

EMC CLARiiON Best Practices for Fibre Channel Storage: CLARiiON Release 26 Firmware Update

Best Practices Planning

Abstract

This white paper presents the best practices for EMC[®] CLARiiON[®] storage systems. The paper acts as a guide to the system settings and configuration options affecting the storage system's performance. For FLARE[®] release 26, this paper includes new information on RAID 6, ALUA, and iSCSI.

September 2007

Copyright © 2006, 2007 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number H2358.2

Table of Contents

Executive summary	5
Introduction	5
Audience	5
What's new.....	5
Considerations for performance	5
Application design	6
Optimizing for sequential or random I/O	6
I/O size	6
Temporal patterns and peak activities	6
Application buffering, concurrency, and asynchronous I/O.....	6
Host file-system effects.....	7
File-system buffering and coalescing.....	7
Minimum I/O size: The file-system request size.....	7
Maximum I/O size.....	7
File-system fragmentation	7
File-system alignment	8
Correcting alignment problems	9
Linux I/O fragmenting	12
Volume managers.....	12
Plaid do's.....	12
Plaid don'ts.....	12
Plaids for high bandwidth	13
Plaids and OLTP	13
Host HBA effects.....	13
HBA limitations	14
PowerPath.....	14
Queuing, concurrency, queue-full	14
MetaLUNs	15
Comparison of metaLUNs and volume managers	15
MetaLUN usage and recommendations.....	16
MetaLUN expansion strategies	18
Storage controller effects	21
CLARiiON storage processors	21
RAID level resources and performance	24
LUN Migration.....	27
Rebuilds.....	28
Hot spares.....	30
Proactive sparing	30
Fastbind	30
The RAID engine cache.....	31
Cache size and performance	31
Scripting cache settings	31
Cache settings.....	31
Balancing load between SPs	33
The back end	33
UltraPoint, DAE2, and SATA-DAE	33
LUN distribution	34
Vault and PSM effects.....	34

Using LUN and RAID group numbering	36
Minimizing disk contention	36
Stripes and the stripe element size	37
CLARiiON RAID 5 stripe optimizations	37
Number of drives per RAID group	37
How many disks to use in a storage system	37
Disk types, speeds, and capacities	38
Considerations for reliability and redundancy	40
Highly available attach	40
Binding RAID groups across buses and DAEs	40
Binding across DAEs	40
Binding across back-end buses	41
Binding with enclosure 0 drives	41
Consistency of data replication	42
ALUA	42
Path management	42
Usage	43
Host considerations	43
OS considerations	44
Definition of columns in table:	44
Performance considerations	44
Sizing the storage requirement	45
Capacity planning	45
Vault drives	45
Actual drive capacity	45
Parity or mirrored protection	45
Performance planning	45
Rule-of-thumb approach	45
Performance Estimate Procedure	46
Resolving performance and capacity needs	48
Sizing example	48
Step 1: Determine the workload	48
Step 2: Determine the required I/O disk load	49
Step 3: Determine the number of drives required	49
Step 4: Determine number and type of storage systems	49
Other considerations	50
CLARiiON software update	50
Conclusion	50
References	50

Executive summary

This white paper presents best practices for achieving high performance and reliability as they pertain to EMC® CLARiiON® Fibre Channel storage systems. These practices may need to be modified and/or tuned for particular storage system I/O workloads.

This paper also explains how to size a CLARiiON storage system for performance and capacity. Additional tuning may be necessary for some storage system I/O workloads.

Introduction

This white paper outlines factors affecting the performance and availability of CLARiiON Fibre Channel block-level storage for both a storage area network (SAN) and direct-attached storage (DAS). The various components in the system that affect *performance* and *availability* are addressed in succession—from the application, to the host, to the storage system itself.

Fundamental concepts that support the conclusions in this white paper can be found in the *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper available on [Powerlink](#)®, EMC's password-protected extranet for customers and partners.

This paper follows the same format as the fundamentals white paper, enabling the latter to be used as a reference. Readers who are unfamiliar with the fundamentals paper are encouraged to read it.

Be sure to read the entire section for each recommendation. Many who have read the first sentence or two in a section and then stopped have become needlessly confused. Most topics have background in the fundamentals paper, and topics that are extended here will be accompanied by explanations.

Audience

The intended audience for this white paper is technical personnel who require guidance with the best approaches to implementing CLARiiON Fibre Channel storage. An understanding of the basics of mirrored and parity RAID is required, as is knowledge of CLARiiON fundamentals.

What's new

This is the best practices for release 26. Major changes to this paper include information about:

- ALUA
- RAID 6
- iSCSI replication

A number of minor changes have also been made to other sections; readers are encouraged to review the entire paper.

Considerations for performance

The default settings for a CLARiiON storage system are designed to provide optimal performance for the largest number of workloads encountered in the real world. However, some workloads require tuning to make the best use of the CLARiiON storage system.

This white paper discusses many factors to consider when designing a system for optimal performance. Please note that the information provided here (including descriptions, rules of thumb, configuration choices, and ranges of behavior) is more useful if you have a full understanding of the workload being applied to your CLARiiON. For this reason, EMC recommends that you use the Navisphere® Analyzer to analyze the workload on a regular basis. Also, remember that there are rarely absolute answers to design choices when configuring a storage system, so when in doubt accept the defaults and use conservative estimates.

Application design

The application design determines much of the behavior of the system. The first best practice to follow is application tuning. No amount of storage system tuning will make up for a badly designed application.

Optimizing for sequential or random I/O

Typically, tuning for sequential I/O pays off in improved bandwidth since the storage system is more efficient with sequential I/O—especially with RAID 5. Tuning for random I/O improves throughput and lowers response times, which improves transaction response times for users.

Writes versus reads

The efficiency of the CLARiiON writes and reads depends on whether the workload is characterized by sequential or random I/O.

Writes consume more CLARiiON resources than reads. Writes to the write cache are mirrored to both storage processors (SPs). Writes to RAID groups with parity (RAID 3, 5, and 6) require the parity calculation, which is the redundant information written to disks. Writes to mirrored RAID groups (RAID 1 and 1/0) require two copies of the data to be written. The overhead of these data protection mechanisms affects the throughput of write operations.

Reads generally consume fewer CLARiiON resources. Reads that find their data in cache (cache hit) consume the least amount of resources and have the highest throughput. However, reads that are not found in cache (cache miss) have a much higher response time than writes because the data has to be retrieved from disks. Workloads characterized by random I/O have the highest percentage of cache misses. Whether a read operation is a cache hit or miss affects the throughput of read operations.

Generally, workloads characterized by sequential I/O have the highest write and read throughput. Sequential writes have lower SP and disk overhead than random writes. Sequential reads have a greater likelihood of a cache hit and of not requiring a disk read.

I/O size

Every I/O has a fixed and a variable resource cost that chiefly depends on the I/O size. For the purposes of this paper, up to 8 KB I/Os are considered small, and greater than 128 KB I/Os are large. Doing large I/Os on a CLARiiON delivers better bandwidth than small-block I/Os. If sufficient drives are present when executing large I/Os, the back-end bus speed is the limiting performance factor. Small-block random access applications such as OLTP bottleneck at the disk drives and rarely achieve back-end bus rates. When designing for OLTP, it is important to use IOPS ratings that are disk-based, not bandwidth ratings that are bus-based.

However, on a CLARiiON storage system, both write caching and pre-fetching are bypassed when I/O reaches a certain size. The decision to use a large request or break it into smaller sequential requests depends on the application and its interaction with the cache. These interactions are discussed in “The RAID engine cache” section.

File system settings can affect I/O sizes as well, as described in the “Host file-system effects” section.

Temporal patterns and peak activities

The operational design of the application—how it is used, when it is used, and when backups occur—affects the storage system load.

For example, storage for a random-access application requires good sequential performance during backups and batch processes.

Application buffering, concurrency, and asynchronous I/O

Application buffering is typically intelligent and offers fast response times. Read EMC Best Practices Planning white papers for Oracle, Microsoft Exchange, and Microsoft SQL Server to determine how to configure these applications for the best effect.

Generally, for OnLine Transaction Processing (OLTP) and messaging applications (applications with a high rate of random access I/O), higher concurrency is better. The storage system gains high throughput when there is a lot of concurrency. Using asynchronous I/O is a common technique to achieve high concurrency. For bandwidth, single-threaded applications are rarely able to make efficient use of more than four effective drives, unless request sizes are greater than 2 MB and volume managers are used. While the best sequential performance is achieved when the sequential process has exclusive access to the drives, the user may still get acceptable sequential performance from Fibre Channel drives with a concurrent random access of fewer than 100 IOPS per drive.

Host file-system effects

At the host level, file systems also affect application I/O characteristics by determining the minimum and maximum I/O request sizes.

File-system buffering and coalescing

Similar to caching on the storage system, buffering is one of the means by which file systems enhance performance. Coalescing is the joining together of many separate file operations into a single larger operation.

Buffering

In most cases, file-system buffering should be maximized, as it reduces load on the storage system. There are, however, some exceptions to this rule.

Traditionally, applications that do their own buffering avoid—or work around—file-system buffering. This is due to the assumption that the application can buffer more intelligently. Also, by avoiding the file system's coalescing, the application has more control over the I/O response time. However, as RAM in 64-bit servers grows to 32 GB and larger, it becomes possible to buffer the entire file system. This can greatly reduce response times for reads, which are buffered. (Writes should use a write-through feature to ensure persistence of committed data.)

Coalescing

File-system coalescing can assist in getting high bandwidth from the CLARiiON, although write cache often makes alignment irrelevant. With write coalescing and full stripe writes, a write can be written to the disks in a single operation when it is properly aligned and is the same size or larger than the RAID stripe. Existing data on the disk is not read, and a recalculation of parity is not needed. Not reading the previous data and parity makes this operation faster.

In most sequential-access operations, use the `maximum contiguous` and `maximum physical` file-system settings (when available) to maximize file-system coalescing.

For example, this technique can be used with backup programs to coalesce 64 KB writes into full-stripe writes, which are very efficient with parity RAID when the write cache can be bypassed.

Minimum I/O size: The file-system request size

File systems often can be configured for a minimum extent size, such as 4 KB, 8 KB, or 64 KB, which is the smallest indivisible request given to the array. Applications that perform I/O at sizes that are smaller than the extent size may cause unnecessary data movement or read-modify-write activity. It is best to consult application and file system documentation for the optimal settings. Raw partitions, whose request sizes are not limited by a file system, do not have this constraint.

Maximum I/O size

If the goal is to move large amounts of data quickly, then a larger I/O size (64 KB and greater) will help. The array is very efficient at coalescing sequential writes in cache to full stripes on the RAID groups, as well as prereading large sequential reads. Large I/O sizes are also critical in getting good bandwidth from host-based stripes since they will be broken into smaller sizes according to the stripe topology.

File-system fragmentation

Fragmented file systems decrease the opportunity for sequential I/O, which reduces overall throughput.

Avoid fragmentation and defragment the file system on a regular basis. Note that if NTFS file systems are formatted at anything but the default extent size, they *cannot* be defragmented with most tools: The API does not allow it. Performing a *file-level* copy (to another LUN or by executing a backup and restore of the file system) is an effective approach to defragmenting.

File-system alignment

File-system misalignment affects performance in two ways:

1. Misalignment causes disk crossings: an I/O broken across two drives (where normally one would service it).
2. Misalignment makes it hard to stripe-align large uncached writes.

The first case is more commonly encountered. Even if the disk operations are buffered by cache, the effect can be detrimental, as it will slow flushing from cache. Random reads, which by nature require disk access, are also affected, both directly (waiting for two drives in order to return data) and indirectly (making the disks busier than they need to be).

A common example is seen in Figure 1. Intel-based systems are misaligned due to metadata written by the BIOS. In an aligned system, the 64 KB write would be serviced by a single drive.

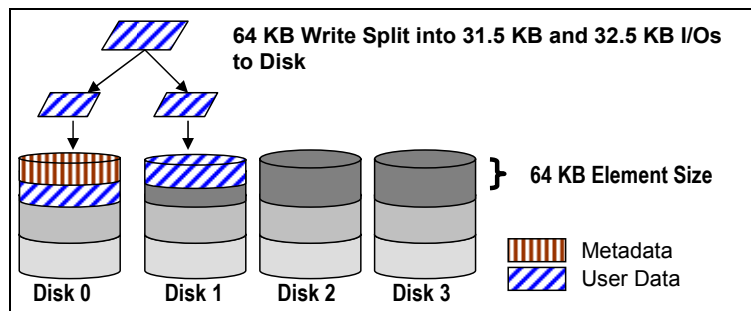


Figure 1. Effect of misalignment with a 63-block metadata area

Disk crossings with small I/O can be significant with some host types due to the issues discussed in the following sections.

Disk crossings have a noticeable effect on response time when a) there is a high percentage of random I/O with block sizes greater than 16 KB and b) the Average Busy Queue Length reported by Navisphere Analyzer for the drives is greater than 4. Workloads that are predominantly aligned to 4 KB or 8 KB boundaries (for example, Exchange and Oracle) will gain a bit of advantage from alignment but because less than 6 percent (for 4 KB) or 12 percent (for 8 KB) of the requests cause crossings (and chances are that they will be completed in parallel anyway) the gains may be barely noticeable in practice. But if and when a specific file system and/or application encourages the use of an aligned address space and the offset is declared, EMC recommends that an operating system disk utility be used to adjust the partitions. The Navisphere LUN bind offset facility should be used with caution, since it can affect layered application sync rates adversely.

Alignment on Intel architecture systems

Intel architecture systems, including Windows 2000/2003, are affected by the placement of metadata on the LUN, which causes the disk partition to be misaligned. This is due to legacy BIOS code, which uses cylinder, head, and sector addressing instead of LBA addressing.

This issue affects Linux systems using Intel architecture, as well as Microsoft Windows NT, 2000, and 2003 systems. This issue also affects VMware systems running on Intel hardware.

The `fdisk` utility, as well as the Windows Disk Manager, places a Master Boot Record (MBR) on each formatted device. An entry in the MBR specifies 63 *hidden sectors* on the device. The addressable partition begins immediately after these 63 hidden sectors. This causes subsequent data structures to become misaligned with respect to the CLARiiON RAID stripe.

On Linux systems, the hidden sector amount depends on the boot loader and/or disk manager software used, but a 63 sector record is most commonly encountered. On VMware, the offset is 63.

In any case, the result causes misalignment for a predictable percentage of I/O. Larger I/O sizes are most affected. For example, assuming the default CLARiiON stripe element size of 64 KB was used, all I/O of 64 KB would cause disk crossings. For I/O smaller than the stripe element size, the percentage of I/O that causes a disk crossing can be computed with this equation:

$$\text{Percentage of data crossing} = (\text{I/O size}) / (\text{stripe element size})$$

The resulting percentage will give you a rough idea of the cost paid to run misaligned. The cost is higher if the cache becomes full.

Correcting alignment problems

You can choose *one* of the following approaches to fix alignment. Use only one of these two methods:

- The Navisphere LUN alignment offset
- A partition utility

Use only one method, not both, for any particular LUN. This statement is repeated intentionally.

Also, when setting up a metaLUN, only the base *component* requires stripe alignment. If using the LUN alignment offset, the metaLUN alignment offset is applied when the metaLUN is created. When expanding a metaLUN, no further adjustment is needed. When using partition methods, the adjustment need only be made when the user partitions the LUN for the first time.

Which method to use

Use the LUN alignment offset method if no host-based utility is available. The LUN alignment offset method causes disk and stripe crossings with some replication software operations: clone sync I/O, SnapView™ Copy On First Write operations, MirrorView™ sync I/O, SAN Copy™ I/O.

Use a host-based partition utility if available.

Avoid using the LUN alignment offset method if SnapView, SAN Copy, or MirrorView will be used on this LUN. Instead, use a host-based disk utility.

The LUN offset

The LUN offset method aligns the *partition* on the *stripe* boundary by *offsetting* the LUN. The LUN starts near the end of the first RAID stripe. In other words, the LUN is offset to the RAID stripe, but the file system (which begins after the MBR) is aligned, as shown in Figure 2.

The downside of the LUN alignment offset is that it *misaligns* the I/O requests made by any software that operates on the raw LUN. CLARiiON replication products operate on the raw LUN, and they will incur disk crossings if the LUN has been offset.

The offset is specified in Navisphere when the LUN is bound, and the value is designated in 512-byte blocks. *Set the LUN alignment offset value to the amount by which your data is offset.* For instance, on a Windows 2003 system, use 63 blocks as the offset value. FLARE® will adjust the stripe so that user data starts at the beginning of the stripe (Figure 2).

