EMC Unified Storage Best Practices for Performance and Availability Common Platform and Block Storage 31.0

Applied Best Practices

Revised: 6/23/2011 4:57 PM

EMC Unified Corporate Systems Engineering
Corporate Headquarters
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Published June, 2011

EMC Unified Storage Best Practices for Performance and Availability— Common Platform and Block Storage 31.0 -- Applied Best Practices

P/N **h8268**

Contents

About this Document

This document provides an overview of the performance and availability Best Practices for the EMC VNX-series. It was developed by the EMC Corporate Systems Engineering group.

Purpose

Information in this document can be used as the basis for general best practices for performance and availability of the VNX platform for both block and file storage.

Audience

This document is intended for VNX storage system administrators, EMC field and sales personnel, partners, and new customers. EMC service and sales organization may use parts of this document in sales kits and other technical documentation.

Scope

This document examines best practices for achieving best performance and availability with EMC® VNX™ series storage systems. It discusses factors influencing VNX-series availability and performance at the host, network, and storage system level using the VNX Block Operating Environment 31.0 (VNX OE Block 31.0). Specific VNX models addressed include:

- VNX5100 (Block storage only)
- VNX5300
- VNX5500
- VNX5700
- VNX7500

This document is designed to address the most common performance and availability related situations. Not all circumstances and platform configuration may be covered. Contact your EMC Sales representative to engage a USPEED professional for very high performance and availability provisioning of VNX storage systems. These systems require deeper analysis, and software tools available only with EMC professionals.

Related documents

The following documents, located on Powerlink.com, provide additional, relevant information. Access to these documents is based on your login credentials. If you do not have access to the content listed below, contact your EMC representative:

- An Introduction to EMC CLARiiON CX4 Disk-Drive Spin Down Technology,
- EMC CLARiiON and Celerra Unified FAST Cache,
- EMC CLARiiON and Celerra Unified Storage Platform Storage Device Technology,
- EMC CLARiiON Asymmetric Active/Active Feature

- EMC CLARiiON Best Practices for Fibre Channel Storage: FLARE Release 26 Firmware Update
- EMC CLARiiON Best Practices for Performance and Availability: FLARE Release 30 Firmware Update
- EMC CLARiiON Global Hot Spares and Proactive Hot Sparing
- EMC CLARiiON Storage Solutions: Microsoft Exchange 2007 - Best Practices Planning
- EMC Data Compression: A Detailed Review
- EMC Networked Storage Topology Guide
- EMC PowerPath Product Guide
- EMC Storage Arrays Configuration Guide
- EMC Unified Affect of Priorities on LUN Management Activities
- EMC Unified Storage Device Technology
- EMC Unified Storage System Fundamentals for Performance and Availability
- EMC VNX Virtual Provisioning: Applied Technology
- EMC® Host Connectivity Guide for Linux
- Native Multipath Failover Based on DM-MPIO for v2.6.x Linux Kernel
- Unified Flash Drive Technology Technical Notes
- Using EMC CLARiiON Storage with VMware Infrastructure and vSphere Environments TechBook
- VLAN Tagging and Routing on EMC CLARiiON,

Questions and comments about this document?

If you have questions or comments about this document, please send them using the 'Feedback to Author' link next to the document title within Powerlink.

Using Best Practices

How to Use Best Practices

This paper covers 'the general case' for getting additional performance and availability from the VNX-series storage system. Before applying any of the recommendations found in this document, the user of this document *must* already have:

- A firm understanding of *basic* storage system operating practices and procedures
- An through understanding of their *own* storage environment, including applications
- The extent of their available resources, including: time, skilled labor, and budget
- Their business's requirements and priorities

Some experience analyzing and designing for performance would also be a helpful.

In some cases, this paper offers more than one recommendation. The recommendation you follow will depend on your application and business priorities.

For example, it may not be possible to achieve both high-performance *and* high-availability with the resources at hand. Likewise, it is possible that the performance best practices for block and file storage may differ. When provisioning a storage system, be prepared to make the hard decision on whether file *or* block, *or* performance *or* availability is more important, and to proceed accordingly.

General Best Practices have been highlighted in the text. In addition, they have been included in a reference at the end of the document.

Finally, tuning parameters and software features change from revision to revision of the EMC file and block operating environments. Techniques and parameters used in tuning the host, network, or storage system on legacy EMC storage systems may no longer work, or have unexpected results on the VNX. Likewise, VNX storage system parameters and tuning techniques may not apply to legacy storage system. Be sure to use the appropriate version of Best Practices for your target storage system series.

File Environment Best Practices

In many places in this document here are references to File Best Practices. This is the companion document to this one describing the separate best practices for VNX OE File 7.0 environment installations.

That document is *EMC Unified Storage Best Practices for Performance and Availability – File 7.0.* It is available on Powerlink.

Legacy Storage System Best Practices

This document specifically addresses the VNX 5000 and 7000-series mid-range storage systems.

If you need guidance with CLARiiON CX4 series storage system's please see *EMC CLARiiON Best Practices for Performance and Availability: FLARE Release 30 Firmware Update.*

CLARiiON CX3 and the earlier CLARiiON series storage systems, please see the *EMC CLARiiON Best Practices for Fibre Channel Storage: FLARE Release 26 Firmware Update* white paper.

These papers are in 'maintenance', and are only updated when patches to the FLARE revision supporting these legacy CLARiiONs changes. Both of these documents are available on Powerlink.

Storage System Fundamentals

EMC Unified Storage System Fundamentals for Performance and Availability is a document written for users who are new to EMC storage systems. At a high level, this document describes how the VNX features work. It also defines the technical vocabulary used in Best Practices and EMC Unified performance and availability oriented documentation. If you are unfamiliar with EMC storage systems, it is *strongly* recommended you read this document.

Terminology

A common vocabulary is needed to understand the sometimes highly technical and EMC product specific recommendations included in this document.

Most of the terms in this document are IT industry standard. However, some are specific to EMC manufactured storage systems. In addition, there are some differences in terms between the VNX File and Block operating environments.

For example, in this document, the term *drive* refers to both *mechanical hard drives* and *Enterprise Flash drives* (*flash drives* or *EFDs*). Flash drives are non-volatile, semiconductor-based storage devices that are sometimes referred to as solid state disks (SSDs) in the IT industry.

A second example is the term *pool*. Virtual Provisioning pool refers to storage implemented on the VNX's Virtual Provisioning feature. Note that other 'pools' exist within the context of the VNX. For example, File Storage Pools and the Reserved LUN Pool. Be sure to verify the context of this frequently used term.

A final example is the common usage of the terms *bus* and *loop*. The VNX's SAS backend has no buses and loops. The terminology of the SAS protocol, which is to have *ports* and *links* is used throughout this document.

To help with the document's vocabulary, a glossary has been included at the end of this document to define EMC and operating specific terms.

Common Best Practices

There are rarely simple answers on how to design, configure, and tune large, complex, multi-vendor, computer-based systems. However, the following are general best practices for getting optimal performance and availability from a VNX-series storage system:

- Read the manual.
- Install the latest firmware.
- Know the workload.
- Use the default settings.
- Resolve problems quickly.

Read the manual: Become familiar with your VNX's hardware by reading the *Introduction to the VNX Series* white paper, and the Hardware and Operational Overview for your model VNX. (For example, the overview for the VNX7500 is the *VNX Model 7500 Systems Hardware and Operational Overview.)* In addition, familiarize yourself with the system's user interface software by browsing the *Unisphere Manager online help*. Many answers to questions can be found there. The help information is directly available on the storage system.

Install the latest firmware: Maintain the most recent firmware release and patch level practical. Stay current with the regularly published release notes for VNX. They provide the most recent information on hardware and software revisions and their effects. This ensures the highest level of performance and availability known to EMC. Have a prudent upgrade policy and use the VNX's Non-disruptive Update (NDU) to upgrade, and maintain the highest patch level available without adversely affecting your workload. Follow the procedure for *VNX Software Update (Standard)* available on Powerlink to update the VNX's firmware to the latest revision.

Know the workload: To implement best practices, you need to understand the storage system's workload(s). This includes knowledge of the host applications. Always remember, that when the workload's demands exceed the storage system's performance capabilities, applying performance best practices has little effect. It is also important to maintain historical records of system performance. Having performance metrics *before* applying any best practices is needed to evaluate results. Accurate record keeping always saves time and labor when tuning. Finally, be aware of any planned changes in the workload or overall system configuration. An application update or major revision can have a big effect on the storage system's workload. This will help to understand and prepare for the change's effect on overall system performance. EMC recommends using Unisphere™ Analyzer to monitor and analyze performance. Monitoring with Analyzer provides the baseline performance metrics for historical comparison. This information can give early warning about unplanned changes in performance.

Use the default settings: Not all workloads require tuning to make the best use of the VNX storage system. The system's default configuration settings have been designed and tested to provide a high level of performance and availability for the largest number of workloads and storage system configurations. When in doubt, accept and use the system's default settings. In addition, use conservative estimates with configuration settings and provisioning when making changes.

Resolve problems quickly: The storage system continuously monitors itself and can be configured to generate alerts, warnings, and centralized, comprehensive logs and reports. Get your system out of degraded mode as quickly as possible. Be proactive. Practice handling common problems, such as a failed drive replacement. Avoid a possibly serious problem later by periodically reviewing the logs and generating and reviewing system status reports.

Chapter 1 Host Best Practices

Host best practices advice on the software and hardware configurations of the server-class computers attached to the storage systems, and the effect they have on overall storage system performance and availability.

Performance

The following sections describe the best practices for the storage system that may be applied to the hosts.

Application tuning

The number of host applications, application's design, their configuration, and modes of execution directly determine the behavior of the overall system. Many enterprise-level applications, such as Microsoft Exchange and Oracle, have integrated performance and availability features. These features can be used to locate bottlenecks within the overall system, and for application tuning.

In many cases, application re-configuration or tuning yields greater performance increases with less time, labor and expense than either network or storage system re-configuration and tuning.

For example, re-indexing a relational database like Oracle to increase its may be performed more quickly, and less expensively than migrating underperforming LUNs to be hosted on faster storage devices.

Different I/O types

The operational design of the host's applications—how they are used and when they are used—affects the storage system load. Being able to describe the I/O of the workload is important to understanding which best practices to apply.

The I/O produced by application workloads has the following broad characteristics:

- Writes versus reads
- Sequential versus random
- Large-block size versus small-block size
- High locality versus low locality
- Steady versus bursty
- Multiple threaded versus single threaded

Writes versus reads I/O

The ratio of writes to reads being performed by the application needs to be known and quantified. Knowing the ratio performed by the application is needed to know which best practices for your cache, RAID group and LUN provisioning to apply to your workload.

Writes consume more storage system resources than reads. Writes going to the storage system's write cache, are mirrored to both storage processors (SPs), and eventually are sent to a storage device via the backend. When writing to a RAID group mirrored or parity data protection

techniques consume additional time and resources. In addition, storage devices including flash drives typically write more slowly than they read.

Reads generally consume fewer storage system resources, if only because most storage devices perform reads faster than writes. Reads that find their data in cache (a *cache hit*) consume the least resources. They can have the lowest host response time. However, reads not found in cache (a *cache miss*) have much higher response times than the hits. This is because the data has to be retrieved from drives.

Sequential versus random I/O

The type of I/O an application performs needs to be known and quantified. Knowing the I/O type determines which best practices for the cache configuration, RAID group and LUN provisioning to apply to your workload.

An application can have three types of I/O:

- Sequential
- Random
- Mixed

How well the VNX handles writes and reads depends on whether the workload is mainly sequential or random I/O.

Small random I/Os use more storage system resources than large sequential I/Os. (See block size below.) Random I/O throughput is affected by many additional factors within the storage system. Applications that only perform sequential I/O have better bandwidth than applications performing random or mixed I/O. Working with workloads with both I/O types requires analysis and tradeoffs to ensure both bandwidth and throughput can be optimized.

Note that use of flash drives is an exception. Flash drives are native random-access devices. They are very efficient at handling random I/O, particularly small block random I/O. See the section below for additional details.

Large block size versus small block size I/O

It is important to know the majority I/O size, and the distribution of I/O sizes, in-use by the workload's applications. This determines which best practices for the cache configuration and RAID group and LUN provisioning to apply to your workload.

Every I/O has a fixed and a variable resource cost that chiefly depends on the I/O size. Note this definition has changed over time, with larger block-sized I/O becoming more common. For the purposes of this paper, up to and including 16 KB I/Os is considered small, and greater than 64 KB I/Os are large. Doing large I/Os on a VNX delivers better bandwidth than doing small I/Os.

A low host response time needs a low access time. Small-block random access applications such as on-line transaction processing (OLTP) typically have much lower access times than applications using sequential I/O. This type of I/O may be constrained by maximum drive I/O operations per second (IOPS).

The use of a smaller or larger I/O block-size is typically application dependent. The decision to use a large request or break it into smaller sequential requests may require reconfiguration at the application level, at the Host Bus Adapter (HBA), and of its storage system LUNs.

High locality versus low locality

It is important to know workload's applications locality when planning for to use secondary caching and storage tiering. This determines the Best Practice for the capacity of the secondary cache and the tier provisioning.

With the inclusion of the secondary caching of FAST Cache and the FAST Virtual Provisioning feature's tiered storage within Virtual Provisioning pools, data locality has become important.

Locality is based on the data set's *locality of reference*. Locality of reference means storage locations being frequently accessed. There are two types of locality, "when written" and "where written."

An application is more likely to access today's data than access data created and written three years ago. *When* data is written, or *temporal* locality refers to the reuse of storage locations within a short period of time. A short duration is considered to be within seconds, minutes at most.

Recently created data is likely residing on a mechanical hard drive's LBAs that are near each other on the drive's sectors and tracks. *Where* data is located, or *spatial* locality refers to the distribution of in-use data within its address space. This may result in data being stored in nearby sectors or sectors on nearby tracks of a mechanical hard drive. However, with random access storage devices like flash drives, it refers to an address space.

Secondary caching and automated tiering exploit locality of I/O requests to achieve higher throughput by ensuring that data with high locality is on Flash drives.. These features assume temporal and spatial locality.

A workload with a data set having high locality of reference gets the best performance with secondary caching and storage tiering. The degree of locality within of the data set is also important to understand. It varies from application to application. The degree of locality is the percentage of the data set receiving the highest usage. A three to five-percent degree of locality is common, but 20-percent is easily possible.

For example, a 1.2 TB database with a 20 percent working data set has about 250 GB of frequently accessed capacity. Index tables within the database likely have a high locality. They are relatively compact, and frequently accessed. Likewise, there will also be tables in the data base that are very large and infrequently accessed; they have low locality. Caches and tiers would be sized capacity-wise to meet the high locality user data making-up the working set.

Note that applications exist with very low locality. For example, a workload with perfectly random I/O would have very low locality. Benchmarking applications, such as the public domain IOMeter™ I/O subsystem measurement and characterization tool are capable of generating perfectly random workloads. Likewise, sequential workloads have no locality.

Steady versus bursty I/O

Knowing the I/O pattern, when, how, and for how long the I/O pattern changes is needed to determine which best practices for the cache configuration to apply to your workload.

I/O traffic can be steady or can vary widely over a short period of time.. This varying I/O is sometimes called *bursty*. The traffic pattern can also change over time, being sporadic for long periods, and then becoming steady. It is common for storage systems to be configured for a random-access application during "business hours" and then to be reconfigured to require good sequential performance during "off hours" backups and batch processing.

Bursty behavior results in *spikes* of traffic. A spike is a sudden and unpredictable, large increase in activity. To manage spikes requires a margin of storage system performance resources be held in reserve. This includes uncommitted SP utilization capacity, I/O bandwidth, and storage capacity. This reserve, is needed to handle the "worst case" demand of the spike. Otherwise, user response times may suffer if spikes occur during busy periods.

Multiple threads versus single thread

It is important to know which I/O threading model is used for your workload's LUNs. This determines which best practices, particularly for flash drive usage, which may apply to your workload.

The degree of concurrency of a workload is the average number of outstanding I/O requests made to the storage system at any time. *Concurrency* is a way to achieve high performance by engaging multiple drives on the storage system. When there are more I/O requests the drives become busy and I/O starts to queue, which can increase response time. However, applications can achieve their highest throughput when their I/O queues provide a constant stream of I/Os.

The way those I/O requests are dispatched to the storage system depends on the threading model.

A *thread* is a sequence of commands in a software program that perform a certain function. Host-based applications create processes, which contain threads. Threads can be *synchronous* or

asynchronous. A synchronous thread waits for its I/O to complete before continuing its execution. This wait is sometimes called *pending*. Asynchronous threads do not pend. They continue executing, and may issue additional I/O requests, handling each request as they complete, which may not be the order in which they were issued.

Single-threaded access means only one thread can perform I/O to storage (such as a LUN) at a time. Historically, many large-block sequential workloads were single threaded and synchronous. Asynchronous single threads can still achieve high rates of aggregate performance as the multiple I/Os in their queues achieve concurrency. *Multithreaded* access means two or more threads perform I/O to storage at the same time. I/O from the application becomes parallelized. This results in a higher level of throughput. In the past, small-block random workloads were multithreaded. However, it is now common to find large-block sequential workloads that are multithreaded.

