Choosing the Right CLARiiON® Data Replication Method: A Performance-Based Approach

EMC Proven™ Professional Knowledge Sharing

Andre Rossouw
EMC Corporation®
Table of Contents

Executive summary............................................................................................................... 3
Introduction.......................................................................................................................... 3
  Audience .......................................................................................................................... 3
  Terminology ..................................................................................................................... 3
General best practices ......................................................................................................... 4
  Local Replication ............................................................................................................. 6
  SnapView Snapshots ....................................................................................................... 7
  SnapView Clones ............................................................................................................ 8
  SnapView Clones with SnapView Snapshots ................................................................. 9
Remote Replication ......................................................................................................... 10
  Incremental SAN Copy .................................................................................................. 11
  Incremental SAN Copy with SnapView Clones ............................................................. 12
  MirrorView/S .................................................................................................................. 12
  MirrorView/S with SnapView Snapshots ....................................................................... 13
  MirrorView/S with SnapView Clones .......................................................................... 14
  MirrorView/S with Incremental SAN Copy ................................................................. 14
  MirrorView/A .................................................................................................................. 15
  MirrorView/A with SnapView Snapshots ..................................................................... 16
  MirrorView/A with SnapView Clones .......................................................................... 17
  MirrorView/A with Incremental SAN Copy ................................................................. 17
Conclusions and Recommendations ................................................................................. 18
  Local data replication .................................................................................................... 18
  Remote data replication ................................................................................................ 18
References ......................................................................................................................... 20
Appendix A ........................................................................................................................ 21
Author’s Biography ............................................................................................................ 22

Disclaimer: The views, processes or methodologies published in this article are those of the author. They do not necessarily reflect EMC Corporation’s views, processes or methodologies.
Executive summary

CLARiiON® based replication software offers customers a wide range of choices for local and remote data replication. The customer can usually define the business requirement driving the need for replication software, but may not know the impact of each different product.

This article explores the performance impact of SnapView™, SAN Copy™ and MirrorView™, used either alone or in combination.

Introduction

CLARiiON Replication Software, previously called layered applications, fall into two broad categories: local replication software and remote replication software.

Local replication software makes replicas of LUNs inside a single CLARiiON. A replica is a copy of data which is present in addition to the original data; this necessarily excludes the LUN Migration feature, which moves data from one LUN to another, and destroys the original LUN. Though SAN Copy can be used to make local replicas, it is much more widely used in the remote replication space, and will be discussed there. The local replication discussion will therefore be confined to SnapView.

Remote replication software makes one or more copies of data on a CLARiiON or CLARiiONs in addition to the one on which the original data is stored; SAN Copy and MirrorView fall into this category.

Audience

This article is suitable for EMC customers, EMC partners, and EMC employees who are interested in learning more about the performance impact of various CLARiiON data replication solutions. This article assumes some familiarity with SnapView, SAN Copy, and MirrorView operation and management.

Terminology

Bandwidth – a measure of the amount of data being transferred, expressed as Megabytes per second (MB/s)

Flushing – writing data held in CLARiiON write cache to the LUNs where the data belongs

Forced flushing – flushing that occurs when write cache becomes 100% full. Because all I/O is suspended until space is made for incoming writes, forced flushing adversely affects performance.

Multi-threaded – a process or application which is capable of performing multiple I/O operations at the same time; the LUN to which that I/O is sent will see a queue depth greater than one.

Random – I/O addressed to disk regions that are not adjacent to each other.
**Response time** – the period of time between making a request and fulfilling the request. Response time is measured in milliseconds (ms).

**Sequential** – I/O addressed to disk regions that are adjacent to each other.

**Single-threaded** – a process or application capable of performing only one I/O operation at a time; the LUN to which that I/O is sent will see a queue depth of one.

**Throughput** – a measure of the number of read and write operations per second, expressed as I/O operations per second (IOPs).

**Utilization** – a measure, expressed as a percentage, of how busy a specific component is. If a disk is running at 50% utilization, then we can reasonably assume that is performing to approximately half of its capability.