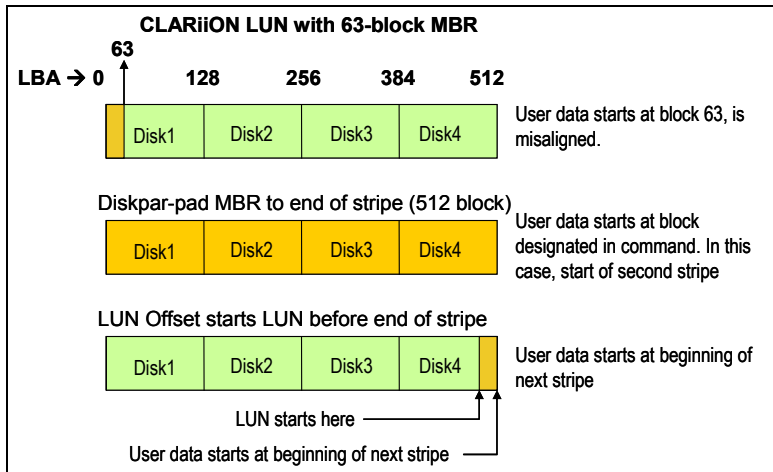


Figure 2. Intel MBR with partition and LUN offset correction

Disk partition alignment

Host-based utilities correct the alignment by padding the beginning of the addressable area so that addressable space begins at the start of a RAID stripe element, or at the beginning of an entire stripe. Since the LUN starts normally, at the beginning of the RAID stripe, replication software operations are also aligned. In fact, for mirroring operations, the alignment of the primary is preserved when the secondary is written, as the padded partition table is written to the destination LUN.

Disk partition alignment and Windows systems

In Windows NT, 2000, and 2003 systems, the utility `diskpart.exe`—part of the Windows Resource Kit—can be used to set the starting offset of a partition. You must do this before data is written to the LUN as `diskpart` rewrites the partition map: All existing data on the LUN will be made unavailable.

For random-access operations or for metaLUNs, set the starting offset in `diskpart` to the stripe element size used to bind the LUN (typically 128 blocks). For high-bandwidth applications, set the starting offset equal to the LUN stripe size.¹

First, obtain the drive number by using Disk Manager. Check the existing starting offset using `diskpart` with the `-i` switch: `diskpart -i x` (where `x` is the drive number) from the command line:

```
C:\>diskpart -i 0
---- Drive 0 Geometry Information ---- <deleted for brevity>
---- Drive Partition 0 Information ----
StatringOffset = 32256
PartitionLength = 40007729664
HiddenSectors = 63
. . .
```

Note the `HiddenSectors` value. This is the amount by which the partition is offset.

1. If disk `X` has data that you don't want to lose, back up that data.
2. If disk `X` is a raw drive, skip to step 4.
3. Delete all partitions on disk `X`, making it a raw disk.
4. Issue `diskpart -s x` (where `x` is the drive number) from the command line.

¹ Refer to “Stripes and the Stripe Element Size” in the *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper for a definition of stripe size.

-
5. Enter the new starting offset (in sectors) and the partition length (in MB). This step writes the new offset to the MBR for that drive and creates the partition. After you enter the starting offset and partition size, the MBR is modified and the new partition information appears.
 6. Issue `diskpar -i x` (where `x` is the drive number) at the command prompt to review the information on the newly created partition.

64-bit Windows systems

In 64-bit Windows systems, MBR-type disks are aligned when created with the defaults; GPT partitions are also aligned by default, though they have a small (32 MB) reserved area that is not aligned.

Disk partition adjustment for Linux systems

In Linux, align the partition table before data is written to the LUN, as the partition map will be rewritten and all data on the LUN destroyed. In the following example, the LUN is mapped to `/dev/emcpowerah`, and the LUN stripe element size is 128 blocks. Arguments for the `fdisk` utility are as follows:

```
fdisk /dev/emcpowerah
x # expert mode
b # adjust starting block number
1 # choose partition 1
128 # set it to 128, our stripe element size
w # write the new partition
```

This method is preferable to the LUN alignment offset method for LUNs that will have a snapshot, clone, or MirrorView image made of them. It is preferred for SAN Copy sources and targets as well.

Disk partition adjustment for VMware

VMware is more complex as there are two cases.

When aligning raw disks or Raw Device Mapping (RDM) volumes, the alignment is done at the Virtual Machine (VM) level. For example, on Windows VMs use `diskpar` to perform the alignment.

For VMFS volumes, the alignment will be done at the ESX Server level using `fdisk`, as well as at the VM level. This is because both the ESX Server *and* the clients will put MBRs on the LUNs. The ESX must align the VMFS volume, and the client systems must align their virtual disks.

To align the ESX Server:

On service console, execute `"fdisk /dev/sd<x>"`, where `sd<x>` is the device on which you would like to create the VMFS

Type `"n"` to create a new partition

Type `"p"` to create a primary partition

Type `"1"` to create partition #1

Select the defaults to use the complete disk

Type `"x"` to get into expert mode

Type `"b"` to specify the starting block for partitions

Type `"1"` to select partition #1

Type `"128"` to make partition #1 to align on 64KB boundary

Type `"r"` to return to main menu

Type `"t"` to change partition type

Type `"1"` to select partition 1

Type `"fb"` to set type to fb (VMFS volume)

Type `"w"` to write label and the partition information to disk

By declaring the partition type as `fb`, ESX Server will recognize the partition as an unformatted VMFS volume. You should be able to put a VMFS file system on it using the MUI or `vmkfstools`. Next, the

virtual disks for each VM must be aligned. For Linux VMs, follow the procedure listed above. For Windows VMs, use the procedure for Windows, above.

Linux I/O fragmenting

With Linux, it is important to avoid concurrent access to multiple large files on a LUN. Otherwise, this can cause many requests from different threads to use different pseudo-devices to access the same underlying device. This interference reduces the coalescing of writes. It is better to use many smaller LUNs, each with a single large file

Dynamic LUN migration and offsets

If you have used a host-based partition tool to align data, that alignment will be preserved when you migrate LUNs. This assumes the LUN stripe element size is the same between LUNs. If the Navisphere Bind Offset is used on the source of the migration, the target LUN must have the Bind Offset applied when it is bound in order to maintain stripe alignment.

Volume managers

The primary performance effect of volume managers is the manner in which CLARiiON LUNs are striped (known as a *plaid* or *stripe on stripe*).

Avoid using host-based RAID implementations that require parity (for example, RAID 3, RAID 5). This consumes host resources for a service (parity protection) that is better handled by the storage system.

Figure 3 shows three different plaid techniques that are discussed in this section.

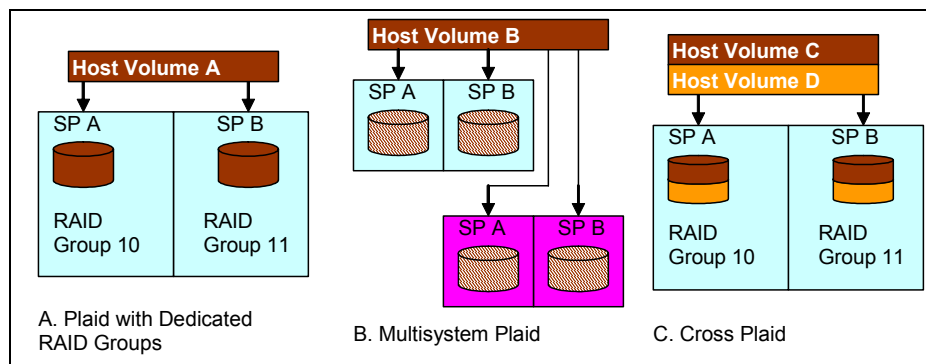


Figure 3. Plaid types

In all cases, the same rules apply:

Plaid do's

Set the host manager stripe depth (stripe element) equal to the CLARiiON LUN stripe size. You can use an integer multiple, but it is best to keep the stripe element at 1 MB or lower. For example, if the CLARiiON stripe size is 512 KB, make the host stripe element 512 KB or 1 MB.

For simplicity, build host manager stripes from base CLARiiON LUNs.

Use LUNs from separate disk groups; the groups should have the same geometry (stripe size, disk count, RAID type, and so on).

Plaid don'ts

Don't stripe multiple LUNs from the same RAID group together. This only causes large disk seeks. If you need to combine multiple LUNs from one disk group, concatenate contiguous LUNs—do not use striping.

Don't make the host stripe element less than the CLARiiON RAID stripe size.

Don't plaid together LUNs from RAID groups with differing RAID types, stripe sizes, or radically different drive counts. The result is not catastrophic, but it is likely to give uneven results.

Plaids for high bandwidth

Plaids are used in high-bandwidth applications for several reasons:

Plaids can increase concurrency (parallel access) to the storage system.

Plaids allow more than one host HBA and CLARiiON storage processor (SP) to service a single volume.

Very large volumes can be split across more than one CLARiiON system.

Increasing concurrency

Plaids are useful when applications are single-threaded (that is, reading a single large file). If the application I/O size fills the volume manager stripe, the volume manager can access the LUNs that make up the volume concurrently.

Distributing across storage processors

A plaid allows you to distribute a high-bandwidth load across both SPs. (Only one storage processor can service a single LUN or metaLUN.) Configuration A in Figure 3 shows this.

Note that you cannot achieve full bandwidth to multiple storage-system ports unless you are using multiple HBAs on the host. To fully drive each active path to a CLARiiON SP, you will need an HBA on the host. For example, if you plan to drive I/O simultaneously through two ports on SP A and two ports on SP B, you will need four HBAs on the host to achieve maximum bandwidth.

Distributing across storage systems

Spanning storage systems, as shown in configuration B in Figure 3, is suggested *only* when file-system sizes and bandwidth requirements warrant such a design. For example, a 30 TB geological information system database that requires write bandwidth that exceeds the write-bandwidth limits of one array is a candidate for a multi-system plaid. Note that a software upgrade or any storage system fault—such as deactivation of the write cache due to a component failure on one storage system—affects the entire file system.

Plaids and OLTP

OLTP applications are hard to analyze and suffer from hot spots. Plaids are an effective tactic to distribute I/O across many spindles. An application that can keep many drives busy benefits from a high drive count.

Note that some volume managers recommend small host stripes (16 KB to 64 KB). This is not correct for CLARiiON LUNs, which use a striped RAID type. For OLTP, the volume manager *stripe element* should be set to the CLARiiON *stripe size* (typically 128 KB to 512 KB). The primary cost of a plaid for OLTP purposes is that most users end up with a cross plaid.

The cross plaid

Disks—and thus disk groups—are getting larger, and users usually end up with several host volumes being built from the same CLARiiON RAID groups (a *cross plaid*—see configuration C in Figure 3).

The rationale for this design is that a burst of random activity to any one volume group is distributed over many drives. The downside is that determining interactions between volumes is extremely difficult.

However, a cross plaid may be effective when:

- I/O sizes are small in size (8 KB or less) and randomly accessed.
- The volumes are subjected to bursts at different times of the day, not at the same time.

Host HBA effects

The topology used for host attach depends on the goals of the system. High availability requires dual HBAs and dual paths to storage. The performance effect of dual pathing lies mainly in the ability of the storage system administrator to balance load across CLARiiON resources.

Keep in mind HBA and driver behaviors when tuning a storage system. The EMC E-Lab™ provides suggested settings for drives and firmware, and these suggestions should be followed.

HBA limitations

The HBA firmware, the HBA driver version used, and the operating system of the host can all affect the maximum I/O size and the degree of concurrency that is presented to the storage array.

PowerPath

If available on the OS, PowerPath® should always be used—whether for a single-attach system through a switch (which allows host access to continue during a software update) or in a fully redundant system.

In addition to basic failover, PowerPath allows the host to connect to a LUN via multiple SP ports—a technique referred to as *multipathing*. PowerPath optimizes multipathed LUNs with load-balancing algorithms. PowerPath offers several load-balancing algorithms, but the default—ClarOpt—is recommended. ClarOpt adjusts for number of bytes transferred, as well as queue depth.

Hosts connected to all current CLARiiON models can benefit from multipathing. Direct-attach multipathing requires at least two HBAs; practical SAN multipathing requires two HBAs, each zoned to more than one SP port. The advantages of multipathing are:

- Failover from port to port on the same SP, maintaining an even system load
- Load balancing across SP ports and host HBAs
- Higher bandwidth attach from host to storage system (assuming the host has as many HBAs as paths used)

While PowerPath offers load balancing across all available active paths, this comes at some cost:

- Some host CPU resources are used during both normal operation, as well as during failover.
- Every active and passive path from the host requires an initiator record, and there are a finite number of initiators per system.
- Active paths increase time to fail over in some situations (PowerPath tries several paths before trespassing a LUN from one SP to the other.).

Because of these factors, active paths should be limited, via zoning, to two storage system ports per HBA for each storage system to which the host is attached. The exception is in environments where bursts from other hosts that share storage system ports are unpredictable and severe. In this case, four storage system ports per HBA can be used.

Finally, see the “ALUA” section of this document for a further discussion of path configuration.

Queuing, concurrency, queue-full

A high degree of request concurrency is usually desirable, and results in a good return on investment. However, if an array’s queues are *too* large, it will respond with a **queue-full** flow control command. The CX3 front-end port drivers return a **queue-full** status command under two conditions:

- The total number of concurrent host requests at the port is 1,984. (Internally, the port value is 2048, but 64 requests are reserved for special commands.)
- The total number of requests for a given LUN at a given port is $(14 * (\text{the number of data drives in the LUN}) + 32$.

The host response to a queue-full is HBA-dependent, but it typically results in a suspension of activity for more than one second. Though rare, this can have serious consequences on throughput if this happens repeatedly.

The practical design limit on concurrency at a front-end port is about 1,600 requests, not 1,984, which allows for ample burst margin. In most installations, the maximum load can be determined by summing the possible loads for each HBA that can access the port and adjusting the HBA LUN settings appropriately. (Some operating system drivers permit limiting the HBA concurrency on a global level regardless of the individual LUN settings.) In complex systems comprised of many hosts, HBAs, LUNs, and paths, it may be difficult to compute the worst-case load scenario (which may never occur in production anyway). In

this case, use the default settings on the HBA and if queue-full is suspected, use Navisphere Analyzer (release 24) to determine if your system is in a queue-full condition by following the steps described below.

HBA queue depth settings usually eliminate the possibility of LUN generated queue-full. For instance, a RAID 5 4+1 device would require $(14 \times 4) + 32$ or 88 parallel requests before the port would issue queue-full. If the HBA queue-depth setting is 32, then the limit will never be reached. A common exception is for RAID 1 (or RAID 10 1+1): For example, if the HBA queue-depth default setting was altered to a larger value (such as 64) to support greater concurrency for large metaLUNs owned by the same host, the RAID 1 device could reach queue-full because its limit is $(1 \times 14) + 32$ or 46 requests.

Queue-full is never generated as a result of a disk drive's queue-depth.

Port statistics are collected for Navisphere Analyzer; this data includes several useful statistics for each individual port on the SP:

- Port queue-full – a counter has been added that shows the number of queue-full signals that have been issued due to port queue overflow (about 1,800 requests)
- Per-port bandwidth and IOPS

Usage

- A high degree of concurrency is desirable, but too much concurrency can result in high drive response times or a queue-full condition.
- Consider all hosts connected to the array, and which LUNs they access.
- If necessary, set HBA throttles to limit concurrent requests to each LUN. Depending on the operating system, you may be able to globally limit the total outstanding requests per HBA or per target. Set the HBA throttles of hosts that share a port so that the total cannot exceed the port queue limit (rule of thumb, 1,600). Keep in mind that multipathing to one LUN via multiple ports on the same SP may multiply concurrency by the port count, because each port counts as a different target.
- If high concurrency is suspected and performance problems occur, check Analyzer port statistics for queue-fulls, and reduce HBA throttles if appropriate.

MetaLUNs

MetaLUNs are a standard feature of the CLARiiON. There are several topics that address when and how to use a metaLUN.

Comparison of metaLUNs and volume managers

On a CLARiiON storage system, metaLUNs are implemented as a layer above the RAID engine, and are functionally similar to the application of a volume manager on a host. However, there are some important distinctions between metaLUNs and a volume manager.

Single SCSI target versus many

To create a volume manager stripe, all of the component LUNs must be made accessible to the host. MetaLUNs require only a single SCSI LUN to be mapped to the host; the host does not see the multiple LUNs, which make up the metaLUN. This benefits the administrator in several situations:

- In hosts with limited LUNs available due to OS limits
- In hosts where adding LUNs causes a renumbering of SCSI devices; often a kernel rebuild is necessary to clean up the device entries

In these cases, using a metaLUN instead of a Volume Manager simplifies administration on the host.

