUNs dedicated to Celerra must be managed within a consistency group. Consistency groups are discussed in the "Storage-system-based consistency groups" section.

- As part of the baseline configuration, control LUNs 0, 1, and 4 must be included in the consistency group. The remainder of the LUNs in the consistency group can be allocated as user LUNs.
- The write intent log must be allocated. The write intent log is discussed in the "Write intent log" section.
- Existing configurations can implement MirrorView/S provided they can meet the first guideline.

MirrorView/Synchronous

MirrorView/Synchronous (MirrorView/S) follows the replication model outlined in the "Replication models" section for synchronous replication. MirrorView software writes new data to both the primary and secondary volumes before the write is acknowledged to the server. The following sections describe the software MirrorView/S software.

Data protection mechanisms

MirrorView/S has mechanisms to protect data loss on the primary and/or secondary image. The fracture log protects primarily against loss of communication with the secondary image. The write intent log protects primarily against interruptions to the primary image. Use of the write intent log is optional. Both of these structures exist to enable partial synchronizations in the event of interruptions to the primary and secondary images.

Fracture log

The fracture log is a bitmap held in the memory of the storage processor that owns the primary image. It indicates which physical areas of the primary have been updated since communication was interrupted with the secondary.

The fracture log is automatically invoked when the secondary image of a mirror is lost for any reason and becomes fractured. The mirror is fractured (system fractured) by MirrorView software if the secondary is not available or it can be administratively fractured through Navisphere Manager or CLI. MirrorView/S system fractures an image if an outstanding I/O to the secondary is not acknowledged within 10 seconds. While fractured, the primary pings the secondary every 10 seconds to determine if communication has been restored.

The fracture log tracks changes on the primary image for as long as the secondary image is unreachable. It is a bitmap that represents areas of the primary image with regions called *extents*. The amount of data represented by an extent depends on the size of the mirror images. Since the fracture log is a bitmap and tracks changed areas of the primary image, it is not possible to run out of fracture log capaity. It may be necessary, depending on the length of the outage and the amount of write activity, to resynchronize the entire volume.

When the secondary LUN returns to service, the secondary image must be synchronized with the primary. This is accomplished by reading those areas of the primary addressed by the fracture log and writing them to the secondary image. This activity occurs in parallel with any writes coming into the primary and mirrored to the secondary. Bits in the fracture log are cleared once the area of the primary marked by an extent is copied to the secondary. This ability to perform a partial synchronization can result in significant time savings.

By default, the fracture log is stored in memory. Therefore, it would be possible for a full resynchronization to be required if a secondary image is fractured *and* an interruption in service occurs on the primary SP. The write intent log protects against such scenarios.

Write intent log

The write intent log is a record stored in persistent memory (disk) on the storage system on which the primary LUN resides. The write intent log consists of two 128-MB LUNs, one assigned to each SP in the

storage system. Each 128-MB LUN services all the mirrors owned by that SP that have the **write intent log** option selected. When the write intent log LUNs are assigned to the SP, they become private LUNs and are under the control of MirrorView software.

During normal operation, the write intent log tracks in-flight writes to both the primary and secondary images in a mirror relationship. Much like the fracture log, the write intent log is a bitmap composed of extents indicating where data is written. However, the write intent log is always active and the fracture log is only enabled when the mirror is fractured.

When in use, MirrorView makes an entry in the write intent log of its intent to update the primary and secondary images at a particular location, and then proceeds with the attempted update. After both images respond that data has been written (governed by normal LUN access mechanisms, for example, written to write cache), it clears previous write intent log entries. For performance reasons, the write intent log is not cleared immediately following the acknowledgement from the primary and secondary images; it will be cleared while subsequent write intent log operations are performed.

In a recovery situation, the write intent log can be used to determine which extents must be synchronized from the primary storage system to the secondary system. For instance, if a single SP becomes unavailable (for example during a reboot or failure), there may be in-flight writes that were sent to the secondary, but not acknowledged before the outage. These writes will remain marked in the write intent log.

Then server software, such as PowerPath®, trespasses the LUN to the peer SP. The remaining SP directly accesses the unavailable SPs write intent log and recovers the recent modification history. The SP then resends the data marked by the extents in the write intent log. This allows for only a partial resynchronization, rather than a full resynchronization because it ensures that any writes in process at the time of the failure are acknowledged by the secondary image. If the entire array becomes unavailable (for example, due to a power failure), then the write intent log is used similarly to facilitate a partial resynchronization from primary to secondary, once the primary array is recovered.

Use the write intent log whenever possible to provide the maximum level of protection for your data. The write intent log is optional, because it involves extra writes to the write intent log LUN. However, release 26 has enhancements that improve performance when using the write intent log. In prior versions, performance would degrade if the storage system write cache neared full. In release 26, there is no impact at lower workloads and only a slight impact (≤10%) on response time at high workloads. Therefore, use of the write intent log should be maximized.

Fracture log persistence

The fracture log resides in SP memory. By itself, it cannot withstand outages, such as an unplanned SP reboot (for example physically removing an SP during normal operation). However, mirrors with the write intent log enabled are able to reconstruct the fracture log in SP memory if there is an outage. Since the data structures in the write intent log achieve a persistent record of the areas on the primary image that have been written to, when the write intent log is enabled, it also provides fracture log persistence— *but only for those mirrors with the write intent log enabled.* In this case, users get the benefit of partial resynchronization after an unplanned outage.

For example, if a mirror is fractured when a failure occurs on the primary system, a partial resynchronization is still possible if the write intent log is enabled. With fracture log persistence, changes to regions on the primary after the mirror is fractured are captured on persistent media, and can be read by the storage processors when they are recovered. Without the benefit of fracture log persistence, changes to the primary since the mirror was fractured are lost and a full resynchronization will be necessary. If the SP is rebooted in a controlled manner through Navisphere, the fracture log is preserved by sending it to the other SP before the reboot.

Fracture log persistence, along with improved performance in release 26, is why the write intent log should be enabled on as many mirrors as possible. If more mirrors exist than can be write intent log enabled, prioritize mirrors based on preference for resynchronization in the instance of an unplanned outage.

Initial synchronization

Link speed is typically adequate to perform initial synchronizations with MirrorView/S because the links are commonly high-bandwidth low-latency links. Twenty images on each SP are able to perform synchronizations at the same time. More can be started, but the 21st image started will be queued until one of the first of the 20 finishes, and so on.

Initial synchronization is tracked using the fracture log. The extents of the fracture log are cleared as the areas they represent are copied to the secondary. You can view progress in Navisphere MirrorView's (MirrorView/S or MirrorView/A) **Mirror Properties** dialog box by clicking the **Secondary Image** tab.

Link requirements

MirrorView/S is qualified for a large variety of SAN topologies that include high-speed Fibre Channel SANs, distance extension solutions including longwave optics, FC/IP converters, and native iSCSI. For comprehensive guidance on supported topologies, see the *EMC Networked Storage Topology Guide* on EMC Powerlink.

It is expected that bandwidth and latency requirements will be governed by the application, since each I/O traverses the link. Under these conditions, links are typically high bandwidth lines with low latency. There are no bandwidth requirements enforced by MirrorView software. However, there are functional requirements for latency, since MirrorView/S will fracture a secondary image if it has an I/O outstanding to the secondary for 10 seconds.

The graph in Figure 12 illustrates the effect of latency on transaction response time plotted against the write component of the transactions. An OLTP simulator was run, where each OLTP transaction consisted of 21 reads and nine writes. The load was generated using 8-KB I/O over 11 data LUNs and 1 log LUN on a CX700. The link is 100 Mb with compression and fast write enabled through an FC/IP device.

Individual results vary for transaction response times in an end-user environment depending on a wide variety of factors, including application concurrency, server hardware and software, connectivity components, storage system model, number of disks, and so on. The graph does not represent any MirrorView or CLARiiON limits, but simply illustrates the effects of latency with all other things equal across the configuration.

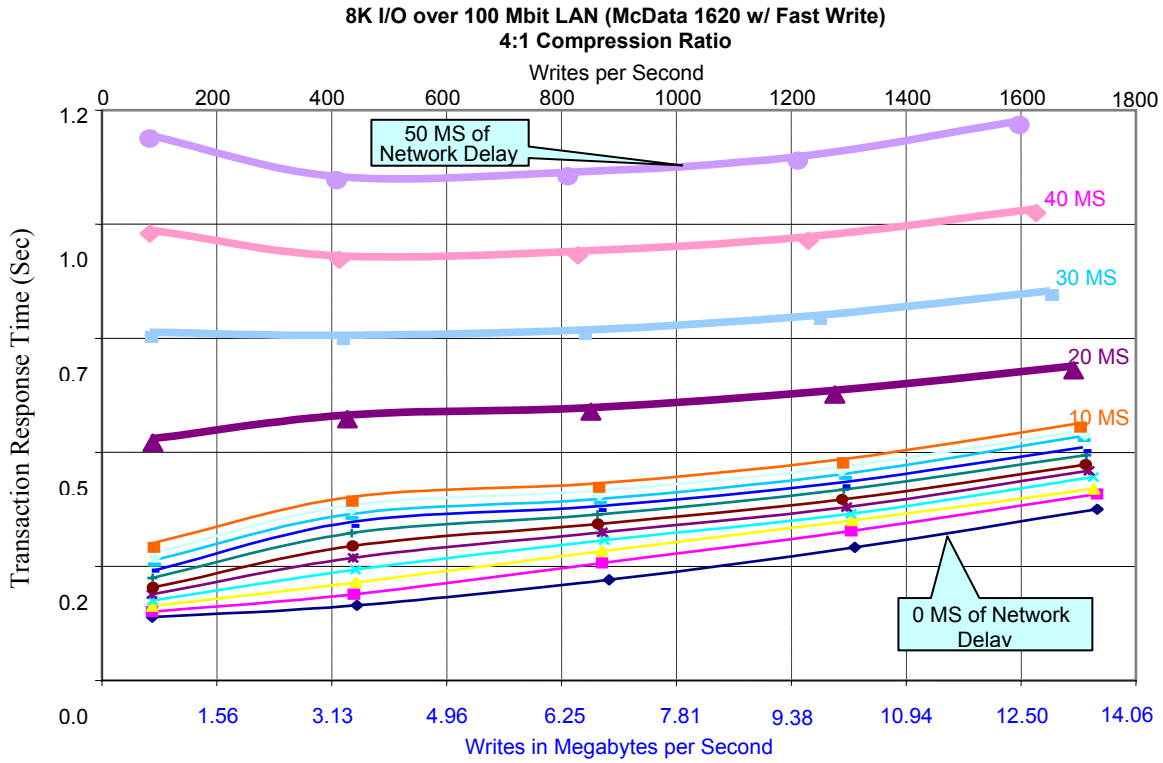

Figure 12. Effects of link latency on transaction response times

When going over short distances with low latency lines, performance differences between using native iSCSI connectivity and using an FC/IP devices should not be very great. As distance and latency are increased, FC/IP devices that offer advanced features such as compression and "fast write" will have performance advantages over native iSCSI connections.

Limits

Table 7 lists the MirrorView/S limits for the CX3 series, CX700, and CX500 running release 26 and the CX600 and CX400 running release 19. For the latest and most comprehensive limits, consult the *EMC CLARiiON Open Systems Configuration Guide for CX3-Series and CX-Series Storage Systems*.

Table 7. MirrorView/S storage system limits

Parameter	Storage system			
	CX3-80, CX3-40, CX700	**CX600**	**CX3-20, CX500**	**CX3-10, CX400**
Maximum allowed mirror images per storage system	200 max	100 max	100 max	50 max
Maximum allowed mirrors per storage system with write intent logs	100 max	50 max	50 max	25 max
Maximum synchronous consistency groups	16 max	16 max	8 max	8 max
Maximum mirrors per synchronous consistency groups	16 max	16 max	8 max	8 max

Notes:
1. Images include primary and secondary images.
2. In release 24 and later, MirrorView/S and SnapView clone limits are independent of each other. In previous FLARE releases, MV/S images and SnapView clones were counted together against a storage system image count limit. Consult the *EMC CLARiiON Open Systems Configuration Guide for CX3-Series and CX-Series Storage Systems* for details.

MirrorView/Asynchronous

MirrorView/A uses a periodic update model for transferring write data to the secondary image. Between updates, areas written to are tracked by MirrorView/A. During an update, the data represented by these areas is transferred to the secondary image. The set of data that is transferred is referred to as a *delta set*.

A delta set may be smaller than the aggregate number of writes that occurred to the primary image, since areas in a bitmap are marked to denote a change occurred to a particular location of the image. If more than one write occurs to the same area, only the current data is sent when the update is started. SnapView technology is used to preserve a point-in-time copy of the delta set during the update.

MirrorView/A is designed to enable distance solutions over lower bandwidth and higher latency lines than would be possible with a synchronous solution. Solutions are often implemented over T3 type lines or the equivalent bandwidth within a larger link. Replication distances can range from 10s to 1000s of miles.

RPO is determined by the frequency and duration of updates. There are three methods for defining the frequency of updates.
- Manual – Each update is initiated by the user
- Start of Last Update – MirrorView starts updates at user defined intervals
- End of Last Update – MirrorView starts updates after a user defined rest period from the completion of the last update

User defined intervals are specified in minutes, with an input range of 0 – 40,320 (28 days). However, typical update intervals range from 15 minutes to hours.

Data protection mechanisms

MirrorView/A uses SnapView technology for data protection on the primary and secondary storage systems. On the primary storage system, SnapView tracks changes on the primary image and creates a point-in-time image that is the source for the delta set transfer.

On the secondary storage system, SnapView creates a protective copy of the secondary image during the update. The protective copy, referred to as the *gold copy*, ensures that there is a usable copy on the secondary storage systems at all times.

All of MirrorView's use of SnapView is autonomous and does not need be managed by the user. Therefore, a SnapView license is *not* required to use MirrorView/A.

Tracking and transfer maps

MirrorView/A uses two bitmaps on the primary image. For each update, one bitmap (the tracking map) tracks changes between updates, and the other bitmap (the transfer map) tracks the progress of the update when transferring to the secondary. Each bitmap is used in a revolving fashion, first as the tracking map between updates and then as the transfer map for the subsequent update.

The tracking and transfer maps are persistently stored on a reserved LUN on a per-mirror basis. As soon as a MirrorView/A mirror is created on a primary image, a reserved LUN is assigned to the image to store the tracking and transfer maps. The reserved LUN is assigned to the source for as long as the mirror exists.

The bits within the tracking and transfer maps represent chunks of the primary image. Chunks are a fixed size, so the number of chunks varies depending on the size of the LUN. Changes to the primary image are tracked at a 2 KB granularity.

This granularity allows smaller inter-site links (T1/T3) to be used efficiently by transferring a data set closer in size to the amount of data changed. MirrorView/A can also coalesce writes for adjacent LUN locations into one write, up to 64 KB, when transferring it to the secondary. For example, two separate 4 KB writes that occur on contiguous block locations on the LUN can be transferred as one 8 KB write from the primary to the secondary.

Between periodic updates, only the bitmap acting as the tracking map is active. When a server's write I/O changes a region for the first time since the last update, a bit is set in the tracking map indicating the change. Any changed regions are transferred to the secondary site during the next periodic update.