**General best practices**

The “CLARiiON Best Practices White Paper” is the reference document for performance-related questions. Several best practices mention the use of CLARiiON Replication Software. The utilization of CLARiiON components needs to be carefully considered before you attempt to add the additional I/O load associated with the Replication Software. Physical disks, in particular, should have a utilization of no more than 50% before the Replication Software is added, in order to achieve utilization below 70% after adding Replication Software.

The figures that follow, using SnapView Session as an example, illustrate why this is so important.

![Figure 1. LUN 100 write response time before and after a SnapView Session is started](image-url)
Figures 1 and 2 show LUN 100, a 512 GB LUN on a 4+4 RAID 1/0 RAID Group. LUN 100 is the target of an Iometer test, which is running a single thread of 4 kB random writes at maximum speed. The utilization of all disks is 100%, LUN throughput is 1200 IOPs, with a response time of 0.75 ms. Forced flushing is occurring at a rate of 760 per second.

After starting a single SnapView Session on the LUN (with a single 4+1 RAID 5 LUN in the Reserved LUN Pool), throughput drops to 60 IOPs and response time climbs to 33 ms. After a short interval, throughput has recovered to around 95 IOPs, and response time is 20 ms. If the Session is allowed to run indefinitely, response time will return to around 0.8 ms, and throughput to around 1150 IOPs.

Contrast this to what happens when the same LUNs are used in a test that allows disk utilization to reach only around 50% before the Session is started. Figures 3 and 4 show the result of that test.
Response time, which was around 0.2 ms, increases to 24.4 ms, and then drops almost immediately to 20.5 ms. Write throughput, which was 442 IOPs, drops to 72 IOPS, and soon increases to 86 IOPs. If this SnapView Session runs indefinitely, throughput and response times will return to their original values. No forced flushing occurs at any time during this test.

**Local Replication**

![Local Replication Diagram]

*Figure 4. LUN 100 write throughput before and after a SnapView Session is started*
Local data replication can be performed with Snapshots only, Clones only, or a combination of Clones and Snapshots. Each of these topics will be discussed separately.

**SnapView Snapshots**

SnapView Snapshots use pointer-based technology to create point-in-time images of Source LUN data. Copy On First Write (COFW) copies data chunks which are to be modified for the first time into a dedicated area called the Reserved LUN Pool (RLP). The data chunks are a fixed size – 64 kB.

The RLP consists of one or more Reserved LUNs, which may be of any RAID type, and may be bound on any type and speed of disk.

A Reserved LUN consists of two different areas, the:
- **map area**: where the pointer information is stored
- **data area**: where the original data chunks are stored

Updates to the map area are required to save a chunk in the RLP. Data in the map area is also kept in SP memory to allow faster access. If there are many simultaneous SnapView Sessions on multiple large Source LUNs, SnapView may need to read map data from the Reserved LUNs; these reads are 64 kB in size.

During a COFW operation, the original chunk is read from the Source LUN and written to the Reserved LUN, and writes are made to the Reserved LUN map area. The host write cannot be performed until these operations are completed, which causes response times to increase dramatically as seen in the previous figures.

Note that a Session can be started, and is usually only started, when the point-in-time data is about to be used. This means that the performance price paid for using SnapView Snapshots is paid at the time the Snapshot is used. When the activity on the Snapshot is complete, the SnapView Session can be stopped, and performance will return to the levels before the Session was started.

We will review 4 types of I/O activity related to the use of Snapshots: reads from the Source LUN, writes to the Source LUN, reads from the Snapshot, and writes to the Snapshot. It is assumed that a SnapView Session is running at the time these I/O activities are performed.

**Reads from the Source LUN**: these reads are sent directly to the Source LUN, so there is no direct performance impact. Note, though, that the additional 64 kB reads caused by COFW activity may cause reads to be somewhat slower. The effect will vary based on the level of write activity on the Source LUN.

**Writes to the Source LUN**: A COFW will be performed when a chunk is written for the first time. The response time of that write will increase. SnapView must consult the LUN map to determine if the chunk has been written before. The time is negligible if the map data is memory-resident; if it must be fetched from disk, it will involve a 64 kB read that can increase the write response time by several milliseconds.