Application buffering, and concurrency

Many Enterprise applications perform their own I/O buffering to coalesce file updates. Applications such as Microsoft Exchange, Microsoft SQL Server, and Oracle use application buffering to intelligently manage I/O and provide low response times.

For example, some databases periodically re-index themselves to ensure low response times. Detailed information on buffer configuration (also referred to as cache configuration) for many specific applications are available on Powerlink. The white paper *EMC CLARiiON Storage Solutions: Microsoft Exchange 2007 - Best Practices Planning* specifically advises on cache configuration for the application.

Application concurrency addresses the conflicting requirements for simultaneous reads and writes within the application to a single object, such as a table row. It attempts to avoid overwriting, non-repeatable reading (reading a previously changed value), and blocking. The higher the I/O concurrency is, then the better the storage system's performance is.

Many applications can be configured to adjust concurrency internally. Review the workload application's configuration documentation for their best practices on concurrency configuration.

Volume managers

Volume Managers will affect how hosts utilize storage system resources.

Contact your EMC Sales representative to engage an EMC USPEED Professional for assistance with Volume Managers.

Host HBAs

More than one HBA is always recommended for both performance and availability. Ideally the HBAs should be separate devices, and *not* a single, multi-ported device. This avoids a single point of failure.

The positive performance effect of HBAs is in their use for *multipathing*. Multipathing is creating more than one data path through the storage network between the host and the storage system. Multiple paths allow the storage system administrator to balance the workload across the storage system's resources. The "PowerPath" section for a description of VNX multipathing.

Keep HBAs and their software driver's behavior in mind when tuning a storage system. The HBA's firmware, the HBA software driver version used, and the operating system version and patch-level of the host can all affect the maximum I/O size and the degree of concurrency presented to the storage system.

The *EMC Support Matrix* service available through Powerlink provides suggested settings for drives and firmware, and these suggestions should be followed which should be implemented in your storage environment..

Host file systems

Proper configuration of the host's file system can have a significant positive effect on storage system performance. Storage can be allocated to file systems through volume managers and the

operating system. The host's file systems may support shared access to storage from multiple hosts.

File system buffering

File-system buffering, sometimes called *file caching*, reduces load on the storage system. However, application-level buffering is generally more efficient than file-system buffering. Buffering should be maximized to increase storage system performance. Note that some Enterprise-level applications can use both file system and application buffering together.

There are, however, some exceptions to the increased buffering advantage. The exceptions are:

- When application-level buffering is already being applied
- Hosts with large memory models

Ensure that application-level buffering and O/S file-system buffering do not work against each other on the host. Application-level buffering assumes the application (for example, Oracle) can buffer its I/O more intelligently than the operating system. It also assumes the application can achieve better I/O response time without the file system's I/O coalescing.

The extent of the file system resident in host memory should be known. With 64-bit operating systems such as MS 2008 R2 and later, hosts can have up to 128 GB of main memory; ESX-hosts up to 1 TB. With these large memory model hosts, it is possible for the entire file system to be buffered. Having the file system in memory greatly reduces the response times for read I/Os, which might have been buffered. This greatly improves performance. Write I/Os should use a write-through feature to ensure persistence of committed data.

File-system I/O size

Coordinating the file-system I/O size with the application and the storage system may result in a positive performance effect.

Minimum I/O size: Ensure the application and file system are not working at cross purposes over minimum I/O size. File systems can be configured for a minimum I/O extent size. This is the smallest indivisible I/O request given to the storage system. Typical values are 4 KB, 8 KB, 16 KB, or 64 KB. Applications performing I/O at sizes smaller than the file system's extent size cause unnecessary data movement or read-modify-write activity.

Note that storage configured as raw partitions, whose request sizes are not limited by a file-system I/O size, do not have this restriction.

Review both the workload's applications and operating system's file-system documentation for recommendations on resolving the optimal minimum I/O size setting. Use the largest minimum I/O size that is practical.

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 storage system is very efficient at handling large block I/O. It coalesces sequential writes in cache to full stripes to the RAID groups, as well as pre-reading large-block 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 coalescing

Host file-system coalescing can assist in getting high bandwidth from the storage system. Larger I/O requests are processed more efficiently than smaller I/O requests. File system coalescing combines many smaller I/O requests into a single larger request to the storage system. In most sequential-access operations, use the maximum contiguous and maximum physical file-system settings (when available) to maximize file-system coalescing. This increases the I/O size to the storage system, which helps improve bandwidth.

Host file-system fragmentation

When the percentage of storage capacity utilization is high, file system defragmentation of host storage can improve performance.

- File system defragmenting is *not* recommended for the following:Virtual Provisioning pool-based LUNs, including FAST tiered pools, and compressed LUNs
- Traditional LUNs bound on flash drive provisioned RAID groups
- Any LUNs that are using the FAST Cache feature

The following sections describe the recommendations for file system defragmentation.

Mechanical hard drive-based storage

Storage using traditional LUNs on mechanical hard drives benefits from host file defragmentation.

A fragmented file system decreases storage system throughput by preventing sequential reads and writes. In a fragmented file system with LUNs hosted on RAID groups provisioned with mechanical hard drives seek more frequently and over a larger portion of the drive than they would if the data were located contiguously on the drive. In general, the longer a file system is in use, the more fragmented it becomes.

Fragmentation noticeably degrades performance when the hard drive's capacity starts to exceed 80 percent. In this case, there is likely to be difficulty finding contiguous drive space for writes without breaking them up into smaller fragments.

It is important to monitor the fragmentation state of the file system. You should regularly defragment the file system hosted on traditional LUNs with defragmentation tools appropriate to the file system. Defragmentation should always be performed during periods of low storage system activity.

Defragmentation tools

EMC does not recommend any specific defragmentation tool. File-system fragmentation occurs independently of the operation of the storage system. The actions of any defrag tool are simply treated as I/O by the storage system.

Before using any defragmentation tool it is prudent to perform a full backup to ensure the safety of the file system. An effective alternative method to tool-based file-system defragmenting is to perform a file-level copy to another LUN, or by executing a backup and restore of the file system.

Defragmentation exceptions

Pool-based LUNs, flash drive-based LUNs, FAST VP LUNs, and FAST Cached LUNs *do not* benefit from file system defragmentation the way traditional LUNs do.

Pool-based LUNs

Thin LUNs should not be defragmented. A pool's allocation algorithms are such that defragmentation of files does not guarantee an increase in available pool capacity or performance. Thick LUNs may receive some benefit.

Thick LUNs reserve their capacity within the pool. Allocation of thick LUN capacity happens on demand. Because of this, there is the potential for some benefit in defragmenting them. More heavily fragmented thick LUNs benefit the most.

It is inadvisable to defragment thin LUNs. Defragmenting a thin LUN may reclaim space for the file system, but it does not return that capacity to the pool, just to the file system. The potential performance benefit of file consolidation also may not be realized. The defragmented files will likely not result in an optimal re-organization within the pool's storage. The highest pool performance comes when data is widely distributed across the pool's RAID groups. A thin LUN defragmentation may compact data that was previously widely distributed into a small portion of a smaller number of private RAID groups. This reduces overall pool performance.

When supported by the host O/S, you can shrink a host LUN to reclaim the defragmented file system capacity for the pool. LUN shrinks should only be used when severely fragmented pool

LUNs have been defragmented. This is because a LUN shrink cannot reduce capacity below 50 percent of the original LUN's capacity.

Flash drive-based storage

LUNs hosted on flash drives *do not* benefit from file system defragmentation. Flash drives are random access devices. There is no advantage in a sequential organization of their LBAs.

FAST VP and FAST Cache-based storage

Pools implementing the FAST VP feature or supported by FAST Cache should *not* be defragmented. Defragmentation makes assumptions about the physical layout and physical locality of data based on the file system's logical locality. This assumption is not correct within a tiered pool or a pool supported by a secondary cache. Depending on the file system's allocation *granularity*, the operation of the defragmentation may have an adverse effect on performance. Granularity is the capacity being internally buffered. Changing the previously algorithmically selected contents of the tiers or in the secondary cache degrades their performance. A defragmentation can undo the effects of a FAST Cache or FAST VP feature's warm-up. A small granularity, for example 4 KB, will result in changes that may require re-warming the tiers or cache. Note that 4KB and 8K the most common file system sizes.

File-system alignment

File system alignment reduces the amount of resources needed when a storage system services an I/O request. A file system aligned with the storage system's RAID group striping has reduced latency and increased throughput. File-system misalignment adversely affects performance in two ways:

- Misalignment causes drive crossings.
- Misalignment makes it hard to stripe-align large uncached writes.

In a drive crossing, an I/O that would normally engage a single drive is split across two drives. This is the most common misalignment case. The splitting of the I/O lengthens its duration. There are two writes, instead of one that would be performed with an aligned file system. Even if the drive operations are buffered by cache, the effect of the double write can be detrimental, as it will slow flushing from cache. Random reads, which by nature require drive access, are also affected. They are affected directly when they must wait for two drives to return data, and indirectly because the RAID group's drives are busier than they need to be.

Alignment has a greater potential positive effect on traditional LUNs. Alignment has a small positive performance effect on Virtual Provisioning pool-based LUNs. The striping across more than one RAID group within the pool reduces its effect.

The common example is shown in Figure 1. Some Intel-based systems are misaligned due to metadata written by the BIOS. For example, the 63 drive block NTFS header at the beginning of a LUN can cause misalignment. In an aligned system, up to a 64 KB write can be serviced by a single drive. However, metadata can cause an offset in the address space of a RAID group stripe. This offset causes I/O requests that would normally only require servicing by a single drive, to require two drives.

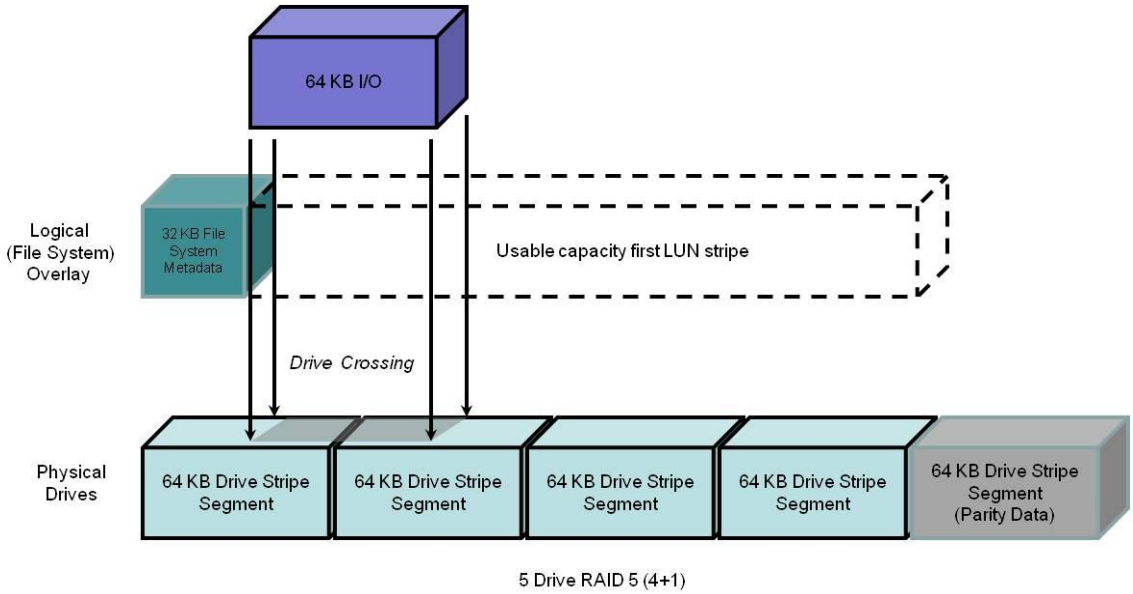

Figure 1 Effect of misalignment with a 63-block (32 KB) metadata area

Knowing the I/O type (random or sequential) and size of the workload is important in understanding the benefits of alignment. The type and size of a data transfer is application-dependent.

With its default 64 KB stripe element size, all I/Os larger than 64 KB will involve drive crossings. To minimize the number of crossings by 64 KB and smaller I/Os, partitions can be aligned to be on a RAID group stripe boundary. Figure 2. Aligned I/O with 65 block (33 KB) Alignment Space addedshows an aligned partition with a 64 KB I/O.

When a specific file system or application encourages the use of an aligned address space, and the offset is declared, EMC recommends using a host operating system drive utility be used to adjust the partitions. The adjustment requires offsetting the start of the LUNs usable data to begin on a drive stripe segment boundary. The Unisphere LUN bind offset facility should be used with caution, since it can adversely affect array replication synchronization rates.

Either host based or Unisphere based, alignment may be used with the same effect. Do *not* use both of them at the same time.

Note that not all I/O workloads require alignment. Misalignment rarely affects applications with small I/Os in comparison to the stripe segment size. Only occasionally will a small I/O result in a disk crossing. It is more likely to 'fit' with its peers on one disk or the next.

For example, workloads of predominantly 4 KB I/Os will see only a small advantage from alignment. As previously noted, 4KB is a common file system I/O block size.

However, expect to see some drive crossings, even with small I/Os. This can result from file headers, file system block size, and file system metadata, which may be part of the workload's I/O. As a result, an aligned partition, and a file system using 4 KB blocks, but serving an application reading a 4 KB header, which then requests a 16 KB I/O from the file can create a larger I/O of an odd size. This originally small-block I/O on an aligned partition could result in occasional drive crossings.

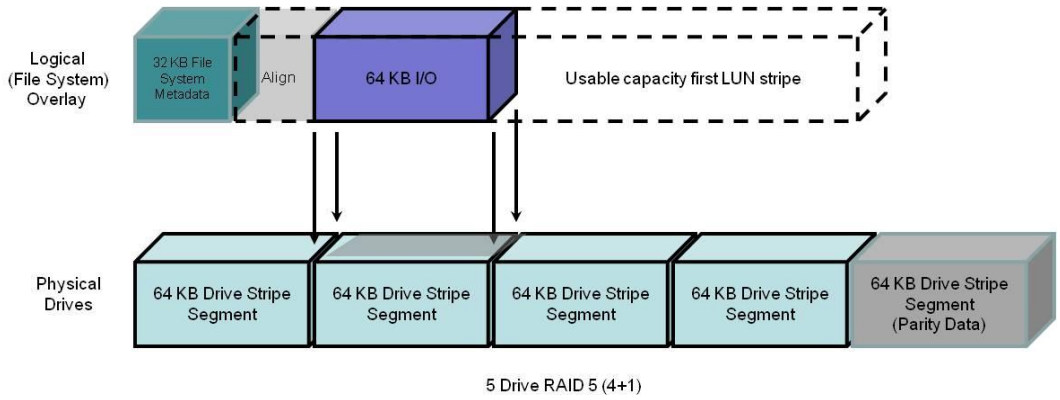

Figure 2. Aligned I/O with 65 block (33 KB) Alignment Space added

Enterprise-level applications such as databases (Oracle, SQL Server, or IBM UDB/DB2) supporting multiple block sizes will see a positive performance effect from alignment when the larger (8 KB and 16 KB) block size is used.

File-system alignment procedure

Detailed information and instructions for performing file-system alignments for host operating systems can be found on Powerlink.

Microsoft-based file-system alignment procedure

For Microsoft-based file systems refer to the white paper *Using diskpar and diskpart to Align Partitions on Windows Basic and Dynamic Disks*. For VMware alignment, the *Using EMC CLARiiON Storage with VMware Infrastructure and vSphere Environments TechBook* is a good source.

Alignment of Microsoft Windows Server partitions may not be necessary. Microsoft Windows Server 2008 LUNs, are automatically aligned. Its partitions are offset automatically by the O/S to 1 MB. This provides good alignment for the element segments used by the storage system. In addition, be aware that Windows Server 2008 defaults to a smaller offset for small capacity drives.

Microsoft Windows Server 2003 and earlier O/S revision partitions may benefit from alignment. Use the DiskPart command utility to align Microsoft Windows Server 2003 SP1 or earlier. To align a basic disk, use the align parameter to create a partition:

```
diskpart> create partition primary align = 1024
```

This makes the partition start at sector 2048. After aligning the drive, assign a drive letter to the partition before NTFS formatting. For more information about using the DiskPart command please refer to Microsoft Windows Server documentation.

Be aware, you cannot use the `align` command for dynamic disks; you must use the `DiskPart` command utility.

Linux file-system alignment procedure

With Linux host O/S installations, align the partition table first using the fdisk utility with instructions provided on the `man` page.

The following procedure using `fdisk` may be used to create a single aligned partition on a second Linux file `sda` or `sdc` file-system LUN utilizing all the LUN's available capacity. In this example, this partition will be:

/dev/nativedevicename.