The transfer bitmap keeps track of the progress of the transfer of data from the primary to the secondary image. At the start of an update, the tracking map and transfer map switch roles, so that the tracking map becomes the transfer map for the update. It already contains the set bits indicating where changes occurred since the last update, so the same information is used to monitor the delivery of the data to the secondary image. At the same time, the existing transfer map becomes the new tracking map for the next update. Each time an update is started, this switching of roles occurs.

As the areas of the LUN represented by the set bits in the transfer map are sent, the bits are cleared from the transfer map. This way, at the end of a successful update, the transfer map is cleared of any set bits.

During the transfer, if a server write is initiated to the primary image for a location not yet transferred, the data area will be copied into the reserved LUN before accepting the write. This preserves the point-in-time image of the changed data from the time the update was initiated. This only occurs the first time a write is initiated into the primary for a location that has not been transferred. This is referred to as copy-on-first-write protection. If a write comes into the primary for an area already transferred, no-copy-on-first-write occurs. The area is marked only in the tracking map and is transferred in the next update cycle.

Delta set

A delta set is a virtual object that represents the regions on the primary mirror LUN that need to be transferred to the secondary mirror LUN. A SnapView snapshot, managed by MirrorView/A software, is used as the source of the transfer to maintain a point-in-time image of the delta set for the duration of the transfer.

Ultimately, the size of the delta set will depend on the number of writes between updates and the locality of those writes on the LUN. Some applications that are random in nature may have to replicate most of the writes that occurred between updates. Other applications, which have high locality, may write to the same location several times between updates and result in small delta sets. Results can vary greatly by application and implementation.

For example, Figure 13 shows expected delta set behavior for a consistent workload. The delta set size peaks before the first update. While the update progresses, I/O continues to the primary volume, so some data lag exists throughout the update process. Delta set peaks remain consistent in this test because of the steady workload.

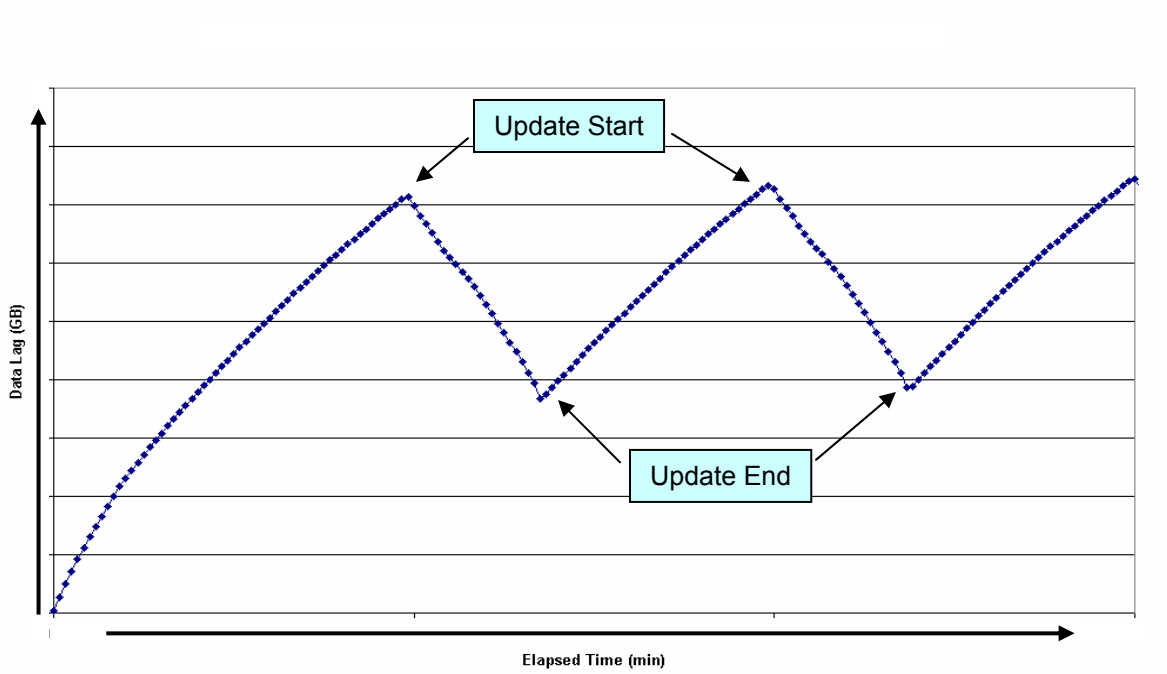

Figure 13. Data lag trend for Oracle 9*i* RAC test over T3 (50 ms latency)

The data for this test was generated using an Oracle Database. Replication was conducted over a T3 line with a 50 ms delay to simulate long-distance replication. The bottleneck for update bandwidth in this test was the T3 link.

Gold copy (protective snapshot)

A gold copy is a virtual point-in-time copy (snapshot) of the secondary image that is initiated before the update begins. It ensures that there is a known good copy on the secondary during an update. The gold copy is managed entirely by MirrorView/A software and is not directly available for server access. Users can take additional snapshots of the secondary image for a usable point-in-time copy of the secondary.

If the update from primary to secondary is interrupted due to a catastrophic failure at the primary site, MirrorView/A uses the gold copy to roll back the partial update on the secondary LUN if the secondary is promoted for use. If the secondary is not promoted, the gold copy is maintained so that the update may complete when the interruption is rectified.

Reserved LUNs

Reserved LUNs are a resource used by SnapView snapshots to store copy-on-first-write data. In cases where SnapView technology is used for point-in-time management, such as in incremental SAN Copy and MirrorView/A, the reserved LUNs also store the tracking and transfer bitmaps necessary to manage the updates.

Reserved LUN resources must be allocated on the primary and secondary storage systems when using MirrorView/A. Reserved LUNs on the primary side are used for storing the bitmaps associated with delta sets and for storing any copy-on-first-write data that is created during the update. On the secondary side, they are used for storing copy-on-first-write data to maintain the gold copy.

When sizing the reserved LUNs for the primary image, consider how much data will change during an update. MirrorView/A maintains optimized copy on first write protection during an update, where only those blocks that haven't been transferred to the secondary are copy on first write protected. Additional consideration needs to be given if there are user initiated SnapView snapshots on the primary image. For standard snapshots, the entire volume is copy-on-first-write protected for as long as the snapshot exists. So

it is expected that capacity requirements for user initiated snapshots are greater than those needed by MirrorView/A.

When sizing the reserved LUNs for the secondary image, the rate of change must be known. Since the primary image is sending only areas that change, every update write from the primary storage system is copied on the first write protect to the secondary. It is best to assume no locality and provide enough reserved LUN capacity to, at the minimum, accommodate the largest number of aggregate writes for any one period. The aggregate number of writes can easily be measured by OS based tools such as I/O Stat and Perfmon or array based tools such as Navisphere Analyzer. If MirrorView/A is being considered for an application that is not yet running, a rule of thumb of 20 percent of the Primary image capacity is commonly used.

Additional space beyond the maximum aggregate writes should also be considered in case there is an interruption to the update process. If I/O continues to the primary image during the interruption, the subsequent delta set after the interruption will be larger than over a normal period. Some additional considerations for MirrorView/A reserved LUN configuration are:

- Place the reserved LUNs on different RAID groups than the primary images. Since it is possible to have tracking and/or copy on first write activity associated with a server write, its best not to have all of the activity queued up for the same set of disks. Often times, several reserved LUNs are configured on their own RAID group.
- Use high performance FC drives for the reserved LUNs. It's common to have several reserved LUNs on the same drives. The tracking and copy on first write activity is a critical component for write response time, so it is important to avoid I/O contention on these drives.
- Use the same configuration on the primary and secondary side. Some end users consider cutting costs on the DR side by using fewer drives or lower-cost drive technology. However, the reserved LUNs plays an important role in copy-on-first-write protecting the secondary images during an update. It's also expected that in a recovery situation, the applications will run with similar performance when failed over to the DR site.

For a more comprehensive look at reserved LUN guidelines for MirrorView and other replication applications, see the white paper *CLARiiON Reserved LUN Pool Configuration Considerations* on EMC.com and Powerlink.

Initial synchronization

The transfer map is used to track initial synchronization. When an initial sync is required, all of the bits in the transfer map are set to indicate transfer. The bits of the transfer map are cleared as the areas they represent are copied to the secondary.

All mirrors can synchronize simultaneously. Resources are shared among active mirrors to evenly distribute bandwidth during the synchronization period.

There is no gold copy (so no protective snapshot) during the initial sync, since there is no usable copy of data until the initial sync is complete. If the initial sync is interrupted, the secondary will be in an *out-of-sync* state and cannot be promoted.

MirrorView/A is intended to operate over lower bandwidth lines. In implementations where the delta set would consume most or all of the link capacity, it can be difficult to perform the initial synchronization. This is prevalent in implementation with large volumes (multiple TBs) and relatively smaller change rates (GBs) over the update period.

Often in these scenarios, the secondary system is brought to the same site as the primary for the initial synchronization and then shipped to the remote site, where an incremental update is performed. The general process to do this is:

1. With primary and secondary on the local SAN, create mirrors, add secondary images, and perform initial synchronization.

2. Admin fracture the mirrors.

    a. An incremental update is optional before the fracture to get the very last possible changes.

3. Transport secondary and install at final site including any zoning and IP address changes. Changing IP addresses and resulting SP reboot do not affect the MirrorView configuration.

4. Bring the secondary either back into the Navisphere domain or manage it with the Navisphere Multi-Domain Management feature if in a different domain.

5. Reestablish the MirrorView Connections in Navisphere and synchronize the Mirrors.

In FLARE release 24 and later, MirrorView/A uses the **Bulk Copy** feature of SAN Copy for the initial synchronization. The bulk copy process is an advanced feature of SAN Copy that minimizes SnapView operations during the initial synchronization. It also provides check pointing, so that initial synchronizations can resume if a link outage occurs.

The Bulk Copy process is completely managed by MirrorView/A software, so there are no changes to user operations. MirrorView/A performs a bulk copy and an immediate incremental update as part of the initial synchronization process. Users may see improved performance during the initial synchronization when compared with prior versions.

Link requirements

MirrorView/A does not have stringent bandwidth and latency requirements. A minimum bandwidth of 128 KB/s is required. This is very likely to be much lower than any practical implementation would require.

MirrorView/A will system fracture during an update if it does not receive acknowledgement from the secondary in five to six minutes. Since server I/O is decoupled from the transfer of data from the primary to the secondary, it is not as important to system fracture within the same time as a synchronous solution.

MirrorView/A is commonly replicated over TCP/IP links to enable long distance replication. Table 8 shows nominal bit rates and bandwidths for commonly used link types. When determining the required link type also factor in TCP/IP overhead, which can lower the effective payload by about 20 percent.

Table 8. Nominal link speeds

Link type	Nominal bit rate (Mb/s)	MB/s	GB/h
T1	1.5	0.18	0.63
T3	45	5.4	18.9
100 Mb	100	11.9	41.9
OC3	155	18.5	65.0

As distance increases, advanced features such as compression and support of large TCP window sizes can often help to optimize link utilization. These features are offered on FC/IP conversion devices, and they are not offered on the native iSCSI ports. So for very long distance replication, like coast to coast of the continental U.S. (~3000 miles), users may opt to employ FC/IP devices, depending on their workload as compared to the link's bandwidth and latency.

Limits

Table 9 lists the MirrorView/A limits for the CX3 series, CX700, and CX500 running release 26, and the CX600 and CX400 running release 19. For the most recent limits guidelines, consult the *EMC CLARiiON Open Systems Configuration Guide for CX3-Series and CX-Series Storage Systems.*

Table 9. MirrorView/A system limits

	Storage system		
Parameter	**CX3-80, CX3-40, CX700, CX600**	**CX3-20, CX500, CX400**	**CX3-10**
Maximum images per storage system	100 max	50 max	25 max
Maximum async consistency groups	16 max	8 max	8 max
Mirrors per async con groups	16 max	8 max	8 max

Note: Images include incremental SAN Copy source LUNs

Storage-system-based consistency groups

MirrorView (both /S and /A) includes the storage-system-based consistency groups feature. A storage-system-based consistency group is a collection of mirrors that function together as a unit within a storage system. All operations, such as synchronization, promote, and fracture, occur on all the members of the consistency group. Once a mirror is part of a consistency group, most operations on individual members are prohibited. This is to ensure that operations are atomically performed across all the member mirrors.

The members of a consistency group can span across the SPs on the CLARiiON storage system, but all of the member mirrors must be on the same storage system. A consistency group cannot have mirrors that span across the storage systems. In addition, although consistency groups are supported for both synchronous and asynchronous mirrors, all of the mirrors in each group must be same type of mirror.

Benefits of storage-system-based consistency groups

Managing all write-order dependent volumes as one entity ensures that there is a *restartable* copy of the application on the secondary system. The copies are restartable because all the volumes are as current as the solution's RPO will allow. Other replication strategies may yield a *recoverable* copy of data at the DR site. Recoverable copies typically require more operations to bring them back online. Consider the example of a DBMS application. When using log shipping in the event of a disaster, data files are restored to a previous point in time (for example, a backup) and then made current by applying logs replicated more frequently.

Restartable copies offer shorter RTOs than recoverable copies in terms of access to current data. A trade-off is that more data may be replicated to constitute restartable copies as both logs and data files are being replicated on a regular basis, as opposed to only replicating the logs.

Consistency groups protect against data corruption in the event of partial failures, for example on one SP, LUN, or disk. With partial failures, it is possible for the data set at the secondary site to become out of order or corrupt.

For example, assume an interruption prevents only one SP from communicating with its secondary volume. If the log volumes reside on the interrupted SP and the database volumes reside on the other SP, it is possible for updates to be written to the database secondaries, but not the log secondaries.

The following sections discuss specific benefits of consistency groups as they pertain to MirrorView/S and MirrorView/A.

MirrorView/S

For MirrorView/S, consistency groups are crucial to performing consistent fracture operations. Figure 14 shows a database where SP A owns the log components and SP B owns the data files. Without consistency groups, an interruption in communication on SP A invokes the fracture log on the log LUNs, and then I/O continues to the log LUNs and the data LUNs. Since there is no outage on SP B, I/O to the data LUNs would still be mirrored to the secondary. If an outage occurs on the primary system during this time, there would not be a usable copy because data would be ahead of the logs at the secondary site.

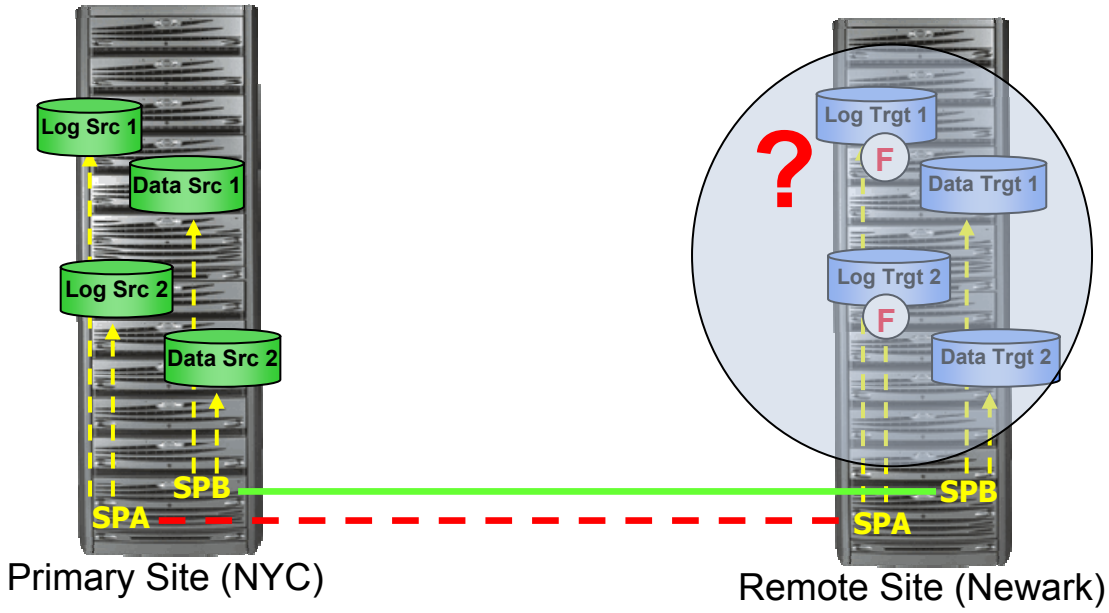

Figure 14. Partial fracture with no consistency groups

Figure 15 shows the same scenario with consistency groups. If one member of the group fractures, then all of the members of the group fracture, and data integrity is preserved across the set of secondary images.