SnapView’s internal operation serializes certain I/O operations. The net effect is a reduction in the number of LUN operations that can be performed simultaneously; this increases the queue length, and therefore increases response time.
**Reads from the Snapshot:** The data will be in the Reserved LUN if the chunk being read was previously the subject of a COFW, and will be read from there. The data will be read from the Source LUN if no COFW has been performed on that chunk.

SnapView consults the map for the LUN to determine if the chunk has been the subject of a COFW. If the map data is memory-resident, the time is negligible; if it must be fetched from disk, it will involve a 64 kB read that can increase the response time of the Snapshot read by several milliseconds.

**Writes to the Snapshot:** If the chunk being written was previously the subject of a COFW, the chunk will already be in the Reserved LUN.

A write to the chunk then involves two operations:
1. duplication of the chunk, which involves a 64 kB read and a 64 kB write
2. the write of the host data

If no COFW has been performed on that chunk, then a COFW must be performed to get the chunk into the Reserved LUN, at which time the chunk duplication and host write can occur. SnapView must consult the map for the LUN to determine if the chunk has been the subject of a COFW. If the map data is memory-resident, the time taken is negligible; if it must be fetched from disk, it will involve a 64 kB read; this can increase the response time of the Snapshot write by several milliseconds.

It is apparent that Snapshots can have a large performance impact on the Source LUN if that LUN is the subject of heavy write activity. If the Snapshot is being used in a way that involves a high level of write activity, the impact on the Source LUN can be very severe.

**SnapView Clones**

SnapView Clones are full copies of Source LUN data, written to a separate Clone LUN. Before a Clone is usable, it must contain an identical copy of the data present on the Source LUN. This copy of the data is obtained by performing a full synchronization.

Once the Clone is synchronized, it may be fractured. Fracturing a Clone stops the updating of Clone information from the Source LUN, and allows the host to access the Clone to be made accessible to a host.

When changes are made to the Clone or the Source LUN while the Clone is fractured, they are flagged in the Fracture Log, which is a bitmap kept in CLARiiON SP memory. The Fracture Log is a fixed size; as a result, the area of the LUN that each Fracture Log entry maps to, called an extent, is proportional to the size of the LUN (for LUNs over 32 GB; smaller LUNs have an extent size of 128 kB). There is a Fracture Log associated with each Clone; these Fracture Logs are protected by saving them to the Clone Private LUNs, 2 dedicated LUNs that must be configured before any Clone operations are allowed.

Incremental updates from the Source LUN to the Clone, or from the Clone to the Source LUN can be performed with the Synchronize and Reverse Synchronize (also called Restore) operations. The extents that have been modified, and are flagged in the Fracture Log, are the only areas that will be copied.

In contrast to Snapshots, the performance price paid for the use of Clone data is paid before the data is to be used – the Clone must be synchronized with the Source LUN. This synchronization involves reads of up to 128 kB each from the Source LUN, and writes of the same size to the Clone.
In accordance with best practices, we are assuming that the Clone and Source LUN are on different RAID Groups, the direct impact of writes to the Clone on the Source LUN is therefore negligible. The rate of synchronization, and its performance impact, can be controlled by the Synchronization Rate setting associated with each Clone. Low, Medium and High are allowed rates. The High rate has such a severe performance impact that we do not recommend its use in most production environments; the Medium rate allows moderately fast synchronization while keeping the additional load within reasonable limits.

When a Clone is not in the fractured state, host writes made to the Source LUN are copied to the Clone, and these Clone writes are the same size as the Source LUN writes. Note, though, that the Clone is treated like an independent LUN, so that a host write to the Source LUN is split by SnapView into a write to the Source LUN and a write to the synchronized Clone, and occupies double the number of write cache pages. This indirect effect will be more pronounced if there are many Clones in the synchronized state.

When a Clone is fractured, reads or writes to the Source LUN have no effect on the Clone, and reads or writes of the Clone have no effect on the Source LUN. This makes a Clone very attractive for activities which involve heavy I/O activity, especially a high level of writes.

**SnapView Clones with SnapView Snapshots**

SnapView Clones can act as the Source LUNs for SnapView Snapshots; note that the reverse is not true, and that Snapshots cannot be cloned.

Among are advantages of using Snapshots of Clones: the Snapshot data may be used for testing, and can be modified without affecting the data on the Clone. If you want to keep the modified data, a Snapshot Rollback operation allows the current Snapshot data to be copied back to the Clone. If desired, a Clone Restore can then be used to copy the data back to the Source LUN.