The procedure is:

```
fdisk /dev/nativedevicename # sda and sdc
n # New partition
p # Primary
1 # Partition 1
<Enter> # 1st cylinder=1
<Enter> # Default for last cylinder
 # Expert mode
b # Starting block
1 # Partition 1
128 # Stripe element = 128
w # Write
```

Aligning Linux file-system very large LUNs

To create an aligned partition larger than 2 TB the GUID Partition Table (GPT) drive partitioning scheme needs to be used. GPT is part of the Extensible Firmware Interface (EFI) initiative. GPT provides a more flexible mechanism for partitioning drives than the older Master Boot Record (MBR) partitioning scheme.

By default, a GPT partition is misaligned by 34 blocks. In Linux, use the **parted** utility to create and align a GPT partition.

The following procedure describes how to make a partition larger than 2 TB. In this example, this partition will be `/dev/sdx`. The `mkpart` command aligns a 2.35 TB partition to a 1 MB starting offset.

Following are the Linux commands needed to create a GPT partition:

```
# parted /dev/sdb
GNU Parted 1.6.19
Using /dev/sdb
(parted) mklabel gpt
(parted) p
Disk geometry for /dev/sdb: 0.000-2461696.000 megabytes
Disk label type: gpt
Minor    Start       End     Filesystem  Name                  Flags
(parted) mkpart primary 1 2461696
(parted) p
Disk geometry for /dev/sdb: 0.000-2461696.000 megabytes
Disk label type: gpt
Minor    Start       End     Filesystem  Name                  Flags
1          1.000 2461695.983
(parted) q
# mkfs.ext3 /dev/sdb1 # Use mkfs to format the file system
```

Availability

The following sections describe the host system availability Best Practices.

PowerPath

Failover is the detection of an I/O failure and the automatic transfer of the I/O to a backup I/O path. The host-resident EMC PowerPath® software integrates failover, multiple path I/O capability, automatic load balancing, and encryption. If available on the OS, we recommend PowerPath—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.

A recommended introduction to PowerPath and its considerations is available in the latest revision of the *EMC PowerPath Product Guide* available on Powerlink.

Port load balancing

PowerPath allows the host to connect to a LUN through more than one SP port. This is known as *multipathing*. PowerPath optimizes multipathed LUNs with load-balancing algorithms. It offers several load-balancing algorithms. Port load balancing equalizes the I/O workload over all available channels. We recommend the default algorithm, ClarOpt, which adjusts for number of bytes transferred and for the queue depth.

Hosts connected to VNX's benefit from multipathing. Direct-attach multipathing requires at least two HBAs; SAN multipathing also requires at least two HBAs. Each HBA needs to be 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 and minimizing LUN trespassing
- Port 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 operations, as well as during failover.
- Every active and passive path from the host requires an initiator record; 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 restricted, via zoning, to two storage system ports per HBA for each storage system SP to which the host is attached. The exception is in environments where bursts of I/O from other hosts sharing the storage system ports are unpredictable and severe. In this case, four storage system ports per HBA should be used.

The *EMC PowerPath Version 5.5 Product Guide* available on Powerlink provides additional details on PowerPath configuration and usage.

Other multipath I/O services (MPIO)

Services other than PowerPath may be used to perform the MPIO function. These applications perform similarly to PowerPath, although they may not have the all the features or as close an integration with the storage system as available with PowerPath.

Microsoft Multi-Path I/O

Microsoft Multi-Path I/O (MPIO) as implemented by MS Windows Server 2008 provides a similar, but more limited, multipathing capability than PowerPath. Features found in MPIO include failover, failback, Round Robin Pathing, weighted Pathing, and I/O Queue Depth management.

Review your Microsoft Server O/S's documentation for information on available MPIO features and their implementation.

Linux MPIO

Linux MPIO is implemented by Device Mapper (dm). It provides a similar, but more limited, multipathing capability than PowerPath. The MPIO features found in Device Mapper are dependent on the Linux release and the revision.

Review the *Native Multipath Failover Based on DM-MPIO for v2.6.x Linux Kernel and EMC Storage Arrays Configuration Guide* available on Powerlink for details and assistance in configuring Device Mapper.

ALUA (Asymmetric Logical Unit Access)

Asymmetric Logical Unit Access (ALUA) can reduce the effect of some front- and back-end failures to the host. It provides path management by permitting I/O to stream to either or both of the storage system's storage processors without trespassing. It follows the SCSI SPC-3 standard for I/O routing. The white paper *EMC CLARiiON Asymmetric Active/Active Feature* available on Powerlink provides an in-depth discussion of ALUA features and benefits.

Host considerations

PowerPath versions 5.1 and later are ALUA-compliant releases. Ensure usage of PowerPath version 5.1 or later, with the host operating system.

PowerPath load balances across optimized paths. It only uses non-optimized paths if all the original optimized paths have failed. For example when an optimized path to the original owning SP fails, it sends I/O via the non-optimal path to the peer SP. If path or storage processor failures occur, PowerPath initiates a trespass to change LUN ownership. That is, the non-optimized path becomes the optimized path, and the optimized path becomes the non-optimized paths.

Not all multipathing applications or revisions are ALUA compliant. Verify that your revision of MPIO or other native host-based failover application can interoperate with ALUA.

When configuring PowerPath on hosts that can use ALUA, the default storage system failover mode is: **Failover Mode 4**. This configures the VNX for asymmetric Active/Active operation. This has the advantage of allowing I/O to be sent to a LUN regardless of LUN ownership. Details on the separate failover modes 1 through 4 can be found in the *EMC CLARiiON Asymmetric Active/Active Feature — A Detailed Review* white paper, available on Powerlink.

O/S considerations

To take advantage of ALUA features, host operating system needs to be ALUA-compliant. Several operating systems support native failover with Active/Passive (A/P) controllers. However, there are exceptions. Refer to the appropriate support guide for O/S support. For example, ALUA supported Linux operating systems would be found in the *EMC® Host Connectivity Guide for Linux, Rev A23* or higher.

Performance considerations

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

The non-optimized path is the alternate path accessed using ALUA. Performance can be 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. This 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 requesting storage processor to be transmitted to the host.

The redirection, from storage processor to peer storage processor and back, increases I/O response time. The duration of the delay is dependent on the overall storage system, 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 Unisphere Analyzer. In addition, metrics exist for total I/O over all paths. These metrics describe the utilization of paths and the differences in performance. Information on how to use Unisphere Analyzer can be found in the Unisphere on-line Help feature.

Queuing, concurrency, queue-full (QFULL)

A high degree of request concurrency results in the best storage system resource utilization. This is achieved by having many host connections per storage processor front-end port. However, if a storage system's queues become full, it will respond with a **queue-full (QFULL)** flow control command. The storage system's front-end port drivers return a QFULL status command under two conditions:

- The practical maximum number of concurrent host requests at the port is above 1600 (port queue limit).
- The total number of requests for a given LUN is (14 * (the number of data drives in the LUN)) + 32

The host response to a QFULL 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 it happens repeatedly. See Host HBAs section.

The best practices 1600 port queue limit allows for ample burst margin. In most installations, the maximum load can be determined by summing the possible loads for each host HBA accessing 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 that are make-up of many hosts, each with several HBAs, it may be difficult to compute the worst-case load scenario. In this case, use the default settings on the HBA and if QFULL is suspected, use Unisphere Analyzer (release 31 or later) to determine if the storage system's front-end port queues are full by following the steps described below.

Typically the queue depth setting for ESX and UNIX is 32. On Windows hosts using the QLogic HBA's and the EMC default driver settings, the queue depth is set to 256. This is called the *Execution Throttle*. This setting is a target setting and may be too high in some cases. If there is indication of QFULL problems, lower it to 32.

HBA queue depth settings usually eliminate the possibility of LUN generated QFULL. For instance, a RAID 5 4+1 device would require 88 parallel requests ((14*4) + 32) before the port would issue QFULL. If the HBA queue-depth setting is 32, then the limit will never be reached. A common exception is the RAID 1/0 (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 1/0 device could reach queue-full because its limit is 46 requests(1*14)+32).

QFULL is never generated as a result of a drive's queue-depth.

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

- Port queue-full – a counter has been added showing the number of QFULL signals issued due to port queue overflow
- Per-port bandwidth and IOPS

Usage

Consider all hosts connected to the storage system, and which LUNs they access. If necessary, set HBA throttles to limit concurrent requests to each LUN. Depending on the operating system, it may be possible to globally limit the total outstanding requests per HBA or per target. Set the HBA throttles of the hosts sharing a port so the total cannot exceed the port queue limit.

Remember that multipathing to one LUN via multiple ports on the same SP may lead to exceeding the port queue limit.

If high concurrency is suspected and performance problems occur, check port statistics reported by Unisphere Analyzer for queue-fulls, and lower the HBA throttles if appropriate.

Storage network attachment

Hosts use:

- Fibre Channel HBAs to connect to Fibre Channel storage networks
- Fibre Channel over Ethernet (FCoE) CNAs to connect to via an Ethernet network to storage
- Network interface card (NIC), iSCSI HBAs, and TCP/IP Offload Engines (TOE) with iSCSI drivers connect to Ethernet network for iSCSI storage network support.

Host bus

Knowing the number, distribution, and speed of the host's buses is important to avoid bottlenecks within the host. The host's I/O bus is sometimes called the *peripheral bus*. Ensure the network adapter's full bandwidth is supported by the host's I/O bus hardware.

Entry-level and legacy hosts may have less internal I/O bus bandwidth than their network adapters or HBAs. For this reason it is important to verify that your host can handle the bandwidth of the network interfaces when using these protocols: Fibre Channel networks that are equal to or faster than 4 Gb/s; 10 Gb/s Ethernet; and 10 Gb/s Fibre Channel over Ethernet (FCoE). In addition, when more than one adapter is present on a host I/O bus, remember that these adapters share the available bus bandwidth, and it is possible for the summed bandwidth requirement of the adapters to exceed the host's available bus bandwidth.

The ratio of network adapter ports to buses needs to be known. A host may have more than one internal bus. The distribution of bus slots accepting network adapters to buses is not always obvious. Review the number of network adapters, and the number of ports per network adapter that are being attached to each of a host's individual buses. Ensure that network adapters are connected to fast (>66 MHz) and wide (64-bit) PCI, PCI-X, and four-lane (x4) or greater PCI Express (PCIe) 1.1 or 2.X host buses. In all cases, the host I/O bus bandwidth needs to be greater than the summed maximum bandwidth of the network adapters to avoid a bottleneck.

Be aware that PCIe 2.0 connectors on host buses may internally provide fewer lanes than the physical connector. In addition, when more than one HBA is slotted the number of available lanes will be divided between the HBAs. This will reduce overall bandwidth to all HBAs.

Fibre Channel HBAs

High availability requires at least two HBA connections to provide redundant paths to the storage network or if directly connected, to the storage system.

It is a best practice to have redundant HBAs. Using more than one single-port HBA enables port- and path-failure isolation, and may provide performance benefits. Using a multiport HBA provides a component cost savings and efficient port management that may provide a performance advantage. Multiport HBAs are useful for hosts with few available I/O bus slots. The liability is a multiport HBA presents a single point of failure for several ports. Otherwise, with a single-ported HBA, a failure would affect only one port.

HBAs should also be placed on separate host buses for performance and availability. Note this may not be possible on smaller hosts that have a single bus or a limited number of bus slots. In this case, multiport HBAs are the only option.

Always use an HBA rated for or exceeding the bandwidth of the storage network's maximum bandwidth. Ensure that legacy 2 Gb/s or slower HBAs are not used for connections to 4 Gb/s or higher SANs.

FC SANs reduce the speed of the network path to the HBA's lower speed either as far as the first connected switch, or to the storage system's front-end port when directly connected. This may bottleneck the overall network when bandwidth is intended to be maximized.

Finally, using the most current HBA firmware and driver from the manufacturer is always recommended. The Unified Procedure Generator (installation available through Powerlink) provides instructions and the configuration settings for HBAs specific to your storage system.

FCoE Converged Network Adapter (CNA)

Contact your EMC Sales representative to engage an EMC USPEED Professional for assistance with CNA performance.

Network interface cards (NIC), TCP/IP offload engines (TOE), and iSCSI host bus adapters (HBA)

Three host devices connect hosts to iSCSI SANs:

- NICs
- iSCSI HBAs.
- iSCSI TOEs

The differences in the devices include cost, host CPU utilization, and features, such as security. The same server cannot use NICs and HBAs to connect to the same VNX or CLARiiON storage system. If there are multiple storage systems, the same host can connect through NICs and HBAs to the separate storage systems at the same time.

iSCSI NICs

NICs are the typical way of connecting a host to an Ethernet network. They are supported by software iSCSI initiators on the host.

Ethernet networks will auto-negotiate down to the lowest common device speed. Using a lower-rated NIC may bottleneck the storage network's bandwidth. Always use a NIC rated for or exceeding the bandwidth of the available Ethernet network. Do not use legacy 10 Mb/s or 100 Mb/s NICs for iSCSI SAN connections to 1 Gb/s or higher Ethernet networks.

iSCSI HBAs

The decision to use an iSCSI HBA versus a TOE, versus a NIC is dependent on the percentage CPU utilization of the host when it is processing workload(s). On small hosts, and hosts with high CPU utilizations, a TOE or iSCSI HBA can lower the host's CPU utilization. However, using an HBA or TOE may increase workload response time. In addition, an iSCSI HBA or TOE costs more than a conventional NIC. On large hosts or hosts not affected by high CPU utilization, we recommend a conventional NIC. Note that to work on an iSCSI SAN, the NIC must support the iSCSI protocol on the host. Check with the NIC's manufacturer for the appropriate driver support.

iSCSI TOEs

A TOE is a faster type of NIC. A TOE has on-board processors to handle TCP packet segmentation, checksum calculations, and optionally IPSec (security) offload from the host CPU on to themselves. This allows the host CPU(s) to be used exclusively for application processing.

In general, iSCSI HBAs are the most scalable interface. On iSCSI HBAs the TCP/IP and iSCSI processing is offloaded to the HBA. This reduces host CPU utilization. An iSCSI HBA also allows booting off an iSCSI target. This is an important consideration when considering diskless host booting. HBAs also typically provide for additional services such as security.

General Recommendations

Redundant NICs, iSCSI HBAs, and TOEs should be used for availability. NICs may be either single or multiported. A host with a multiported NIC or more than one NIC is called a *multihomed* host. Typically, each NIC or NIC port is configured to be on a separate subnet. Ideally, when more than one NIC is provisioned, they should also be placed on separate host buses. Note this may not be possible on smaller hosts having a single bus or a limited number of bus slots, or when the on-board host NIC is used.

All NICs do not have the same level of performance. This is particularly true of host on-board (mainboard) NICs, 10 Gb/s NICs, and 10 Gb/s HBAs. For the most up-to-date compatibility information, consult the *EMC Support Matrix* (ESM), available through *E-Lab Interoperability Navigator* (ELN) at: http://elabnavigator.EMC.com.

Finally, using the current iSCSI initiator, NIC, TOE, or iSCSI HBA firmware and driver from the manufacturer is always recommended.

Chapter 2 Network Best Practices

Network best practices advise on the software and hardware configurations of the iSCSI and Fibre Channel network infrastructure that attaches hosts to storage systems and their effect overall storage system performance and availability.

A recommended introduction to storage system networks and networking performance considerations can be found in the *EMC Networked Storage Topology Guide* available on Powerlink.

Performance

The following sections describe the best practices for the storage system that may be applied to the storage network.

iSCSI protocol

Avoiding iSCSI network congestion is the primary consideration for achieving iSCSI LAN performance. It is important to take into account network latency and the potential for port oversubscription when configuring your network. Network congestion is usually the result of an ill-suited network configuration or improper network settings. Ill-suited may be a legacy CAT5 cable in-use on a GigE link. Network settings include IP overhead and protocol configuration of the network's elements.

For example a common problem is a switch in the data path into the storage system that is fragmenting frames.

As a minimum, the following recommendations should be reviewed to ensure the best performance.

Simple network topology

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

It is common for network contentions, routing inefficiency, and errors in LAN and VLAN configuration to adversely affect iSCSI performance. It is important to profile and periodically monitor the network carrying iSCSI traffic to ensure the consistently high Ethernet network performance.

In general, the simplest network topologies offer the best performance. Minimize the length of cable runs, and the number of cables, while still maintaining physically separated redundant connections between hosts and the storage system(s).

Avoid routing iSCSI traffic as this will introduce latency. Ideally the host and the iSCSI front-end port are on the same subnet and there are no gateways defined on the iSCSI ports. If they are not on the same subnet, users should define static routes. This can be done per target or subnet using *naviseccli connection –route.*

Network latency

Latency can contribute substantially to iSCSI-based storage system's performance. As the distance from the host to the storage system increases; a latency of about 1 millisecond per 200 kilometers (125 miles) 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 200 km distance. EMC recommends increasing the number of streams to maintain the highest bandwidth with these long-distance, sequential I/O workloads.