At the time of the interruption, MirrorView fractures all of the mirrors in the consistency group due to the communication failure of those mirrors on SP A. This allows I/O to continue to the primary volumes, but not to the secondary volumes. While MirrorView performs the fracture operation, it briefly holds write I/Os to members of the consistency group until that particular member is fractured. After each member is fractured, I/O is allowed to continue to that volume.

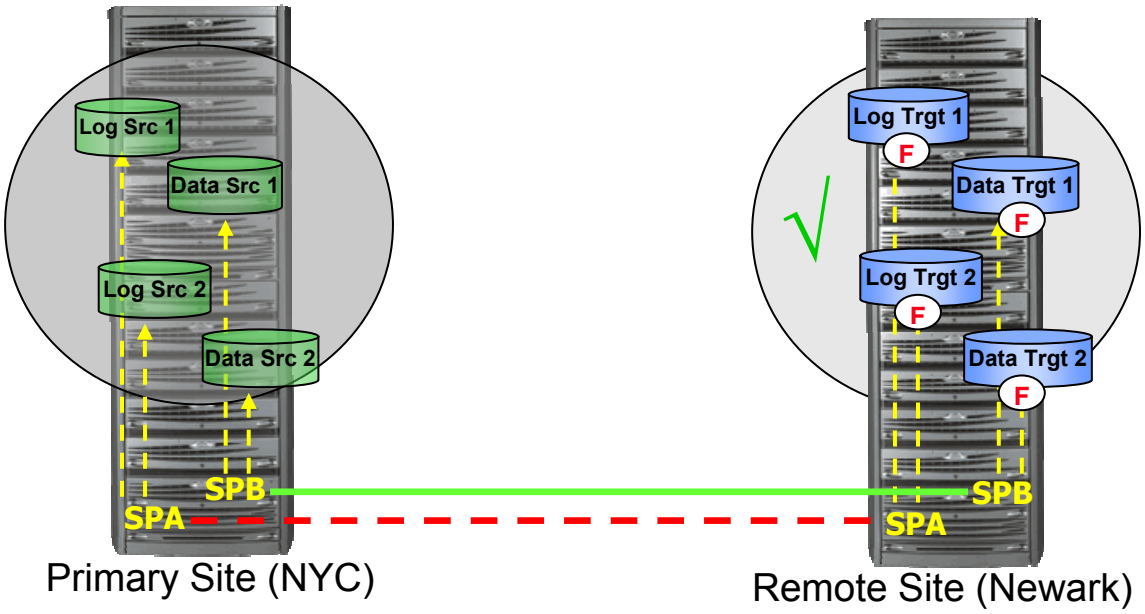

Figure 15. Consistency group fracture with partial interruption

For a database application, as the fracture log is invoked on the log LUN, the storage system acknowledges the log write to the server. The server then issues the subsequent write to the data volume. If the data volume write request is issued to the storage system before it is fractured, the storage system holds the I/O until the fracture of the data LUN can occur. Once fractured, the storage system acknowledges the write to the server.

MirrorView/A

In addition to managing fractures, consistency groups for MirrorView/A play a key role in the update process. At the start of an update, a point-in-time copy is created that serves as the source for the point-in-time transfer. When there is a set of write-order-dependent volumes, the point-in-time copy must be started simultaneously across all volumes. As shown in Figure 16, at the start of an update, I/O is held by FLARE for each mirror until the point-in-time session has been established for that volume. Once the point-in-time session is created, I/O resumes.

Similarly, before updates are applied to the secondary images, a gold copy is created for all secondary mirrors. These gold copies remain in place until members of the consistency group complete the transfer of the delta set. This ensures that changes are applied to all the secondary images or none of them, and that there is always a consistent point-in-time image available to promote. If the secondary images are promoted with gold copies in place, MirrorView rolls the contents of the gold copies back to the secondary images and makes them available for direct server access as primary images. The rollback process has instant restore capabilities, which means that users have access to their data while the rollback process runs in the background.

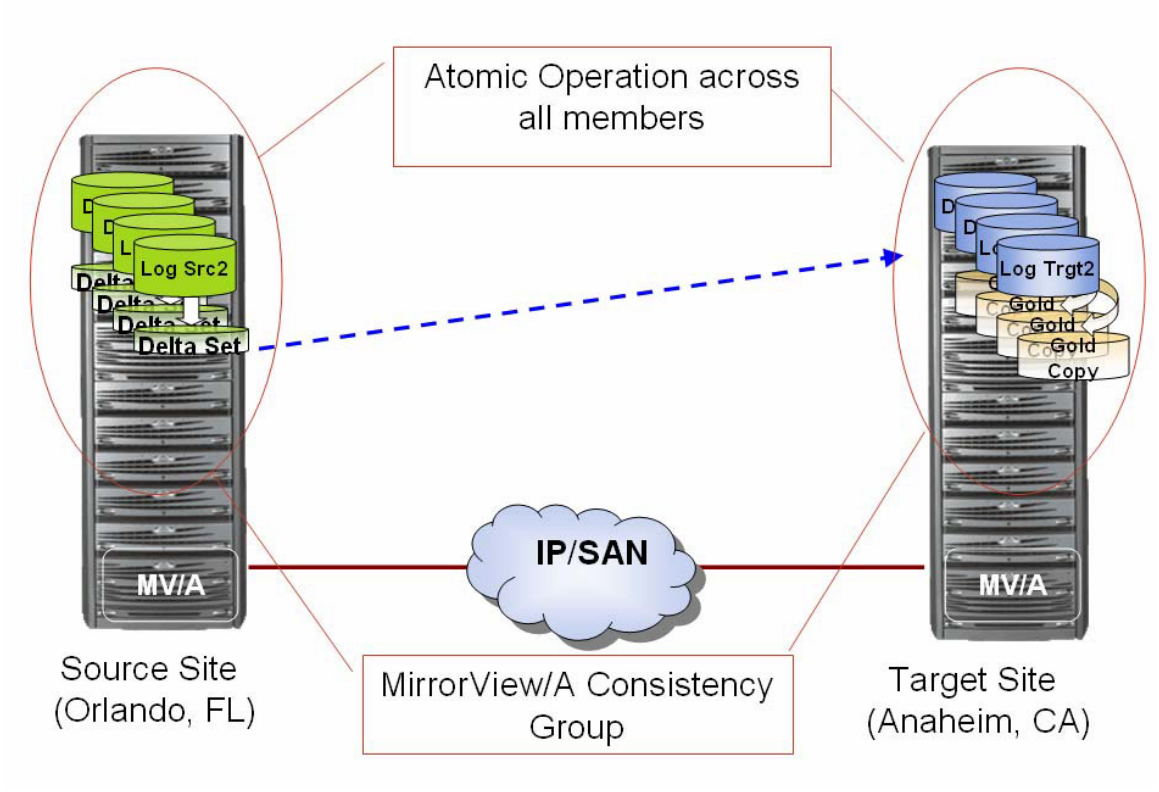

Figure 16. MirrorView/A storage-system-based consistency group atomic operations

Managing storage-system-based consistency groups

You can manage consistency groups using Navisphere Manager or Navisphere CLI. Figure 17 illustrates how it is possible to perform operations such as synchronization, fracture, and so forth from Navisphere Manager.

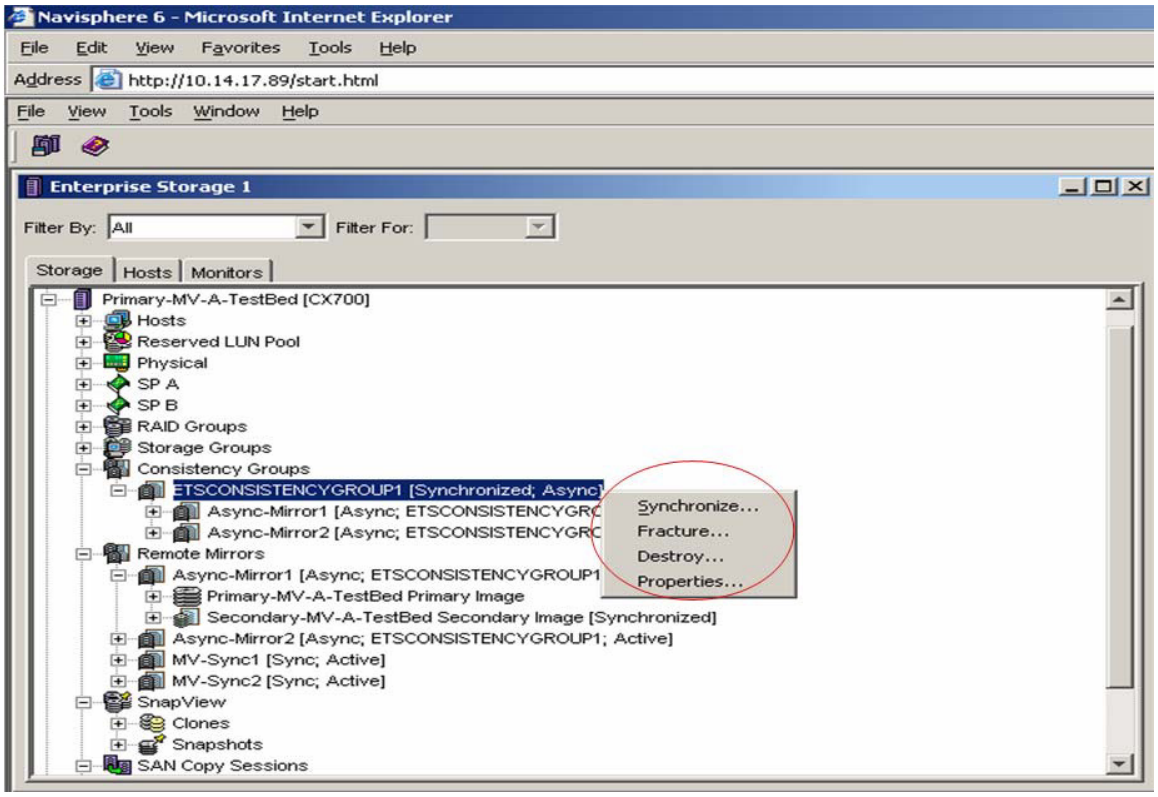

Figure 17. Using Navisphere Manager to manage MirrorView/A storage-system-based consistency groups

To automate consistency group operations, you can use Navisphere CLI. Figure 18 shows the Naviseccli CLI syntax.

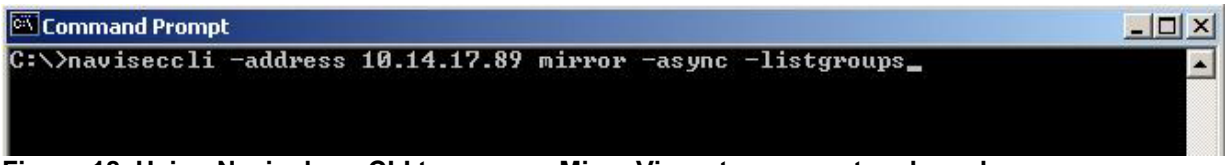

Figure 18. Using Navisphere CLI to manage MirrorView storage-system-based consistency groups

Consistency group promote options

Like individual mirrors, all members of the consistency group must be in the synchronized or consistent state for the group to be promoted. To facilitate a normal promote, all members must be in the synchronized state and communication must exist between storage systems. The various promote options are as follow:

- **Normal promote –** When promoting the secondary images of a group, MirrorView determines if connectivity exists between the primary and secondary storage systems, and then checks to see if all the mirrors within are synchronized. If these conditions are met, the promote proceeds and the current set of primaries become secondaries of the group. If there is no connectivity or MirrorView determines that any of the primaries do not match the secondaries, then it will fail and the user can select from the other promote options listed below. Unlike individual mirrors, a new consistency group is *not* created for the promote. Therefore, after a normal promote, the consistency group ID is the same before and after the promote.

- **Force promote –**Force promote is used if the conditions for normal promote are not met. It is a "brute force" promote that ignores most errors preventing a normal promote, such as the secondary image state being consistent, the mirror being in a fractured state (admin or system fracture), a failed link, and so forth. In cases where connectivity still exists between the primary and secondary storage systems, the secondary images are placed as primary images of the same consistency group. The original primaries are placed in the group as secondary images. If connectivity does not exist, force promote acts like Local Only promote.
- **Local only promote –** This promotes the consistency group on the secondary site without adding the primary images as secondary images of the mirrors. A new group is created on the secondary storage system, and the original secondary images are placed in the new consistency group as primary images. The primary images on the primary storage system remain in the original consistency group. Both consistency groups will be in the **Local Only** state. Although this option is not available for individual MirrorView/S mirrors, it is available for MirrorView/S consistency groups.
- **Cancel –** Do not proceed with the promote.

MirrorView management

Required storage system software

All CLARiiON arrays ship with the FLARE operating environment installed. The FLARE OE contains all basic and optional software components. Optional applications, such as MirrorView, are enabled by installing software enabler packages. MirrorView/A and MirrorView/S have separate enabler packages, which can be added via the Non-disruptive Upgrade (NDU) process. Figure 19 shows what the various enablers look like in the **Storage System Properties** dialog box.

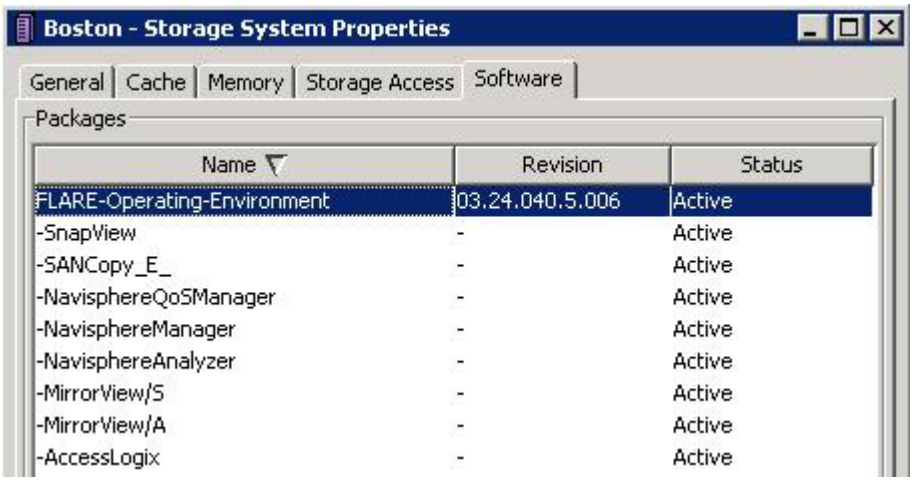

Figure 19. Storage System Properties (Software tab)

Once the enabler is installed, all relevant menus and dialogs become available to the user within Navisphere Manager, and the storage systems will respond to feature specific CLI commands.