No volume manager

Not all operating systems have volume manager support. Microsoft Windows Server 2000/2003 clusters using Microsoft Cluster Services (MSCS) cannot make use of dynamic disks. MetaLUNs are a solution for providing expandable, striped, and concatenated volumes for these systems.

Replication of the volume

If the volume is to be replicated on the storage system using SnapView, MirrorView, or SAN Copy, a useable image requires consistent handling of splits. A metaLUN will simplify replication.

Volume access sharing

When a striped or concatenated volume must allow shared access between hosts, and a volume manager will not permit shared access, a metaLUN can be used. The metaLUN is placed in both hosts' storage groups.

Storage processor bandwidth

An important distinction between a volume manager volume and a metaLUN is that a metaLUN is addressed entirely by one storage processor on one CLARiiON storage system. If very high bandwidth is required for a single volume, a volume manager is still the best approach, as the volume can be built from LUNs on different SPs. A volume manager allows the user to access storage at an aggregate bandwidth of many storage processors.

Volume managers and concurrency

As pointed out in the "Plaid for high bandwidth" section, the use of a host-striped volume has the effect of multithreading large requests (those that consist of more than one volume stripe segment). This increases concurrency to the storage system. There is no multithreading effect with a metaLUN, as the multiplexing of the component LUNs is done on the storage system.

MetaLUN usage and recommendations

The three types of metaLUNs are striped, concatenated, and hybrid. This section presents general recommendations. For those who want more detail, the following subsections address strategies for creating metaLUNs and the relative merits of each type.

When to use metaLUNs

Given the previous volume manager discussion, you should use a metaLUN in the following cases:

- When massive storage consolidation is necessary (many drives per volume are needed)
- When LUN expansion is a requirement

You can control the following factors when setting up a metaLUN: component LUN type, metaLUN type, and stripe multiplier.

Component LUN type

The LUN type you bind for inclusion in a metaLUN should reflect the I/O pattern expected for the metaLUN. For instance, match the I/O pattern with the recommendations made in this white paper for different RAID types. ("RAID level resources and performance" provides more information.)

When binding component LUNs, follow these rules:

- Always use the default stripe element size (128 blocks) when binding LUNs for use in metaLUNs.
- Always activate read and write cache.
- Ensure that the write-aside size for component LUNs is 2048. (Write-aside is covered in "The RAID engine cache.")
- Avoid using LUNs from RAID 5 groups of fewer than four drives (in other words, use 3+1 or larger).
- Use RAID 1/0 groups of at least four drives (the new 1+1 is not a good choice for metaLUNs).
- Do not use the component LUN offset to adjust for stripe alignment. MetaLUNs have their own offset value.

MetaLUN type

In general, you should use striped metaLUN components wherever possible as they yield the most predictable performance. Concatenation of a single LUN to a metaLUN is intended for convenience; this may be appropriate for expanding a volume that is not performance-sensitive.

Hybrid metaLUNs combine concatenation with striping. The approach is used to overcome a cost of striped expansion (it is slow). A striped metaLUN can be expanded by *concatenating another striped component*.

This preserves the predictable performance of a striped component, and allows you to expand a striped metaLUN without restriping existing data (performance is affected while the restriping operation is under way). Figure 4 illustrates this point.

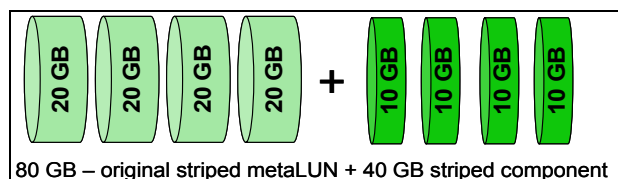


Figure 4. Hybrid-striped metaLUN

Ideally, the LUNs in the expansion stripe set are distributed over RAID groups of the same RAID type and geometry as the original striped component. The most direct way to achieve this is to use the same RAID groups as the base component. The RAID groups are expanded first, in order to make space available. This approach is illustrated in “MetaLUN expansion strategies.”

RAID group expansion is more efficient and has far less influence on host performance than a metaLUN restripe (set the process to medium priority).

MetaLUN stripe multiplier

The stripe multiplier determines the metaLUN stripe segment size:

$$\text{stripe multiplier} * \text{base LUN stripe size} = \text{metaLUN stripe segment size}$$

The metaLUN stripe segment size is the *largest I/O any component LUN will receive*.

Both high bandwidth performance and random distribution call for metaLUN stripe elements of about 1 MB. Also, the underlying RAID groups may be expanded. We want to make sure the metaLUN stripe element is big enough to write full stripes to expanded component LUNs (Table 1).

Use these rules to set the stripe multiplier:

- Unless using RAID 0, use groups of at least four disks to make up the RAID groups hosting the component LUNs.
- Determine the number of effective disks for the group size chosen. For example, six-disk RAID 1/0 is 3 (3+3). Five-disk RAID 5 is 4 (4+1).
- Select the multiplier for the number of effective disks from Table 1.

Table 1. MetaLUN stripe multipliers

Effective disks in component RAID group	MetaLUN stripe multiplier	MetaLUN stripe element
2	8	1024
3	6	1152
4	4	1024
5 – 7	3	960 - 1344
8	2	1024

If in doubt, use 4 for the metaLUN stripe multiplier. This is the default, and a good choice in most cases.

MetaLUN alignment offset

If you plan to use SnapView or MirrorView with your metaLUN, leave the metaLUN alignment offset value at zero. Use disk utilities to adjust for partition offsets.

MetaLUNs and rebuilds

In general, for SATA and FC drives, the increased IOPS available to multi-RAID group metaLUNs (compared to individual RAID group LUNs) during normal operation outweigh the relatively infrequent

rebuild penalty that is incurred if one of the component RAID groups experiences a disk failure. For ASAP rebuild, the percentage decrease in a metaLUN's throughput can be roughly approximated by the percentage of drives involved in the rebuilding RAID group relative to the metaLUN's full drive width. However, if the rebuild setting is High, Medium, or Low, the rebuilding RAID group will itself incur less than 10 percent degradation, and the overall metaLUN will run very close to production speed.

CLI example: Creating a metaLUN

In the following example code, we create a striped metaLUN using base LUN 30. There is no create command for metaLUNs; you create a metaLUN by expanding an existing FLARE LUN. All the LUNs designated in the command are FLARE LUNs of the same RAID type and the same capacity. LUN 30 will become the base—the new metaLUN will keep 30 as its identifier.

```
metalun -expand -base 30 -lus 31 32 33 -name P1H00 -elszm 4 -type S
```

The expansion type is set to **S**, for striped, and the stripe multiplier (known in this context as the element size multiplier) is set to 4, because the LUNs are built on 4+1 RAID 5 groups with four effective drives.

MetaLUN expansion strategies

There are several strategies for using metaLUNs for a long-term expansion plan. To develop a strategy, you must identify your goals. The goals of the approach presented in the following section are:

- Distribution of localized bursts of otherwise random data over many disk drives
- Good sequential/bandwidth performance
- Efficient use of capacity
- Flexible expansion of devices

These goals apply to the majority of metaLUN users.

Expansion model initial configuration

The rules for the initial setup of this solution are illustrated in Figure 5. The rules are:

- Deploy required drives for initial deployment capacity.
- Create modest-size RAID groups:
 - For RAID 1/0, use four or six drives.
 - For RAID 5 or RAID 3, use five drives.
- Organize the RAID groups into *sets* of four to eight groups. (Use more groups per set if very high rates of random I/O are required.)
- For each metaLUN, determine the RAID group set to which it will belong.
- Define the component LUN size for each planned metaLUN by dividing metaLUN size by the number of RAID groups in its RAID group set.
- Create a component LUN for each metaLUN from *each* RAID group in its set.
- Form metaLUNs from LUNs distributed across all the RAID groups in their respective sets.

Figure 5 is an example of a set of metaLUNs and their RAID group set.

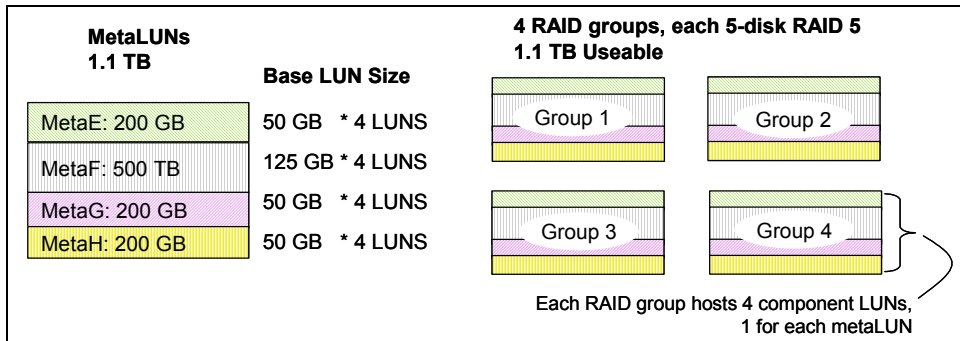


Figure 5. Initial distribution of storage for metaLUNs

Note that in Figure 5, each metaLUN consists of one LUN per RAID group. Thus, each LUN's load is evenly distributed across all RAID groups in *that set*. However, these metaLUNs are fenced off from data access to *other* RAID group sets.

Why use RAID group sets? If we do not allow a metaLUN to extend outside of its set, we can determine a level of fencing, controlling interactions at the drive level. For instance, one RAID group set may be for a large number of file servers, while another is used for RDBMS data tables—and an ordinary pair of RAID 1 groups may be used as the RDBMS log devices. Figure 6 illustrates this.

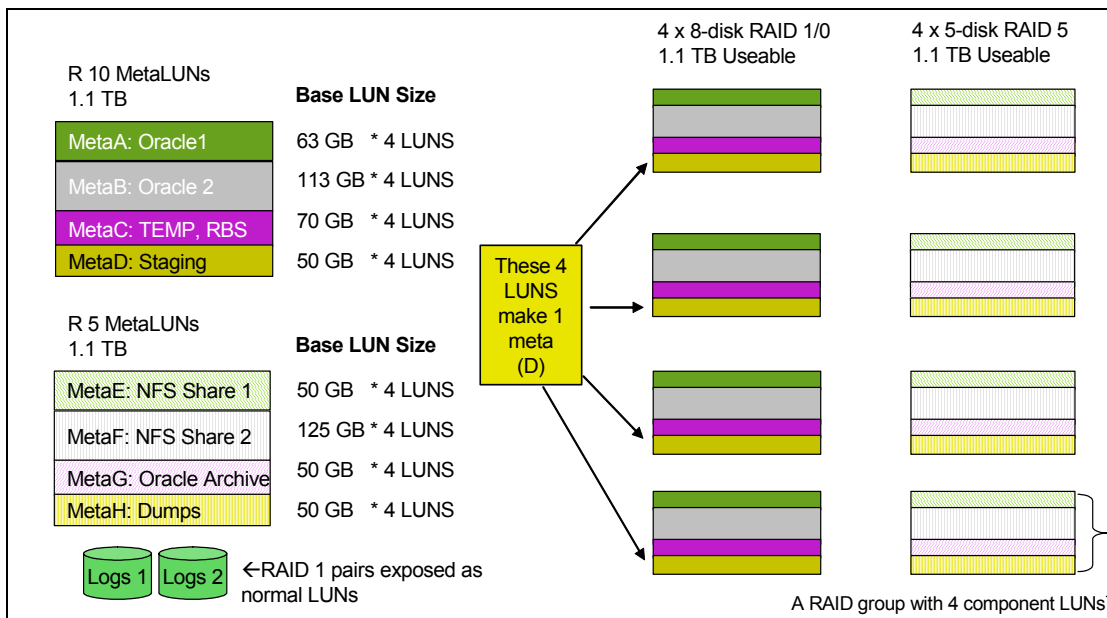


Figure 6. Example of data fencing with RAID group sets and metaLUNs

In the example shown in Figure 6, access to the NFS share metaLUNs will not interfere with the Oracle servers' access to their data tables or to their logs.

Expansion model expansion procedure

The next step is to set up the strategy for expansion. The goals for expansion are:

- Maintain distribution of data across many drives.
- Use capacity efficiently.

The approach to achieve these goals is:

- When capacity is *anticipated* for a metaLUN, add drives to existing RAID groups in the set.
- Bind expansion LUNs on the RAID groups of the metaLUN's set.
- Add expansion LUNs to metaLUNs as a new striped component.

MetaLUN expansion example

The approach used in this example adheres to the original goals for the metaLUN configuration—I/O distribution across all disk drives.

In the first step, the IS department determines that capacity usage for Meta A is over its watermark—85 percent—and notifies the users of that metaLUN. IS receives a request for an additional 160 GB by the end of the week. The operator of this system adds two drives to each RAID group in the set on which metaLUN A resides. RAID group expansion set to medium priority has a very minimal effect on performance. Storage per group is increased by one drive's capacity (66 GB), as shown in Figure 7.

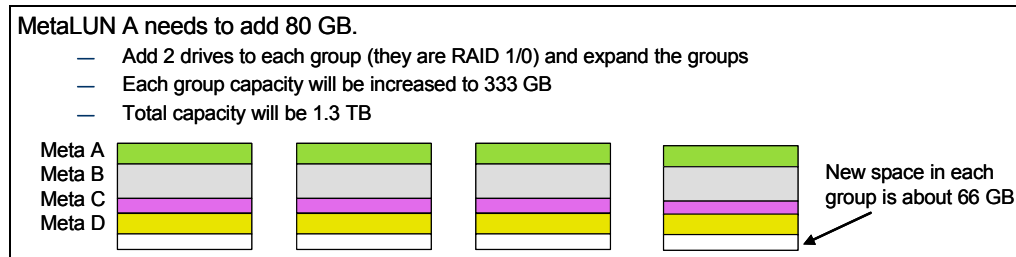


Figure 7. Expansion for metaLUNs: Step 1

The next step is to bind a LUN in each RAID group of the metaLUN's set. The total amount by which they must expand is 160 GB, and there are four RAID groups in the metaLUN set, so $160/4 = 40$. A 40 GB LUN must be bound on each RAID group in the set.

The last step is to expand the metaLUN using the four newly bound LUNs. The operator designates the LUNs to be added and sets the expansion as *concatenated*. Because the expansion LUNs are all the same size, Navisphere concatenates a new *striped component* to the metaLUN, consisting of these LUNs (Figure 8).

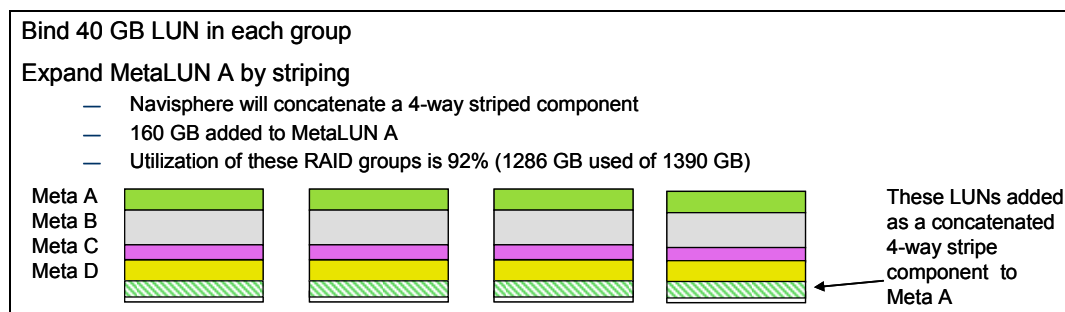


Figure 8. Expansion for metaLUNs: Step 2

The following is an example CLI command for expanding a metaLUN by concatenating a new striped component. The metaLUN identifier is 30. FLARE LUNs 34, 35, 36, and 37 all have the same RAID type and capacity:

```
metalun -expand -base 30 -lus 34 35 36 37 -type C
```

The expansion type is set to C, for concatenated. Navisphere will stripe the LUNs together in a new component to be added to the existing metaLUN, metaLUN 30.

MetaLUN base LUN stacking

When creating multiple metaLUNs from a set of RAID groups as in the previous example, rotate the RAID group in which you locate the base LUN for each metaLUN. This distributes the *hot edge* of a database, file system, or even of a backup process among the disk drives, as shown in Figure 9. Each color stripe denotes a different metaLUN; the base LUN for each meta is on a different RAID group.