Bandwidth-balanced configuration

A balanced bandwidth iSCSI configuration is when the host iSCSI initiator's bandwidth is greater than or equal to the bandwidth of its connected storage system's ports. Generally, configure each host NIC or HBA port to only two storage system ports (one per SP). One storage system port should be configured as active, and the other to standby. This avoids oversubscribing a host's ports.

Network settings

Manually override auto-negotiation on the host NIC or HBA and network switches for the following settings. These settings improve flow control on the iSCSI network:

- Jumbo frames
- Pause frames
- TCP Delayed ACK

Jumbo frames

Using jumbo frames can improve iSCSI network bandwidth by up to 50 percent. When supported by the network, we recommend using jumbo frames to increase bandwidth.

Jumbo frames can contain more iSCSI commands and a larger iSCSI payload than normal frames without fragmenting or with less fragmenting depending on the payload size. On a standard Ethernet network the frame size is 1500 bytes. Jumbo frames allow packets configurable up to 9,000 bytes in length.

The VNX series supports 4,000, 4,080, or 4,470 MTUs for its front-end iSCSI ports. It is not recommended to set your storage network for Jumbo frames to be any larger then these.

If using jumbo frames, all switches and routers in the paths to the storage system must support and be capable of handling and configured for jumbo frames. For example, if the host and the storage system's iSCSI ports can handle 4,470-byte frames, but an intervening switch can only handle 4,000 bytes, then the host and the storage system's ports should be set to 4,000 bytes.

Note that the File Data Mover has a different Jumbo frame MTU than the VNX front-end ports. The larger Data Mover frame setting should be used. See the *File Best Practices* for details.

Pause frames

Pause frames are an optional flow-control feature that permits the host to temporarily stop all traffic from the storage system. Pause frames are intended to enable the host's NIC or HBA, and the switch, to control the transmit rate.

Due to the characteristic flow of iSCSI traffic, pause frames should be *disabled* on the iSCSI network used for storage. They may cause delay of traffic unrelated to specific host port to storage system links.

TCP Delayed ACK

On MS Windows, and ESX-based hosts, *TCP Delayed ACK* delays an acknowledgement for a received packet for the host.

TCP Delayed ACK should be *disabled* on the iSCSI network used for storage.

When enabled, an acknowledgment is delayed up to 0.5 seconds or until two packets are received. Storage applications may time out during this delay. A host sending an acknowledgment to a storage system after the maximum of 0.5 seconds is possible on a congested network. Because there was no communication between the host computer and the storage system during that 0.5 seconds, the host computer issues Inquiry commands to the storage system for all LUNs based on the delayed ACK. During periods of congestion and recovery of dropped packets, delayed ACK can slow down the recovery considerably, resulting in further performance degradation.

Note that delayed ACK cannot be disabled on Linux hosts.

General iSCSI performance recommendations

The following general recommendations apply to iSCSI usage:

- When possible, use a dedicated LAN for iSCSI storage traffic, or logically segregate storage traffic to its own subnet or virtual LAN (VLAN).
- Avoid routing iSCSI traffic as this will introduce latency.
- Use the most recent version of the iSCSI initiator supported by EMC, and the latest version NIC driver for the host supported by EMC; both are available on the EMC E-Lab Interoperability Matrix.
- Configure iSCSI 1 Gb/s (GigE) and 10 Gb/s (10 GigE) ports to 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 GigE speeds.
- Use the appropriately rated optical fiber type for the distance at 10 GigE networks.. Note that with optical cables there is no agreed color for the specific type: OM1 through OM4. Its recommend that you use:
  - Shortwave Optical OM2 at ≤50m
  - Shortwave Optical OM3 for <380m.
- Use jumbo frames and TCP flow control for long-distance transfers or with networks containing low-powered servers.
- Use a ratio of 1:1 SP iSCSI ports to NICs on GigE SANs for workloads with high read bandwidths. 10 GigE networks can use higher ratios of iSCSI ports to NICs.
- Ensure the Ethernet connection to the storage system is equal to or exceeds the bandwidth of the storage system's iSCSI front-end port.
- Always check the latest Host Connectivity Guides, available on Powerlink, for the most up-to-date configuration information for connecting iSCSI
- Be careful not to oversubscribe either the target or initiator ports
- Configure the VNX Management ports and the iSCSI ports on different subnets.

Availability

The following sections cover network availability Best Practices.

Fibre Channel

At least two paths between the hosts and the storage system are required for high availability. Ideally, the cabling for these paths should be physically separated. In addition paths should be handled by separate switching, if not directly connecting hosts and storage systems. This includes redundant, separate HBAs, and attachment to both of the storage system's storage processors. Path management software such as PowerPath and dynamic multipathing software on hosts (to enable failover to alternate paths and load balancing) are recommended.

For device fan-in, connect low-bandwidth devices such as tape, and low utilized and older, slower hosts to edge switches or director blades.

FCoE protocol

Contact your EMC Sales representative to engage an EMC USPEED Professional for assistance with FCoE performance.

Additional Information

For additional information on FCoE, see the *Fibre Channel over Ethernet (FCoE) TechBook* available on Powerlink.

iSCSI protocol

iSCSI SANs on Ethernet do not have the same reliability and built-in protocol availability as Fibre Channel SANs. Although they do handle longer transmission distances and are less expensive to setup and maintain.

If you require the highest availability, for a SAN under 500m (1640 ft.) a SAN based on the Fibre Channel protocol is recommended.

Redundancy and configuration

Ideally, separate Ethernet networks should be created to ensure redundant communications between hosts and storage systems. The cabling for the networks should be physically, and as widely separated as is practical. In addition paths should be handled by separate switching, if not directly connecting hosts and storage systems.

Separation

We recommend that you use a physically separate storage network for iSCSI traffic. If you do not use a dedicated storage network, iSCSI traffic should be either separated onto separate LAN segments, or a *virtual* LAN (VLAN).

With VLANs, you can create multiple *virtual* LANs, as opposed to multiple *physical* LANs in your Ethernet infrastructure. This allows more than one network to share the same physical network while maintaining a logical separation of the information. OE Block 31.0 and later support VLAN tagging (IEEE 802.1q) on 1 Gb/s and 10 Gb/s iSCSI interfaces.

Logical Separation

Ethernet connections to the storage system should use separate subnets depending on if they are workload or storage system management related.

Separate the storage processor management 10/100 Mb/s ports into separate subnets from the iSCSI front-end network ports. It is also prudent to separate the front-end iSCSI ports of each storage processor onto a separate subnet.

Do this by placing each port from SPA on a different subnet. Place the corresponding ports from SPB on the same set of subnets. The 10.x.x.x or 172.16.0.0 through 172.31.255.255 private network addresses are completely available.

For example, a typical configuration for the iSCSI ports on a storage system, with two iSCSI ports per SP would be:

A0: 10.168.**10.10** (Subnet mask 255.255.255.0; Gateway 10.168.10.1)
A1: 10.168.**11.10** (Subnet mask 255.255.255.0; Gateway 10.168.11.1)
B0: 10.168.**10.11** (Subnet mask 255.255.255.0; Gateway 10.168.10.1)
B1: 10.168.**11.11** (Subnet mask 255.255.255.0; Gateway 10.168.11.1)

A host with two NICs should have its connections configured similar to the following in the iSCSI initiator to allow for load balancing and failover:

NIC1 (for example, 10.168.**10**.180) - SP A0 and SP B0 iSCSI connections
NIC2 (for example, 10.168.**11**.180) - SP A1 and SP B1 iSCSI connections

Note that 128.221.0.0/16 should never be used because the management service ports are hard configured for this subnet.

There is also a restriction on 192.168.0.0/16 subnets. This has to do with the configuration of the PPP ports. The only restricted addresses are 192.168.1.1 and 192.168.1.2 and the rest of the 192.168.x.x address space are usable with no problems.

VLANs

VLANs are a convenient way to create separation within an Ethernet network.

VLAN tagging with the compatible network switch support isolates iSCSI traffic from general LAN traffic; this improves SAN performance by reducing the scope of the broadcast domains. Note that the number of VLANs that may be active per iSCSI port is dependent on the LAN's bandwidth. A 10 GigE can support a greater number.

VLAN tagging can add more bytes per packet on Ethernet switches and switch port MTUs must be adjusted accordingly. This is a particular issue when enabling and configuring Jumbo Frames.

For more information about VLANs and VLAN tagging, please refer to the *VLAN Tagging and Routing on EMC CLARiiON* white paper available on Powerlink.

Chapter 3 Storage System Platform Best Practices

Storage system platform best practices advise on the software and hardware configurations of the 5000 and 7000 VNX series storage systems, and affect overall storage system's hardware and operating environment performance and availability.

A recommended introduction to the storage system can be found in the white paper, *Introduction to the VNX Series*. This paper is available on Powerlink.

Performance

The following sections describe the best practices for the storage system that may be applied to the storage system.

The diagram below may be helpful to new users to understand the relationship between the storage system's major components. The order of presentation generally follows the diagram below, proceeding from left to right.

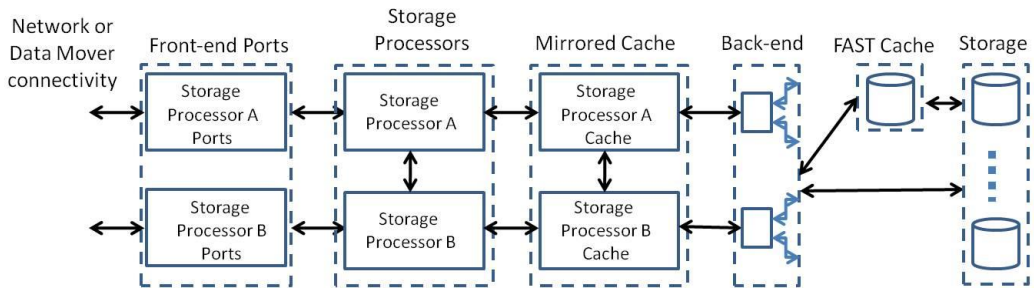

Figure 3. Storage System Conceptual Block Diagram

Front-end ports

Front-end ports connect the storage system to the storage network or the Data Mover. Three types of protocols, each with different speeds and performance characteristics are available with UltraFlex modules installable in the storage processors. Each module has two or more ports of the same type, either:

- Fibre Channel
- Fibre Channel over Ethernet (FCoE)
- iSCSI.

Connections to Fibre Channel and iSCSI hosts are supported simultaneously. However, a single host can only be connected to either a storage system's Fibre Channel ports or iSCSI ports; it cannot be connected to both types at the same time.

General front-end port performance

For best performance distribute the load equally across two or more ports per storage processor to avoid long failover times.. For high availability, hosts should always have at least two front-end port connections to each storage processor on the storage system from two separate HBAs or NICs.

Splitting the workload's bandwidth needs between both SPs lowers SP utilization. For the production workload, the front-end bandwidth of the storage system's ports should ideally be about the same for both SPs. This results in lower response time. Balancing the bandwidth may require trespassing LUNs between SPs or adding additional front-end connections to a host.

Adding additional ports helps to achieve the full storage system bandwidth performance in a wider range of configurations.

Note that the nature of the workload's I/O is an important consideration. For small-block random workloads, adding front-end ports make little performance difference.

Fibre Channel ports

The VNX series with Disk Processor Enclosures (DPEs) includes eight built-in ports of 8 Gb/s Fibre Channel host I/O. Storage Processor Enclosure (SPE) based VNX models do not include these ports as standard. To maximize a storage system's available bandwidth, connecting via Fibre Channel ports to 8 Gb/s Fibre Channel networks is recommended.

Note that File-based and unified storage system installations connect to the Data Mover by Fibre Channel ports.

Note that a single host can be connected to a single storage system by FC and FCoE ports at the same time.

The use of additional front-end connections for hosts can reduce the number of Fibre Channel switches needed in the storage network. All available bandwidth should be distributed across all the available front-end Fibre Channel ports. For high availability, two HBA's and four paths – two to SPA and two to SPB need be configured..

Usage

Contact your EMC Sales representative to engage an EMC USPEED professional for assistance with Fibre Channel front-end port performance.

Additional Fibre Channel ports

Additional Fibre Channel ports are configurable with all models. The additional Fibre Channel ports may reduce the number of other protocol ports configurable.

For example, The VNX7500 will technically support a maximum of 16 Fibre Channel front end ports per storage processor (32 total in the system). That configuration would require four (4) 8Gbs Fibre Channel I/O Modules. Because there are only five available I/O slots on the VNX7500, to get the maximum number of Fiber Channel ports the storage system must sacrifice one of its SAS I/O modules. That is, it must be configured with only one SAS IO Module– providing four backend ports for the storage system. A VNX7500 storage system can support a maximum of eight backend ports.

Fibre Channel over Ethernet (FCoE) ports

FCoE ports allow Fibre Channel and Ethernet networks to share a common infrastructure.

Usage

Contact your EMC Sales representative to engage an EMC USPEED professional for assistance with FCoE front-end port performance.

iSCSI ports

All VNX models offer an option of GigE at 1 Gb/s and 10 GigE.. In addition 10 GigE iSCSI ports are in either copper or fiber optic cabling are available. To maximize IOPS in iSCSI communications, connect 10 Gb/s iSCSI ports to only 10 Gb/s infrastructure Ethernet networks.

Auto-negotiation

iSCSI front-end ports do *not* auto-negotiate downward to all Ethernet speeds. The GigE iSCSI ports will auto-negotiate to 100 Mb/s and 1000 Mb/s only. They *do not* support 10 Mb/s connections. The 10 GigE iSCSI ports will *not* auto-negotiate downward; they will only run at 10 Gb/s. Do *not* connect 10 GigE iSCSI ports to GigE infrastructure. Check with the EMC Support Matrix (ESM) available through the E-Lab interoperability navigator for the supported Ethernet speeds and configurations.

Multi-protocol connections

When making connections to a host using iSCSI, only that protocol can be used. Note that Fibre Channel and FCoE connections from a single host to the same storage system *can* be made at the same time. Storage systems can always be connected to different hosts using separate protocols. In addition, a host cannot be connected to the same storage system through NICs *and* iSCSI HBAs at the same time. Either NICs *or* iSCSI HBAs must be used.

iSCSI processing requires more SP CPU resources than Fibre Channel processing. Workloads made up of small I/O requests with a high cache hit rate can cause high SP CPU utilization. T*his is particularly true with 10 GigE iSCSI.* EMC recommends a Fibre Channel connection for workloads with the highest transaction rate.

A single UltraFlex iSCSI controller drives a pair of iSCSI ports. The two ports share the controller's total bandwidth and its IOPS. The write bandwidth is not shared equally between the ports when both ports are used at the same time.

Usage

Contact your EMC Sales representative to engage an EMC USPEED professional for assistance with iSCSI front-end port performance.

Storage processors

The CPU 'horsepower' of the peer Storage Processors is a resource that still needs to managed. The storage processors are very capable of high performance when correctly sized model-wise to their workloads. However, their capability is finite. The highest storage system performance is achieved when the CPU utilizations of both storage processors is as equal as is practical.

CPU utilization may easily be maintained at *up-to* 70-percent for either or both storage processors. It is not prudent for *both* SPs to have utilizations of greater than 70-percent at the same time, for long periods of time. This is because in the event of a complete or partial SP failure, neither SP has enough margin to completely accommodate its peer's load without a significant performance degradation.

For example, in the unlikely event of an SP failover, neither SP would have enough margin to take-up its peer's load. In the possible event of a LUN trespass, the peer SP's assumption of the additional load may result in unacceptably higher host response times when both storage processors are in excess of 70-percent utilization.

Workload balancing

There is a performance advantage to evenly distributing the workload's requirement for storage system resources across all of the available storage system's resources. This balancing is achieved at several levels, where the storage processor (A or B) is the primary divisor. Work load can be balanced by:

- LUN ownership

- LUN I/O
- Feature utilization

Balancing through LUN ownership

Balancing across storage processors is performed by LUN provisioning in anticipation of the workload's requirements. The "The LUNs" section has more information. Note it is not the number of LUNs assigned to each Storage Processor, it is the total I/O being handled by the assigned LUNs that is used in balancing.

For example, fewer heavily utilized LUNs assigned to SP A may be balanced by a greater number of moderately utilized LUNs assigned to SP B.

Balancing through LUN I/O

One component of workload balancing LUN I/O across storage processors is through the system's front-end ports is largely performed by PowerPath in addition to a careful assignment of ports and zoning to hosts. The "PowerPath" section has more information. However, front-end ports are owned by storage processors. The number of active ports and their utilization directly effects storage processor utilization. Try to achieve a roughly equal distribution of the I/O between the two Storage Processors. This will result in a lower average host response time for the storage system.

Balancing through feature utilization

The features that are associated with LUNs also put demands on the storage processor's CPUs. Certain features have higher processing requirements than others. Below is a partial list of features requiring additional CPU resources:

- FAST Cache
- FAST Virtual Provisioning (FAST VP)
- LUN Compression

The processing load of some features is shared between the peer storage processors. However, many features resource needs are primarily associated with LUNs; LUNs using features that may require additional CPU resources should be evenly distributed between storage processors.