Management topology

Management network connectivity

Network connectivity requirements for configuring MirrorView are determined by Navisphere Manager and/or CLI. Navisphere Manager and Navisphere CLI communicate with the storage systems using ports 80/443 or alternately, 2162/2163. Furthermore, network connectivity between systems on these ports is required when the systems are in the same Navisphere domain. Once configured, MirrorView

communicates in-band, via the data path, between the storage systems; any communication needed, such as to facilitating a promote operation, occurs in-band.

For example, assume a user stops I/O to a primary volume that is replicated with MirrorView/S until the secondary image is in the *synchronized* state. A command can then be issued to the secondary storage system to promote the secondary image. This command is sent through the secondary storage system's management LAN port. MirrorView on the secondary system then communicates in-band with MirrorView on the primary system to reverse the image roles. The process could then be reversed by issuing the promote command to the current secondary (former primary) storage system.

Navisphere domains

MirrorView can be used between systems in the same Navisphere domain and/or with systems that are not in the same Navisphere domain. Navisphere Manager UI must be at release 19 or greater to use the multidomain management feature for systems that are not in the same domain.

If there is no storage system in the environment running release 19 or greater, the off-array UI on the management server can be used to conduct multidomain activities. Figure 20 shows the Navisphere Manager UI configured to manage multiple domains.

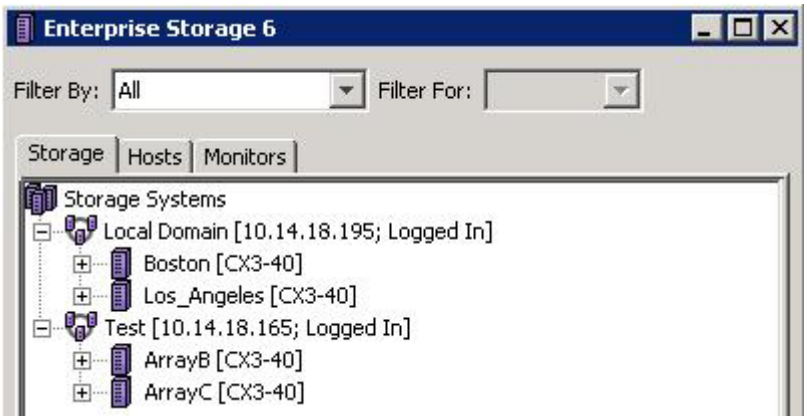

Figure 20. Navisphere multidomain management

Each domain is mutually exclusive and has its own set of users and user credentials. MirrorView can be configured between systems in different domains provided the user has manager or administrator rights in each of the system's domains that they wish to configure.

Management software

The Navisphere Task Bar offers a suite of wizards to guide users through a multitude of operations. One wizard in this suite is the MirrorView Wizard. Figure 21 shows the Navisphere Task Bar with the Replication tab selected.

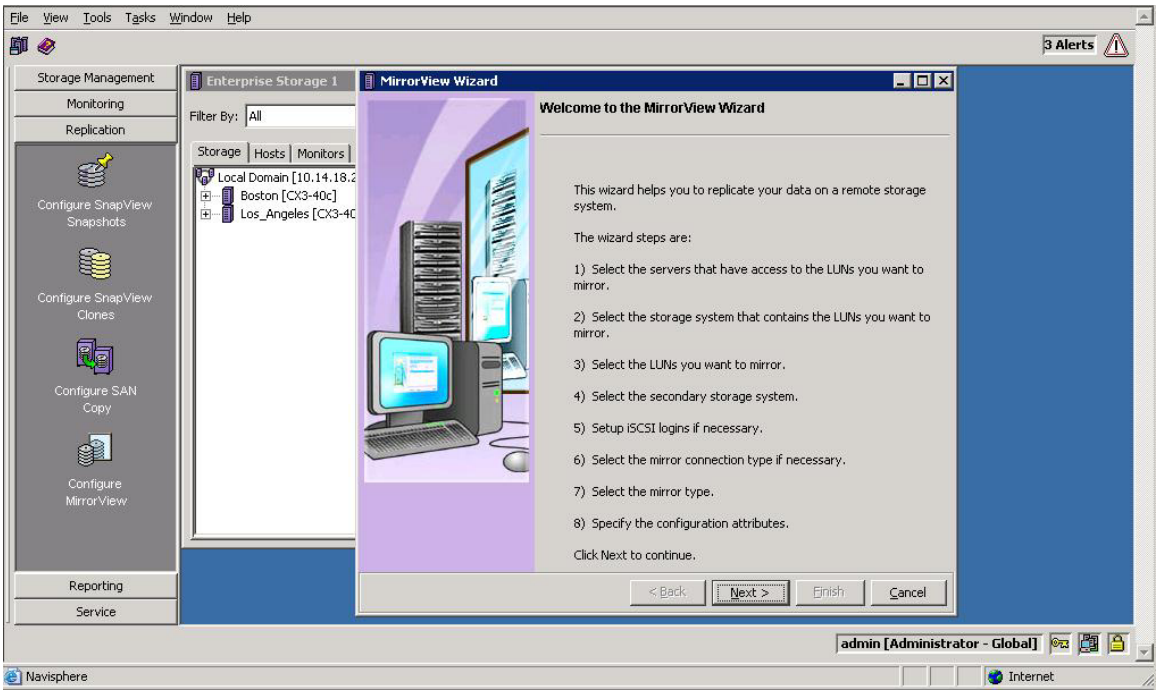

Figure 21. Navisphere Task Bar

The MirrorView Wizard uses a task-oriented approach to guide users through the process of creating mirrors. Along with creating the mirrors, the wizard automatically performs any needed setup operations such as creating iSCSI connections, MirrorView Connections, and creating/assigning write intent log and reserved LUNs.

The object-based approach to creating mirrors is also still available in Manager. Advanced users who have very specific requirements or have a set of conditions that are not covered by the wizard may use this approach.

Navisphere CLI offers a command line approach. All configuration options are available through the CLI and all security features for authentication, authorization, privacy, and audit are shared with Navisphere Manager.

For a comprehensive description of management operations, the following documents and white papers are available on EMC.com and Powerlink except where noted:

- *Navisphere Manager Help* (Powerlink only)
- *Securely Managing EMC CLARiiON Storage Systems* white paper
- *Domain Management with EMC CLARiiON Storage Systems – Release 19 Firmware* white paper

The following sections discuss the features, benefits, and assumptions of the MirrorView Wizard, object-based management, and CLI.

MirrorView Wizard

The wizards in the Navisphere Task Bar use a server-centric approach. This ease-of-use feature changes the traditional approach to storage management, which is to manage the environment from the storage system out to the server. The basic steps for the wizards are **select a server > select a storage system > select a LUN(s) > perform an action.** So the first assumption is that a server will be attached and a LUN assigned prior to using the MirrorView Wizard.

Before using the wizard, physical connections must be established between the two CLARiiON storage systems. This includes proper LAN preparation, SAN zoning, and Navisphere Domain configuration, tasks that are not included in the MirrorView Wizard.

In the wizard, the user is first guided through the LUN selection process by selecting the desired server and storage system. One or many LUNs may be selected. Then, the wizard performs the following discrete operations based on user input and existing conditions:

1. Select the remote storage system.
2. If necessary, create iSCSI connections.
3. Select the **MirrorView** connection type if both iSCSI and Fibre Channel exist.
4. If necessary, create a MirrorView connection.
5. Select a mirror-type: synchronous or asynchronous.
6. Select or bind a RAID group(s) for the secondary image(s).
7. Bind secondary image LUN(s) on the remote storage system.
8. Configure the remote mirror.
9. If multiple source LUNs are selected, perform consistency group configuration.
10. Assign the mirror LUN to a secondary server.
11. Configure the write intent log (in MirrorView/S) or the reserved LUN (in MirrorView/A).

Other than selecting the source LUN, remote storage system, and the secondary image RAID group, the wizard requires minimal input from the user. To simplify the process, there is a default naming convention for objects created by the wizard, based on the storage system and source LUN names. Users can edit these names in the wizard on the fly.

Mirror LUN properties

The wizard binds a secondary LUN in all instances. The wizard sets all secondary LUN properties to match those properties of the source LUN. The default name for the mirror LUN is Image Of <primary_server_ name><primary LUN name>. The default name may be changed simply by modifying the default name in the **Secondary LUN Name** field in the wizard, as shown in Figure 22.

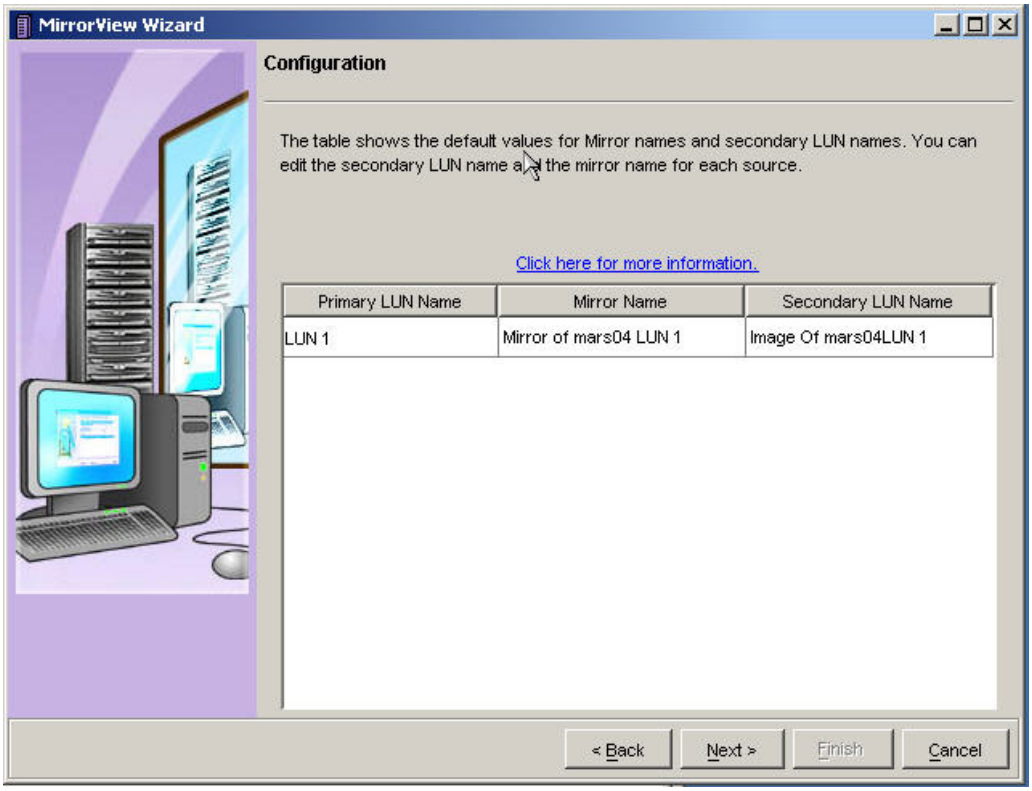

Figure 22. Mirror LUN name

Remote mirror configuration

The remote mirror wizard creates and configures the remote mirror with minimal input from the user. In fact, there are only two or three parameters the user may alter based upon the type of mirror chosen, as shown in Table 10.

Table 10. Remote mirror user configurable properties

Mirror property	Default value	Mirror type
Sync rate	Medium	Both
Mirror name	"Mirror of <primary server> <primary LUN>"	Both; See Figure 22 above
Update interval	4 hours	Asynchronous only
Consistency group name	"Group of <primary server> <primary LUN1> <primary LUN2> primary LUN3>…"	Both; only when multiple LUNs are selected

Additional parameters are automatically configured by the wizard with no user input, as listed in Table 11. In addition, the contents of the primary LUN are automatically copied to the secondary LUN upon completion of the wizard.

Table 11. Default remote mirror properties

Mirror property	Default value	Mirror type
Minimum required images	0	Both
Quiesce threshold	60	Synchronous only
Use write intent log	Selected	Synchronous only
Recovery policy	Automatic	Both

Assumption – Hidden property values are not commonly changed by customers.

Implication – Customers that require specific settings for these hidden properties may not change them using the wizard.

Alternative – Modify the property value after mirror creation with the wizard or use the traditional, object-based method to configure a mirror.

Consistency group configuration

The MirrorView Wizard automatically configures a consistency group when you select multiple primary LUNs. The consistency group default name is listed in Table 10 and may be modified by the user in the wizard. The wizard automatically creates the consistency group with the selected name and adds each remote mirror to the group.

Storage Group configuration

The MirrorView Wizard optionally allows the user to assign the secondary LUN to a server, for example, for disaster recovery preparedness. The wizard automatically performs the storage group operations if the user selects a secondary server as shown in Figure 23.

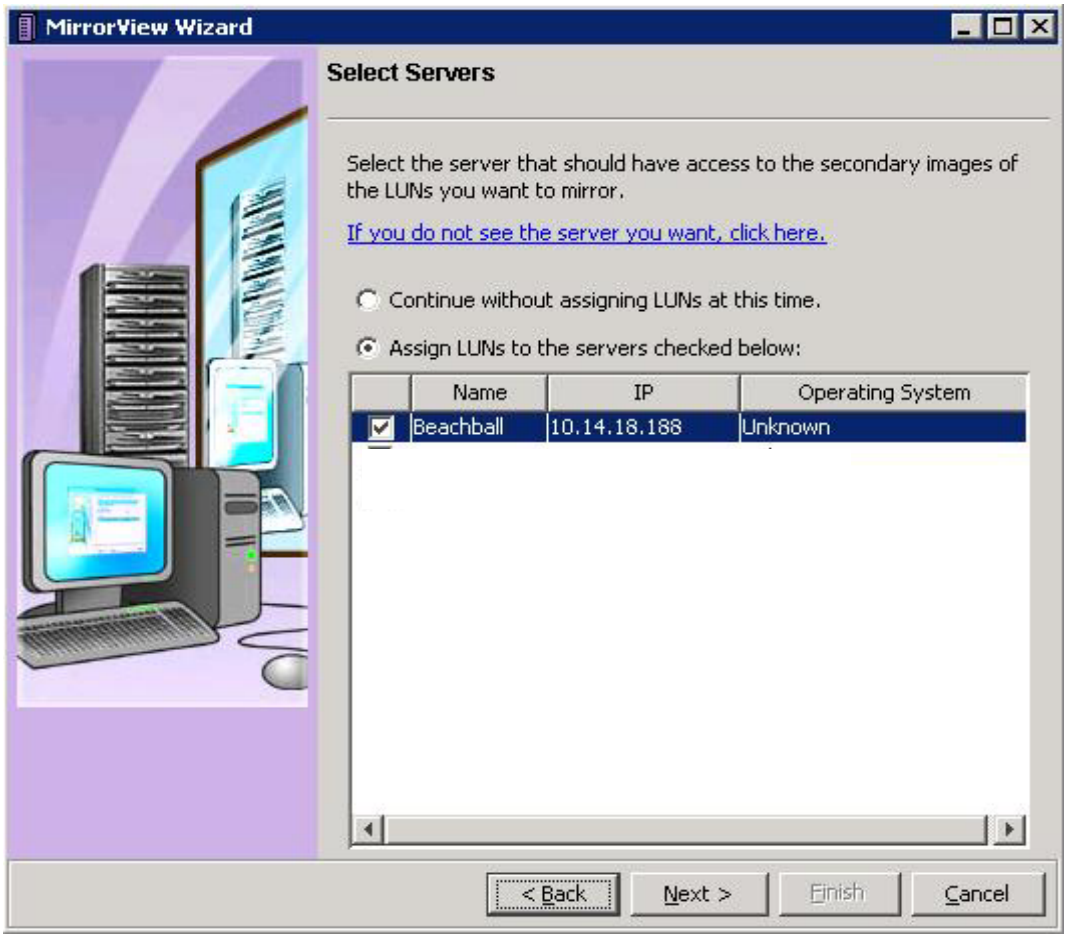

Figure 23. Select server for secondary image

The specific steps performed by the wizard vary based on the existence of a storage group for the selected server.