I/O activity on either the Snapshot or the Clone will have no direct effect on the Source LUN. Note that the COFW activity on the Snapshot caused by synchronizing the Clone can result in an increase in write cache activity, and high levels of I/O activity in the Reserved LUN Pool. This activity will be more pronounced with larger LUNs, because the Clone granularity will be larger.

As an example, consider a single 4 kB host write to a 256 GB source LUN. If a Snapshot is running on the LUN, and the host write addresses a new chunk, a COFW will be performed and 64 kB of data will be copied to the RLP.

If a Clone on the Source LUN is fractured at the time the host write is performed, then that extent will be flagged in the Fracture Log. When the Clone is resynchronized, 1 MB of data will be copied from the Source LUN to the Clone. Each unique 64 kB area of that 1 MB will cause a COFW on a Snapshot of the Clone, so a total of 16 COFWs will be performed, and 1 MB of data will be copied into the RLP.
Remote Replication

Remote replication uses SAN Copy and MirrorView software, either alone or in combination. Local copies of LUNs used for MirrorView and SAN Copy can be made with SnapView; we will discuss environments using various combinations of CLARiiON Replication Software.

SAN Copy

SAN Copy migrates data from one location to another. SAN Copy Sessions may be either Full or Incremental.

The full SAN Copy operation consists of reading from the Source LUN, filling data buffers, and writing that data to the Destination LUN. The Destination LUN must be offline; the Source LUN must be offline or must be mounted read-only. If this is not possible, then a point-in-time copy of the Source LUN may be used as the source of the SAN Copy transfer. That point-in-time copy may be either a Snapshot or a Clone; the choice will impact performance.

When a SAN Copy session is run directly off the Source LUN, the performance impact on the host which owns that LUN is likely to be minimal. The host will either not be accessing the LUN at all, in which case there is no impact, or will be reading only. The extra I/O load on the Source LUN therefore consists only of the reads performed to fill the SAN Copy buffers. Read response times may increase dramatically, especially if a throttle value of 8 or higher is used; the reads are large, 512 kB by default, and will be performed rapidly, leaving little opportunity for host reads to occur.
If a point-in-time copy is used as the SAN Copy Source LUN, it may be either a Clone or a Snapshot. The performance impact of the SAN Copy portion of the data transfer will be negligible; the major impact is caused by the SnapView software implementing the point-in-time copy. For a discussion of that impact, see the SnapView Snapshot and SnapView Clone sections.

**Incremental SAN Copy**

Incremental SAN Copy allows a Destination LUN to be incrementally updated with changes made to the Source LUN. It tracks changes made to the Source LUN by using a modified SnapView Session, and copies the data to the Destination LUN from a modified SnapView Snapshot of the Source LUN.

The latter point is important; data is not read directly from the Source LUN, but from the Snapshot, and is copied to the Destination in I/O block sizes which do not exceed 64 kB, the size of a SnapView chunk. Unlike full SAN Copy, Incremental SAN Copy allows the Source LUN to be online at all times.

The modified Session running against the ISC Source LUN may be in either of 2 states: unmarked or marked. When the Session is in the unmarked state, changes made to the Source LUN are flagged in a bitmap located in the map area of the Reserved LUN. This bitmap determines which data will be copied to the Destination LUN in the next update cycle.

When the Session is in the marked state, changes are also flagged to the bitmap; in addition, if a chunk that is flagged to be copied in this update cycle is changed on the Source LUN, a COFW takes place. Note that while the ISC Session will always be in the marked state when transferring data, being in the marked state does not imply that data is being transferred.

An ISC Session can be marked at 12:00 pm, for example, to take a point-in-time copy of the Source LUN data at that time, and the transfer could be started much later, let’s suppose at 8:00 pm. This means that there have been 8 hours of COFW activity before the transfer starts, with the corresponding impact on the Source LUN.
**Incremental SAN Copy with SnapView Clones**

An Incremental SAN Copy (ISC) Session can be in one of 2 states: the unmarked, or normal state, and the marked state. The behavior of ISC with a SnapView Clone depends on the Session’s state.