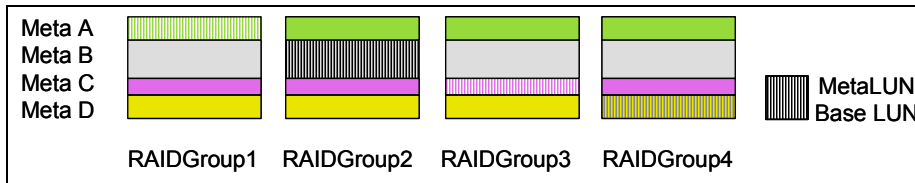


Figure 9. MetaLUN base LUN stacking

Storage controller effects

This section guides the user in matching the required performance with the appropriate CLARiiON hardware.

CLARiiON storage processors

Storage processors in EMC's CX3 line of arrays differ from those in previous models in that they support 4-Gb/s and 2-Gb/s connections to hosts and disks; in fact, both speeds can be present in the same storage system. The lower end models, that is, the CX3-20 and the CX3-40, are comparable in bandwidth performance to the older CX500 and CX700, respectively, when 4 Gb/s disks are configured. Configuring 2 Gb/s disks results in half the available bandwidth so the workload should be studied carefully before upgrading only the SPs. Disk bound OLTP applications are essentially unaffected by the loop speed, so attention should focus mostly on backup and DSS bandwidth behavior.

The characteristics of the CX3 family of storage processors are as follows:

Table 2. CX3 family characteristics

Components/ Connectivity	CX3 Model 10c	CX3 Model 20f	CX3 Model 20c	CX3 Model 40f	CX3 Model 40c	CX3 Model 80
Processor architecture per SP ²	1 processor 1.8 GHz	1 processor 2.8 GHz	1 processor 2.8 GHz	2 processors 2.8 GHz	2 processors 2.8 GHz	2 processors 3.6 GHz
Physical memory per SP	1 GB	2 GB	2 GB	4 GB	4 GB	8 GB
Max cache	310 MB	1053 MB	1053 MB	3016 MB	3016 MB	6875 MB
Max write cache	310 MB	1053 MB	1053 MB	2500 MB	2500 MB	3072 MB
Front-end 4-Gb/s FC ports per SP ³	2	6	2	4	2	4
Front-end 1-Gb/s iSCSI ports per SP ²	2	n/a	4	n/a	4	n/a
Back-end 4-Gb/s FC ports per SP ⁴	1	1	1	4	2	4
Max drives per storage system	60	120	120	240	240	480

² All models in the CX3 series have two SPs.

³ Back-end ports connect the storage system to other DAEs.

⁴ Application available only on the front-end FC ports.

Components/ Connectivity	CX3 Model 10c	CX3 Model 20f	CX3 Model 20c	CX3 Model 40f	CX3 Model 40c	CX3 Model 80
Min drives per storage system	5	5	5	5	5	5
Max initiators per storage system	128	256	256	256	256	512
Max H/A hosts per storage system	64	128	128	128	128	256
Max LUNs per storage system	512	1024	1024	2048	2048	2048
Max RAID groups per storage system	30	60	60	120	120	240
Max drives per RAID group	16	16	16	16	16	16
Max LUNs per RAID group	128	128	128	128	128	128

* See the “Additional Fibre Channel ports” section for further information.

Front-end ports

Front-end ports on CX3 arrays will run at 4 Gb/s if the host and fabric environments support and are configured for that speed. If the infrastructure is designed with 2 Gb/s switches and components, the CX3 ports will run at that speed. If the critical workload is primarily small block I/O, the slower speed will have a negligible effect on throughput. If the workload is larger I/O or sequential and needs high bandwidth, 2 Gb/s ports limit the bandwidth to about 180 MB/s per port versus 360 MB/s for 4 Gb/s (provided that the back-end loops and disk drives are sufficient to meet the 4 Gb/s demand). When multiple ports are driving high bandwidth, the total array-write bandwidth (with write cache on), is determined by SP memory speed, not port speeds. Memory speed is dependent on the SP and varies within the CLARiiON product line.

Table 3 describes the rates, for the CX3 series, at which SP memory speed limits host bandwidth performance.

Table 3. CX3 series typical SP memory speed

Typical SP speed (MB/s)	CX3 Model 10 series	CX3 Model 20 series	CX3 Model 40 series	CX3 Model 80 series
Writes	300	400	450	550
Reads	670	670	1280	1500

The front-end ports can be the bottleneck for write-cache-off bandwidth, and read bandwidth in general, when sufficient drives are available to achieve the limit.

Additional Fibre Channel ports

The CX3-20f and CX3-40f models have four FC ports per SP. In a CX3-20f, these four ports provide additional front-end connections for hosts, for a total of six per SP, reducing the need for FC switches. All available bandwidth can be distributed across six ports. The back-end configuration does not change from the original single bus (for example, two FC loops).

For CX3-40f, two ports are used on the front end for a total of four per SP, and two are used as additional back-end ports for a total of four per SP. This provides several benefits.

- In a 4 Gb/s host environment, four hosts can be directly attached instead of two, with matching back-end capability.
- In a 2 Gb/s infrastructure (with 2 Gb/s back-end devices as well as 2 Gb/s ports) full CX3-40 bandwidth can be achieved as per the 4 Gb/s infrastructure provided there are sufficient drive DAEs to connect all four loops).
- In configurations with both 4 Gb/s and 2 Gb/s devices, four loops in the back end make it easier to isolate the slower drives without sacrificing as much 4 Gb scalability.

The CX3-40f does not deliver an appreciable increase in net bandwidth over the CX3-40. When additional large block sequential devices are added to 4 Gb/s back-end DAEs, the processor bottlenecks before the busses bottleneck. There is also true with small block random I/O, because it is insensitive to bus speed.

However, the CX3-40f does offer more configuration options than the CX3-40. The CX3-40f can reduce the net system effect of common loop-saturating operations. These operations include streaming host workloads, and internal operations such as rebuilds, LUN migration, and replication synchronization. In addition, it is easier to evenly distribute LUNs or RAID groups over multiple busses on the CX3-40f.

Usage

- The extra ports help to achieve the full array bandwidth performance in a wider range of configurations. They do not increase previous SP maximums, since the memory bandwidth remains the same.
- For small-block random workloads, extra buses and ports make little performance difference.
- For mixed environments, separate DAEs of 2 Gb/s drives from 4 Gb/s drives, keeping the 2 Gb/s drives on as few buses as possible.
- For best performance distribute the load equally across all front-end and back-end ports.

iSCSI ports

The CX3 10c, 20c, and 40c models offer four iSCSI ports and two FC ports per storage processor. Connections to both types of hosts are supported simultaneously. The white paper *Best Practice Considerations for CLARiiON CX3 FC/iSCSI Storage Systems* available on EMC.com and EMC Powerlink provides an in-depth discussion of these ports and their features.

SP considerations

A single controller on an SP drives a pair of iSCSI ports. The two ports share the total bandwidth and the IOPS capability of the controller. The controller can drive both ports at once to full GbE bandwidth for reads at about 200 MB/s. Write bandwidth is about 150 MB/s. iSCSI requests are 64 KB maximum regardless of host I/O size, which may affect bandwidth performance and cache bypass settings. In cases where network latency is not an issue (for example, direct host-to-array connections), a single 64 KB stream runs at about 40 MB/s, increasing to 80 MB/s at 512 KB. Multiple parallel threads are required to reach the maximum bandwidth of an iSCSI-based SAN.

iSCSI processing requires more SP CPU power than FC processing. An SP's CPUs may become saturated with iSCSI traffic at a lower IOPS load level than with FC traffic. The controller's IOPS capability only limits array performance with workloads that have very high cache hit rates on small block requests. These types of workloads are very rare. (They are more characteristic of benchmark tests and not real-world applications.)

Block size and workload determine iSCSI performance. For large block and sequential workloads on 1 GbE networks, bandwidth is limited by the ports' capability of about 100 MB/s per iSCSI port. (Direct connection on 10 GbE is not supported.) For small block random workloads, iSCSI ports scale linearly to about 5000 IOPS per storage processor (10,000 per array). To achieve higher rates, EMC recommends that you use FC connections. In these rare cases, sharing the load between one port on each of two different CLARiiON's SPs provides higher performance than sharing the load between both ports of the same SP. FC should be used when you anticipate high bandwidth workloads.

Network and latency considerations

Both bandwidth and throughput rates are subject to network conditions and latency.

It is common for network contention, routing inefficiency, and VLAN misconfiguration to adversely affect iSCSI performance. It is important to profile and constantly monitor the network carrying iSCSI traffic to ensure the best iSCSI connectivity and SAN performance. In general, simple network topologies offer the best performance.

Latency can contribute substantially to iSCSI system performance. As the distance from the host to the CLARiiON increases, a latency of about 1 ms per 125 miles (200 km) is introduced. This latency has a noticeable effect on WANs supporting sequential I/O workloads. For example, a 40 MB/s 64 KB single stream would average 25 MB/s over a 125-mile distance. EMC recommends increasing the number of streams to maintain the highest bandwidth with these long-distance, sequential I/O workloads.

Layered applications with iSCSI

Release 26 supports layered applications over iSCSI. Layered applications that can run over iSCSI include SAN Copy, MirrorView/Synchronous, and MirrorView/Asynchronous. The following performance can be expected from iSCSI with layered applications:

- Incremental SAN Copy performs at the same rate over iSCSI as with FC.
- Single MirrorView/Asynchronous sessions perform at the same rate as FC-based sessions.
- MirrorView/Synchronous mirrored writes with iSCSI perform similarly to FC sessions. However, iSCSI synchronizations are slower due to single thread bandwidth limitations.
- Full SAN Copy uses the full 100 MB/s of available bandwidth, up to a latency of 10 ms.
- Initial synchronizations occur at 40 MB/s per session when run as single streams, and running multiple streams decreases the available bandwidth.

iSCSI usage notes

The following general recommendations apply to iSCSI usage:

- Use FC ports for applications with high bandwidth requirements, including high performance remote replication.
- Use the most recent version of the iSCSI initiator supported by EMC, and the latest version NIC driver for your host.
- Configure iSCSI ports to Gigabit Ethernet full duplex on all network devices in the initiator-to-target path.
- Use CAT6 cabling on the initiator- to-target path whenever possible to ensure consistent behavior at Gigabit Ethernet speeds.
- Use jumbo frames and/or TCP flow control for long distance transfers or with networks containing low-powered servers.
- Use a ratio of 1:1 iSCSI ports to NICs for workloads with high read bandwidths.
- As a best practice, when using two connections make one connection to each controller and leave the other one open. Use port numbers 0, 2 first; then use ports 1, 3.

RAID level resources and performance

CLARiiON storage systems support RAID levels: 1, 3, 5, 6, and 1/0. Refer to the *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper on EMC.com and EMC [Powerlink](#) to learn how each RAID level delivers performance.

RAID levels are divided into two types, parity and non-parity. The parity RAID types include RAID levels 3, 5, and 6. The non-parity RAID types are levels 1 and 1/0. Non-parity RAID groups are typically called *mirrored* RAID groups.

Each RAID level has its own resource utilization characteristics, performance, and data protection characteristics. However, for certain workloads, a particular RAID level can offer clear performance advantages over others.

RAID resource utilization characteristics

The different RAID levels have different levels of storage resource utilization.

With mirrored RAID groups, the storage capacity of the RAID group is half the total of drives in the group. For example, a four-disk RAID 1/0 would be described as RAID 1/0 (2+2). If this RAID group were made up of four 146 GB, 10,000 rpm FC drives, the raw (unformatted) storage capacity of the group would be 292 GB.

With parity RAID groups, the data-to-parity ratio in a RAID group is different depending on the RAID level chosen. This ratio describes the number of drives within the group dedicated to parity and cannot be used for data storage. The data-to-parity ratio depends on the number of drives in the RAID group. The maximum number of drives in a group is 16. The minimum number is dependent on the RAID level. Note that low throughput limits the usefulness of RAID groups with the minimum number of drives.

For RAID levels 3 and 5 the ratio is $N:1$; for RAID level 6 the ratio is $N:2$, where N is the number of data drives in the RAID group.

For example, for a five-disk RAID 5 group, the ratio is 4:1; for an eight-disk RAID 6 group, the ratio is 6:2. These two examples would be described as RAID 5 (4+1) and RAID 6 (6+2), respectively. Using the RAID 5 (4+1) example, with five 146 GB, 10,000 rpm FC drives in the group, the raw storage capacity of the group would be 584 GB with 292 GB being used for parity.

When to use RAID 1

Use of RAID 1 is not recommended. RAID 1 groups are not expandable. Use RAID 1/0 (1+1) groups as an alternative.

When to use RAID 3

For workloads characterized by sequential reads, RAID 3 delivers several MB/s of higher bandwidth than then the alternatives. RAID 3 delivers the highest read bandwidth under the following conditions:

- Disks create the bottleneck, such as when there are a small number of disks for each back-end loop.
- Sequential streams are larger than 2 MB.
- The file system is unfragmented or using raw storage.
- The block size is 64 KB or greater.

RAID 3 should only be used in special cases, because RAID 3 can bottleneck at the parity drive on random writes. Also, when more than one RAID 3 group is actively running sequential reads on a bus, the bus rapidly becomes the bottleneck and performance is no different from RAID 5. In general, RAID 5 usage is recommended over RAID 3.

When to use RAID 5

RAID 5 is generally recommended and preferred over RAID 3. RAID 5 is favored for messaging, data mining, medium-performance media serving, and RDBMS implementations in which the DBA is effectively using read-ahead and write-behind. If the host OS and HBA are capable of greater than 64 KB transfers, RAID 5 is a compelling choice. These following applications are ideal for RAID 5:

- Random workloads with modest IOPS-per-gigabyte requirements
- High performance random I/O where writes are less than 30 percent of the workload
- A DSS database in which access is sequential (performing statistical analysis on sales records)
- Any RDBMS tablespace where record size is larger than 64 KB and access is random (personnel records with binary content, such as photographs)
- RDBMS log activity
- Messaging applications
- Video/media

When to use RAID 6

RAID 6 is new to release 26.

RAID 6 offers increased protection against double drive failures in a parity RAID. It has similar performance to RAID 5, but requires additional storage for the additional parity calculated. This additional storage is equivalent to adding an additional drive, not available for data storage, to the RAID group. RAID 6 can be used as an alternative to RAID 5 when the need for increased reliability outweighs the overhead of the additional parity drive. The white paper *EMC CLARiiON RAID 6 Technology - A Detailed Review* available on EMC.com and EMC Powerlink provides an in-depth discussion of this RAID topology and its features.

RAID 6 groups can be four to 16 drives. A small group is up to six drives (4+2). A medium group is up to 12 drives (10+2), with large groups being the remainder. Small groups stream well. However, small random writes destage slowly and can affect the efficiency of the system write cache. Medium groups perform well for both sequential and random workloads.

The ratio of data to parity is an important consideration when choosing a RAID 6 group size. There is an overhead in capacity and I/O performance equivalent of two drives with RAID 6. For example, a RAID 5 five drive (4+1) would need to migrate to a RAID 6, six-drive (4+2) group to have the same data capacity. In addition, this RAID 6 group would have 20 percent greater IOPS, if it is not unsaturated. The optimal RAID 6 groups are 10 drive (8+2) and 12 drive (10+2) sized.

It is useful to compare RAID 6 to RAID 5 performance for random and sequential workloads, and rebuilds.

For random workloads, RAID 6 performs the same as RAID 5 with regard to read operations. Random writes are different. The additional parity drive over RAID 5 increases the RAID 6 back-end workload by 50 percent for writes. This affects the performance in a CLARiiON as the number of drives scales. In addition, the additional overhead of RAID 6 could lead to a full cache state earlier than on RAID 5. However, as long as workload can be destaged from cache without forced flushing, RAID 5 and RAID 6 have similar behavior from a host response time point of view.

For sequential workloads with the same number of drives, read performance is nearly identical. Sequential write workloads are about 10 percent lower for RAID 6.

Performance-wise, the comparison between RAID 6 and RAID 1/0 is similar to a RAID 5 versus RAID 1/0 comparison. That is, the increased storage capacity of the parity RAID versus the efficiency of the mirrored RAID with random, small-block writes. However, RAID 6 can suffer a double drive failure on any two drives in the group. The failure of a RAID 1/0 drive and its mirroring drive would be fatal for a RAID 1/0 group. When deciding, RAID 6 has a capacity advantage over RAID 1/0, while RAID 1/0 continues to offer superior small-block write performance.