The server is already configured in a storage group – The wizard places the newly bound LUNs in the server's storage group.

The server is not configured in a storage group – The wizard creates a storage group for the server, places the server in the storage group, and places the newly bound LUNs in the storage group. In this case, the wizard automatically creates a storage group named "SG_<server name>." If you need a specific storage group name, you may change it using the traditional **Storage Group Properties** dialog box.

Write intent log configuration (MirrorView/S)

The MirrorView Wizard checks for the existence of the write intent log LUNs on the system and will bind them on both storage systems appropriately if needed, based on the following policy.

The RAID group used for each write intent log LUN is determined by the following criteria:

- Does not contain the source LUN
- Contains fewest server-visible LUNs
- Contains fewest clones and mirror images
- Contains the most free space
- Contains the fewest LUNs

The general Default Owner property is assigned based on the RAID group from which the write intent LUN is bound:

- For even numbered RAID groups, the default owner is SP-A.
- For odd numbered RAID groups, the default owner is SP-B.

The naming convention used is virtual disk sequence_number, where the sequence number is generated based on the existing LUN names in the storage system. Sequence number generation uses the following formula:

- The number starts from the largest number found at the end of existing LUN names. For example, if the existing LUNs in the storage system have names LUN 0 to LUN 10, LUN 50 to LUN 59, then the sequence number starts at 60, giving the first LUN the name "Virtual Disk 60."
- Each time the wizard is run, the above method is used to find a new starting sequence number, if necessary.
- If no trailing numbers exist in the LUN names (for example, if all LUN names were changed to not include a number), the sequence numbering starts at 0.

Reserved LUNs (MirrorView/A)

MirrorView/Asynchronous uses SnapView snapshot technology to track changes to the primary source LUN in between update intervals. As such, Reserved LUNs are required for both the primary and secondary LUNs in an asynchronous mirror relationship. The MirrorView Wizard automatically configures the reserved LUNs on both storage systems with no input from the user.

The following attributes are used to create the LUNs:

- Reserved LUN capacity for the primary and secondary images is set at 20 percent of the image LUN size.
- Two Reserved LUNs are bound for each primary and secondary LUN selected in the wizard.
- The total overhead storage space is divided evenly among the number of Reserved LUNs.
- The RAID group used for each Reserved LUN is determined by the following criteria:
    1. Does not contain the Source LUN
    2. Contains fewest server-visible LUNs.

3. Contains fewest clones and mirror images
4. Contains the most free space
5. Contains the fewest LUNs

The general Default Owner property is assigned based on the RAID group from which the Reserved LUN is bound:

1. For even-numbered RAID groups, the Default Owner is SP-A
2. For odd-numbered RAID groups, the Default Owner is SP-B

The naming convention used is the same as for write intent log LUNs; *Virtual Disk sequence_number*

For example, assume a user wants to create a mirror of LUN 12, which has a capacity of 200 GB. Existing LUNs on both systems range in LUN number and name from LUN 0 – LUN 50. The wizard will configure the reserved LUN pool on the primary storage system with two 20 GB LUNs and secondary storage system with two 20 GB LUNs. The reserved LUN names on both arrays will be "Virtual Disk 51" through "Virtual Disk 54".

The wizard always creates two new Reserved LUNs for each source LUN and never uses unallocated LUNs in the pool. Two Reserved LUNs for each source LUNs allow the reserved snapshots to expand over time.

Navisphere Manager object menus

Object-based management options are also available within Navisphere Manager. Object-based menus offer more of a "manual" approach to mirror management, but also offer all configuration options. Object-based menus can be used to create mirrors as well as change mirror properties of any mirror regardless of which method was used to create it.

To access object menus, simply right-click the object of interest, whether it is a storage system, mirror, consistency group, or image, and all relevant management options will be presented. Figure 24 shows the storage system level object menu from which any mirror-related operation can be launched. Other MirrorView object menus are available on the LUN and remote mirror objects in the tree structure.

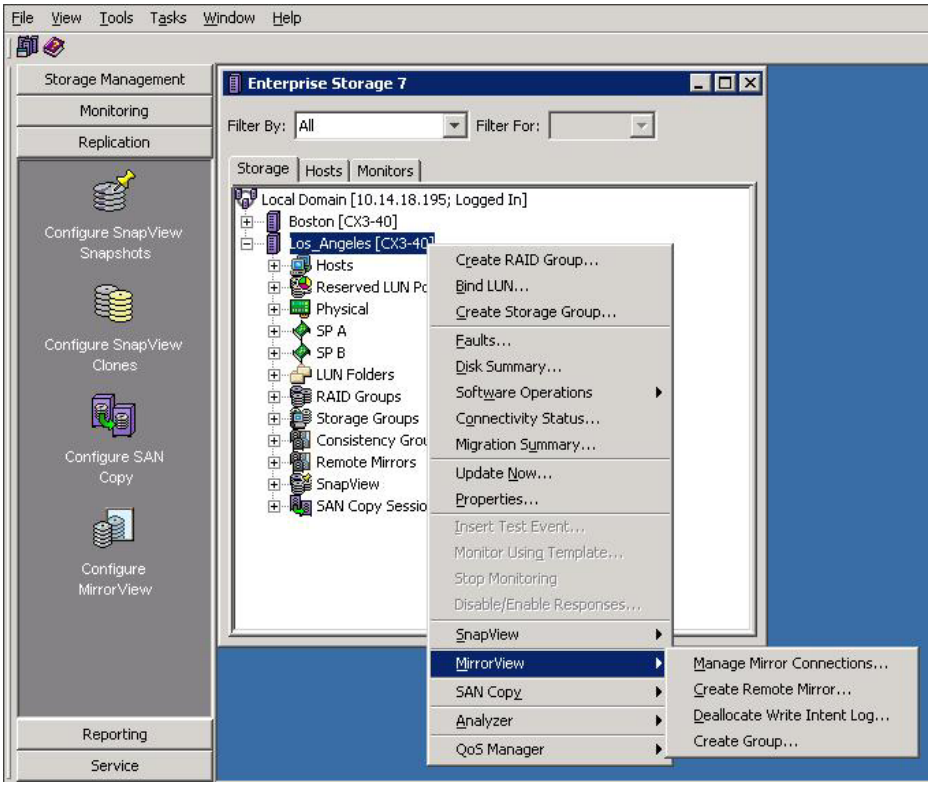

Figure 24. MirrorView right-click menu options in Navisphere Manager

Creating a mirror in this manner is a two-step operation. First, create the mirror while selecting the primary image LUN. Then, add a secondary image to the mirror. Figure 25 shows the object-based dialogs for creating a sync mirror and adding a secondary image. Here users have the option to specify values other than default for Minimum Required Images, Quiesce Threshold, use of the write intent log, initial sync required, and Recover Policy.

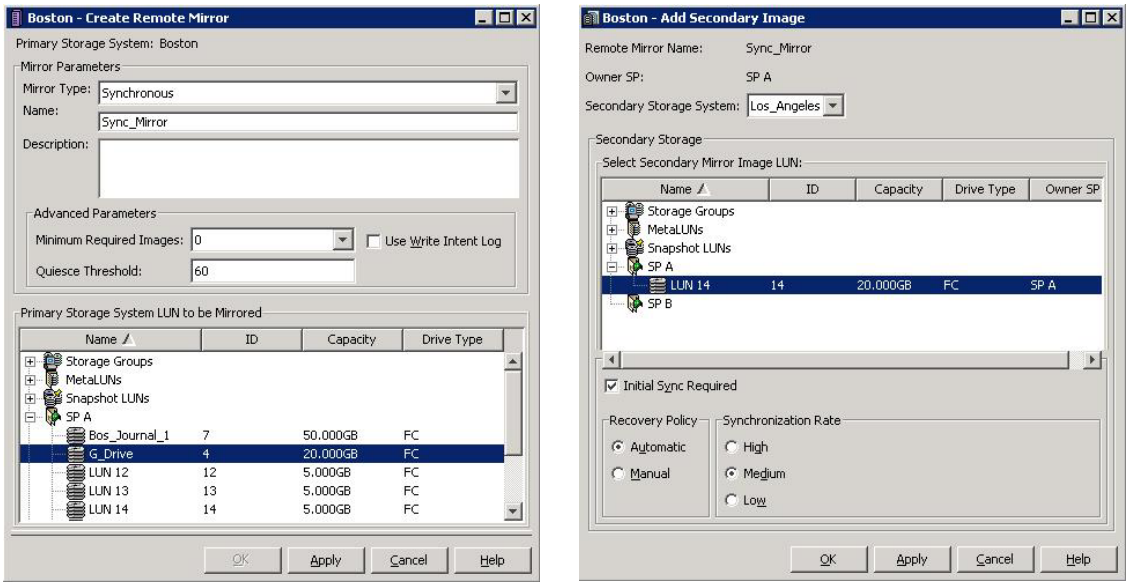

Figure 25. Create Remote Mirror and Add Secondary Image dialog boxes

Figure 26 shows the object menus for creating an asynchronous mirror and adding a secondary image. For asynchronous mirrors, users can configure minimum required images, initial sync required, recovery policy, and the update type in the object dialog boxes, where these items are shown in the wizard.

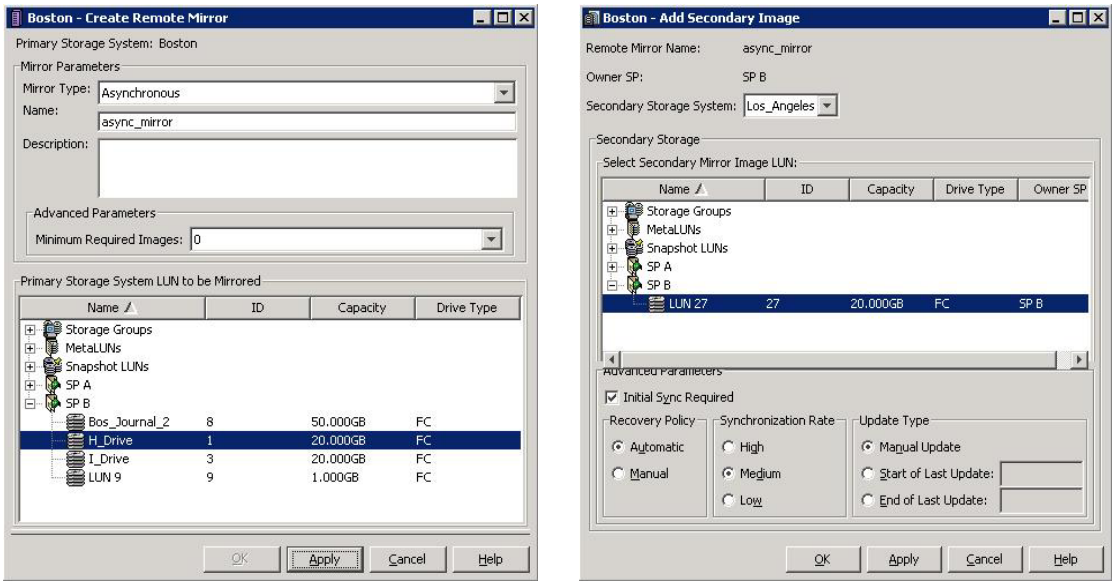

Figure 26. Create Remote Mirror and Add Secondary Image dialog boxes

Consistency groups have similar dialogs as shown in Figure 27. Note that for asynchronous mirrors, you can designate the **Recovery Policy and Synchronization rate** at the consistency group level. Values set at the consistency group level override the same settings for individual mirrors in the consistency group. For synchronous mirrors, you set the recovery policy at the consistency group level, but you set the synchronization rate at the individual mirror level.

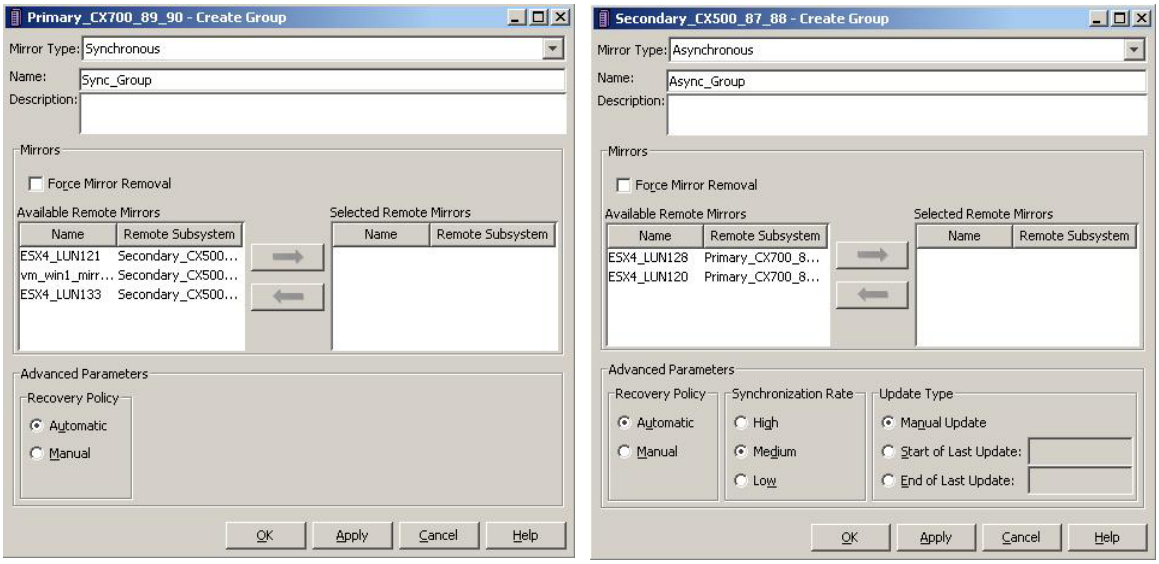

Figure 27. Create Group dialog boxes for synchronous and asynchronous mirrors

The **Force Mirror Removal** option is not used under normal conditions. However, under recovery conditions (where one of the storage systems is unavailable due to a disaster), it can be used to remove mirrors from the consistency group.

MirrorView failure recovery scenarios

When a failure occurs during normal operations, MirrorView and FLARE implement several actions to facilitate recovery. MirrorView tries to achieve three goals in the recovery of failures: minimize the duration of data unavailability, preserve data integrity, and when possible, complete a recovery transparent to the user.