Do *not* give ownership of all compressed LUNs to a single storage processor. Divide the compressed LUNs ownership between storage processors to reduce the likelihood of a single storage processor being required to perform all LUN compressions or decompressions within the storage system..

Also note that installed applications, such as MirrorView™, and SnapView™ use CPU resources. Balance the use of active installed applications across storage processors as well.

Mirrored Cache

The allocation of read and write cache and the parameters for their operation can be tuned to achieve optimal performance for individual workloads.

Memory Utilization

The storage processor's operating environment, features, installed applications, and read/write cache use a shared memory capacity called *system memory*. The amount of system memory available for read/write cache depends on the VNX model.

The storage system's memory is a resource that must be managed for best performance results. The capacity of the read/write cache, particularly the write cache, determines system ability to smoothly handle bursts of write activity within a workload.

Per Storage Processor Memory	VNX5100	VNX5300	VNX5500	VNX5700	VNX7500
System Memory (MB)	4000	8000	12000	18000	24000
Maximum Read/Write Cache (MB)	801	3997	6988	10906	14250

Table 1 Maximum cache allocations VNX O/S Block 31.0 and File 7.0

Allocating read and write cache recommendations

Having the largest possible cache is an important performance optimization. Always allocate *all* the remaining memory after system memory is allocated to features and applications to the read/write cache.

Although it is possible to configure all of available memory (Maximum Cache from Table 1) to either read cache or write cache, this is *not* recommended.

Note that a write cache allocation applies to both SPs; it is mirrored. Read cache in not mirrored. To use the available SP memory most efficiently, ensure the same amount of read cache is allocated to both storage processors.

All workloads are different. Individual workloads may require adjusting the ratio of read to write cache to reach optimal performance. The amount of memory used by read or write cache can be changed at any time without disruption. Note that that the read/write cache will temporarily be disabled and performance will be lower..Start with the recommended cache settings and adjust the ratio in response to the local storage environment's needs.

Read cache

It is generally easier to allocate the less important (performance-wise) read cache first.

Read cache typically needs a smaller amount of capacity to operate efficiently. In addition, its typical usage is less varied across the majority of workloads. Write cache is different because its usage affects performance more heavily, and can range more widely. Set the read cache first, and then allocate the total remaining capacity to write cache.

It is advisable to have at least 100 MB of read cache for the block-only VNX5100, and at least 256MB of read cache for File-enabled systems. This is the amount of read cache where the read-ahead becomes efficient for the majority of workloads. Only rarely and with serious consideration should more than 10-percent, or more than 1024 MB, of available memory be allocated to the read cache. In addition, very few workloads benefit from very large read caches. A 1024 MB read cache is considered a very large read cache. The following table is the recommended initial read caches for each of the VNX models. It applies to both block and file configured storage systems. Start with this amount of read cache. Adjust the capacity as the workload may require.

Per Storage Processor Memory	VNX5100	VNX5300	VNX5500	VNX5700	VNX7500
Recommended initial Read Cache (MB)	100	400	700	1024	1024

Table 2 Recommended initial Read cache allocations VNX O/S Block 31.0 and File 7.0

Write cache

It is always more important to maximize the storage system's write cache size.

If performance is lagging from a too small write cache, you can lower the write cache watermarks to free-up cache pages more quickly. This may make-up for a too small write cache. Also, you can allocate all of the available memory to be used as write cache. That is, operate with no read cache. Not having a read cache allocation will adversely affect the storage system's sequential

read performance, but having a larger write cache will have a generally more positive effect on overall system performance.

Read/write cache allocation example

For example, the VNX5300 without licensed features or applications installed has 3997 MB of memory usable as cache per storage processor. Assume an average 60:40 read to write I/O mix for the workload, with a notable amount of the read I/Os being sequential. Further, assume that no features will be enabled on the storage system. (See section below.)

Applying the general recommended that 400 MB is a 'good' initial amount of read cache to have, the cache might be allocated per storage processor as: 400 MB read cache and 3597 MB write cache

System Memory's effect on thread and write cache

Enabling features and applications may decrease the amount of system memory to be less than the Maximum Cache capacity. Features and installed applications that use system memory include:

- FAST Cache
- FAST VP
- Thin Provisioning
- Compression
- SnapView
- MirrorView

Note that all features and applications do not allocate memory at the same time in their lifecycle. Some allocate memory upon their Non-Destructive Upgrade (NDU), while others allocate on first use. Check with your feature and application's documentation and release notes for details.

The feature enabler and application installers will maintain the configured ratio of read to write cache, although the overall capacity of the read/write cache will be reduced. The capacity of the storage system's read/write cache is the remainder of the SP's memory *after* system memory has been allocated.

For example, if your ratio of write to read cache is 80:20, before FAST Cache is enabled, it still be 80:20 after FAST Cache is enabled. However, after the FAST Cache is enabled there will be a smaller overall cache capacity.

In addition, note that enabling some features or changing the read/write cache settings causes the read/write cache to be temporarily disabled as the SP's memory is re-allocated. Enable features during 'off-peak' periods of storage system utilization to minimize the uncached effect on host I/O response time.

There are very many combinations of feature enablement's, provisioning, and parameterization. It is not possible to describe or predict the cache capacities resulting from every cache allocations and combinations of features usage. Inspect the read/write cache allocations after the enablement of features or the installation of applications to ensure that enough cache capacity remains to meet the workload's expected needs.

Cache page size recommendation

Changing the cache page size must be performed through Unisphere or the CLI. During this change, the cache is disabled.

If there is a mostly homogenous environment, such that all the hosts are running the same underlying O/S, file system and application suites, then setting the page size to match the I/O size of the reads/writes to the storage system's drives will improve performance. However, if the hosts have different O/Ss, mixed applications, or mixed file systems, it is recommended to leave the cache page size to its default value.

In environments where the storage system is supporting targeted search and manipulation of small block data, such as online transaction processing (OLTP), financial transactions, etc., where the data being requested from storage would typically be 4 or 8KB, the 8 KB page size is ideal.

In environments where large block access patterns are normal, such as big data analytics, video streaming and continuous data capture (e.g. IP network traffic monitoring and packet captures), where the major application works with fundamentally large blocks of data users may choose to configure a 16KB cache page size.

The default cache page size is 8 KB. It is not recommended to change the cache page size from the default. This cache page size has been tested to provide good performance across the largest number of workloads.

Low and high watermark recommendation

Watermarks help manage write cache flushing. The goal of setting watermarks is to avoid forced flushes while maximizing write cache hits. By adjusting the watermarks, cache can be tuned to decrease response time while maintaining a reserve of cache pages to match any sudden increases in the workload's I/O.

VNX storage systems have two watermarks: high and low. These parameters work together to manage the cache flushing conditions. High water flushing activates when the percentage of dirty pages reaches a pre-set limit. This limit is called the *high water mark*. The low watermark is when flushing stops. Watermarks only apply when write cache is enabled.

Between the watermarks

The difference between the high watermark and the low watermark determines the rate and duration of flushing activity. The larger the difference is, the less often you will see watermark flushing. Watermark flushing does not start until the high watermark is passed. Note that if there is a wide difference between the watermarks, when flushing does occur, it will be very heavy and sustained. This is called *forced flushing*. Heavy flushing increases LBA sorting, coalescing, and concurrency in storage, but it may have an adverse effect on overall storage system performance. The smaller the difference between the watermarks, the more constant the flushing activity. A lower rate of flushing permits other I/Os (particularly reads) to execute.

FAST Cache will speed write cache flushing time for data that has been promoted into it due to its ability to handle writes faster than the mechanical drive-based storage..

Above the high watermark

The margin of cache above the high watermark is set to contain bursts of write I/O and to prevent forced flushing.

In a bursty workload environment, lowering both watermarks will increase the cache's "reserve" of free pages. This allows the system to absorb bursts of write requests without forced flushing.

Below the low watermark

The amount of cache below the low watermark is the smallest number of cache pages needed to ensure a high number of cache hits under normal conditions. . It is also the point at which the storage system stops high water or forced flushing. The number of pages below the watermark can drop when a system is not busy due to idle flushing.

Initial recommended watermarks

The recommended VNX watermarks for OE Block 31.0 and later are shown in the following table.

Generally, with the mid-model VNX and lower, the high watermark should be 20 percent higher than the low watermark. This is particularly applicable when high write re-hit and reads from write cache hit rates are observed. Otherwise, the high watermark may be set to be 10 percent higher than the low watermark. Be prepared to adjust these initial watermarks to more closely fit your workload.

VNX Model	VNX Recommended Write Cache Watermarks	
	High	Low
VNX7500 VNX5700 VNX5500	60	50
VNX5300 VNX5100		40

Table 3 Recommended watermark settings, OE Block 31.0

Additional mirrored cache information

Further information is available in the EMC Unified Storage System Fundamentals for Performance and Availability white paper available on Powerlink.

Back-end

The VNX series implements a 6 Gb/s Serial Attached SCSI (SAS) backend. The SAS protocol is an evolution over the Fibre Channel protocol used in the backend of previous generations of EMC mid-range storage.

VNX back-end ports overview

The VNX's SAS expanders and links constitute the back end. Technically, the backend is a hierarchy of two classes of SAS expanders: Fanout and Edge. The VNX uses SAS expander hardware as a switch to simplify the storage system's configuration so it can be scaled drive-wise with a small amount of latency while still providing the same bandwidth for increasing workloads.

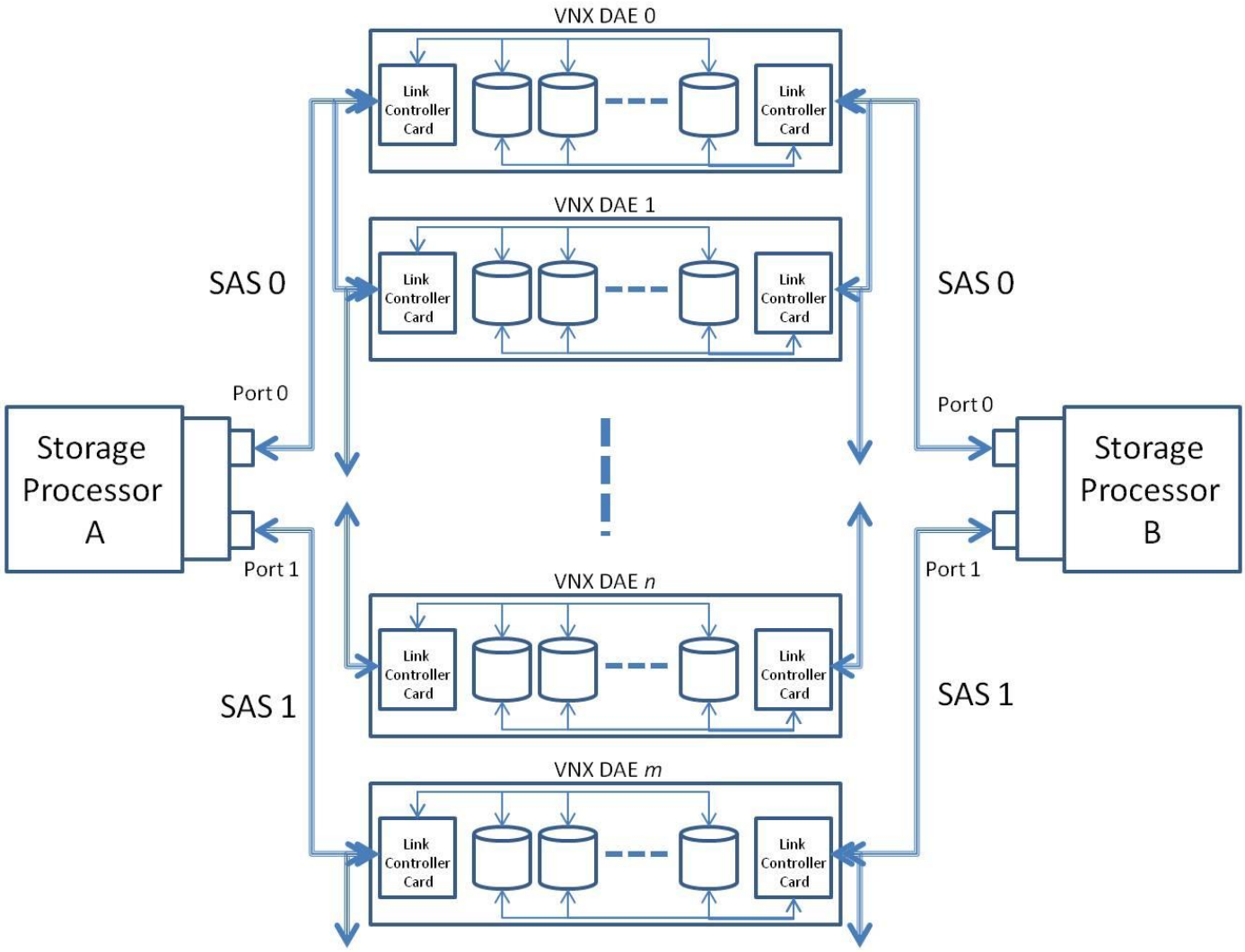

Figure 4, VNX Back-end Conceptual Diagram

The figure above shows a high-level, block diagram of the VNX back-end. This figure is a conceptual diagram and is not intended to be comprehensive. It is also not intended to represent any particular VNX model storage system.

Maximum number of VNX enclosures per port (shown in the figure as *n*) is 10. The maximum possible number of drives per port hosted by the enclosures is 250.

Back-end SAS ports

The VNX series has from one to four fanout expanders per storage processor, depending on the model. A fanout expander has two *wide*-ports (0 and 1). Ports support links to devices. A device is typically a SAS drive, but can also be another expander. Wide ports support multiple SAS links (sometimes called *lanes*).

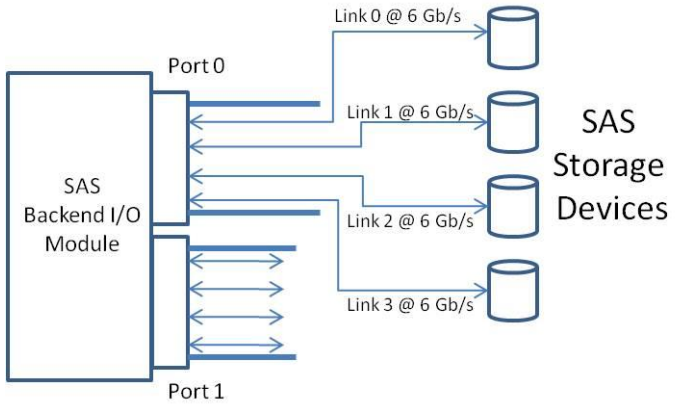

Figure 5. VNX SAS Backend Port Conceptual Diagram

Each VNX port is made-up of four links. Each link is capable of 6 Gb/s (1200 MB/s nominal). Note that the practical link bandwidth is dependent on the storage device on the link. The expander can aggregate the bandwidth and balance the traffic of the links within the port. The performance available is determined by the actual application and environment. This architecture can produce 2.4 GB/s between expanders. It can reach about 750 MB/s on a link with an ideal workload. Note this is more than twice the bandwidth of legacy CLARiiON 'buses'.

The number of drives simultaneously addressed within the storage system is limited by the number of links within the physical ports integrated into the fanout expander.

For example, with two wide ports, each with four lanes, a single storage processor can simultaneously issue commands to up to eight storage devices at the same time.

The wide ports also provide redundancy. Loss of any individual port lane degrades performance, but does not cause failure of the overall wide port. Only one of four lanes is needed for continuous operation.

In the VNX each of the fanout expander's ports is connected to a separate edge expander. An edge expander is in each Disk Array Enclosure (DAE). Actually, there are two edge expanders, one connected to each SP. See Figure 4, VNX Back-end Conceptual Diagram. These edge expanders are referred to as Link Connector Controllers (LCCs). Edge expanders connect directly to the SAS storage devices in the DAE and one other expander. The connection to another edge expander creates a daisy-chain of edge expanders/DAEs.

a.k.a. "Buses" and "loops"

The daisy-chain of LCCs may also be called a *bus*. The connection between an LCC and a storage device may sometimes be called a *loop*. This is the legacy CLARiiON terminology. Note that 'bus' and loop are not terms used in the SAS protocol specification.

SAS back-end port balancing

There is a general performance advantage to evenly distributing the I/O to the storage devices evenly across all the available back-end ports. Note, it is the I/O and not the number of drives or their capacity that needs to be spread across the backend ports. In addition with four backend-port models of the VNX, there is a slight performance advantage with one port over the other.

The SAS backend has different operational characteristics from legacy Fibre Channel protocol CLARiiONs in terms of the technique used for spreading I/O across the backend. To a degree, "bus balancing" is *less critical* with the VNX, because of the SAS protocol, and the increased I/O bandwidth of the VNX's backend.

For example, each SAS port can address four storage devices at the same time on a port, with the expanders optimizing the I/O at the protocol-level. The legacy Fibre Channel back-end could only address a single storage device on its bus.

However, additional attention to distributing the usage of storage system resources can improve backend performance. This is particularly true for the highest-performing storage devices such as flash drives. Flash drives can fully leverage the VNX's higher speed backend. If not provisioned carefully, in some cases they can saturate the VNX's more capable backend.