Due to its double-drive protection, RAID 6 groups are good candidates to use the High rebuild rate instead of the default and common ASAP rebuild rate for RAID 5. Large-size RAID 6 groups rebuild at approximately the same rate as large RAID 5 groups. Medium-size RAID 6 groups require about 10 percent longer to rebuild than RAID 5 groups with the same number of data drives. Small-size RAID 6 groups take 25 percent longer than the same size RAID 5 group.

RAID 6 rules of thumb

- Ten drive (8+2) and 12 drive (10+2) RAID 6 groups have good data-to-parity ratios, IOP capability, and streaming characteristics. These group sizes should be the first candidates for consideration.
- Sixteen drive (14+2) has the best RAID 6 data-to-parity ratio and scales random workloads the best. However, it does not run sequential streams better than eight drive (6+2), 10 drive (8+2), and 12 drive (10+2) RAID 6 groups.

- RAID groups of four drives (2+2) are not recommended. The data-to-parity ratio is too low, despite the 33 percent increase in IOPS over a similar RAID 5 group of three drives (2+1).
- If the read/write ratio does not exceed drive capability, six drive (4+2) RAID groups have 20 percent more available IOPS over a RAID 5 group of five drives (4+1).
- The overhead of RAID 6 over RAID 5 needs to be considered when making replacement choices. A RAID 6 group of 10 (8+2) drives is not equivalent in performance to two RAID 5 groups of five drives (4+1).
- RAID 6 groups lend themselves to “bus balancing.” A 10-drive (8+2) group can be evenly distributed over two DAEs; five drives per DAE. Twelve drive (10+2) groups can be evenly distributed over three or four DAEs.

When to use RAID 1/0

RAID 1/0 provides the best performance on workloads that use very small, random, and *write-intensive* I/O. A write-intensive workload’s operations consist of greater than 30 percent random writes. Some examples of random, small I/O workloads are:

- High-transaction-rate OLTP
- Large messaging installations
- Real-time data/brokerage records
- RDBMS data tables containing small records that are updated frequently (account balances)

A RAID 1/0 also offers performance advantages during certain degraded modes. These conditions are when write cache is disabled or when a disk has failed in a RAID group.

LUN Migration

The LUN Migration facility is used to change a LUN’s RAID group topology, or to move a hot LUN to a less-used set of spindles.

LUN Migration is like clone synchronization in that a speed is and the data transfer is done through the SP’s cache subsystem. The available transfer speeds are: ASAP, High, Medium, and Low. High, Medium, and Low transfer rates have little effect on the production workloads, because they are implemented as 1 MB sequential transfers every 0.3s, 0.6s, and 1.2s, respectively, or about 10 GB/hr, 5 GB/hr, and 2.5 GB/hr.

The effect of a LUN migration on an individual drive depends on the number of drives in the source and destination LUNs. The transfer rates are designed to allow production workloads to continue executing during the migration process. As a reference, the SP’s background sniffing policy is set at 512 KB/s per drive, about half the rate of a Low migration on a 4+1RAID 5.

ASAP LUN Migration with default cache settings should be used with caution. It is recommended to run at the High speed, unless migration time is critical.

The ASAP setting emphasizes the migration process, and can consume considerable SP resources at the expense of application throughput/bandwidth/responsiveness. The workload can be considered a simple 1 MB block size sequential copy from the original LUN to the new LUN. For a 4+1 RAID 5 LUN on 15k rpm 4 Gb/s drives, the rate is about 150 MB/s (a little less than an hour for a 500 GB LUN). Since the cache subsystem is utilized, the transfer behavior can be largely affected by cache settings (SP read/write on/off, LUN read/write on/off, max prefetch and prefetch multipliers, the write aside value, and watermarks), as well as the drive widths of the source and destination.

If the destination device cannot write as fast as the source can read, the write cache will fill for the duration of the migration, causing forced flushes and poor response times for other LUNs on the same SP.

An effective way to ensure that an ASAP LUN Migration does not consume the SP is by setting the cache write-aside to 511 (the current default is 1023) at the destination LUN (*before* the migration operation is initiated, since the target will become invisible once it starts). In this case, the 4+1 RAID 5 device will migrate at about 40 MB/s (144 GB/hr) with no effect on the write cache. Alternatively, the write cache

can be disabled at the destination LUN, but as the migration proceeds host writes to the source must be mirrored to the (write-disabled) target drive.

When migrating to a destination LUN that has a greater capacity than its source, an internal consistency operation occurs in the “new” space on the destination LUN. On the 4+1 RAID 5 above, it runs at about 40 MB/s (144 GB/hr). To avoid the time penalty associated with this operation, consider using the concatenate operation under metaLUNs to create the correct sized source LUN, and then copy it to its destination. Alternatively, copy the source to the same sized destination, and then perform a LUN expansion of the destination.

Rebuilds

Rebuild times depend on a number of factors, including the rebuild rate (ASAP, High, Medium, or Low), presence/location of an appropriate hot spare, disk type, disk size, bus speed, application load, and RAID group topology. When a hot spare is used for the rebuild, there is an additional “equalize” operation that occurs when the faulty drive is replaced and the hot spare is copied to it.

Baseline ASAP rebuild/equalize rates with no application load for several common RAID group configurations using 15k rpm Fibre Channel drives on a single 4 Gb/s loop are listed in Table 4.

Table 4. Baseline rebuild rates.

ASAP FC no load 4 Gb/s loop	4+1 R5	8+1 R5	4+4 R10
Rebuild rate (writes to hot spare)	2.5 GB/minute	1.6 GB/minute	3 GB/minute
Equalize rate (hot spare to new drive)	3.75 GB/minute	3.75 GB/minute	3 GB/minute
Rebuild drive bandwidth	42 MB/s	28 MB/s	50 MB/s
Equalize drive bandwidth	65 MB/s	65 MB/s	50 MB/s
SP CPU utilization	15 percent	17 percent	4 percent

Note: The Rebuild rate in the table is for the target drive, *not* LUN capacity. A 4+1 RAID 5 group consisting of 146 GB drives will rebuild in about an hour, which equates to about 530 GB of user capacity. The rates above are the same for FC 15k rpm 4-Gb/s drives that are 73, 146, and 300 GB.

Equalize rates on RAID 5 are faster than the rebuild itself since only two drives are active (the hot spare and the new drive) and there is no need to recompute parity or data from the other drives. Equalize on RAID 10 is identical to the rebuild, only in the opposite direction.

For approximate RAID 5 rebuild rates between 4+1 and 8+1, subtract 0.25 GB/hr per additional drive from the 4+1 rate. Above 8+1, subtract 0.1 GB/hr per additional drive from the 8+1 rate.

RAID 6 groups require more time to rebuild than a RAID 5 group with the same number of data drives. This difference decreases as the size of the group increases. Small groups require 25 percent, midsize groups 10 percent, and large groups 5 percent additional time to rebuild over RAID 5.

The CX3 series of arrays have similar rebuild times for a given topology, since parity and mirroring computations are extremely short relative to the disk latencies. The CX300/500/700 series (that is based on 2 Gb/s buses) uses a different algorithm for RAID 5 rebuilds. They run about half as fast as the CX3 series for that topology. The RAID 10 algorithm is the same for all the processors and there is very little variation from the above table.

When the hot spare is on a different loop from the surviving drives in a parity RAID group, the rebuild rate is about 30 percent faster than the same-bus rate in the table, because the data reconstruction operations have better concurrency with the hot spare writes – but we do not recommend that you design degraded mode application criteria that depend on that fact without careful consideration. This prevents the situation

where a user benchmarks a (fast) rebuild to a hot spare on another bus, but then encounters a (slower) production rebuild to the same bus with markedly differing results.

It should be noted that when all drives of a parity RAID group are on the same bus, a significant portion (greater than 50 percent) of the available bus bandwidth is consumed by a rebuild. If this has a detrimental effect on bandwidth-sensitive applications elsewhere on the array, consideration should be given to configuring the RAID groups across buses.

If a hot spare is on a 2 Gb/s loop and the surviving drives are on 4 Gb/s, the rebuild rate is basically unchanged because of thread concurrency. If all drives are on 2 Gb/s, however, the bottleneck will be at the bus (200 MB/s theoretically, about 180 MB/s practically) instead of the drives.

Rebuild and equalize operations do not use the read or write cache subsystems.

The effect of ASAP rebuild on application degradation depends on the workload mix. Three typical profiles and their degraded characteristics are listed in Table 5:

Table 5. ASAP rebuild effect on application load

Topology	Workload profile	Normal LUN operation	Degraded LUN operation	Drive rebuild rate
4+1 R5	64 KB seq read 1 thread	145 MB/s	8 MB/s*	40 MB/s
4+1 R5	64 KB seq write 1 thread	110 MB/s	95 MB/s	38 MB/s
4+1 R5	2:1 r:w 8 KB random 16 thread 100 percent disk utilization	800 IOPS	400 IOPS	21 MB/s
4+1 R5	2:1 r:w 8 KB random 70 percent disk utilization	500 IOPS	250 IOPS	21 MB/s
4+4 R10	64 KB seq read 1 thread	190 MB/s	60 MB/s*	30 MB/s
4+4 R10	64 KB seq write 1 thread	120 MB/s	90 MB/s	27 MB/s
4+4 R10	2:1 r:w 8 KB random 16 thread 100 percent disk utilization	1400 IOPS	1100 IOPS	8 MB/s
4+4 R10	2:1 r:w 8 KB random 70 percent disk utilization	1150 IOPS	850 IOPS	14 MB/s

*Sequential read.

Sequential writes destage from cache with multiple threads per drive and large coalesced block sizes that compete with the rebuild operation. Sequential reads (the bold values in Table 5) do not generally generate a similar parallel thread load per drive, nor is the block size augmented. Hence sequential read operations like LUN backups run significantly slower during ASAP rebuild than in normal operation. This can be considerably improved by setting the variables **prefetch multiplier** to 32 and **segment multiplier** to 4. 64 KB sequential reads then prefetch 2 MB at a time in 256 KB chunks, which increase the 4+1 RAID 5 value from 8 MB/s to about 45 MB/s and the RAID 10 value from 60 MB/s to 110 MB/s.

If an ASAP rebuild cannot be tolerated by an application, rebuild time can be paced with the High, Medium, or Low setting. The rebuild rates are much slower than ASAP, but even at the High setting production workloads are only competing with the rebuild for 10 percent of the time. Note that RAID 6 is particularly efficient at rebuilding. See Table 6 for more information.

Table 6. High, medium, and Low rebuild characteristics

Characteristic	Topology	High	Medium	Low
Rebuild rate* (drive)	RAID 5	12 GB/hr	6 GB/hr	3. GB/hr
	RAID 6	12 GB/hr	6 GB/hr	3. GB/hr
	RAID 1/0	18 GB/hr	11 GB/hr	7 GB/hr
Rebuild active (%)	RAID 5	10 %	5 %	2.5 %
	RAID 6	10 %	5 %	2.5 %
	RAID 1/0	10 %	6 %	4 %

*Depending on the drive activity, the rates may be faster. In the extreme case of no other disk activity to the RAID group for the duration of the rebuild, High, Medium, and Low all behave like ASAP.

It is also interesting to note that SATAII-drive sequential behavior is similar to FC drives, except that you must multiply the rebuild rates in Table 6 by 0.75 for ASAP no-load. (A 500 GB SATAII drive rebuilds in about 4.5 hours.) Use a multiplier of 0.5 for the application loads. CX300/500/700 SATA RAID 3 and RAID 5 4+1 320-GB 7200-rpm drives rebuild in about 6 hours. Use a multiplier of 0.33 for the application loads. 10k rpm FC drives run at about half the rebuild speed of 15k rpm drives (they are 2 Gb/s devices). Use a multiplier of 0.65 for the application loads.

Hot spares

The replacement algorithm is the smallest hot spare that can accommodate the size of the failed drive that is selected as the target. Secondly, hot spares on the same loop as the failed drive are preferred over other like-size drives. Any drive not under control of a BCC (2 Gb/s SATA, which must have its own hot spares) can act as a hot spare for a normal DAE, so be careful in SATAII/LCFC/FC mixed environments to have at least one spare available for each type. LCFC or SATAII can be used to spare FC, but with obvious performance degradation. One hot spare for every two DAEs of a given drive type is a reasonable rule of thumb. They can be positioned anywhere except the Vault.

Proactive sparing

CLARiiON supports the proactive initiation of drive substitution when an SP detects drive errors. Proactive sparing runs at the equalize rate, unless the drive actually fails – in which case the hot spare rebuild is completed by the appropriate RAID algorithm.

Fastbind

CLARiiON supports the fastbind feature for binding a LUN. Through this feature a LUN is accessible to the user immediately after the command to bind the LUN returns.

Binding a LUN requires initialization of the parts of the disks on which the LUN resides. Previously, a LUN was not available for use until this initialization was completed. LUN initialization could take several hours, depending on the size of the LUN and the type of disks involved. A fastbind initialization is done in the background, and on-demand when un-initialized parts of the LUN are accessed.

The initialization process has an effect on performance and you should be aware of this, especially when benchmarking. In these situations, you should wait until the background initialization is complete before running any tests. On an otherwise idle set of disks, it is possible to detect background initialization through Navisphere Analyzer by looking at disk statistics. Start and completion of LUN initialization is also noted in the SP Event Logs.

An average disk write size of over 512 KB indicates that background initialization is in progress. On disks with other workloads, this indicator is still correct but not comprehensive. The average write size could be brought down below 512 KB because of other small requests. For reference, examples of time it takes to do background initialization on an idle array for LUNs that occupy the whole RAID group are included in Table 7. This time depends on the size and type of disks in the RAID group. It is not affected by the number of disks in the RAID group. It is also not linear in the fraction of the LUN initialized— the second half of the LUN takes about 10 percent longer to initialize than the first half.

Table 7. Time taken to complete background initialization⁵

Type of disk	Size of disk (GB)	Time taken for background initialization to complete on an idle CX3-40 (Hours)
15k 4 Gb/s FC	146	1.25
7.2k 4 Gb/s SATA II	500	8.5

The RAID engine cache

This section discusses the proper cache page size to use, how to use prefetch size, where to set watermarks, and other techniques, such as SP balancing. The *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper on [EMC Powerlink](#) has general information on these features and their effect on performance.

Cache size and performance

For SPs with 1 GB or less of available cache memory, use 20 percent of the memory for read cache and the rest for write cache. For larger configurations, use as much memory for write cache as possible while reserving at least 200 MB for read cache.

Scripting cache settings

In many environments, production workloads vary significantly during different times of the day. For instance, from 8:00 A.M. to 5:00 P.M. the load may be interactive OLTP, from 5:00 P.M. to 8:00 P.M. it may change to multithreaded sequential DSS for reporting, and after 8:00 P.M. the backup streams begin. Navisphere CLI scripts can be used to adjust cache parameters if desired to tune to the various workloads. Some parameters, such as SP read cache on/off, SP read cache size, LUN read and write cache on/off, LUN prefetch settings, and watermark settings can be changed non-intrusively. Others, such as SP write cache size and page size require SP write cache disable, which significantly affects write response times for the minute or so that the operation takes to complete.

Cache settings

The cache parameters listed below for CLARiiON storage systems *all have default settings that will benefit most users*. The cache's operation and its default settings are outlined in the *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper available on [EMC Powerlink](#).

Caches on or off

Most workloads benefit from both read and write cache; the default for both is **on**.

To save a very small amount of service time (a fraction of a millisecond to check the caches when a read arrives), turn off read caching on LUNs that will not benefit from it. For example, LUNs with very random read environments (no sequential access) will not benefit from read cache. Also, LUNs with more than a

⁵ Measurements were done on an idle CX3-40 using RAID5 (4+1) groups. Size of RAID group and type of array should not have a significant affect on these times.

dozen simultaneous sequential streams (often the case in DSS) can benefit without read cache (or by disabling prefetch) to avoid thrashing. Some users turn it off when syncing clones in order to get an intermediate bandwidth between ASAP and High with significantly reduced SP utilization. Use Navisphere CLI scripts to turn on read cache for LUNs when preparing to perform backups.

Write caching is beneficial in all but the most extreme write environments, and deactivation of write cache is best done using the per-LUN *write-aside* setting (refer to “Write-aside size”).

Page size

For workloads where I/O size is very stable, you gain some benefit by setting the cache page size to the request size seen by the storage system—the file system block size or, if raw partitions are used, application block size.

In environments with varying I/O sizes, the 8 KB page size is optimal.

Systems that make heavy use of SnapView, MirrorView/A, or incremental SAN Copy benefit from a 16 KB cache page size due to internal paging of 64 KB blocks. If the application workload is predominantly small block, the watermarks should be moved to 60/40.