The following sections describe various failure scenarios of primary components and secondary components. In these scenarios, failures may occur in the path between the production server and the primary storage system, to the primary or secondary image LUN itself [3], to a single SP, or to an entire CLARiiON storage system. For the purpose of these discussions, the term *primary storage system* refers to the storage system containing the primary image and *secondary storage system* refers to the storage system containing the secondary image. "Appendix C: Storage-system failure scenarios" contains much of the same information in a table format for quick reference.

Primary component failure and recovery

Failures in the production server to storage system path, primary image(s), or primary storage system affect application performance and availability. The nature of the failure may dictate that little corrective action is needed. It may also dictate that the secondary image be promoted and used as the production image until the primary site (or image) can be repaired. The following sections describe various failure scenarios within the primary storage system and its components.

Path failure between server and primary image

If a server loses its I/O path(s) to a storage processor, LUNs are trespassed to the alternate SP. This is typically enacted by server-based path management software such as PowerPath. With SP ownership changed, the primary storage system will send a request to the secondary storage system to trespass the secondary image. This way, SP ownership is consistent between the mirrored pairs on the primary and secondary storage systems. Between updates, MirrorView/A may not trespass the secondary volume until the start of the next update. Once trespassed, the update proceeds.

Release 26 FLARE adds support for the CLARiiON Active/Active feature. CLARiiON Active/Active is a failover strategy based on the Asymmetric Logical Unit Access (ALUA) standard. If there is explicit or implicit trespass activity on the primary LUN, the mirror will follow the primary LUN's SP ownership. For additional information on the CLARiiON Asymmetric Active/Active feature, see the white paper *EMC CLARiiON Asymmetric Active/Active Feature* on EMC.com and Powerlink.

Primary image LUN failure

Release 26 FLARE has new internal I/O redirection features that make it more resilient to certain types of LUN failures than previous releases. When a failure prevents an SP from accessing its LUNs on the back end, FLARE internally redirects I/Os to the other SP, provided that the other SP can access the LUNs.

In previous releases, back-end failures affecting just one SP (for example an LCC failure) would result in a LUN trespass by the host failover software. In release 26, the I/O redirection under this condition occurs below the front end and MirrorView drivers. Therefore, the host continues to access the LUN through the original SP, and mirroring activity continues on the same SP. Once the failure is corrected, the original SP resumes control of back-end I/O to the LUN.

3 Many media failures are themselves automatically detected and corrected by lower-level CLARiiON software, so this scenario is probably the least likely.

Masking the back-end failure internally provides a faster and less complex failover and recovery process. Server I/O and mirroring traffic remain balanced across the connectivity environment. Also, there is no interruption to the mirroring process, which reduces the number of fractures and resynchronizations.

If the LUN failure occurs in such a way that neither SP can access the LUNs, then the mirror will become fractured and server I/O to the LUN is rejected until the LUN becomes available again. During this process, the user may choose to promote the secondary image, and redirect server I/O to the secondary image.

If the secondary image state is consistent and/or fractured at the point in time of the failure on the primary image, a full resynchronization of the originally-primary-now-secondary image would be required, since there would be no other way to guarantee byte-for-byte synchronicity.

For MirrorView/A, if the primary image fails during an update and the user promotes the secondary image, any changes to the secondary are rolled back from the protective snapshot. If the user fixes the error on the primary without promoting the secondary, the update continues from the point it was interrupted.

Storage processor (SP) controlling primary image failure

If the SP owning the primary image fails, the MirrorView LUN trespasses to the surviving SP on the primary storage system per direction of the server failover software. At the same time, MirrorView on the primary system will issue a trespass command to the secondary to trespass the secondary image.

For MirrorView/S, the write intent log is used to avoid a full resynchronization. Mirrors not using the write intent log will have to do a full synchronization. With the improvements in write intent log performance in release 26, it should be enabled in almost all instances. For MirrorView/A, a full synchronization is not required since tracking/transfer data is stored on persistent media.

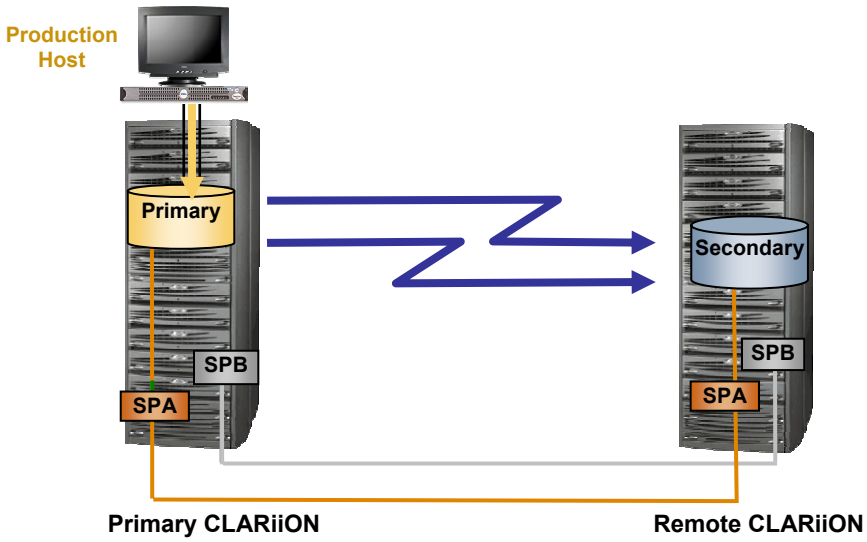

Figure 28. Before the SP failure: Both images are controlled by SP A

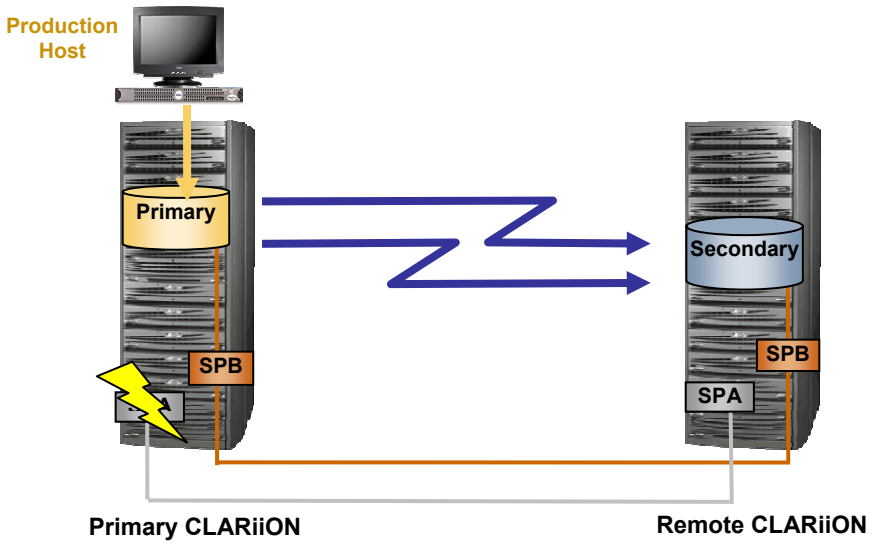

Figure 29. Following the SP failure: Both images are trespassed to SP B

Primary image storage system failure

If the storage system containing the primary image fails, server I/O to the LUN will fail until the storage system becomes available again. As with the primary image LUN failure, the user may choose to promote the secondary image, and redirect server I/O to this image. However, if the secondary image is promoted while communication to the original primary storage system is unavailable, it will not be possible to add the original primary image in the new mirror as a secondary image. Instead, it is necessary to add[4] the original primary device subsequently (when that storage system comes back online) to the new mirror and then perform a full synchronization. Alternatively, a different LUN on the original primary array can be added as the secondary image for the newly promoted primary (as shown in Figure 31), which likewise would require a full synchronization.

4 Before adding the original primary image to the new mirror, the old mirror must be force destroyed. In this case, the force destroy option is used because changes were made to the mirror while to primary system was unavailable.

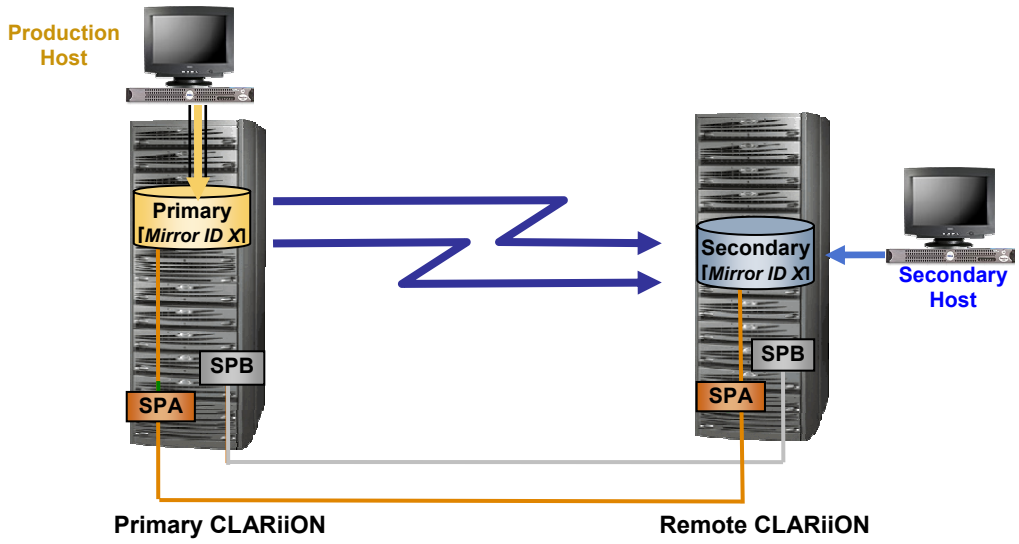

Figure 30. Before the primary storage system failure: Primary image (on the primary CLARiiON) has mirror ID X

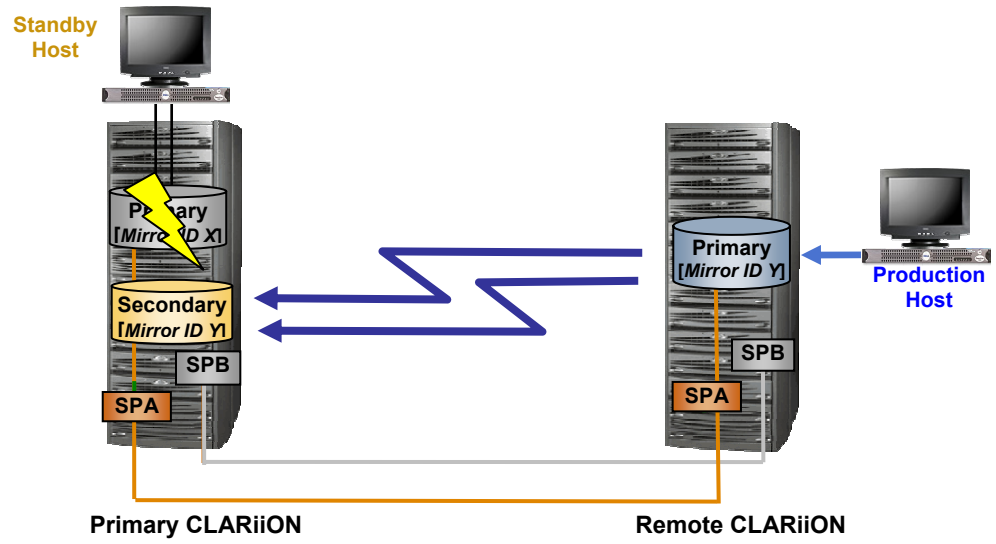

Figure 31. Following the primary storage system failue: Newly promoted primary image (on the remote CLARiiON) has mirror ID Y

Secondary component failure and recovery

Loss of secondary components reduces or eliminates disaster recovery protection of the mirror. When a primary image determines that it is not possible to communicate with a secondary image, it marks the secondary as *unreachable* and discontinues attempts to write to the secondary. However, the secondary image remains a member of the mirror.

Secondary image LUN failure

The I/O redirection capability (as described in the "Primary image LUN failure" section) of release 26 will allow mirroring to continue if the LUN failure only affects the current SP. The primary image would continue mirroring to the secondary image on the original SP. Prior to release 26, a secondary LUN failure under these conditions would result in the mirror becoming admin fractured.

If the secondary image LUN fails in such a way that affects both SPs, then the image is admin fractured. At that point, the administrator can choose whether to repair the LUN or remove it from the MirrorView relationship altogether. If the LUN is repaired, it can be resynchronized once it is available for use. If another LUN is selected to serve as the secondary image, it must first be fully synchronized.

Storage processor (SP) controlling secondary image failure

If the SP owning the secondary image fails, it is system fractured until the SP comes back online. Once the SP comes back online, the mirror is automatically resynchronized—provided that the **auto recovery** policy is selected. If **auto restore** is not enabled, then the administrator must intervene by explicitly starting the synchronization[5]. Additionally, if the same SP on the primary storage system also fails—resulting in the trespassing of the primary LUN to the peer SP and, likewise, the trespassing of the secondary LUN to its peer SP—the secondary image is automatically resynchronized (again, provided auto-restore is enabled).

Secondary image storage system (or link) failure

If the storage system containing the secondary image fails, the secondary image is system fractured. Similarly, if the link between the primary and secondary storage system fails, the secondary image is also system fractured. In either event, once the issue has been resolved and the secondary storage system is back in communication with the primary storage system, the secondary image is automatically resynchronized if **auto recovery** policy is enabled.

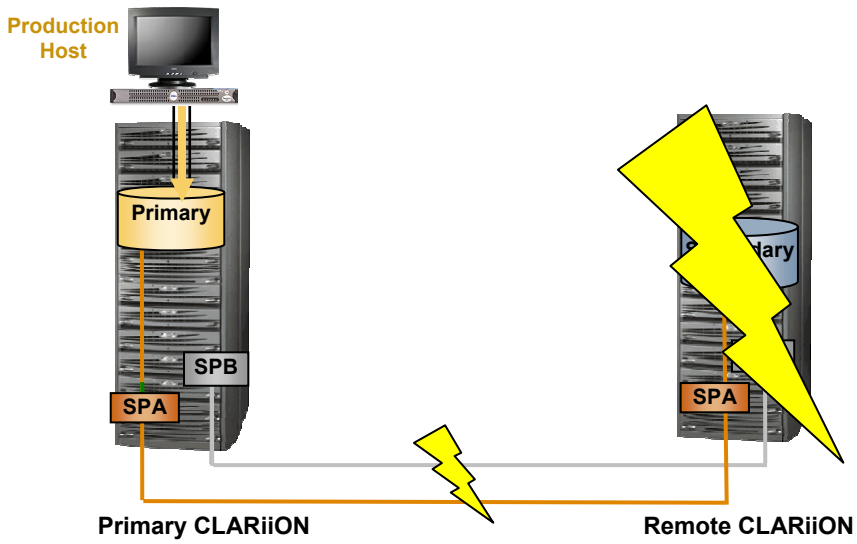

Figure 32. Failure of link between storage systems or of secondary storage systems temporarily suspends mirroring

5 Users are given the option to have the resynchronization occur automatically by enabling **auto restore**. If they prefer, they can manually initiate this themselves. This is a property of the mirror, and can be changed at any time.

Full reserved LUN (MirrorView/A only)

On the primary storage system

If the reserved LUN space for a mirror fills up during an update, MirrorView software aborts the update operation and the mirror transitions into an admin-fractured state. For the MirrorView/A session, the point-in-time view of the primary mirror LUN created at the start of the update operation is stopped while the information in the tracking map is maintained. In this condition, the user has an option of adding more reserved LUNs and manually starting the MirrorView/A session update operation. The contents of the tracking map are persistently maintained. Starting the update operation will result in incremental update of the secondary mirror with the changes on the primary mirror since the last successful update.