**Unmarked**: a Clone on the Source LUN will have a moderately high impact on Source LUN response times while it is synchronizing, a slight impact while it is synchronized, and a negligible effect when it is fractured.

A Clone on the Destination LUN will have no effect on the Source LUN, no matter what the Clone state may be.

**Marked**: a Clone on the Source LUN will have a more pronounced effect on Source LUN response times when it is synchronizing, a slight impact while synchronized, and almost no effect when fractured.

If the Destination LUN Clone is synchronizing while an update is in progress, it will cause increased response times on the Destination LUN. This, in turn, will cause the update cycle to take longer, with a corresponding increase in overall COFW activity on the Source LUN. A synchronized Clone on the Destination LUN will have a minimal effect on the Destination LUN, with a correspondingly low effect on the performance of the Source LUN. Again, a fractured Clone will have no effect at all.

**MirrorView/S**

MirrorView/S (MV/S) operates synchronously – writes to the Primary Image are immediately copied to the Secondary Image, and a host write cannot be completed until an acknowledgement has been received from the secondary CLARiiON.
In environments with Fibre Channel connections between primary and secondary CLARiiONs, the latency of the connection is likely to be very small, especially where distances are very short. When a WAN link, with associated FC/IP conversion hardware, is used between the participating CLARiiONs, the link latency becomes much more significant.

Each write to the Secondary Image will require 3 to 4 round trips, and the time taken for each round trip will typically increase by 1 to 2 ms because of the conversion from Fibre Channel to TCP/IP and back. This effect can be reduced if the conversion hardware allows local acknowledgements (the ‘Fast Write’ or ‘SmartWrite’ used in some products), but will still add the latency equivalent to 1 round trip. If the propagation delay due to distance is added, the increase in write response times on the Primary Image will increase by a minimum of 1 ms per 50 miles of separation. MirrorView/S distances do not usually exceed around 75 miles because of this.

The optional Write Intent Log (WIL) feature of MV/S adds a write to the WIL LUN for each write to the Primary Image. This write will be a cached write if we adhere to best practices, but will increase host write response time by a small amount.

**MirrorView/S with SnapView Snapshots**

SnapView Snapshots can be used on either the Primary Image or Secondary Image of a MV/S mirror, with similar effects. As discussed in the section on SnapView Snapshots, COFW activity can increase the response time of Source LUN writes significantly. Because MV/S host writes cannot continue until the secondary CLARiiON acknowledges the write, a Snapshot on the Secondary Image affects host write response times slightly more than a Snapshot on the Primary Image.

This increased write response time will be unacceptable to many host applications; MV/S is not often used with active SnapView Sessions during normal operating hours. The use of Snapshots during relatively quiet periods may be acceptable, especially if only reads are being performed from the Snapshot. Writes to a Snapshot, as noted before, have a high performance cost.

EMC Corporation
MirrorView/S with SnapView Clones

A MirrorView/S mirror can be in one of three states: fractured, synchronizing, or synchronized (the normal state). Each of these states will exhibit different behavior when Clones are used with the mirror.

Fractured mirror: The Primary Image behaves like an ordinary LUN because no I/O is being sent to the Secondary Image. For a discussion of the effect of a Clone on a LUN, see the SnapView Clones section. The Secondary Image will not receive any I/O, so the Clone will have no impact on the mirror.

Synchronizing mirror: A Secondary Image with a synchronized Clone will see a significant increase in write response times, which directly affect the mirror’s performance. If the Clone is synchronizing at the same time the mirror is synchronizing, the effect will be much more pronounced.

Synchronized mirror: Synchronized Clones on the Primary Image or Secondary Image have a slight effect on mirror performance if the CLARiiON is experiencing a moderate I/O load. If the CLARiiON was very busy before the Clones were added, the effect will be more pronounced.

Observe CLARiiON best practices when configuring Clones with MV/S mirrors. A synchronizing Clone will affect performance of a synchronized MV/S mirror indirectly because of the reads that take place from the Clone Source LUN; if the Clone Synchronization rate is Medium or Low, the effect will be insignificant.