Be careful when applying a 2 KB cache page size. Sequential writes to parity RAID groups with more than five drives may be affected. “CLARiiON RAID 5 stripe optimizations” has more information.

The HA Vault option and write cache behavior

The HA Cache Vault option, found on the **Cache** tab of the storage system properties dialog box, is on (selected) by default.

Several failure types can cause the write cache to disable and dump its contents to the Vault. One type of failure is that of a Vault drive. If the user *clears* the **HA Cache Vault** selection, then a Vault disk failure will *not* cause write cache to disable. Since a disabled write cache significantly affects host I/O, it is desirable to keep the write cache active as much as possible. Thus, the user has this option.

Why is it an option? Clearing this selection exposes the user to the possibility of data loss in a *triple*-fault situation: If a Vault drive fails there will be no dump. If another Vault drive fails before the first one has been replaced *and* the array experiences a total power failure, the contents of the cache will be lost. Similarly, if after the initial Vault drive failure there is a power outage and then a second drive fails during or after the subsequent dump, data will be lost.

The user must make the decision based on the relative benefit versus risk.

Prefetch settings

The default setting for prefetch (**Variable**, with segment and multiplier set to 4) causes efficient cache behavior for most workloads. Changing the multipliers can have unintended effects. It is best to work with a CLARiiON SPEED expert in cases where sequential performance requires tuning.

High and low watermarks and cache flushing

The CLARiiON design has two global settings called *watermarks*—high and low—that work together to manage cache flushing. For most workloads, the default settings: 80 percent high watermark and 60 percent low watermark afford optimal behavior.

When modifying, the high and low watermarks, the values typically increased or decreased by the same amount. That is, if the high watermark is decreased by 20 percent, the low watermark is increased by 20 percent.

If write bursts are causing enough forced flushes to affect host write workloads-- decrease the watermarks. Lowering the watermarks causes more aggressive flushing, which allows more free cache pages to absorb bursts, at the cost of fewer read hits from the write cache. For smaller CLARiiON configurations, reducing the watermarks to 60 percent high watermark and 40 percent low watermark can help reduce forced flush conditions.

Write-aside size

This setting specifies the largest write request that will be cached. Write requests larger than this size will bypass the write cache. The write-aside size is a per-LUN setting.

Write-aside helps keep large I/O from taking up write cache mirroring bandwidth, and makes it possible for the system to exceed the write cache mirroring maximum bandwidth. The cost is that I/O that bypasses cache will have a longer host response time than cached I/O.

The Navisphere CLI **getlun** command displays the write-aside size for a LUN. To change the write-aside size, use the Navisphere CLI **chglun** command.

To exceed the write cache mirroring bandwidth, there must be sufficient drives to absorb the load. Furthermore, if a parity RAID is used, ensure that:

- I/O is equal to or a multiple of the LUN stripe size *and*
- The host generates enough concurrency to keep the LUNs busy
- I/O is aligned to the stripe *and*
- The rules for Full Stripe Writes are met (see “CLARiiON RAID 5 stripe optimizations”)

These conditions for parity RAID are crucial and cannot be stressed enough. Getting I/O to align for effective write aside can be difficult. If in doubt, use write cache.

The tradeoff for doing write-aside is as follows:

- The data written this way is not available in cache for a subsequent read.
- The response times for writes are longer than for cached writes.

Note that Full SAN Copy uses a 512 KB transfer size. The default write-aside value is 2048 blocks, or 1 MB. Therefore, Full SAN Copy writes are cached by default. Multiple Full SAN Copy sessions may tax the write cache on the target array; in this case, the write-aside value can be fixed at 1023 blocks (511.5 KB) or the write cache disabled at the target LUNs. Note, however, that only 2+1, 4+1, and 8+1 RAID 5 or 4+1 and 8+1 RAID 3 topologies will be able to take advantage of full stripe writes in this case, provided there is no LUN bind offset

Balancing load between SPs

In order to avoid scenarios where one SP becomes a bottleneck while the other SP has significant spare capacity, distribute LUNs across SPs so that the load across the two SPs is balanced. Navisphere Analyzer can be used to detect load imbalances.

The back end

This section addresses the disk hardware, assignment of drives to RAID groups, and the distribution of LUNs.

UltraPoint, DAE2, and SATA-DAE

UltraPoint™ refers to the CLARiiON DAE. UltraPoint DAEs use a switch within the DAE to access the drives. This enables the CLARiiON series system to test drives in isolation before they are brought online. Also, the disk can be diagnosed and isolated more effectively than the loop implementation of the DAE2. (The DAE-SATA also uses a switch, to go from Fibre Channel loop to SATA.) In Navisphere, the UltraPoint DAEs will show up as type DAE-2P.

The benefits of the fault isolation are greatest when the entire disk loop uses UltraPoint or DAE-SATA DAEs. Also, the benefits are greatest for systems with large drive counts.

The approach for using UltraPoint with existing systems focuses on expansion strategies.

- When upgrading storage processors, new loops should be populated entirely with UltraPoint or SATA DAEs.
- The ability of the CLARiiON to migrate LUNs without disruption can be used to equalize load across loops after an upgrade.
- When existing systems add capacity, concentrate the new drives on the least loaded loop. Also, if the system can be taken out of service, the DAE2s can be swapped out for DAE2Ps. (UltraPoint DAEs are compatible with the previous models of CLARiiON Fibre Channel disk drives.)

4 Gb/s and 2 Gb/s back ends

CLARiiONs support 4-Gb/s Fibre Channel loops in the back end. Disks with 2-Gb/s Fibre Channel interfaces can also be used.

The DAE-SATA connects to the SP through a 2-Gb/s interface. When 2 Gb/s disks and DAE-SATAs are used, the whole bus (pair of loops, one to each SP) reverts to 2 Gb/s speed.

A 4-Gb/s back end helps in high bandwidth workloads. For these applications, use only 4 Gb/s disks to service the high bandwidth LUNs. Published performance numbers for CLARiiON are generated with 4 Gb/s back-end buses; it may not be possible to achieve these bandwidth numbers if some or all of the back-end buses are running at 2 Gb/s.

LUN distribution

For the purposes of this discussion:

- *Back-end bus*: The redundant pair of Fibre Channel loops (one from each SP) by which **all** CLARiiON storage systems access disk drives. (Some CLARiiON storage systems have dual back-end buses—a total of four fiber loops, some have four back-end buses.)
- *Partitioned RAID group or partitioned LUN*: A RAID group partitioned into multiple LUNs, or a LUN from such a RAID group.
- *Dedicated RAID group or dedicated LUN*: A RAID group with only one LUN.

For efficient distribution of I/O on Fibre Channel drives, distribute LUNs across RAID groups. When doing distribution planning, take the capacity of the LUN into account. Calculate the total GB of high-use storage, and distribute the capacity appropriately among the RAID groups.

In applications that demand high bandwidth, care should be taken to distribute the LUNs involved across the back-end loops so that the capacity of all back-end loops can be used. This can sometimes lead to scenarios where one uses partially populated disk enclosures. In the CLARiiON CX3-80 example, if a system needs 80 disks on an array with eight back-end loops, then the disks would need to be distributed in eight disk enclosures to make full use of all the back-end loops.

Additionally, balance load across SPs. To do this, assign SP ownership: The default owner property for each LUN specifies the SP through which that LUN is normally accessed.

When partitioning SATA drive RAID groups on CX300/500/700, keep all LUNs from each RAID group owned by a single SP.

When planning for metaLUNs, note that all LUNs used for a metaLUN will be trespassed to the SP that owns the base LUN; their original default owner characteristic will be overwritten. Thus, when planning for metaLUNs, designating pools of SP A and SP B LUNs assists in keeping the balance of LUNs across SPs even.

Vault and PSM effects

In the CLARiiON, the first five drives in the base DAE are used for several internal tasks.

The cache Vault

Drives 0 through 4 are used for cache Vault. The cache Vault has significant I/O during the following cases:

- When the system is disabling write cache (dumping cache to the Vault).
- When the system is recovering from a power failure (no host access during this period).
- When the Vault is rebuilding (following a failure of a Vault drive: disks 0 - 4).

The primary effect on host access during a failure is due to write cache *disabling*. If the write cache is disabled, overall performance is affected, with host writes returning at much higher response times

(especially for RAID 5). Access to the Vault drives (both reads and writes) will be slowed during the dump, which completes within a few minutes.

During power failure recovery, the saved write cache contents stored in the Vault are written to their destination LUNs.

A RAID group that is *rebuilding* will be fairly busy. This effect is lessened if the HA Vault option is cleared in Navisphere. In that case, the cache stays enabled during a Vault drive rebuild, at the expense of a small amount of protection for cached data. “The HA Vault option and write cache behavior” section has more details.) The rebuild of a Vault drive is more significant than rebuild of disks in other RAID groups as the Vault is a parity-protected region, so all Vault drives are busy during a rebuild. Also, other system access to the FLARE database and PSM is ongoing (see below).

It is recommended that you avoid using the Vault drives for response-time critical data. This type of information includes:

- Microsoft Exchange data volumes
- Host boot volumes
- Cluster quorum drive volumes
- RDBMS tables expecting heavy load

If the system has 30 drives or less, or the user requires the storage on the Vault drives, it’s recommended that the load not exceed 100 IOPS per FC drive or 50 IOPS per SATA drive. The average busy queue length (ABQL) per drive cannot exceed 3.

Do not place Snaps, Clones, and Mirrors on the Vault drives. They incur substantial internal processing that increases the ABQL significantly.

Any data LUN on the Vault drives will increase the time it takes to get the write cache enabled again after Vault drive failure due to the rebuilding of the data unless the **HA Vault** option is cleared. This means that *all* LUNs in the array will see poor write response times for a longer period of time. On systems with SATA drives in the Vault with **HA Vault** checked, this period may be up to 8 hours since the equalize is included in the disable period.

FLARE database drives and PSM

A subset of the five drives containing the Vault also contains: the operating system files for the SP, the Persistent Storage Manager (PSM), and the FLARE configuration database.

Once booted, the operating system files are accessed very infrequently. However, Navisphere uses the PSM for storing critical array state information as well as statistical data and software update binaries. Although SP access to the PSM is light, maintenance of good access time to the drives is necessary to maintain a properly performing array. Heavy user and/or layered applications such as Snaps and Clones can interfere with the timing of state changes and result in overall sluggish responsiveness (including turnaround times of Navisphere Manager commands).

On CLARiiONs with 120 hard drives or more, the Vault is often not configured for 100 percent non-disruptive FLARE access. On CLARiiONs with 60 or more hard drives, the Vault may be user-configured as a RAID 5 4+1 and used for low-traffic archiving or file system space.

On CLARiiONs with 60 hard drives or less where the Vault drive capacity is a significant percentage of the overall system total, there are several choices. A RAID 5 4+1 used for low-traffic archiving or file system space is the preferred choice (keeping IOPS in mind, especially for SATA drives), but consider that a 3+3 RAID 10 device (including one non-Vault drive) can help reduce the ABQL over RAID 5 if there is a large load of random small block writes that cannot be provisioned elsewhere. Alternatively, the five drives may be split into a RAID 10 1+1 (first two drives) and a RAID 5 2+1 (next three drives) to at least partially reduce the ABQL while gaining some capacity over the fully mirrored 3+3.

Creating RAID groups that extend beyond the Vault drives decreases the available capacity of the RAID group by 33 GB per drive.

Using LUN and RAID group numbering

This suggestion does not help performance but does assist in the administration of a well-designed system. Use RAID group numbering and LUN numbering to your advantage.

For example, number LUNs so that all LUNs owned by SP A are even numbered and LUNs owned by SP B are odd numbered.

A scheme to extend this is to use predictable RAID group numbering, and extend the RAID group number into the LUN number. This will facilitate selection of LUNs for metaLUNs. The RAID group number embedded in the LUN number allows you to select LUNs from multiple RAID groups (Table 8).

Table 8. Example of RAID group and LUN numbering

RAID group	LUN	Default owner
10	100	SP A
	101	SP B
20	200	SP A
	201	SP B
30	300	SP A
	301	SP B

For example, if selecting LUNs with which to extend FLARE LUN 101 into a metaLUN, choose LUNs 201 and 301 because all three LUNs belong to the same SP (and metaLUN components will all be trespassed to the same SP as the base LUN anyway). Also, now the I/O for the new metaLUN 101 will be distributed across three RAID groups.

The SP assignment scheme in Table 8 is called AB ownership, since the LUNs are alternately assigned to SPA and then SPB in each raid group. AB ownership works on FC and SATA drives because they are truly dual ported, but is highly discouraged for CX300/500/700 SATA since performance can be significantly affected by their single-port (multiplexed) access, especially in bandwidth applications. (This is the principle reason for default ownership alternating by RAID group at bind time in Navisphere Manager rather than alternating by LUN, as was the case in the past.) AB ownership within a RAID group does not guarantee that SP utilization will be balanced, especially when LUN assignments to hosts are done ad hoc without consideration of anticipated workload. It is also generally more difficult to analyze performance metrics when there is widespread AB ownership on the array.

Minimizing disk contention

As drive sizes continue to increase, partitioned RAID groups are more common, and it becomes more difficult to optimize disk behavior. The CLARiiON design is quite flexible and will deliver good performance, even with a significant amount of disk contention. However, for high-performance environments, the following guidelines apply.

Backup during production

Environments that require sequential reads (online backups) concurrent with production will get very good results with RAID 1/0 groups, as the read load can be distributed across many spindles. RAID 5 can also deliver good read throughput while under moderate load, such as messaging applications. Such arrangements should be tested before deployment. Keep write loads from saturating write cache while backing up—the higher priority of cache flushes will slow read access.

Reserved LUN pool and clone LUNs

It is not wise to place reserved LUN pool LUNs on the same drives as the source LUNs you will snap. Write operations will result in very high seek times and disappointing performance.

The same holds true for clone LUNs: put them on disk groups separate from the LUNs they are cloning.

SATA drives are not generally recommended for use in RAID groups for reserved LUN pools or clones. While they can work in some cases, their lower performance may make them a bottleneck, especially when source LUNs are bound on Fibre Channel RAID groups and the host is writing at a high rate.

Stripes and the stripe element size

The stripe element size is 128 blocks or 64 KB. There is no reason to change it.

CLARiiON RAID 5 stripe optimizations

Sometimes EMC personnel will recommend that only 4+1 or 8+1 RAID 5 groups be bound. This is usually not necessary and is often based on an incomplete understanding of the CLARiiON RAID technology. For most cases, the RAID group sizes used to bind a required collection of drives should be based on factors other than performance. The basis for the 4+1/8+1 myth is in stripe-write (bandwidth) optimizations.

Parity RAID types achieve their best performance when the entire stripe is in SP memory and the parity is computed on the fly. This is called a Full Stripe Write and is sometimes referred to as Modified RAID 3 (MR3) when applied to RAID 5. The requirements to achieve MR3 optimization for RAID 5 have been misunderstood for some time. Many EMC personnel believe that the CLARiiON RAID optimizations work only with a 4+1 or 8+1 stripe, which is not true—*MR3 can work for any size RAID 5 group*.

MR3 occurs when an I/O fills the RAID stripe, whether because the I/O was large and aligned on the stripe, or if sequential I/O is cached until it fills the stripe. However, the process has the following requirements:

- The cache page size is 4 KB or, for stripes of 512 KB and larger, 8 KB or 16 KB.
- The stripe element size is 64 KB (128 blocks) or smaller, and a multiple of 8.

For example, with a 12+1 RAID 5 group and a 64 KB stripe element, the stripe size is $12 \times 64 \text{ KB} = 768 \text{ KB}$. For MR3, a cache page size of 8 KB or larger must be used, as a 4 KB page is too small.

Number of drives per RAID group

For high bandwidth workloads, RAID 5 groups with more than 10 drives should be avoided. These large RAID groups result in the following:

- Longer rebuild times for parity RAID groups
- Longer synchronization times as the SP waits for multiple devices to complete an I/O
- Additional seek latency as all the disks align on the same stripe for a particular I/O

If many drives are needed to satisfy a LUN's performance requirement, use a metaLUN with smaller disk groups.

Large spindle counts

Distribution of data across many disks is effective for random I/O workloads where the hosts generate enough concurrent requests to keep the disks busy. High concurrency can be achieved when applications have concurrent processes or threads or if the processes use random I/O.

A large disk count allows concurrent requests to execute independently. For workloads that are random and bursty, striped metaLUNs are ideal. MetaLUNs that share RAID groups ideally have their peaks at different times. For example, if several RDBMS servers share RAID groups, activities that cause checkpoints should not be scheduled to overlap.