Balancing is achieved at both the physical and logical levels. Physically, it requires installing or selecting storage devices in enclosures attached to separate backend ports. At a logical level, this involves creating LUNs that distribute their I/O evenly across the backend ports.

To achieve physical distribution may require physically relocating drives between DPEs and DAEs. Ideally this should be done when the storage system is first provisioned or with drives and slots that are not in use. Be aware, drives that have already been provisioned into RAID groups or Virtual Provisioning pool cannot be removed or relocated without deleting the storage object.

VNX model	SAS Backend Ports per Storage System
VNX5100	2
VNX5300	2
VNX5500	2
VNX5700	4
VNX7500	4 or 8

Table 4, SAS Back-end Ports per Storage System

Four Backend SAS port and greater bus usage

VNX models with greater than two backend-SAS ports per storage processor (see Table 4, SAS Back-end Ports per Storage System) have a performance advantage when using specific ports, if *all* ports are *not* in-use. When *all* ports are in use, there is *no* performance advantage in using one port over the other.

- If you are using only two backend busses on a VNX5700, you should use ports 0 and 2, or 1 and 3 for the best performance.
- If you are using four or fewer backend ports on s VNX7500, you should alternate ports on the SAS backend I/O module for highest performance.

Back-end port balancing example

Assume a VNX5300, which has two (2) back-end ports per storage processor. Using five (5), 200 GB flash drives toward a goal of creating a 800 GB (raw) RAID level-5 FAST VP flash drive tier requires the drives be distributed between Backend port 0 (Bus 0) and Backend port 1 (Bus 1). The VNX5300 is a DPE-type storage system. Assume the DPE and a single DAE are the only enclosures in the storage system. One possible physical distribution of the drives is as follows:

1. Enclosure 0, Bus 0, Disk 4 # DPE: Flash drives installed next to system drives
2. Enclosure 0, Bus 0, Disk 5
3. Enclosure 0, Bus 0, Disk 6
4. Enclosure 1, Bus 1, Disk 0 # DAE
5. Enclosure 1, Bus 1, Disk 1

This distribution places three of the flash drives on backend port 0, and two on backend port 1. While this distribution may not be exactly evenly balanced, the I/O to the RAID group made-up

from these drives will be more evenly distributed across the backend. For more information, see the RAID groups section.

VNX Disk Enclosures

VNX DAEs refer to the DAEs delivered with VNX series storage systems. The VNX DAE is a 6 Gb/s SAS device. The maximum number of drives for each array is determined by counting the total number of Disk Processor Enclosure (DPE) and DAE drive *slots* in the storage system model. Note that enclosures with different drive form factors can be hosted on the same storage system using different enclosures. There is a maximum of 10 enclosures per backend SAS port. The table below shows drive capacities of the DPEs and DAEs for the VNX.

VNX Drive Enclosures Format and Capacity			
Enclosure Type	Part Name	Drive Form factor Supported	Number Drive Slots
DPE	DPE7	3.5″	15
DPE	DPE8	2.5″	25
DAE	DAE5S	2.5″	25
DAE	DAE6S	3.5″	15
DAE	DAE7S	2.5″	60

Table 5, VNX DAEs VNX OE Block 31.0

The physical drive format supported does not correlate to capacity.

For example, a 2.5” SAS drive can have the identical capacity of a 3.5” SAS drive.

The relationship of capacity to format is important to understand, if *storage density* is a priority. Storage density is a measure of the quantity of information in GBs that can be stored in a given volume of a storage system. Almost two 2.5” format drives can occupy the volume of a single 3.5” format drive in a storage system drive enclosure. The smaller 2.5” format drives of the same speed and capacity would have equivalent performance and a higher storage density than larger 3.5” format drives. In addition, the 2.5” drives use less power than their 3.5” counterparts. Greater storage density generally equates to greater power savings per GB of storage. The exception is that 3.5” flash drives have lower power consumption than 2.5” format SAS drives.

Maximum Slots

Some combinations of DAEs will create a configuration where the DAEs offer more drive slots than the VNX model supports. The storage system will not support these configurations. That is, the user will not be able to install a higher count of DAEs and corresponding slots than the maximum drive slots supported on a given storage system. Note that this maximum refers to the drive slots available, regardless of whether drives are installed.

It is possible to provision a storage system with a combination of DAEs or DAEs and DPEs that cannot reach the maximum number of supported drives. Choose the combination of DPEs and DAEs that will maximize the number of drives available during the storage system’s lifetime, not just for immediate usage.

For example, a VNX5300 has a maximum drive count of 125. A 125 drive system can be provisioned in more than one way. The table below shows a few alternative provisioning options; others are possible.

VNX5300 Example DPE/DAE Provisioning			
	Example #1 Number Enclosures	Example #2 Number Enclosures	Example #3 Number Enclosures
DPE7		1	1
DPE8	1		
DAE5S	4		4
DAE6S		7	
Total Slots:	125	120	115

Table 6, VNX5300 Example DPE/DAE Provisioning

Note from the table that Examples #2 and #3 cannot reach the storage system model's maximum specified 125 drives. Adding even the smallest enclosure to the example configuration, a 15-slot DAE5S, would exceed the model's 125 slot maximum..

Enclosure provisioning recommendations

The following table summarizes the recommendations for usage of the VNX enclosures according to the most used criteria.

VNX Enclosure Provisioning		
	Selection Criteria	Drive Types
DPE7	High Performance	15K rpm SAS Flash
	High Capacity	NL-SAS
DPE8	Storage Density	SAS
DAE6S	High Performance	15K rpm SAS Flash
	High Capacity	NL-SAS
DAE5S DAE7S	Storage Density	SAS

Table 7, VNX Enclosure Provisioning Criteria, OE Block 31.0

Within the enclosures, high performance is achieved by using 15K RPM SAS or flash drives. High capacity is achieved by using 2 TB NL-SAS drives. High density is achieved by using 10K RPM SAS drives. See the Mechanical hard drives section for details on performance and capacity usage.

FAST Cache

The VNX series supports an optional performance-enhancing, feature called FAST Cache. The FAST Cache is a storage pool of Flash drives configured to function as a secondary I/O cache. With a compatible workload, a FAST Cache increases performance in the following ways:

- Reduces host response time
- Lower RAID group drive utilization
- FAST Cache supports both pool-based and traditional LUNs

FAST Cache Overview

The storage system's primary read/write cache optimally coalesces write I/Os to perform full stripe writes for sequential writes, and prefetches for sequential reads. However, this operation is generally performed in conjunction with slower mechanical storage. FAST Cache monitors the storage processors I/O activity for blocks that are being read or written multiple times from storage, and promotes those blocks into the FAST Cache.

The FAST Cache is provisioned with fast, flash drives. Entire flash drives, must be allocated in pairs to FAST Cache. Once a block has been promoted, FAST Cache handles the I/O to and from that block. FAST Cache reduces read activity to the backend as well as writes. It also allows the storage processor's write cache to flush faster, because its flushing to high-speed flash drives.. This allows the primary cache to absorb a greater number of non-FAST Cache write I/Os. These optimizations reduce the load on mechanical drives and as a result improve overall storage system performance.

The increase in performance provided by the FAST Cache is dependent on the workload and the configured cache capacity. Workloads with high locality, small block size, random read I/O and high concurrency benefit the most. Workloads made up of sequential I/Os benefit the least.

The operation of FAST Cache creates a small storage processor CPU overhead. FAST Cache should not be used on systems that have regular very high storage processor utilization (> 70-percent) for long periods of time.

In addition, the closer the FAST Cache capacity matches the high-locality portion of the workload(s) working data set the less system memory will be consumed. See the Memory Utilization section. .

Finally, if the workload is already efficiently using the primary read-write cache as shown by high cache hit percentages, a secondary cache will provide very little performance improvement.

Additional information

Further information is available in the *EMC CLARiiON and Celerra Unified FAST Cache* white paper available on Powerlink.

Candidate conditions for use of FAST Cache

The following conditions determine if FAST Cache will be beneficial for your storage system:

- Storage Processor Utilization is under 70-percent
- There is evidence of regular forced Write Cache Flushing
- The majority I/O block size is under 64K
- RAID groups having drive utilization consistently greater than 70-percent
- The workload has a clear majority of read I/O over writes
- The active LUNs have a high percentage of read cache misses
- A higher than acceptable host response time is being observed

Workload's affect FAST Cache performance

The following factors affect the FAST Cache's potential to improve performance:

- Locality of reference
- Capacity of the Working Data set
- I/O Size
- I/O Type

Locality and the overall capacity of the working data set have the greatest effect on FAST Cache performance.

Locality is discussed in the High locality versus low locality section.

The capacity of the working data set is also important to know. It varies from application to application. A three to five-percentage of active data is common, but a 20-percent active portion of user data is easily possible.

For example, a 1.2 TB database with a 20 percent working data set has about 250 GB of frequently accessed capacity.

Storage architects and administrators should confer with their application's architects and analysts to determine the capacity of the working data set to more accurately determine the appropriate capacity of the FAST Cache.

The size of the I/O can affect performance. FAST Cache performs best with I/O sized between 4 KB to 32 KB in capacity. This is the ideal block-size for the FAST Cache's underlying flash drives.

The I/O type (random or sequential) also affects the performance. FAST Cache is designed for optimizing random I/O that has a high-degree of locality. High locality improves the efficacy of the FAST Cache. It reduces the movement of data in and out of the cache. Data is moved into the cache based on the cumulative number of accesses. The more random and the less sequential the nature of the workload's I/O, the better the feature's performance. Sequential I/O does not fit this pattern.

The Storage Processor's read/write cache complements the FAST Cache for handling sequential I/O.. The storage system's primary cache is optimized for sequential reads with its read cache pre-fetching when an appropriate read cache capacity is provisioned. Sequential writes are typically handled with very efficient direct-to-disk processing.

FAST Cache effect on read/write cache capacity

Every attempt should be made to 'right-size' the FAST Cache.

Larger caches hold more data, which increases the chance of a *cache hit*. Hits are when a read or write request can be serviced by the FAST Cache's Flash drives. A miss is a request serviced by the VNX's drives allocated to main storage. Cache hits within the FAST Cache have a very low host response time compared to typical mechanical drive storage access.

However, the FAST Cache requires system memory for its metadata. In addition, it requires additional storage processor CPU resources for its operation.

Read/write cache is a primary storage system performance asset. The FAST Cache enabler proportionally decreases the primary read/write cache's capacity to maintain the configured ratio and uses the memory capacity for FAST Cache metadata. However, while FAST Cache does reduce the available capacity of the read/write cache; this will in most cases be offset by faster flushing of read/write cache to FAST cache.

FAST Cache provisioning

Within the available maximum FAST Cache size for the storage system, the FAST Cache should be large enough in capacity to contain an application's working data set.

FAST Cache uses a RAID level 1 paired drive provisioning to provide both read and write caching, in addition, to mirrored data protection. All drives of the FAST Cache must be the same capacity.

To determine how many flash drives that will need to be installed and configured, a good rule of thumb is:

- If there is forced read/write Cache flushing and high LUN response time, then configure the maximum FAST Cache configuration.
- If there is no forced Write Cache flushing and LUN response times averages less than 15ms then a smaller number of flash drives can be provisioned.

When practical, it is recommended that *at least* four flash drives total be used in a FAST cache. More flash drives of lower capacity cache perform better than fewer of larger capacity. This is because larger number of drives the superior concurrency and less storage processor contention resulting in greater efficiency in the caching role.

Vertically provision the FAST Cache drives. (See VNX Enclosures and drive placement section.) Locate the primary and secondary drives of the mirrored RAID 1 pair on different back-end ports, to improve availability. The order the drives are added into FAST Cache is the order in which they are bound, with the first drive being the first primary; the second drive being the first secondary; the third drive being the next second pair's primary and so on.

By default, pools that contain Flash drives have FAST Cache disabled, all mechanical-based drive pools have it enabled. Data in a flash drive tier will not be tracked and thus not promoted to FAST cache. Using the FAST Cache generally will not help performance and is not an efficient use of the FAST Cache's capacity. The capacity of the FAST Cache and the maximum number of Flash drives used to provision the cache are dependent on the storage system's model. Note that *only* certain Flash drive configurations are supported, *not* every possible configuration of Flash drives.

For example, large 800 GB FAST Cache's are not available on entry-level VNX models. In addition, there are model restrictions; for the VNX5100 model, FAST Cache and Thin Provisioning of thin LUNs are mutually exclusive.

FAST Cache Actual Capacity

The capacity of the FAST Cache will be less than the raw capacity of its drives. This is because FAST Caches is implemented on flash drives using a RAID 1 configuration. They have the same data protection overhead as traditional LUNs using RAID-level 1.

For example, creating a read/write FAST Cache from a pair of 200 GB Flash drives results in approximately a usable capacity of 183 GB. This cache is then divided equally between the two storage processors, resulting in 86 GB per storage processor.

See the *EMC CLARiiON and Celerra Unified FAST Cache* white paper available on Powerlink for the available configurations.

FAST Cache warm-up

For optimal FAST Cache performance, the cache needs to be needs to contain the data that the application is currently re-using. *Warm-up* is the filling of the FAST Cache with candidate data. Depending on the workload, the initial population of the cache may take some time.

The efficient operation of the FAST Cache depends on locality and frequency of access of the user data making up its contents; the higher the locality and frequency of access the higher the caching efficiency. Initially, the I/O response time with FAST Cache will be variable. The empty cache will have a higher response time than as with no FAST Cache at all. When partially filled, some I/Os will have the very short FAST Cached response time, while others will have the uncached response time. As the cache warms-up the response time shall become on average the lower cached duration. Likewise, the time it takes for the cache to warm up depends on locality and frequency of data access within the workload.

The lowest average response time for a FAST Cached workload occurs when the maximum number of active addresses has been copied into the FAST Cache. That results in the highest hit rate inside the FAST Cache. Assuming the workload reads or writes the same addresses with a range of addresses with required frequency, promotions will be made in the VNX's background processing. The hit rate increases as more blocks are promoted which decrease the average response time.

Warm-up can take from several minutes to several hours; depending on the pattern of the read and write requests. If the working set being cached is smaller in capacity than the FAST Cache, a wide range of warm-up times are possible. If the working set is much larger than the FAST Cache, there may be very few hits or reuses of addresses in the FAST Cache. Warm-up can take a long time or possibly never complete under this condition.

Enabling FAST Cache

It is recommended that FAST Cache be enabled (installed) during periods of low or no activity.

Creating a Fast Cache will disable the storage system's read/write cache until it is complete. The storage system's write cache has to be flushed in order to zero and then reconfigure it. While the read/write cache is disabled overall performance will adversely affected.

The time it takes to fully configure the FAST Cache that depends on the cache's size and any workload activity on the storage system. Larger FAST Caches take longer to configure than smaller. On a 'quiet' system with low activity and small FAST Caches the configuration can take several minutes. Configuring a large FAST Cache on a loaded storage system may take more than an hour.

Disabling FAST Caching

Disabling FAST Caching of private LUNs is recommended. In general, FAST Cache does not provide a performance benefit to private LUNs. However, other than occupying FAST Cache capacity better used servicing host LUNs, promoting the contents of private LUNs into the FAST Cache also does not adversely affect overall system performance.

Installed applications such as MirrorView, SAN Copy, and SnapView create reserved LUNs that may end up promoted into the FAST Cache. These LUNs are grouped together into the Reserved LUN Pool. The Reserved LUN Pool does not benefit from FAST Cache. The Write Intent Log (WIL) and SnapView clone reserved LUNs (CPL) reserved LUNs (sometimes called "private" LUNs) already have optimizations for priority in the storage system's primary write cache. The re-hit rate of the RLP is actually very high, but it is over such a narrow capacity, it gets served by the storage system's read/write cache.. The greater performance gain is achieved by enabling FAST Cache for the source LUN.

WIL and CPL have optimizations to keep the data in write cache. Reserved LUNs do not. Disabling FAST Cache on the MirrorView WIL and the CPL is recommended. This will avoid their unnecessary promotion into the FAST Cache. This promotion would consume capacity in the FAST Cache, which would be better used for user data.

An exception is MetaLUN components. These LUNs are private LUNS. Enabling FAST Cache for a MetaLUN requires all the components to be cached for consistent high performance. Do *not* disable FAST Cache for MetaLUN components.

Cached Virtual Provisioning pools

FAST Cache provides a very effective way of increasing Virtual Provisioning pool performance.

Multiple Pools

FAST Caching is enabled at the pool level with Virtual Provisioning. Any number of pools may utilize FAST Cache and be cached at the same time. However, the entire pool, *not* individual LUNs, will have FAST Cache enabled or disabled, depending on the user selection.

Since not all LUNs may benefit from FAST Cache, it may be more efficient to create more than one Virtual Pool when FAST Cache is enabled. This way, FAST Cache can be enabled for the pool(s) containing LUNs that will benefit from FAST Cache, while FAST Cache is not enabled for the pool(s) containing the LUNs that are not suited for FAST Cache.