MirrorView/S with Incremental SAN Copy

Observe CLARiiON best practices when configuring Clones with MV/S mirrors. A synchronizing Clone will affect performance of a synchronized MV/S mirror indirectly because of the reads that take place from the Clone Source LUN; if the Clone Synchronization rate is Medium or Low, the effect will be insignificant.
Incremental SAN Copy run directly off a MV/S Primary Image or Secondary Image is likely to have a severe impact on response times – see the ISC section for further information. Note: we strongly recommend that ISC not be used directly off a MV/S image.

The only alternative is to use a Clone of the MV/S image as the Source LUN for ISC – the impact on the mirror will then be the same as that seen for a mirror with Clones. The Clone will be fractured before the ISC update is started, so ISC will have no direct effect on the mirror. Bear in mind, though, that the Clone granularity will have a direct effect on the amount of data transferred over the link by ISC.

Another option is to use a Snapshot of the mirror image as the Source LUN of a full SAN Copy Session. Note, though, that this introduces more COFW activity than the use of ISC directly, and is discouraged for the same reasons that the use of Snapshots with MV/S mirrors is discouraged.

**MirrorView/A**

MirrorView/A (MV/A) makes use of Incremental SAN Copy to transfer data from the Primary Image (which acts like an ISC Source LUN) to the Secondary Image (which acts like an ISC Destination LUN), and makes use of a SnapView Session, started on the Secondary Image before an update cycle starts, to protect Secondary Image data in the event of a failed update cycle.
As a result, MirrorView/A acts exactly like Incremental SAN Copy when it is not actively transferring data to the Secondary Image, and acts exactly like Incremental SAN Copy with a Session running on the Secondary Image when it is actively transferring data to the Secondary Image.

Note that there is one important difference between MV/A and ISC – an ISC Session can be marked by the user, and remain marked for an unlimited period of time before the update cycle is initiated, whereas MV/A automatically marks its ISC Session only when an update cycle starts. As a result, MV/A COFW activity on the Source LUN only occurs when absolutely needed, which minimizes the overall performance impact.

**MirrorView/A with SnapView Snapshots**

SnapView Snapshots may be made of MV/A primary or secondary images. The state of the MV/A mirror when the SnapView Session is started on the Primary Image will determine the performance impact of the SnapView Session.

On the primary CLARiiON, MV/A uses an ISC Session to track and transfer data to the Secondary Image; on the secondary CLARiiON, MV/A creates a SnapView Snapshot and Session shortly before an update cycle begins.

In both cases, data copied by MV/A activity is likely to be shared with SnapView. If a Snapshot is created on the Primary Image when the update cycle starts, or on the Secondary Image just before the update cycle starts, little to no extra COFW activity will be seen. This makes the point-in-time copy of the mirror image available for a very small additional cost in performance.

Note that this only holds true if the Snapshots are not written to; write activity to a Snapshot causes additional COFW activity, and therefore performance impact, on the SnapView Source LUNs.
SnapView Clones may be made of primary or secondary MV/A images. The performance impact of Clones depends on a number of factors: CLARiiON utilization, the state of the Clones, and the state of the MV/A mirror.

Clones made of either the Primary or Secondary Image will need to be synchronized initially; as seen in the discussion of Clones. The synchronization activity will involve sequential reads of up to 128 kB from the Clone Source LUN.

There are 2 MV/A mirror states to be investigated: the normal state, when tracking, but no COFW activity, is occurring; and the updating state, when COFW activity will be occurring.

**Normal MV/A state:** Synchronization of a Clone at the Medium rate increases response time by between 30% and 50% on the Primary Image. As noted in the discussion on Clones, the High rate causes such severe performance degradation that we do not recommend its use.

**MV/A updating:** synchronization of a Clone during MV/A updates increases response times by around 30% to 50%. When the Clone is synchronized, the only effect is caused by the additional writes to the Clone LUN, and this will increase Primary Image response times during the first few minutes of the update cycle.
MirrorView/A, like MirrorView/S, does not allow a multi-hop environment. If you need to have replicas of data at two or more remote sites, ISC may be used from the Primary Image or Secondary Image to the second remote site.

Because SnapView only keeps one unique copy of each chunk per session, and shares common chunks across sessions, there is an advantage to using MV/A along with ISC.