How many disks to use in a storage system

There are plateaus in performance where adding disks does not scale workload linearly. These are addressed in detail in "Sizing the storage requirement" but the following are some rough guidelines for *strictly maximizing performance*. The drive counts presented in Table 9 are for *concurrently active* drives, under constant and moderate-to-heavy load.

Table 9. System high-efficiency/high-performance drive counts

CX3 model	IOPS per drive*	Bandwidth per drive**	Maximum drives
CX3-80	300	60	480
CX3-40	180	30	240
CX3-20	120	20	120
CX3-10	60	20	60

* For IOPS, the bottleneck moves from the disks to the SP at these drive counts.

** For bandwidth, the bottleneck moves from the BE loops to the SP at these drive counts.

These considerations are for clients whose absolute top priority is performance. As drives are added to the systems listed in Table 9 performance increases; however, the increase is not be linear.

Disk types, speeds, and capacities

The CLARiiON supports two types of hard drives, Fibre Channel (FC) and Serial ATA (SATA). FC drives are available in two rotational speeds, and two interface transfer rates. SATA drives have a single rotational speed and interface transfer rate. Each of the supported drive types comes in more than one capacity. Table 10 shows the hard drives available with the CLARiiON.

Table 10. CLARiiON hard drives

Hard drive type	Rotational speed (rpm)	Transfer rate (Gb/s)	Capacity (GB)
Fibre Channel	15,000	4	146
			73
	10,000	2	300
			146
			73
SATA	7,200	4	750
			500

Fibre Channel drives

Fibre Channel drives are enterprise-class devices. They contain performance enhancing features such as on-disk firmware capable of highly efficient queue reordering, buffering, and advanced seek optimizations. In addition, they meet strict operating environment and reliability standards.

FC type drives have short data access times. They perform very well with workloads characterized by random I/O. As a result, the 15k rpm drives offer about a 20 percent to 30 percent real-world increase in maximum random load on a system.

SATA drives

SATA drives are modestly performing, low-cost, high capacity devices. They contain features such as queue reordering and buffering that support their use in RAIDs. SATA has more relaxed operating environment and reliability standards than FC hard drives.

SATA drives behave very well with single threaded sequential workloads. In addition, they are very economical with workloads that require high storage densities. For sequential workloads, SATA drives achieve greater than 80 percent of the bandwidth of FC drives.

The SATA drives have been used in random environments where the IOPS requirement was modest. When comparing random I/O performance to FC drives, depending on the workload, SATA drives have about one-third to one-half the performance. The greatest performance difference occurs with smaller I/O sizes at higher thread counts.

Rotational speed

For any hard drive, higher rotational speed results in higher performance. A high rotational speed allows the drive to seek linearly along the tracks in less time. Furthermore, the higher-rpm drives include improvements in lateral seeks.

For random read performance, a faster drive rotational speed means less latency, as the cache cannot buffer random reads. For random writes, which are absorbed by the SP write cache, the effect of higher rotational speed is seen in faster cache flushing. A cache that flushes faster allows a higher rate of I/O to the storage system before watermark and forced flushing cause service times to increase.

Sequential operations benefit from high rotational speeds, but the effect will be less.

Transfer rate and drive capacity

The effect of a faster transfer rate is very small compared to a hard drive's rotational speed. Transfer rates increase with rotational speed. However, the drive buffer and the Fibre Channel and SATA protocols limit the practical transfer rate of the drives. In addition, the layers of system software and hardware between the drive and the host further reduce the effect of the drive's internal transfer rate on the overall external transfer rate.

A higher transfer rate does allow for more hard drives to be added to a back-end loop before overall performance is adversely affected. Overall performance depends on the number of spindles. Having more smaller drives will provide better performance than fewer larger drives at the same speed. A higher transfer rate supports a greater number of spindles.

For example, with the introduction of the 300 GB FC drives, the CLARiiON SPEED (performance guru) community was flooded with queries: What is the performance of the 300 GB drive? The answer: Exactly the same as the other 10k rpm FC drives. However, since performance tracks spindle count, the user must take account of performance needs as well as capacity requirements when choosing drives.

Mixing disk types in a DAE

SATA drives and Fibre Channel drives cannot be mixed in a DAE.

A mix of drive speeds and sizes of the same hard drive type may be used within a DAE. If drive speeds are mixed within a DAE, the placement becomes a planning issue. If a storage system uses disks of different capacities and/or speeds (for example, 146 GB and 73 GB, or 10k and 15k rpm) within any enclosure, then EMC recommends that you place them in a logical order, such as the following:

- Place drives with the highest capacity in the first (left-most) slots, followed by drives of lower capacities.
- Be aware that in bandwidth applications adding a 2-Gb/s interface drive on a bus with all 4-Gb/s interface drives will reduce the overall bus speed to 2 Gb/s with a possible effect to performance.

SATA drives and Fibre Channel drive types cannot be mixed in a DAE.

RAID group partitioning and CX300/500/700 SATA drives

When partitioning a RAID group with CX300/500/700 SATA drives, it is best to assign all LUNs from that RAID group to the same SP. This allows the highest bandwidth to the drives.

SATA drives as mirror targets and clones

SATA drives are not the drives of choice for clone and MirrorView targets. They can be used in some cases when the workload has been carefully shown to fit SATA performance characteristics. The performance profile of a disaster recovery (DR) storage site should mirror that of the primary site, so SATA drives are appropriate only if the primary site uses SATA storage. Careful planning of a DR site includes planning for sufficient performance and extended operations.

When used for clone or mirror targets, the performance effect of SATA drives must be considered with care and attention to detail. When in sync, the SATAs will receive all the writes of the primaries, but flush them slower, resulting in more load on the write cache. While syncing, they have the concurrent write load plus the sync load to absorb, further stressing the cache. Also, any reads against in-sync SATA drives will result in contention, with low sequential read rates and slower cache flushes. Periodic updates, such as with MirrorView/Asynchronous, will load the cache during the update.

In a system that is not being stressed (for example, the write cache is not hitting watermark flushes), the use of SATA drives as targets compared to FC drives will have a modest effect on system performance. Keep the IOPS load for SATA drives below the figures given in Table 6. The system should be monitored to ensure the use of SATA targets has not driven cache utilization consistently over the high watermark.

In a system that is already experiencing a heavy cache load, synchronization or establishment of a clone implemented on SATA drives could cause the write cache to fill. This would cause forced flushes for other LUNs being written—with a negative effect on other LUNs in the system. If Navisphere Analyzer shows extended periods of 100 percent dirty pages, SATA workloads should be suspected immediately. Write cache at individual SATA LUNs can be disabled as a means of controlling cache while alternatives (more drives, different RAID type, restricting write throughput by applying Navisphere Quality of Service Manager limits, or migration to FC disks) are considered.

Similarly, with SATAs as a synchronous mirror target for Fibre Channel primaries, the cache will flush more slowly on the target than at the source (due to slower drives being used). The target system's cache must be monitored, as forced flushes will result in an increase in response time for mirrored writes. Asynchronous MirrorView and incremental SAN Copy also write random I/O to the target, so carefully consider the use of SATA drives in that role as well.

The latest replication recommendations can be found on [EMC Powerlink](#).

Considerations for reliability and redundancy

A reliable and redundant storage network starts with SAN design, which is beyond the scope of this white paper. However, some aspects of storage system design—such as disk and RAID types chosen—are inherently storage-system subjects. In addition, the presentation of communication paths between the host and the disk array can be used to ensure reliability and redundancy.

A point to remember is that dynamic LUN migration makes it possible to change RAID and disk types after initial implementation.

Highly available attach

Highly available attach methodologies are definitely a best practice for implementing CLARiiON storage. The *EMC CLARiiON Open Systems Configuration Guide* outlines the attach methodologies and equipment that are supported.

The Navisphere Server Utility (NSU) offers a tool set to validate the high availability of the storage system attaches. EMC recommends using the NSU to periodically ensure the high availability of the attaches. Refer to the white paper *EMC CLARiiON High Availability (HA)* on [EMC.com](#) and [Powerlink](#) for more information.

Binding RAID groups across buses and DAEs

Engineers who have experience with older SCSI-based systems expect to bind parity RAID groups with each disk in a separate enclosure. This was done when each enclosure was served by a separate SCSI bus. Considerations for availability due to SCSI failure semantics no longer apply. However, there are several things to consider when binding disks.

Binding across DAEs

Few subjects cause as much concern and confusion in the field as the binding of disks across DAEs. Is there a performance advantage? Is there a redundancy advantage? In both cases, it depends on the RAID configuration, and in all cases *the differences are slight*.

Parity groups (RAID 3, RAID 5, RAID 6)

Binding parity RAID groups such that each drive is in a separate DAE does not help performance. However, there is a small increase in data availability in this approach. However, this is very unwieldy; if very high availability is required, use RAID 1/0. Refer to “Binding across back-end buses,” which follows.

RAID 1/0 groups

There is absolutely no advantage in binding a RAID 1/0 group in *more than* two DAEs, but it is not harmful in any way.

Binding across back-end buses

All current CLARiiON systems except the CX3-10, CX3-20, and CX300 have more than one set of dual back-end FC loops with which they attach to the DAEs. A disk group can be made up of drives from one, both, or all buses. The standard racking of DAEs alternates the buses across adjacent DAEs, with the DPE or first DAE being bus 0, the next DAE bus 1, and so on.

Parity groups (RAID 3, RAID 5, RAID 6)

Parity groups of 10 drives or more benefit from binding across two buses, as this helps reduce rebuild times. For example, bind a 10-drive RAID 5 with five drives in one DAE, and the remaining five drives in the DAE above it.

Mirrored groups (RAID 1, RAID 1/0)

Binding mirrored RAID groups across two buses increases availability and keeps rebuild times lower. This technique ensures availability of data in two (rare) cases of double failure: an entire DAE⁶ or redundant back-end bus (dual-cable failure). Bind the drives so that the primary drives for each mirror group are on the first back-end bus, and the secondary (mirror) drives are on the second back-end bus. Binding across buses also has a minimal but positive effect on performance.

When creating the RAID group (or defining a dedicated LUN in the bind command), use Navisphere CLI to bind across buses. When designating the disks, Navisphere CLI uses the disk ordering given in the `createrg` or `bind` command to create Primary0, Mirror0, Primary1, Mirror1, and so on, in that order. Disks are designated in `Bus_Enclosure_Disk` notation. Here is an example of binding the first two drives from enclosure on each bus:

```
Navicli -h <ip address> createrg 55 0_1_0 1_1_0 0_1_1 1_1_1
```

Binding with enclosure 0 drives

In a total power-fail scenario, the Standby Power Supply (SPS) supplies battery-backed power to the SPS and Vault disks. This allows the storage system to save the contents of the write cache to disk.

However, the power to the non-Vault disk storage system enclosures (DAEs) is not maintained. When the storage system reboots, LUNs that had I/O outstanding are checked, using the background verify process, to verify that no writes in progress resulted in partial completions. The background verify is a fairly low-intensity process.

However, a LUN bound with some drives in the Vault enclosure (DPE, enclosure 0, or the first DAE, depending on the model) and with some drives outside of the Vault enclosure *may* require a rebuild, which is a more disk-intensive process. This affects performance to some degree on reboot.

To avoid a rebuild on boot, create groups with all disks either in or out of the Vault enclosure. If groups are split, follow these guidelines:

1. Do not split RAID 1 groups across the Vault enclosure and another DAE.
2. For parity RAID make sure at least two drives are *outside* the Vault enclosure.
3. For RAID 1/0, make sure at least one mirror (both the primary and secondary drive in a pair) is outside the Vault enclosure.

⁶ The active components of a DAE are redundant (power supply, cooling, LCC). The passive components are redundant as well, and are not subject to mechanical or thermal stress during normal operations.

Consistency of data replication

The use of SnapView, MirrorView, and SAN Copy are out of the scope of this paper, but the concept of consistency is very close to that of redundancy, and thus warrants attention here. (Refer to the EMC white papers covering these other products.)

When implementing mirroring or replication of any kind (including backups) of *multi-LUN-related data*, the engineer *must* take consistency into account. Multi-LUN-related data is data that spans more than one LUN and is interrelated. Examples are:

- Components of a relational database: data tables, logs, TEMP, and RBS
- Volumes in host volume sets (VERITAS, AIX LVM)
- Components of a messaging database
- Shared file systems

Multi-LUN consistency depends on certain behaviors in the storage system. For example, writes to a database that spans LUNs must remain consistent across those LUNs. The three cases in which the storage design must address this requirement are clone fractures, snapshot starts, and mirroring.

MirrorView/A offers *in-array consistency groups*, which ensure that associated LUNs are updated following consistent splits. MirrorView/S has consistent fractures. SnapView offers consistent splits of clones and consistent start of snapshot sessions.

ALUA

Asymmetric Logical Unit Access (ALUA) is new to release 26.

ALUA can mitigate the effect of some front- and back-end failures to the host. It provides a new way of path management, by permitting I/O to stream to either or both CLARiiON storage processors. It follows the SCSI SPC-3 standard for I/O routing. The white paper *EMC CLARiiON Asymmetric Active/Active Feature* available on EMC.com and EMC Powerlink provides an in-depth discussion of ALUA features and benefits.

Path management

ALUA provides access to LUNs from the front-end ports of either CLARiiON storage processor. This does not change the traditional CLARiiON LUN ownership model. However, with ALUA I/O requests directed at the ports of storage processors without ownership of the destination LUN are automatically redirected. This supports native failover mechanisms such as MPIO—which is the main benefit to ALUA. For PowerPath users, additional, ALUA host-based features are available with the use of PowerPath version 5.1 or later.

The capability for redirection creates two types of paths. The direct path from host to owning SP to LUN is called the *optimal* path. The path to the non-owning SP is the *non-optimal* path. The non-optimal path provides alternate access to the LUN. The non-optimal path is the alternative path used by host-based failover software. The I/O response time is typically increased when using the non-optimal path.

CLARiiON ALUA automatically manages storage processor ownership of LUNs. Excessive use of the non-optimal path triggers a transfer of LUN ownership. The transfer of LUN ownership from one storage processor to another is called a *trespass*. There are two types of trespass. An *explicit* trespass is when the optimal path is configured manually through the Navisphere GUI or CLI. An *implicit* trespass is when LUN ownership is changed, through ALUA.

An implicit trespass is triggered when the great majority of I/O requests for a LUN occur over the non-optimal path. In this case, the LUN is trespassed to the peer storage processor so the path with the most active I/O requests becomes the optimal path. High thresholds of detection are used to avoid excessive changing of LUN ownership and high rates of trespassing. This implicit trespassing of LUNs ensures the shortest I/O response time.

Usage

Before ALUA is enabled, LUN ownership and loads should be balanced across storage processors as outlined in this white paper. That is, initially I/O traffic is routed over the manually defined optimal path, which should be the normal operational mode of the system. In addition, LUN load balancing should be performed across ports on the owning storage processor.

Verify that all I/O requests are using the manually defined optimal paths before enabling ALUA.

ALUA is enabled per initiator by setting the Failover Mode to 4. This is a new initiator setting.

For best performance, verify that all requests are using optimal paths following any SAN changes affecting paths, and after any SAN upgrades.

Host considerations

PowerPath version 5.1 is the first ALUA-compliant release. Ensure that your version is 5.1 or later, and reference Table 11 for host compliance.

PowerPath will load balance across optimized paths. It will only use non-optimized paths if all optimized paths have failed. If path or storage processor failures occur, PowerPath shall initiate a trespass to change LUN ownership. That is, the non-optimized paths will become the optimized paths, and failed previous optimized paths become non-optimized.

In addition, MPIO and other native host-based failover applications can interoperate with ALUA.

OS considerations

To take advantage of ALUA features, host software needs to be ALUA-compliant. Several operating systems support native failover with Active/Passive (A/P) controllers. Table 11 shows the current ALUA status of some common OSs.

Table 11. ALUA-compliant operating systems

O/S	PowerPath with ALUA	Native with ALUA	PowerPath with A/P	Native with A/P
MS Windows Server 2007	No (Q4 '07)	Yes	Yes	No
W2K3	PP 5.1	No	Yes	No
Win2K	PP 5.1	No	Yes	No
HP-UX 11i v1 and v2	No (Q4 '07)	No	PP 5.1	Yes (LVM)
HP-UX 11i v3	No (Q4 '07)	Yes	PP 5.1	No
Solaris 9/10	PP 5.1	Yes	Yes	Yes
Linux (RH and SuSE)	No (Q4 '07)	SLES 10 SP1	Yes	Yes
AIX	No	No	Yes	No
VMware	No	No	No	Yes

Definition of columns in table:

- **PowerPath with ALUA:** ALUA features are provided when the specified PowerPath release is used with the noted operating system.
- **Native with ALUA:** ALUA features are provided when using the noted operating system alone (PowerPath is not required.)
- **PowerPath with A/P:** Standard Active/Passive failover (not ALUA) is provided by PowerPath with the noted operating system. (PowerPath issues trespass commands to enable alternate paths by changing SP LUN ownership.)
- **Native with A/P:** Standard Active/Passive failover (not ALUA) is provided when using the noted operating system alone. (OS issues trespass commands to enable alternate paths.)