On the secondary storage system

The amount of reserved LUN capacity required on the secondary side equals the amount of changes applied to the secondary. For example, if 20 percent of the contents of the primary LUN changes, the reserved LUNs must have a minimum capacity equal to the 20 percent change.

If, during the update, the reserved LUNs on the secondary system are full, the MirrorView/A session will transition into an admin-fractured state. The point-in-time view of the secondary mirror LUN in the form of a gold copy is maintained. User intervention is required to allocate more storage to the reserved LUNs and restart the update operation. The update will continue from the point it was interrupted.

Using SnapView with MirrorView

SnapView is CLARiiON's storage-system-based replication software for local replicas. SnapView is licensed separately from MirrorView/S and MirrorView/A. It supports both pointer-based replicas, called snapshots, and full binary copies, called clones. SnapView benefits users by providing business continuance copies for local rapid recovery and parallel processing operations such as back up, reporting, or testing.

When used with MirrorView, SnapView can provide local replicas of primary and/or secondary images. It allows for secondary access to data at either the primary location, secondary location, or both, without taking production data offline.

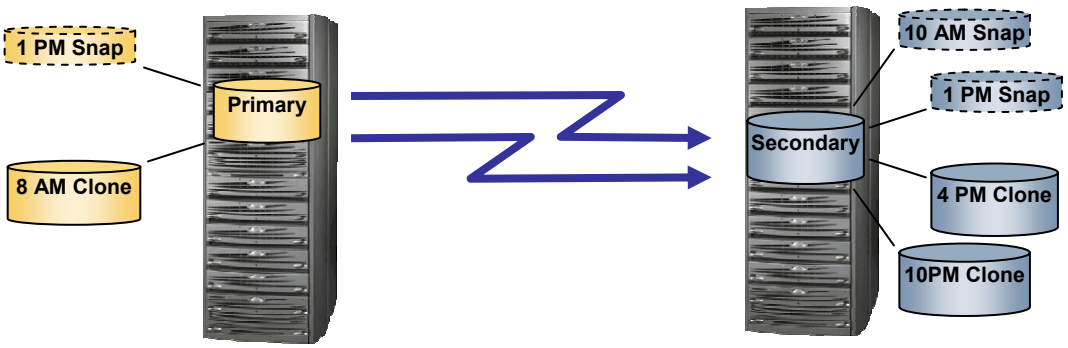

Figure 33. Clones and snapshots of primary and secondary images

For in-depth information on SnapView architecture and basic operations, see the following white papers on EMC.com and Powerlink:

- *EMC CLARiiON SnapView Clones – A Detailed Review*
- *EMC SnapView SnapShots and Snap Sessions Knowledgebook*

SnapView replicas of MirrorView primary images follow standard operating guidelines as described in the above white papers. SnapView replicas of MirrorView secondary images have to be produced very carefully because secondary image states can change frequently.

All SnapView replicas should represent points in time where the secondary image is in the **consistent** or **synchronized** state. During MirrorView synchronizations, the secondary image LUN is not in a usable state. Secondary images are in the **synchronizing** state during MirrorView/S synchronizations and the updating state during MirrorView/A updates. You should not start SnapView snapshots or fracture clones while the secondary image is in this state.

You can use the **Manual** MirrorView Recovery policy to coordinate SnapView activities with fractured mirrors. The Manual option requires user intervention to start synchronizations after a system fracture. Snapshots can be started or clones fractured before initiating the mirror synchronization.

SnapView and MirrorView consistency features are also compatible. Use SnapView's "consistent start" and "consistent fracture" operations when creating SnapView replicas of a MirrorView/S consistency group. SnapView consistent operations cannot be performed on the consistency group, but can be conducted across all members of the group. Consistent operations can also be used with MV/A, but are not required if you are starting sessions or fracturing clones between updates.

Clone of a mirror

Support for clone of a mirror provides users with full binary copies of their mirrored volumes. Full binary copies provide several benefits, especially in the area of performance. When a clone is fractured, it has little to no performance impact on production volumes. Clones reside on separate disks than the production volumes, so higher workloads can be maintained without directly affecting the production workload.

LUN eligibility

Any mirror image *can* be a clone *source;* however, a clone *image* of another source *cannot* serve as a mirror image. The LUN can become a clone source or mirror image in any order. For example, an existing clone source could be added to a mirror as a secondary image. Alternately, a mirror secondary image could also be made a clone source.

Limit dependencies

Clone and MirrorView/S limits are independent of one another in release 24 and later. The number of mirror images and clones for each LUN and each storage system are counted separately against the LUN and storage system limits. Release 24 also introduced an increase in the total number of clone images per storage system for the CX3 series. The CX3-20, CX3-40, and CX3-80 models support 256, 512, and 1024 clones, respectively. The number of clones for source LUN remains 8, while the number of mirror images per LUN remains 1.

Clone reverse synchronizations

To preserve the secondary image, clone reverse synchronizations (syncs) are allowed for primary images only. Secondary image LUNs can be reverse synchronized only after they are promoted to the role of primary image.

During a clone reverse sync, SnapView copies data from a clone image back to the source LUN. Servers are allowed read/write access to source LUNs during the reverse sync process, because SnapView manages the copy process along with incoming I/O from the server.

Until the reverse synchronization completes, a copy of the source LUN is not usable to another server. As a safeguard against copying an unusable image, the secondary image of a reverse syncing clone source must be fractured for the duration of the reverse sync. This is enforced by the software to prevent MirrorView from presenting a **consistent** secondary image state while the data isn't usable. If the mirror is part of a consistency group, then the group must be fractured.

Promoting a secondary image during a reverse sync is possible with the **Local Only** promote option. The local only promote option places the secondary image LUN in a new mirror as the primary image. The original primary image LUN cannot be demoted to secondary while the reverse sync process is active.

Finally, in the case of MirrorView/A only, a reverse sync cannot be performed on a primary image if the image is *rolling back*. An image is rolling back if it was recently promoted and MirrorView/A is recovering it from the gold copy. It is possible to reverse sync the primary image once the rollback process is complete.

Clone state "Clone Remote Mirror Synching"

The **Remote Mirror Synching clone** state was added to help users coordinate mirror-sync and clone-sync-and-fracture operations. During mirror sync operations the secondary image LUN is not in a usable state. This is true for initial syncs, MirrorView/S partial syncs, and MirrorView/A updates when the secondary image is in the **synchronizing** state.

Clones that are in-sync during a mirror sync will be in the **Clone Remote Mirror Synching** state as shown in Figure 34. This clone state indicates that the source LUN, also serving as a secondary image, is not in a consistent state and should not be fractured.

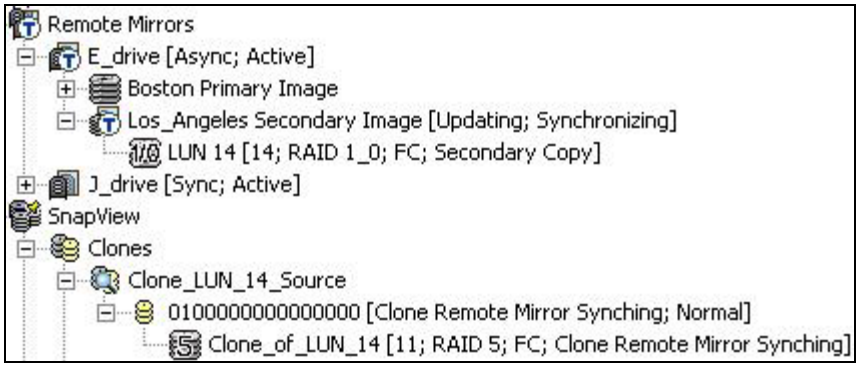

Figure 34. Remote Mirror Syncing Clone state

iSCSI connections

iSCSI connections define initiator/target pairs for iSCSI remote replication between CLARiiON storage systems. Similar to zoning in a Fibre Channel environment, they dictate which targets the initiators log in to. Since all necessary iSCSI connections are automatically created with the MirrorView Wizard or the MirrorView Connections dialog box, it is rare that it's necessary to explicitly manage iSCSI connections for MirrorView.

iSCSI connections are defined per front-end port of the storage system with the initiating ports. Once set up, those ports are able to log in to another storage system's set of defined target ports. For MirrorView, connections must be made for the primary storage system to the secondary system and for the secondary system back to the primary system. This is required because mirroring can be bidirectional between systems and primary/secondary images can reverse roles, resulting in a reversal of replication direction.

Figure 35 shows a connection defined for the storage system "Boston". This connection allows the MirrorView ports (A-3, B-3) to log in to target ports of the "Los Angeles" system. The IP addresses listed in the Target Portal column represent the MirrorView ports of the Los Angeles system. The connections shown were automatically generated by MirrorView when the MirrorView connection was enabled.

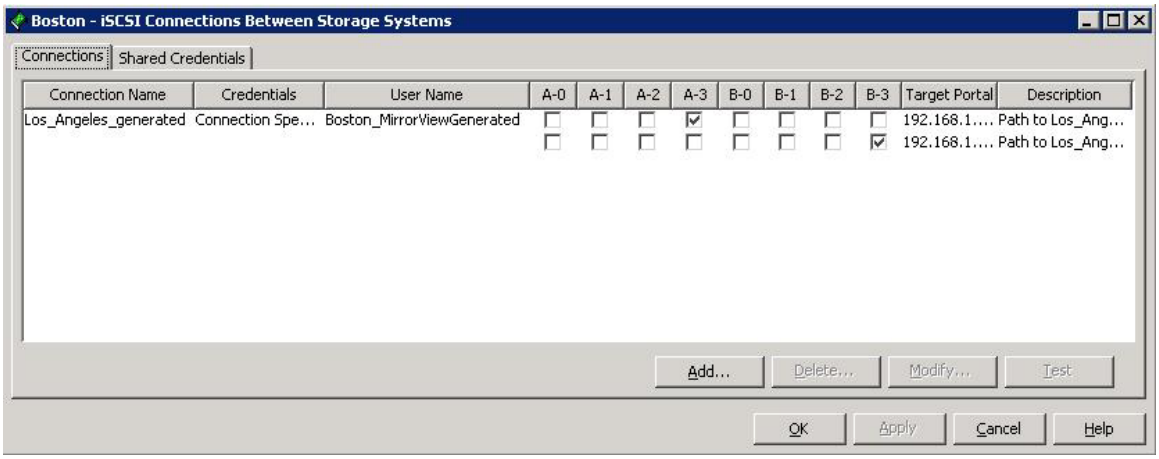

Figure 35. iSCSI Connections Between Storage Systems dialog box

To manually add an iSCSI connection, use the **Add Connection** dialog box, which is shown in Figure 36. In this dialog, initiator target pairs and CHAP authentication methods are identified. The dialog can be launched in the UI by going to Path Array **> iSCSI > Connections Between Storage Systems** and clicking the **Add…** button. Checkboxes are used to choose which front-end ports are wanted to log in to the target port. The target port is selected using the drop-down menu. One or more ports can assigned to log in to each target port.

When creating iSCSI connections, the user has three options for CHAP authentication:

- Shared – A general set of CHAP credentials that can be used by any or all of the connections of a storage system
- Connection Specific – CHAP credentials that are defined specifically for the connection being created
- None – No authentication

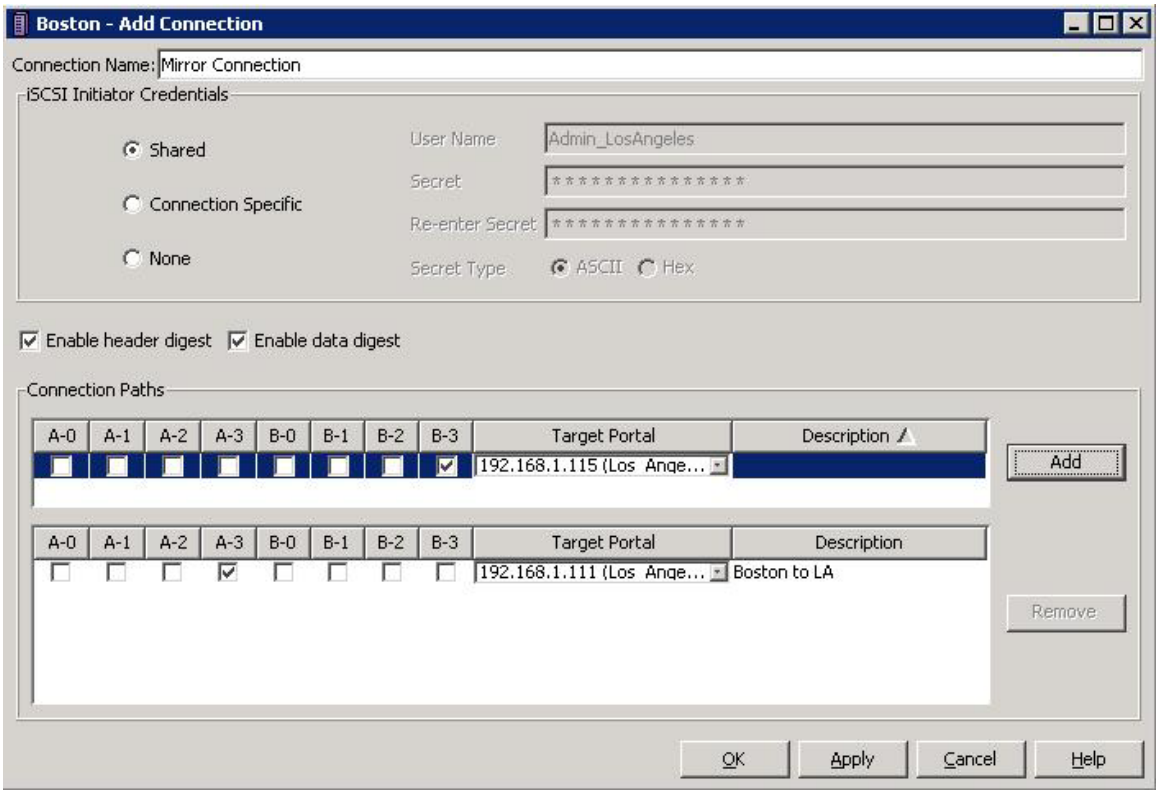

Figure 36. Add Connection dialog box

Connection-specific CHAP credentials are used when an iSCSI connection is created by the MirrorView Wizard or MirrorView Connections dialog box. The connection shown in Figure 35 shows the connection name and username format based on the system names used when automatically generated. A CHAP username and secret is generated by the system when the connection is created.

System-generated CHAP secrets are unique for each connection created. Corresponding entries in the target system's CHAP management are made for the user by the system when the iSCSI connection is created. Users that must define their own CHAP username and secrets can edit the system-generated entries by using the Modify button on the iSCSI Connections Between Storage Systems dialog box (Figure 35). Corresponding changes must be made to the target system's CHAP management (Figure 38).

The need for CHAP credentials is determined by the target system. There is an option in the iSCSI port properties page for requiring initiator authentication. Figure 37 shows the iSCSI Port Properties page, which is launched from the storage system right-click menu and going to **iSCSI** > **Port Management**, selecting the desired port, and clicking **Properties**. The option for requiring initiator authentication is at the bottom of the dialog box.