Virtual Provisioning pools with flash drives

With Virtual Provisioning pools, including FAST VP pools, that include a flash drives, data on the pool's flash drives is *not* cached by the FAST cache feature.

If only a few flash drives are available for provisioning a storage system, using them as a FAST Cache is recommended over creating a flash FAST VP tier or a flash homogenous pool. This is because the limited flash drive resource can be used to benefit the entire storage system's performance, and not just a single pool. When five or more flash drives are involved, consideration is needed. If individual pool-based LUN high performance is needed, create a flash drive tier or a traditional flash RAID group. Otherwise, create a FAST Cache. See the Virtual Provisioning: Pools section.

Additional Information

Review *EMC CLARiiON, Celerra Unified, and VNX FAST Cache—A Detailed Review* for additional information, including the supported configurations of the FAST Cache capacity that results, and additional provisioning restrictions. This document is available on Powerlink.

Physical Storage

The VNX series supports several types of physical storage devices both mechanical and semiconductor-based in more than one physical form factor.

Physical Storage Overview

The VNX series supports several storage device types. The table below shows the broad categories of the drives supported.

VNX Supported Storage Devices			
Drive Technology	**Type**	**Drive Speed (RPM)**	**Physical Form Factor**
Flash Drives	SAS	N/A	3.5”
Mechanical Hard Drives	SAS	15K	3.5”
		10K	3.5”
			2.5”
	NL-SAS	7.2K	3.5”

Table 8 Supported Storage Devices VNX OE Block 31.0

The different drive technologies have different performance, capacity, and storage density characteristics. These all affect provisioning decisions.

Flash drives have more bandwidth and IOPS than mechanical hard drives. The IOPs and bandwidth are directly related to drive speed, where 15K RPM drives have higher IOPS and bandwidth than lower speed drives. The 2.5” mechanical hard drives in a few cases have slightly higher throughput, but the same IOPS as the 3.5” form factor mechanical hard drives of the same speed.

There is a range of available drive capacities. Generally, the speed and capacity of a drive are inversely proportional. Near-Line SAS (NL-SAS) drives have the highest capacity and the lowest speed. Flash Drives have the lowest capacity and the highest speed. SAS drives of either drive speed can have the same capacity ranging between that of NL-SAS and flash drives.

In legacy EMC storage systems, only a single physical form factor storage device was available. On the VNX there are two formats: 2.5” and 3.5”. The 2.5” drives allow for putting more capacity into a storage system using a smaller volume of space.

All available VNX RAID levels are supported by all the drive type. Each RAID level and RAID group size has a different amount of user data capacity. RAID level 5 groups offer the highest ratio of user data capacity. RAID level 6 has a somewhat lower ratio, but offers the highest level of data protection.

Flash drives

Flash drives are a semiconductor-based storage device. They are the highest performing type storage device on the VNX. They offer a very low response time and high throughput in comparison to traditional mechanical hard disks and under the proper circumstances can provide very high throughput. To fully leverage the advantages of Flash drives, their special properties must be taken into consideration to match them to the workload that best fits them.

Flash drive performance overview

As a storage device, flash drives have different characteristics from mechanical hard drives. Flash drives offer the best performance with highly concurrent, small-block, read-intensive, random I/O workloads.

Flash drives have modest capacity compared to available hard drives, and a much higher cost per GB of capacity.

The capacity and provisioning restrictions of the Flash drives in comparison to conventional hard disks may restrict their usage to modest-capacity LUNs or special features such as FAST Cache and FAST VP. Generally, Flash drives provide their greatest performance advantages over mechanical hard drives when LUNs have:

- A drive utilization greater than 70 percent
- A queue length greater than 12
- Average response times greater than 10 ms
- An I/O read-to-write ratio of 60 percent or greater
- An I/O block-size ranging from 4KB to16KB

Flash drive provisioning

Unlike legacy CLARiiON storage systems, on the VNX series, flash drives may be provisioned together in enclosures with *any* other type drive.

There is no restriction on the number of flash drives that may be hosted on any VNX model, other than the model's maximum drive count. However, flash drives are high-performance storage devices that place a greater demand on storage system resources than mechanical hard drives. Provision flash drives by distributing them as widely as is practical across the available SAS back-end ports.

For lowest response time, do *not* fully populate a SAS backend port with flash drives. Distribute flash drives evenly across all the SAS back-end ports. Up to 12 flash drives per SAS back-end port are suggested for bandwidth dependent workloads. For the highest possible throughput, we suggest that a SAS back-end port have only about five (5) flash drives.

Flash drive provisioning for capacity

Currently, flash drives have only modest capacity compared to mechanical hard drives. When practical, provision flash drive-based RAID groups as RAID 5. RAID level 5 groups offer the highest ratio of user data capacity to data protection of all the RAID levels.

Flash drive provisioning for performance

Workloads with specific I/O characteristics benefit the most from flash drive-based storage. In general, overall Flash drive performance is higher than mechanical hard drives. Specifically, read performance is higher than write performance on Flash drives. As such, Flash drives are best utilized with workloads having a large majority of read I/Os to write I/Os.

The following I/O type recommendations should be considered in using Flash drives with parity RAID 5 groups using the default settings:

- **Random reads**: Flash drives have the best random read performance of any storage device. Throughput is particularly high with 32 KB or less block sized I/O. Throughput decreases as the block size increases from the flash drive's page size.
- **Sequential reads**: When four or greater threads can be guaranteed, Flash drives have up to twice the bandwidth of SAS mechanical hard drives with large-block, sequential reads. This is because the Flash drive does not have a mechanical drive's seek time.
- **Random writes**: Flash drives have superior random write performance over SAS hard drives. Throughput decreases with increasing block size.

- **Sequential writes**: When not cached through the storage processor's write cache, flash drives are somewhat slower than SAS mechanical hard drives with single threaded write bandwidth. They are higher in bandwidth to SAS drives when high concurrency is guaranteed. Avoid deploying flash drives on workloads with small-block sequential writes such as in data logging.

The more concurrent the workload, the higher the throughput with Flash drives. Ensuring appropriate balance of Flash-based LUN utilization between storage processors will increase concurrency.

Flash drives and cache OE Block 31.0

By default, with OE Block 31.0 flash drive-based LUNs have both their read and write cache set to *off*. Note the performance described in the previous sections applies to the default cache-off configuration for Flash drive-based LUNs.

However, many workloads that do not risk saturating the cache can benefit from cache-*on* operation. In particular, small-block sequential I/O-based workloads, which are not ideal for flash drives, can benefit from *cached* flash drives. The sequential write performance can be improved by write-cache coalescing to achieve full-stripe writes to the flash RAID groups. With write-cache *ON*, back-end throughput, if properly provisioned for, will be higher than available with hard drives. In addition, read cache may also be enabled, although its effect is dependent on how successfully the workload can leverage the read-ahead for a large read cache capacity.

Flash drive performance over time

Flash drives do not maintain the same performance over the life of the drive. The write IOPS of a flash drive is better when the drive is new, compared to when it has been in service for a period of time. This is characteristic of all SSD-type Enterprise storage devices. The difference in write speed is the result of the drive's management of its internally reserved capacity. It is *not* individual locations "wearing out." The internally used capacity is used for 'house keeping' processes such as garbage collection and wears levelling. Once the new drive's reserved capacity has been written at least once, garbage collection becomes the norm - -after which the drive will have a steady-state performance level.

The performance metrics used as a rule-of-thumb in this document are a conservative per-drive IOPS number that is achievable in the long run, with a broad range of I/O profiles for Flash drives that have been in service over a long period.

Additional information

An in-depth discussion of Flash drives can be found in the *An Introduction to EMC CLARiiON and Celerra Unified Storage Platform Storage Device Technology* and the Unified Flash Drive Technology Technical Notes white paper available on Powerlink.

Mechanical hard drives

Several different mechanical hard drives are available on the VNX series. These storage devices have different performance and capacity characteristics that make them appropriate to a wide range of workloads.

The VNX series supports the following types of storage devices:

- SAS 15k rpm and 10k rpm hard drives
- NL-SAS 7.2k rpm hard drives

Mechanical hard drive provisioning

There are very few provisioning restrictions with mechanical hard drives on the VNX series compared to legacy storage systems. Their usage requires the following provisioning restrictions:

- Mechanical hard drives require the same type drive hot spares.
- RAID groups are made-up of the same type hard drives only.

For best performance, provision the drives conservatively in terms of IOPS and bandwidth. Distribute the drives evenly as practical across all SAS back-end ports.

SAS hard drives

SAS drives are the main performance storage device on the VNX series. Knowing the workload's I/O characteristics is important in choosing between the 15K rpm and 10K rpm SAS drives.

SAS drive provisioning for capacity

SAS drives have moderate to high capacity compared to all VNX storage devices. Provision SAS drive-based RAID groups as RAID 5.

SAS drive provisioning for performance

Generally, all workloads will benefit from provisioning with SAS drive-based storage.

The SAS hard drives with 15K rpm rotational speed have a reduced access time over the 10K rpm drives. The 15K rpm SAS drive will be about 30-percent faster than a 10K drive in performing I/O, on a *per drive* basis. However, with cached I/O and the standard RAID level 5 (4+1) grouping this difference becomes smaller. Use of 15K rpm SAS drive is recommended only when responses need to be maintained in workloads with strictest response time requirements.

The following I/O type recommendations should be considered in choosing between 10K and 15K SAS drives (both drives using parity RAID 5 groups and the default settings):

- **Sequential reads**: SAS drives of either speed provide consistent performance for 8KB and larger sequential reads and writes.
- **Random reads**: SAS hard drives with 15K rpm rotational speed have a reduced service time with random reads. SAS hard drives with 10K rpm rotational speed have a lesser random read I/O capability.
- **Sequential writes**: Similar performance with either drive speed with block sizes 8KB and larger.
- **Random writes**: SAS hard drives with 15K rpm rotational speed have slightly better performance than 10K rpm drives. The difference in write performance is greater than the difference in read performance between the two speeds.

SAS hard drives offer the good performance with single threaded and moderately concurrent, 16KB and larger block I/O workloads. Note that for large block random I/O, 15K rpm SAS drives may be a better and more economical solution than flash drives because of their higher capacity and comparable bandwidth.

SAS hard drives have moderate capacity compared to other available storage devices, and an intermediate cost per GB of capacity. The 10K rpm drives have a lower cost per GB than the 15K rpm drives.

NL-SAS hard drives

NL-SAS drives are the primary bulk- storage device on the VNX series. These drives provide very high per drive capacity with modest performance.

NL-SAS drive provisioning for capacity

NL-SAS drives have the highest capacity of all VNX storage devices. The use of RAID 6 is *strongly* recommended with NL-SAS drives with capacities of 1 TB or larger.

NL-SAS drive provisioning for performance

Knowledge of the workload's access type is needed to determine how suitable 7.2k rpm NL-SAS drives are for the workload. NL-SAS drives are economical in their large capacity; however they do not have the high performance nor the same availability characteristics as SAS hard drives.

The following I/O type recommendations should be considered in using NL-SAS drives with parity RAID 5 groups (for close comparison) using the default settings:

- **Sequential reads**: NL-SAS drive performance approximates 10K rpm SAS hard drive performance.
- **Random reads**: NL-SAS drive performance is about half of the 15K SAS drive
- **Sequential writes**: NL-SAS drive performance is comparable but lower than 15K rpm SAS performance.
- **Random writes**: NL-SAS drive performance is about half of the 15K rpm SAS drive. It decreases in comparison to SAS performance with increasing queue depth.

Additional information

For additional information on VNX mechanical hard drives can be found in the *EMC Unified Storage Device Technology* whitepaper available on Powerlink.

Drive performance estimates

Estimating the performance of drives within a workload requires a good understanding of the workload, and the performance characteristics of the drives. For example, as a minimum the following information is needed:

- Threading model of the workload
- Type of I/O (random or sequential)
- I/O block size
- Candidate drive type performance

Rule-of-thumb approach

This section provides the basic information on candidate drive type performance. The IOPS per drive or MB/s per drive, depending on the I/O type is an important part is in a drive performance estimate. Use the guideline values provided in the tables below for estimation purposes.

These values are intentionally conservative. They are intended to give simplistic measure of performance. Extending the estimate to include for RAID group response time (for each drive), bandwidth, and throughput need to account for the I/O type (random, sequential, or mixed), the I/O size, and the threading model in use are not covered here.

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.

Random I/O Performance

Small-block (16 KB or less per I/O) random I/O, like those used in database applications and office automation systems, typically require throughput with an average response time of 20 ms or better. At an average drive-queue depth of one or two, assume the following per drive throughput rates:

Drive type	IOPS
SAS 15K rpm	180
SAS 10K rpm	150
NL-SAS 7.2K rpm	90
Flash drive	3500

Table 9 Small block random I/O performance by drive type

In cases of random I/O sizes greater than 16 KB, there will be a steady reduction in the IOPS rate. The IOPs for flash drives falls-off more quickly than mechanical hard drives.

The number of threads can increase the IOPS.

For example, a single threaded application has one outstanding I/O request to a drive at a time. If the application is doing 8 KB random reads from a 10K rpm SAS drive, it will achieve 130 IOPS per drive. The same application, reading a drive through 12 simultaneous threads of the same size, can achieve 240 IOPS at 50 ms response time per I/O. Flash drives are particularly well-suited to multi-threading.

When architecting for optimal response time, limit the drive throughput to about 70 percent of the throughput values shown in Table 9 Small block random I/O performance by drive type by relaxing response time and queue depth ceilings. If a response time greater than 50 ms and a drive queue depth of eight or more is allowable, the table's drive throughput can be increased by 50 percent more IOPS per drive.

For random I/O requests 64 KB to 1MB, as the block size increases, so does the per-drive bandwidth.

Note the number of threads has a large effect on large block random bandwidth.

Sequential I/O Performance

For 64 KB and greater block sizes running single thread sequential I/O, RAID group striping makes bandwidth independent of the drive type. Use 30 MB/s per drive for 10K rpm SAS as conservative design estimate.

Mixed I/O Performance

In mixed loads for mechanical hard drives, the pure sequential bandwidth is significantly reduced due to the head movement of the random load, and the random IOPS are minimally reduced due to the additional sequential IOPS. The sequential stream bandwidth can be approximated using the values in Table 9 Small block random I/O performance by drive type and the random load can be approximated by using 50 percent of Table 9 Small block random I/O performance by drive type's IOPS. Aggressive prefetch settings (prefetch multiplier 16, segment multiplier 16) improve the sequential bandwidth at the expense of the random IOPS. Increasing the random load queue depth increases its IOPS at the expense of the sequential stream bandwidth.

Additional drive performance information

Contact your EMC Sales Representative to engage an EMC USPEED Professional for specific drive performance information.

Utilization recommendation

As drive utilization increases response time likewise increases according to Little's Law. At about 66-percent utilization response times increase dramatically. It is *not* recommended to use the full Rule-of-Thumb IOPS for performance planning.

A drive that has 180 IOPS, (15K RPM SAS), will be at near full utilization under that that throughput. The response times of individual IOs can be very large. It's prudent to plan for

about 2/3 of Rule-of-Thumb IOPS for normal use. This leaves more than enough margin for bursty behavior and degraded mode operation.

Availability

Availability refers to the storage system's ability to provide user access to their applications and data in the case of a hardware or software fault (sometimes called a *degraded* state or mode). Midrange systems like the VNX-series are classified as *highly available* because they provide access to data without any single point-of-failure.. Generally, some aspect of storage system performance in degraded mode is lower than during normal operation. The following optimizations to the VNX's platform configuration can improve performance under degraded mode scenarios.

The following sections cover the storage system's availability Best Practices.

Back-end

The following sections describe Back-end availability best practices.

Enclosures and drive placement

Drive placement within enclosures at the RAID group level of organization can have an effect on storage system availability.

Horizontal Provisioning

Single DAE Provisioning is the practice of restricting the placement of a RAID groups within a single DAE (or DPE). This is sometimes called *horizontal* provisioning. Single DAE provisioning should be the default method of provisioning RAID groups. Owing to its convenience and High Availability (HA) attributes, Single DAE provisioning is the most commonly used method.

In *Multiple DAE Provisioning*, two or more DAEs are used. An example of multiple DAE provisioning requirement is where drives are selected from one or more other DAEs because there are not enough drives remaining in one DAE to configure a desired RAID topology. Another example is SAS back-end port balancing. The resulting configuration may or may not span back-end ports depending on the storage system model and the drive-to-enclosure placement.

Vertical Provisioning

Multiple DAE provisioning can also be used to satisfy *performance requirements* specifically related to back-end port behavior. . The SAS protocol's built-in ability to simultaneously access more than one drive at the same time on a single port results in a much smaller positive effect on VNX series storage system's performance than with legacy CLARiiONs

With vertical provisioning, a RAID group (see the RAID groups section.) is intentionally provisioned spanning more than one DAE to gain parallel access to two or more back-end ports. These cases arise when:

- There are well-defined, higher-than-normal bandwidth goals that require careful load distribution
- Spikes in host I/O activity are known to cause backend port bottlenecks
- Heavy write cache destage operations on a RAID group interfere with host reads
- Heavy large block I/O requests on a backend port interfere with small block I/O response time

Note the above cases are considered exceptional behavior. In addition, users will not gain the typical performance benefits attributed to vertical provisioning in Virtual Provisioning pools.