Performance considerations

The optimized path is the normal operation path. ALUA has no effect on optimized path performance.

Performance is adversely affected on the non-optimized path.

Host I/O requests received over non-optimized paths are received by the storage processor not owning the destination LUN. These requests are then forwarded to the peer storage processor owning the LUN. The owning storage processor executes the I/O as though the request had been received directly. When the I/O completes, data or acknowledgements are forwarded back through the peer to be transmitted to the host.

The redirection, from storage processor to storage processor, and back increases I/O response time. The duration of the delay is dependent on the overall disk array and storage processor workloads and the size of the I/O. Expect a 10-20 percent decrease in maximum IOPS, and up to a 50 percent decrease in bandwidth with non-optimum path usage.

Monitoring ALUA performance

A number of metrics have been created to describe requests arriving over optimized versus non-optimized paths. This path usage can be monitored through Navisphere Manager. Navisphere Analyzer likewise

provides information on path usage. In addition, metrics exist for total I/O over all paths. These metrics describe the utilization of paths and the differences in performance.

Sizing the storage requirement

Storage sizing consists of calculating the right number of drives for capacity, and calculating the correct number of drives and the right storage system for performance.

Capacity planning

Now is the time to determine the RAID type and disk-group size (this affects capacity in parity RAID types). Once that is known, the performance calculation can be made. EMC personnel have tools to determine the exact usable capacity of a provisioned system. These tools take into account various factors that affect final capacity, such as the extended redundancy data stored with each sector on a CLARiiON storage system.

Vault drives

The first five drives in a CLARiiON contain the Vault. Vault drives can be used just as any other drives on the system. However, Vault drives have less usable capacity (by about 33 GB). When bound with other, non-Vault drives, all drives in the group are foreshortened to match the Vault drive capacity. It is thus more efficient for capacity utilization to bind the Vault drives together as a group (or groups).

Actual drive capacity

Accessible capacity may vary because some OSs use binary numbering systems for reported capacity. Drive manufacturers consider a gigabyte to be 1,000,000,000 bytes (base 10 gigabyte). A computer OS may use base 2 (binary) and a binary gigabyte is 1,073,741,824 bytes. Many clients are surprised to find a 73 GB drive holds only 66 GB.

Also, CLARiiON uses eight additional bytes per 512 byte sector for storing redundancy information. This 520- byte sector reduces the usable capacity by a small margin.

Parity or mirrored protection

The use of parity or mirroring to protect data from disk failure will also reduce usable storage. Mirroring always requires 50 percent of total storage. The percentage of parity RAID space overhead is determined by the number of disks in the group and the type of parity RAID used. Parity RAID 3 and RAID 5 have a single drive of overhead. RAID 6 has a two drive overhead.

For example, a five-drive RAID 5 (4+1) has a 20 percent overhead. It has the equivalent of four drives available for storage and the overhead of one drive for parity.

Performance planning

Performance planning or forecasting is a science that takes considerable knowledge.. The threading model of the workload, the type of I/O (random or sequential), the I/O size, and the type of drive all affect the performance you will observe. The *EMC CLARiiON Fibre Channel Storage Fundamentals* white paper contains detailed information on the factors affecting performance.

Rule-of-thumb approach

To begin performance estimation, a rule of thumb is used for IOPS per disk drive and MB/s per disk drive. Use the guideline values provided in Table 12 for this estimate. This is a conservative and intentionally simplistic measure. It should be noted this is only the beginning of an accurate performance estimate; estimates based on the rule of thumb are for quickly sizing a design. More accurate methods are available to EMC personnel.

Table 12. Data access rate guidelines for disk drives

Drive Type	Drive speed (RPM)	IOPS*	Bandwidth** (MB/s)
SATA	7,200	80	8
Fibre Channel	10,000	140	10
Fibre Channel	15,000	180	12

*IOPS 2 KB to 8 KB random requests.

**Bandwidth assumes 128 KB and larger sequential requests.

SATA drives are not recommended for *sustained* random workloads at these rates, due to their duty cycle limitations.

Remember these are rules of thumb.

Most installations will have more than a single thread active, but want to keep response times below 20 ms. To create a more conservative estimate, these response time sensitive applications may want to perform calculations assuming 120 or fewer IOPS per FC drive. In rare cases of random I/O sizes greater than 256 KB, there may be much fewer IOPS. In cases of well-behaved sequential access, the rate may be well above 300 IOPS for FC drives, even for large I/O sizes.

For example, a single threaded application has one outstanding I/O request to the disk at a time. If the application is doing 8 KB random reads from a 10k rpm drive, it will achieve about 120 IOPS at 8 ms per I/O. The same application, reading a disk through 12 simultaneous threads of the same size, can achieve 215 IOPS at 50 ms per I/O.

Performance Estimate Procedure

The approach for a quick estimate is:

1. Determine the workload.
2. Determine the disk load.
3. Determine number of disk drives required.
4. Determine number and type of storage systems.

Determining the workload

This is often the most difficult part of the estimation. Many clients do not know what the existing loads are, let alone loads for proposed systems. Yet it is crucial for the client to make a forecast as accurately as possible. Some sort of estimate must be made.

The estimate must include not only the total IOPS, but also what percentage of the load is reads and what percentage is writes. Additionally, the predominant I/O size must be determined.

Determining the disk load

Note that the IOPS values in Table 12 are *disk IOPS*. To determine the number of disk IOPS implied by a host I/O load, adjust as follows for parity or mirroring operations:

Parity RAID 5 and 3: Disk IOPS = Read IOPS + 4*Write IOPS

Parity RAID 6: Disk IOPS = Read IOPS + 6*Write IOPS

Mirrored RAID: Disk IOPS = Read IOPS + 2*Write IOPS

For example, if a RAID 1/0 group size is 2+2 (four disks in group), and the I/O mix is 50 percent random reads and 50 percent random writes with a total host IOPS of 10,000:

$$\text{IOPS} = 0.5 * 10,000 + 2 * (0.5 * 10,000)$$

$$\text{IOPS} = 15,000$$

For bandwidth calculations, when you expect large or sequential I/O to fill LUN stripes, use the following approaches, where the write load is increased by a RAID multiplier:

Parity RAID 5 and 3: Disk MB/s = Read MB/s + Write MB/s * (1 + (1/ (number of disks in group)))

Parity RAID 6: Disk MB/s = Read MB/s + Write MB/s * (1 + (2/ (number of disks in group)))

Mirrored RAID: Disk MB/s = Read MB/s + Write MB/s * 2

For example, if a RAID 5 group size is 4+1 (five disks in group), the read load is 100 MB/s, and write load is 40 MB/s:

$$\text{Disk MB/s} = 100 \text{ MB/s} + 40 \text{ MB/s} * (1 + (1/5))$$

$$\text{Disk MB/s} = 148 \text{ MB/s}$$

Calculate the number of disk drives required

You must make both a performance calculation and storage capacity calculation to determine the number of drives in the storage system.

Performance capacity

Divide the total IOPS (or bandwidth) by the per-disk IOPS value provided in Table 12. The result is the approximate number of drives needed to service the proposed I/O load. If performing random I/O with a predominant I/O size larger than 8 KB, increase the disk count by 20 percent.

Storage capacity

You also need to calculate the number of drives required for your storage capacity. Ideally, the number of drives needed for the proposed I/O load is equal to the number of drives needed to satisfy the storage capacity requirement. See the previous section “Capacity planning” for an explanation of a storage capacity estimate. Use the *larger* number from the Performance capacity and Storage Capacity estimates.

Furthermore, the vault requires five drives, and it is prudent to add one hot spare drive per 30 drives (rounded to the nearest integer) to the drive count. It is important to remember not to include the vault and hot spare drives into the performance calculation when calculating the operational performance.

Total disk drives

Total Approximate Drives = RAID Group IOPS / (Hard Drive Type IOPS) + Large Random I/O adjustment + Hot Spares + Vault

For example, if a RAID group was previously calculated to execute 8,000 IOPS, the I/O is 8 KB requests, and the hard drives specified for the group are SATA drives (see Table 12):

Total Approximate Drives = $8,000 / 80 + 0 + ((8,000 / 80) / 30) + 5$

Total Approximate Drives = 108

Calculate number and type of storage systems

Once the number of drives is estimated, they must be matched to a storage system or set of systems that supplies performance, capacity, and value to the client.

For performance-sensitive deployments, refer to Table 9 to select the storage system whose high-efficiency/high-performance drive count—the sweet spot—best fits the client's requirement. Divide the number of drives by the high-efficiency/high-performance drive count to determine the number of storage systems necessary to service the required disk drives effectively.

For deployments where performance is not critical, use up to the maximum drive count for the chassis. Divide the number of disk drives by the chassis maximum to determine the number of storage systems to deploy.

Storage systems

Storage Systems = Approximate Drives / Storage System Drive Count

For example, if a high-performance storage system requires approximately 135 drives (see Table 9):

Storage Systems = $135 / 180$

Storage Systems = 1 (CLARiiON CX3-40 solution)

Resolving performance and capacity needs

The number of disks in a system is determined by the performance and capacity needs. The method described previously calculates the minimum number of disks needed to meet performance requirements.

A separate estimate is required using different disk sizes to calculate the number of disks needed to meet capacity requirements. The final number of disks used is determined by interpolating to determine the number of drives that meet both performance and capacity requirements.

Sizing example

The following example uses the procedure just described.

Step 1: Determine the workload

Having an accurate description of the workload is the first, most important step. Any error or ambiguity in the workload will adversely affect the sizing calculations and may result in a system unsuitable for the client's needs.

The following example includes the minimum workload requirements needed to perform a sizing estimate.

A client has a storage requirement for an object database. The application is expected to execute random I/Os. The profile is:

- 70 percent random reads
- 30 percent random writes
- Predominant I/O size is 8 KB
- Total host IOPS is 20,000
- Usable capacity required is 10 TB

The client is price-sensitive, so a large parity RAID is desired. However, the write percentage (30 percent) is high enough that RAID 1/0 should be considered as well.

Step 2: Determine the required I/O disk load

Calculate the IOPS for the three RAID types being considered: RAID 5, RAID 6, and RAID 1/0.

The disk load is:

$$\text{RAID 5 (8+1): } 0.7 * 20,000 + 4 * 0.3 * 20,000 = 38,000 \text{ disk IOPS}$$

$$\text{RAID 6 (8+2): } 0.7 * 20,000 + 6 * 0.3 * 20,000 = 50,000 \text{ disk IOPS}$$

$$\text{RAID 1/0 (8+8): } 0.7 * 20,000 + 2 * 0.3 * 20,000 = 28,000 \text{ disk IOPS}$$

From an IOPS perspective, RAID 1/0 would be the choice in this step.

Step 3: Determine the number of drives required

Calculate the number of hard drives for each RAID type needed assuming 15,000 rpm FC drives with hot spares and vault drives included in the total. The performance capacity is the IOPS divided by the drive IOPS (from Table 10).

$$\text{RAID 5 (8+1): } 38,000/180 + ((38,000/180)/30) + 5 = 211 + 7 + 5 = 223 \text{ disk drives total}$$

$$\text{RAID 6 (8+2): } 50,000/180 + ((38,000/180)/30) + 5 = 278 + 11 + 5 = 292 \text{ disk drives total}$$

$$\text{RAID 1/0 (8+8): } 28,000/180 + ((28,000/180)/30) + 5 = 156 + 9 + 5 = 166 \text{ disk drives total}$$

From a number of drives perspective, RAID 1/0 would be the choice in this substep.

The number of drives needed to achieve performance needs to be resolved with the available hard number of drives needed to meet the storage requirement. This calculation is performed using data disks only. Do not include vault and hot spare disks in the calculation. Assume 73 GB FC drives with a formatted capacity of 66 GB.

$$\text{RAID 5 (8+1): } 8/9 * 211 * 66 = 12.4 \text{ TB}$$

$$\text{RAID 6 (8+2): } 8/10 * 278 * 66 = 14.7 \text{ TB}$$

$$\text{RAID 1/0 (8+8): } \frac{1}{2} * 156 * 66 = 5.1 \text{ TB}$$

From a capacity perspective, RAID 5 would be the choice in this substep.

Step 4: Determine number and type of storage systems

The previous steps result in Table 13. This table is used to compare the RAID options toward making a storage system selection.

Table 13. Sizing estimate calculation results

RAID type	Disk load (IOPS)	Performance capacity (drives)	Storage capacity (TB)	Total drives
RAID 5	38,000	211	12.4	223
RAID 6	50,000	278	14.7	292
RAID 1/0	28,000	156	5.1	166

Using Table 9, determine which CLARiiON model most closely matches the calculated performance capacity, storage capacity, total drives, and stated requirements.

In this example, a CX3-40 with RAID 5 would be a good candidate. A reasonable RAID 5 solution would be 211 drives for data, seven hot spares, and five drives for the vault (and archive space). Adding a DAE containing 15 SATA drives for backup would make the final number of drives 238.

Note that the RAID 5 solution would need to move to a larger CX3-80, if a requirement for clones, snaps, or mirrors is added.

Other considerations

A short discussion of why the RAID 1/0 and RAID 6 solutions were not chosen and how to make them successful candidates may be useful.

The RAID 1/0 solution has the needed IOPS for the client's system. However, it was not selected, because it does not meet the needed storage capacity. A RAID 1/0 would require more than 300 76-GB FC drives to meet the client's storage capacity requirement. This would require a CX3-80 to house them. A possible RAID 1/0 solution would be to use higher capacity drives. Either 15k rpm or 10k rpm 146-GB FC drives could be used to provide the needed 10 TB storage capacity. (A 146 GB FC drive formats to 133 GB.) Using larger drives would provide an approximately 200-drive RAID 1/0 solution.

The RAID 6 solution more than meets the client's storage capacity requirement. In addition, it provides greater data security than the RAID 5 solution. It was not selected because the IOPS and total drive count would require the larger CLARiiON CX3-80 to house it. (The client is stated to be "price sensitive.") However, if added requirements for growth, clones, snaps, and mirrors occur, the RAID 6 solution becomes competitive.

The final choice of a CLARiiON model involves weighing many factors. Capacity, performance, availability, growth, and cost objectives must all be considered. This example introduces you to the most important factors involved in the decision.

CLARiiON software update

EMC has developed a process to ensure a successful software update to the CLARiiON storage systems. A detailed guide is available under the CLARiiON Tools section on EMC [Powerlink](#). High availability (HA) of the attaches is especially important when performing online software updates (also commonly referred as non-disruptive upgrade or NDU). Data loss and data unavailability situations during NDU are primarily caused by the lack of HA configuration for the attaches. EMC recommends using the Navisphere Server Utility to validate HA prior to performing a software update on the storage system.

Conclusion

Performance, reliability, redundancy, and capacity are the primary considerations in choosing storage.

With a CLARiiON Fibre Channel storage system, getting very good performance—what a large number of our customers need—is fairly straightforward. To squeeze the last bit of performance from the storage system, the operator must be aware of host-based effects and the geometry of the RAID configuration. The applications analysis and host memory configuration *must* be done before tuning the storage system.

Reliability is built into the storage system—with greater than 99.999 percent availability. Redundancy is easy to achieve, but more costly than non-highly available configurations. Much of the redundancy of the CLARiiON storage system is built in—dual SPs, hot-swappable disks, redundant RAID, and hot spare disks. The choices made in the field are in the attach methodology and RAID types chosen. For the cost-conscious client, the costs of *not* building redundancy into the attach configuration must be explained. RAID 1/0 is the gold standard for redundancy.

Determining capacity is straightforward using EMC tools. Use the performance white papers as a guide to determine the number and model of systems to deploy for any given I/O profile and load.

References

The following white papers are available on EMC.com and Powerlink:

- *EMC CLARiiON Asymmetric Active/Active Feature - Applied Technology*
- *EMC CLARiiON Fibre Channel Storage Fundamentals*
- *EMC CLARiiON High Availability (HA) - Best Practices Planning*
- *EMC CLARiiON RAID 6 Technology - A Detailed Review*
- *Practice Considerations for CLARiiON CX3 FC/iSCSI Storage Systems*