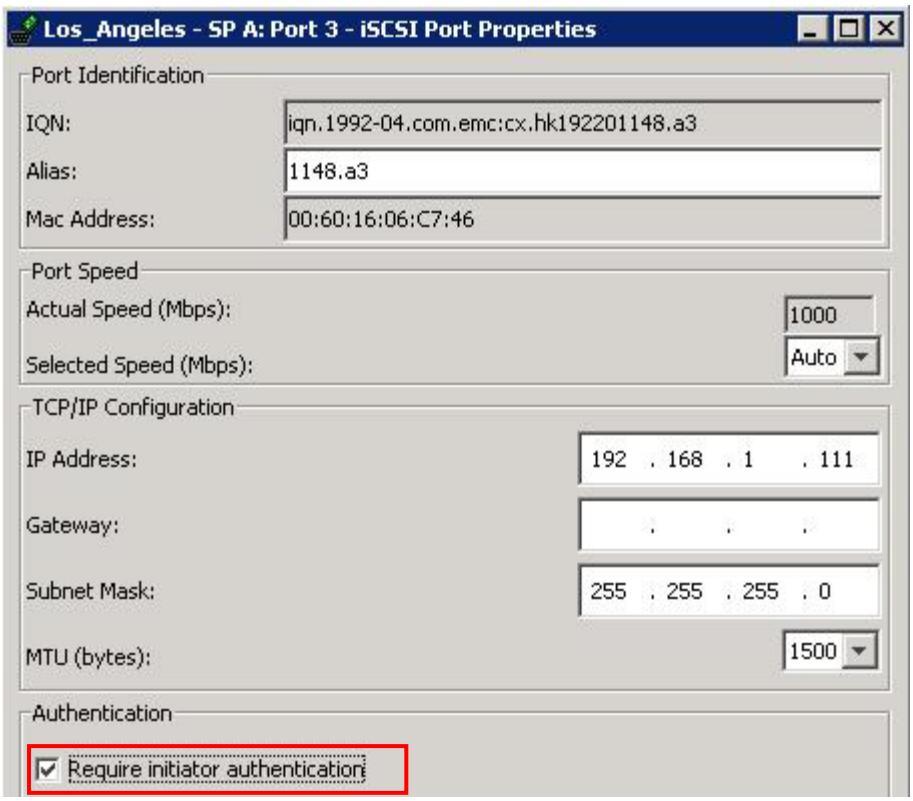

Figure 37. iSCSI Port Properties dialog box

When initiator authentication is required, initiators must be defined within the target system's CHAP management dialog. Figure 38 shows the iSCSI CHAP Management dialog box for the Los Angeles system. The entries shown are the system-generated entries created for the MirrorView ports of the Boston storage system. Initiator and CHAP user/secret entries can be manually added from this dialog, however, it's rare that manual entries are required for MirrorView ports, since the entries are automatically created as part of creating the iSCSI connection.

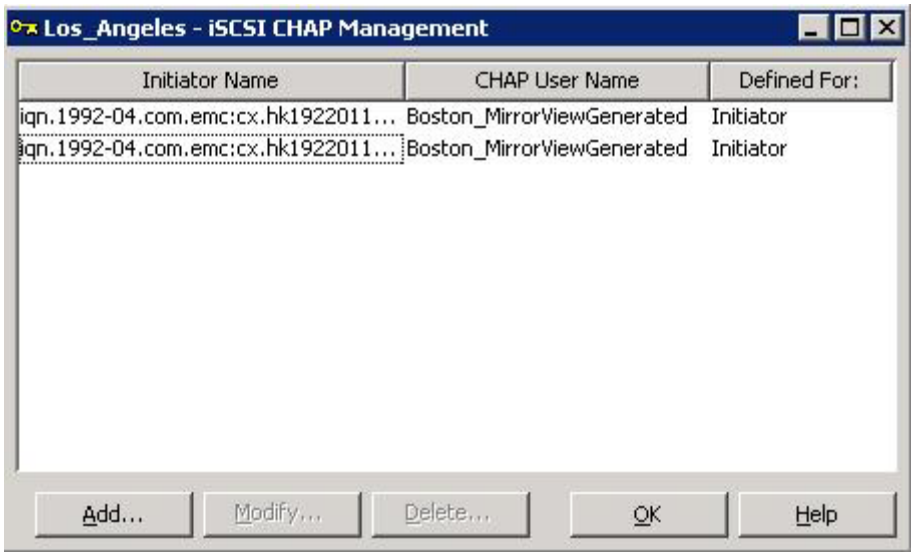

Figure 38. iSCSI CHAP Management dialog box

When iSCSI connections are created by the system, CHAP credentials are always entered regardless of whether they are required by the target system. This way, if the user did not initially have initiator authentication required, the credentials are already in place if they were to enable it later.

Conclusion

MirrorView offers two products that can satisfy a wide range of recovery point requirements. MirrorView/S offers a zero data loss solution typically implemented over high bandwidth, low latency lines. MirrorView/A offers a periodic update solution with recovery points in minutes to hours. MirrorView/A is typically implemented over long distances on low bandwidth, higher latency lines.

Both products offer a consistency group feature for managing write ordering over a set of volumes. Consistency groups can lower RTO by maintaining restartable copies of a data set on the remote storage system. Ease of management is achieved with the ability to manage all remote mirrors in the consistency group as a single unit.

References

White papers

The following papers are available on EMC.com and Powerlink, EMC's customer- and partner-only extranet:

- *EMC CLARiiON MetaLUNs: Concepts, Operations, and Management*
- *EMC Virtual LUN Technology – A Detailed Review*
- *CLARiiON Integration with VMware ESX Server*
- *Navisphere Task Bar Explained – A Detailed Review*
- *EMC CLARiiON SnapView Clones – A Detailed Review*
- *EMC SnapView SnapShots and Snap Sessions Knowledgebook*
- *CLARiiON Asymmetric Active/Active Feature*

Product documentation

- *EMC CLARiiON Open Systems Configuration Guide for CX3-Series and CX-Series Storage Systems*
- *Navisphere Manager Help*
- *MirrorView/Synchronous Release Notes*
- *MirrorView/Asynchronous Release Notes*
- *Using MirrorView/Synchronous with Celerra for Disaster Recovery* technical module
- *MirrorView/Synchronous Command Line Interface (CLI) Reference*
- *MirrorView/Asynchronous Command Line Interface (CLI) Reference*
- E-Lab Navigator

Appendix A: Remote mirror conditions and image states

Mirror states

State	Meaning
Active	The remote mirror is running normally
Attention	The mirror's secondary image is fractured, and the mirror is configured to generate an alert in this case. The mirror continues to accept server I/O in this state. The event code ID for a mirror moving into the **Attention** state is **0x71050129.**

Secondary image states

State	Meaning
Synchronized	The secondary image is identical to the primary image. This state persists only until the next write to the primary image, at which time the image state becomes **Consistent**.
Consistent	The secondary image is identical to either the current primary image or to some previous instance of the primary image. This means that the secondary image is available for recovery when you promote it.
Synchronizing	The software is applying changes to the secondary image to mirror the primary image, but the current contents of the secondary are not known and are not usable for recovery.
Out-of-Sync	The secondary image requires synchronization with the primary image. The image is unusable for recovery.
Rolling Back (MirrorView/A only)	A successful promotion occurred where there was an unfinished update to the secondary image. This state persists until the Rollback operation completes.

Image condition

Along with an image state, an image will have an image condition that provides more information.

Condition	Meaning
Normal	The normal processing state.
Admin Fractured	The administrator has fractured the mirror, or a media failure (such as a failed sector or a bad block) has occurred. An administrator must initiate a synchronization.

System Fractured	The mirror is system fractured
Synchronizing	• MirrorView/A or MirrorView/S is performing an initial synchronization. • MirrorView/S is synchronizing after a fracture.
Updating (MirrorView/A)	Mirror is performing periodic update.
Waiting on admin	The mirror is no longer system fractured. It is a temporary condition for automatic recovery mirrors. For manual recovery mirrors, the administrator can now initiate the synchronization command
Queued to be Synchronized (MirrorView/S)	Synchronize command has been received, but the maximum number of synchronizations are in progress (20 per SP). The mirror will synchronize when one of the in-process synchronizations completes.

Appendix B: Consistency group states and conditions

Consistency group states

State	Meaning
Synchronized	All the secondary images are in the Synchronized state.
Consistent	All the secondary images are either in the Synchronized or Consistent state, and at least one is in the Consistent state.
Quasi Consistent (MirrorView/A)	A new member that is not consistent with existing members is added to the consistency group, which automatically starts an update. After the update completes, the consistency group is again consistent.
Synchronizing	At least one mirror in the group is in the Synchronizing state, and no member is in the Out-of-Sync state.
Out-of-Sync	The group may be fractured, waiting for synchronization (either automatic or via the administrator), or in the synchronization queue. Administrative action may be required to return the consistency group to having a recoverable secondary group.
Scrambled	There is a mixture of primary and secondary images in the consistency group. During a promote, it is common for the group to be in the scrambled state.
Empty	The consistency group has no members.
Incomplete	Some, but not all of the secondary images are missing, or mirrors are missing. This is usually due to a failure during group promotion. The group may also be scrambled.
Local Only	The consistency group contains only primary images. No mirrors in the group have a secondary image.
Rolling Back (MirrorView/A)	A successful promotion occurred where there was an unfinished update to the group. This state persists until the Rollback operation completes.

Consistency group conditions

MirrorView/S

Condition	Meaning
Active	The normal processing state.
Inactive	The group is not accepting I/O; this is normal during group promotion.
Admin Fractured	The administrator has fractured the group, or a media failure (such as a failed sector or a bad block) has occurred. An administrator must initiate a group synchronization.
System Fractured	Some or all of the members are system fractured.
Waiting on sync	The group is waiting on a Synchronize command. Consistency groups with the manual recovery option require the user to initiate the synchronization. If after a system fracture, it is only a temporary condition for automatic recovery groups before the system initiates a synchronization.
Invalid	Temporary condition while a group fracture is in progress;

	the group is partially fractured. If the state is incomplete, the condition is irrelevant and is thus set to invalid. If this state persists, try fracturing the group and synchronizing it.

MirrorView/A

Condition	**Meaning**
Normal	The normal processing state.
Initializing	The group is performing an initial synchronization.
Updating	The group is performing a periodic update.
Admin Fractured	The administrator has fractured the group, or a media failure (such as a failed sector or a bad block) has occurred. An administrator must initiate a group synchronization.
System Fractured	Some or all of the members are system fractured.
Waiting on admin	The group is no longer system fractured. It is a temporary condition for automatic recovery groups. For manual recovery groups, the administrator can now initiate the group synchronization.

Appendix C: Storage-system failure scenarios

The following table details CLARiiON storage-system failure scenarios during various states or conditions. Primary image failures are covered first, followed by secondary image failures.

CLARiiON storage-system failure scenarios

Primary image failure		
Event	**State/Condition**	**Action**
Primary CLARiiON storage system totally fails.	Any	Option 1: Administrator promotes a secondary LUN on a secondary storage system. After application data recovery, applications on a standby server can start up and access required data. **Note**: Any writes that are in progress when the primary storage system fails may not propagate to the secondary storage system. Also, if the remote image was fractured at the time of the failure, any writes since the fracture will not have propagated.
		Option 2: Repair and reboot the primary storage system, then synchronize the secondary LUN(s) using the write intent log. If the write intent log is disabled, then a full synchronization is necessary.
Primary CLARiiON storage system's secondary SP fails.	Any	Repair the failed SP to restore high-availability protection for the data. Access to the mirrored LUN data is uninterrupted.
Primary CLARiiON storage system's controlling SP fails.	Server attempts I/O request to the LUN.	The I/O request fails, and software on the server retries the request to the secondary SP. This results in a trespass to the secondary SP, which then synchronizes the remote LUN (if necessary) based on the write intent log. The I/O request is coordinated with the synchronization.
Primary CLARiiON storage system's controlling SP fails and reboots. No server I/O attempted while the SP is unavailable.	Any	All primary LUNs are checked for a current relationship with their secondary LUNs.
	Primary LUN is in-sync with secondary LUN.	No action
	Primary LUN is synchronizing with secondary LUN.	If a full synchronization is interrupted, then it continues from the last checkpoint. If a fracture log-based synchronization is interrupted, then a full synchronization is necessary unless the mirror is using the write intent log, in which case the partial synchronization can be resumed. In either case, if auto sync is enabled, then synchronization starts automatically. Otherwise, an administrator must start the synchronization.
	Primary LUN is out-of-sync with secondary LUN.	A full synchronization is necessary. If auto sync is enabled, then synchronization starts automatically. Otherwise, an administrator must start the synchronization.

Primary image failure		
Event	**State/Condition**	**Action**
	Primary LUN is consistent with secondary LUN.	For a primary LUN using the write intent log, regions in the log marked potentially out-of-sync are read from the primary LUN and written to the secondary LUN. For a primary LUN **not** using the write intent log, the secondary LUN is marked out-of sync and a full synchronization is necessary. If auto sync is enabled, then synchronization starts automatically. Otherwise, an administrator must start the synchronization.
Path failure - Primary LUN trespass from controlling SP to peer SP	LUN is consistent or in-sync.	I/O is quiesced during the trespass. The fracture log (MV/S) or Reserved LUNs (MV/A) are transferred to the peer SP.
	Primary LUN is synchronizing with secondary LUN.	Synchronization continues on the new controlling SP.
	Secondary LUN is fractured.	Fracture log or Reserved LUNs transfer from the previous controlling SP to the secondary SP.
Back end failure	Any	If I/O can be redirected to the other SP, host access and mirroring continue on original SP.
Media error: data write		An error is returned to the server. Any defined secondary images are administratively fractured.
Media error: write intent log modification (MV/S)		Disable use of write intent log for the mirror. Log the error and alert an administrator to review. The mirror is admin fractured.
Reserved LUNs full (MV/A)	MV/A Update	Current point-in-time image is stopped. Mirror becomes admin fractured. User must add capacity to RLP and resume update. Update will include all changes made to the primary up to the restart of the update.
Secondary image failure		
Event	**State/Condition**	**Action**
Secondary CLARiiON storage system totally fails	Any	Repair and reboot the secondary storage system. The secondary LUN is fractured. To rejoin the mirror, synchronization (either full or based on a fracture log) of the secondary LUN is necessary for all states *except* in-sync.
Secondary CLARiiON storage system's SP fails	Any	No action Repair the failed SP to restore high-availability protection for the data.
Secondary CLARiiON storage system's controlling SP fails or reboots		Primary storage system detects the failure and fractures the secondary LUN. If the failed SP is unavailable for an extended period, an administrator should manually trespass the primary LUN to its secondary SP so mirroring can continue. When the failed SP returns to service, synchronization is necessary (unless the LUN is in-sync). If the LUN is consistent or synchronizing, the system uses the fracture log to perform synchronization. Otherwise, a full synchronization is necessary.
LUN trespass from the controlling SP to the secondary SP		No action This is the expected result when the primary storage system trespasses for any reason.

Primary image failure		
Event	**State/Condition**	**Action**
Back end failure	Any	If I/O can be redirected to the other SP, host access and mirroring continue on original SP.
Media error: data write		Secondary image will be marked as administratively fractured. When repairs are complete, administrator starts synchronization.
Reserved LUNs Full (MV/A)	MV/A Update	Mirror becomes admin fractured. Point-in-time image of secondary is maintained. User must add more capacity to RLP and resume update. Update will continue from where it was interrupted.