MV/A uses a modified SnapView Session and Snapshot. The RLP saves the COFW data generated, and once it is there, it will not need to be copied again. This implies that for an ISC Session started at the same time as a MV/A update (if the ISC Session is on the Primary Image), or shortly before the MV/A update (if the ISC Session is on the Secondary Image), the performance price has already been paid by MV/A, and the ISC Session is almost free (in performance terms).

Conclusions and Recommendations

Local data replication

Most local data replication should be performed with Clones, particularly if the replica will be extensively written. A Clone is usually a better choice if the Source LUN is being written to at a high rate of change, for example if more than around 25% of the Source LUN would be changed during a SnapView Session.

SnapView Snapshots are a good choice if the replica must be available immediately. A Snapshot is an effective choice if the Source LUN is not subject to a high rate of change, and the local replica will not be written to extensively.

Keep in mind, though, that most reads from a Snapshot will be serviced by the Source LUN. A Clone is the best choice if the Source LUN is already too busy to handle the additional read load.

Remote data replication

SAN Copy in full mode moves large quantities of data in the shortest possible time. If the Source LUN must be accessible for normal write operations while the SAN Copy transfer is active, then a Clone of the Source LUN should be used as the source for the SAN Copy transfer.

Using a Snapshot as the source for a full SAN Copy transfer is not the same as using Incremental SAN Copy, even if the full LUN must be copied – Incremental SAN Copy works at a smaller transfer granularity (64 kB), and copying the data will take longer than a full copy.

Incremental SAN Copy is a good choice in environments where rapid, incremental updating of destination LUNs is important, and where LAN/WAN bandwidth between sites is limited. Increased activity on the source CLARiiON and LUN, in the form of COFWs, is the price paid for these advantages.

MirrorView/S is a good choice for Disaster Recovery when used over short distances. If it will be used with local replication software, Clones are a better choice than Snapshots under most circumstances. If MV/S data must be transferred to additional sites, Incremental SAN Copy can be used; carefully consider the performance impact of COFWs. If high performance at the production site is the overriding concern, and LAN/WAN bandwidth is not an issue, then a Clone of the MV/S image may be used as the source for an ISC transfer.
MirrorView/A can be used in environments where high performance of the Primary Image LUNs is not a critical factor. MV/A will allow the achievement of a relatively inexpensive Disaster Recovery solution with a modest bandwidth requirement between sites if the COFW activity does not push performance beyond acceptable limits.

Either SnapView Snapshots for local replication, or Incremental SAN Copy for copying data to a larger number of remote sites are good choices – MV/A has already absorbed the performance cost of the COFW activity.
References

Clone of Mirror Performance Study FLARE Release 24 (Technical Note for CSPEED) 3/26/07


EMC CLARiiON Best Practices for Fibre Channel Storage (Engineering White Paper) November 2005

EMC CLARiiON Fibre Channel Storage Fundamentals (Engineering White Paper) 10/31/2003
Appendix A

Testing

All testing was performed on the following equipment:

3 x CX3-80 CLARiiONs, each with
   30 x 300 GB 10 krpm FC disks
   Write cache set to 256 MB
   Read caches set to 128 MB
   Watermarks set to 60/80, page size 8 kB
   FLARE 3.24.080.5.011

3 x Windows hosts
   2.5 GHz Xeon
   2 GB RAM
   2 x Emulex LP10000 FC HBA

2 x Cisco 9216 switches

All physical disks were zeroed before LUNs were bound to remove the effects of background zeroing. Initial verify was disabled on all LUNs; SNiiFFER was allowed to run because it is a normal part of an operating environment.

RAID types used were 4+1 RAID 5, 4+4 RAID 1/0, 2+2 RAID 1/0, and 1+1 RAID 1/0. LUN sizes were 512 GB, 256 GB, and 32 GB.
Author’s Biography

Andre has worked in the IT industry since CP/M was a state of the art operating system for small computers. His roles have included Technical Support Engineer for a repair center environment, Customer Service Engineer, course developer, and instructor. He has achieved a number of Certifications including several EMC Proven Professional Specialist and Expert level certifications, A+, and Network+. He was previously an MCNE, MCNI and SCO Authorized Instructor. Andre lives in North Carolina with his wife, daughter, and a garden full of squirrels.