DAE availability

At a high-level, a DAE is made-up of the following components:

- 2x Link Controller Cards (LCC)
- Slots
- Drives

The LCC is the device that connects the drives in a DAE to one SP's SAS back-end port. A peer LCC connects the DAE's drives to the peer SP. (See Figure 5. VNX SAS Backend Port Conceptual Diagram.) In the case of a single LCC failure, an SPs LUNs on the DAE RAID groups will be without enough drives to operate. All the horizontally provisioned LUNs on the RAID groups in the DAE will lose connectivity. In a single DAE LCC failure, the peer storage processor still has access to all the drives in the DAE. Because of this, RAID group rebuilds are avoided. The storage system automatically uses its lower director capability to re-route around the failed LCC and through the peer SP, until it is replaced. The peer SP experiences an increase in its bus loading while this redirection is in-use. The storage system is in a degraded state until the failed LCC is replaced. When direct connectivity is restored between the owning SP and its LUNs, data integrity is maintained by a background verify (BV) operation.

Request forwarding's advantages of data protection and availability result in a recommendation to horizontally provision. In addition, note that horizontal provisioning requires less planning and labor.

If vertical provisioning was used for compelling performance reasons, provision drives within RAID groups to take advantage of request forwarding. This is done as follows:

- RAID 5: At least two (2) drives per SAS back-end port in the same DAE.
- RAID 6: At least three (3) drives per back-end port in the same DAE.
- RAID 1/0: Both drives of a mirrored pair on separate backend ports.

See the SAS back-end port balancing section for further details and an example.

FAST Cache

It is required that flash drives be provisioned as hot spares for FAST Cache drives. Hot sparing for FAST Cache works in a similar fashion to hot sparing for traditional LUNs made up of flash drives. See Hot spares section. However, the FAST Cache feature's RAID 1 provisioning affects the result.

If a FAST Cache Flash drive indicates potential failure, proactive hot sparing attempts to initiate a repair with a copy to an available flash drive hot spare before the actual failure. An outright failure results in a repair with a RAID group rebuild.

If a flash drive hot spare is not available, then FAST Cache goes into degraded mode with the failed drive. In degraded mode, the cache page cleaning algorithm increases the rate of cleaning and the FAST Cache is read-only.

A double failure within a FAST Cache RAID group may cause data loss. Note that double failures are extremely rare. Data loss will only occur if there are any dirty cache pages in the FAST cache at the moment both drives of the mirrored pair in the RAID group fail. It is possible that flash drives data can be recovered through a service diagnostics procedure.

System drives

The first four drives, 0 through 3 in a DPE or in the DAE-OS of SPE-based storage system models VNX models are the system drives. The system drives may be referred to as the *Vault drives*. On SPE-based storage systems the DAE housing the system drives may be referenced as either DAE0 or DAE-OS. On the VNX series with OE Block 31.0 *only* SAS drives may be provisioned as system drives. These drives contain files and files space needed for the:

- Saved write cache in the event of a failure
- Storage system's operating system files

- Persistent Storage Manager (PSM)
- Operating Environment (OE) -- Configuration database

The remaining capacity of system drives not used for system files can be used for user data. This is done by creating RAID groups on this unused capacity.

System drive provisioning

Special attention is needed when the unused capacity of the system drives are used for user data.

System drive's and Virtual Provisioning pools

No part of a system drive's capacity may be used in Virtual Provisioning pools.

System drive capacity provisioning

System drives on the VNX-series for OE Block 31.0 and File 7.0 use about 192 GB of their per-drive capacity for system files. The portion of the drives not used for the system files is user configurable. A RAID group for traditional LUNs may be created there.

It is *not* recommended to combine system drives with non-system drives in RAID groups. The system files already consume capacity on the system drives. Binding the system drives with non-system drives would truncated the non-system drives capacity to the same capacity as a system drive.

To provision system drives for the highest useable capacity they can be configured as a four drive RAID level 5 (3+1) and used to host traditional LUNs.

System drive performance provisioning

User workloads may be applied to LUNs on the system drives. However, it is possible for user LUNs created on the system drives to adversely affect overall system performance, or storage system availability.

Heavily used LUNs should not be placed on the system drives. These LUNs include:

- Reserved LUN Pool
- Mirror LUNs

When LUNs are provisioned on a RAID group composed of the system drives do not assume the full bandwidth of the RAID group will be available. Plan bandwidth utilization for LUNs on system drives as if they are sharing the drives with an already busy LUN. This will account for the operating environment's vault drive utilization. Heavy user usage of the system drives can cause delays in the storage system's accessing to the vault drive's files.

The table below shows the maximum recommended host I/O loads for the system drives.

Maximum Vault drive host I/O loading		
System Drive Type	**Max. IOPS**	**Max. Bandwidth (MB/s)**
15k rpm SAS	150	10
10k rpm SAS	100	10

Table 10 Maximum host I/O loads for vault drives, OE Block 31.0

Note that with OE Block 31.0 flash drives are not permitted to be provisioned as system drives.

System drive data protection

The system files on the vault drives have their own internal data redundancy scheme implemented at the operating environment level. The system files do not use hot spares. User

LUNs created on the capacity of the vault drives not used by the system files are eligible for hot sparing.

Hot spares

Hot spares are drives provisioned to automatically replace failing drives or drives that have abruptly failed.

Proactive sparing is the process of automatic replacement with a hot spare when a drive indicates it is likely to fail. Failure indication comes from OE Block's continuous diagnostics. This is essentially a drive-to-drive copy operation. It is very fast and uses a minimum of storage system resources.

Note that a hot spare is a temporary replacement for a failing or failed drive. A RAID group operating with a hot spare is in degraded mode. Degraded mode ends when the failed drive has physically been replaced, rebuilt, and the hot spare is returned to the pool of available hot spares. A hot spare may or may not be identical to the drive it is replacing. This may result in a difference between performance from normal operation.

Proactive sparing on the VNX automatically selects hot spares from a group of drives designated as hot spares. Hot spares are global. Global means they are shared by all the same type, in-use drives on the storage system.

Careful provisioning of hot spares ensures the efficient use of available capacity and best performance until a replacement drive is installed. Otherwise, it is possible to have too many hot spares, or for a less appropriate drive to be used as a hot spare replacement.

For example, a 10K rpm SAS drive could automatically be selected as a spare for a failing 15RPM SAS drive. This would result in a noticeable RAID group performance degradation.

Hot spares are specially provisioned drives. There is no operating environment enforcement to allocate any drives as hot spares. However, it is *strongly* recommended that hot spares be provisioned.

Hot spare algorithm

The appropriate hot spare is chosen from the provisioned hot spares algorithmically. If there were no hot spares provisioned of appropriate type and size when a drive fails, no rebuild occurs. (See the Drive Rebuilds section.) The RAID group with the failed drive remains in a degraded state until the failed drive is replaced; then the failed drive's RAID group rebuilds.

The hot spare selection process uses the following criteria in order:

1. **Failing drive in-use capacity -** The smallest capacity hot spare drive that can accommodate the in-use capacity of the failing drive's LUNs will be used.
2. **Hot spare location -** Hot spares on the same back-end port as the failing drive are preferred over other like-size hot spares.
3. **Same Drive type –** Hot spare must be of the same drive type.

Failing drive in-use capacity

It is the in-use capacity of the failing drive's that determines the capacity of the hot spare candidates. Note this is a LUN-dependent criterion, not a raw drive capacity dependency. This is measured by totalling the capacity of the drive's bound LUNs. The in-use capacity of a failing drive's LUNs is not predictable. This rule can lead to an unlikely hot spare selection. For example, it is possible for a smaller capacity hot spare to be automatically selected over a hot spare drive identical to, and adjacent to the failing drive in the same DAE. This occurs because the formatted capacity of the smaller, hot spare (the highest-order selection criteria) matches the in-use capacity of the failing drive's LUNs more closely than the identical hot spare drive.

Note that a hot spare's form factor and speed are *not* a hot spare criteria within the type.

For example, a 3.5" format 15K rpm drive can be a hot spare for a failing 2.5" 10K rpm SAS drive.

Hot spare location

Ideally, the automatically chosen hot spare will be of the same speed, and capacity as the failing drive. This would prevent any loss of performance while the RAID group is in degraded mode while operating with a hot spare in the RAID group. To influence this type of selection, ensure an appropriate hot spare is located on the same backend port of the drives you'd like it to replace. This may require provisioning that segregates drives by type to different backend ports to make the most economical use of the fewest hot spares. Note that the backend port attachment should always be verified when provisioning hot spares.

Same Drive type

Generally, a drive can only hot spare for the same type drive. That is:

- NL-SAS hot spares can only hot spare for NL-SAS drive.
- SAS hot spares can only hot spare for another SAS drive.
- Flash hot spares can only hot spare for another Flash drive.

System drives

System drives do not require hot spares and should not be included in the hot sparing ratio. However, user LUNs located on system drives will be re-built to available hot spares should a system drive fail.

Hot spare provisioning

Performance in degraded mode, and overall availability need to be considered with provisioning hot spares. High availability requires that there be one hot spare for every in-use 30 drives (1:30).

For mechanical hard drives, hot sparing does not take into account drive speed. Use the highest speed drive practical for a hot spare. This will avoid a lower speed drive being selected as a hot spare, which may affect LUN performance until the failed drive is replaced. In addition, in an ASAP Rebuild, the lower speed drive would extend the rebuild's duration.

For example, it's possible for a 10K rpm SAS drive to be used as a hot spare for a failing drive in a 15K rpm SAS-based RAID group. While the 10K rpm hot spare is substituting for the failed 15K rpm drive, the 15K rpm-based RAID group would have lower performance.

If users believe they have sound data protection policies and procedures in place (frequent backups, an already historically sufficient number of hot spares, etc.), they may choose to increase the ratio of in-use drives to hot spares.

For example, for storage environments provisioned for high storage density, with a large number of the same drive, rounding to one hot spare per two DAEs *may* be an alternative. A large number of drives are 32 or more in-use RAID groups.

Hot sparing best practices summary

The following summarizes the hot spare best practices:

- Have at least one hot spare drive of the same type, speed, and maximum needed capacity as the drives on the storage system.
- Position hot spares on the same back-end ports containing the drives they may be required to replace.
- Use the highest speed mechanical hard drive that is practical as a hot spare.
- Maintain a 1:30 ratio of hot spares to in-use drives for highest availability.

Additional information

An in-depth discussion of hot sparing can be found in the *EMC CLARiiON Global Hot Spares and Proactive Hot Sparing* white paper available on Powerlink.

High availability, high performance hot sparing example

The following is an high availability example of hot sparing. Assume a VNX5300, which has two (2) backend SAS ports. The storage system has the following enclosures:

- 1x DPE8
- 2x DAE5Ss

The storage system is to be provisioned with the following hard drives:

- 5x 200 GB flash drives
- 17x 7.2k rpm, 2 TB NL-SAS hard drives
- 25x 10K rpm, 300 GB SAS hard drives

How many hot spares should be used to complete the provisioning for this storage system? Where should they be positioned?

Configuring the storage system for hot spares

The following table summarizes the hot spare provisioning of the storage system for availability.

DPE/DAE	Back-end port	System drives	Data drives	Hot spares	Total DAE drives
DPE	0	4x SAS	20x SAS	1x SAS	25
DAE 1	1	0	15x NL-SAS	0	15
DAE 2	1	0	1x NL-SAS 4x Flash Drive	1x NL-SAS 1x Flash Drive	7
* System drives are not hot spared, they have their own protection.			**Total drives:**		47
			Total SAS drives		25
			SAS hot spares ratio		1:20*
			Total NL-SAS drives		17
			NL-SAS hot spare ratio		1:16
			Total Flash Drives		5
			Flash Drive hot spare ratio		1:4
			Effective hot spare ratio:		1:15*

Table 11 Hot spare provisioning example

A DPE-type enclosure is always on backend port 0. Twenty-five 300 GB 10K RPM SAS drives are provisioned on it. Four of these hard drives are used as the system drives. Note that hot spares are *not* provisioned for the system drives. This provides for 4x (4+1) SAS RAID groups. One of the SAS drives is a hot spare. Being on backend port 0 with the SAS drives, it has port affinity for sparing with them.

DAE1 is provisioned on backend port 1. It is provisioned with 15x 7.2k rpm 2 TB SAS drives. This provides for 1x complete (6+2) RAID group and part of another.

Configure DAE2 to be on backend port 1. It is provisioned with the remaining 4x NL-SAS drives. One NL-SAS drive completes the partial NL-SAS RAID group in DAE1. The storage system can have 2x (6+2) NL-SAS RAID groups. The second NL-SAS drive is a hot spare. In addition, this DAE is provisioned with all the flash drives. This includes four flash drives intended for use as a FAST Cache. A fifth flash drive is for use as a hot spare. Note that the NL-SAS hot spare cannot be as a spare for a failing SAS drive. A SAS drive cannot be a spare for a failing NL-SAS drive. Also that flash drives can only hot spare for flash drives.

In this provisioning the ratio of drives to hot spares is lower than 1:30. This was done intentionally. It provides for a higher level of performance in degraded mode should any of the several drive types in use fail.

Drive Rebuilds

A drive rebuild replaces the failed drive of a RAID group with an operational drive created from a hot spare or a replacement drive.

One or more LUNs may be bound to the RAID group with the failed drive. If the failing drive is part of Virtual Provisioning pool, data from a great many LUNs may be on the failed drive. For the rebuild replacement to occur, there must be a hot spare provisioned that is the correct type and capacity to replace the failed drive or the failed drive may be physically replaced from on-site stores to be rebuilt directly.

If an appropriate hot spare is not available, the RAID group remains in a degraded mode until the failed drive is physically replaced. It is always prudent to minimize the length of time a RAID group is in degraded mode. *Both* availability and performance are decreased in degraded mode.

The failed drive's RAID group rebuilds the LUN contents of the failed drive from parity or its mirror drive depending on the RAID level.

When a hot spare is used, a rebuild is a two-step process: rebuild and equalize. They occur when a drive fails. During the rebuild step, all data that is part of user LUNs located on the failing drive are rebuilt to the replacement (either the actual replacement drive or the Hot Spare) either from parity or their peer drive with mirror RAID types. If using a Hot Spare, the data is rebuilt to the Hot Spare and when the replacement drive is installed, the data is Equalized from the Hot Spare to the newly installed replacement drive.

Rebuild considerations

The duration of a rebuild depends on a number of factors, including:

- In-use capacity of failed drive
- Rebuild priority
- Presence and physical location a hot spare
- Hot spare drive type
- RAID group type
- Current workload

In-use capacity

Only the capacity of the failed drive being used for user data is rebuilt. This can range from a small fraction to the entire capacity of the drive.

Rebuild priority

A LUN rebuild is a prioritized operation. Priority has the biggest influence on how quickly a drive is rebuilt. The priority allocates the amount of storage system resources dedicated to the operation. The available priorities are:

- ASAP (As Soon As Possible)
- High
- Medium
- Low

Once the rebuild is complete, the RAID group operates normally. Performance and availably may be partially or wholly restored, depending on the type of hot spare in-use. During the equalize step, which occurs after the faulty drive is physically replaced, the hot spare is copied by an Equalize operation to the replacement for the original failed drive. After equalize the RAID group is no longer in a degraded mode. After equalize, the hot spare becomes available as a hot spare for any other failing drive.

Rebuild priority rates

During a rebuild, overall storage system performance may be adversely affected. The Low, Medium, and High priorities make the most economical use of the storage system's resources during the rebuild process. Low reduces the effect of a rebuild to almost nothing, high somewhat larger. The lower the priority, the longer the duration of the rebuild, which extends the period in degraded mode. High is the default and recommended rebuild priority to use, unless restoring a LUN from degraded mode in the shortest period of time is critical.

The table below shows the rule-of-thumb rebuild rate for 10K rpm SAS drives in RAID 5 and RAID 6 groups. Calculate the duration by multiplying the number of data drives in the group by the in-use capacity of the LUNs.

Priority	Parity RAID Rate (MB/s) per drive
Low	3.3
Medium	6.2
High	12.0

Table 12 Mirrored and parity RAID SAS drive rebuild rates, OE Block 31.0

The ASAP priority uses all available storage system resources to complete a rebuild in the shortest period of time. The reallocation of resources may have an adverse effect on other workloads. It is *not* recommended to start more than one (1) ASAP rebuild at a time on a storage processor.

An ASAP rebuild has different rates depending on the RAID type, the hard drive type, and for parity RAID types, the number of drives in the RAID group. It also depends on the location of the drives – a rebuild of a Parity group at ASAP engages all the drives in the group, and the aggregate bandwidth can approach or even hit the maximum bandwidth of a SAS backend-port. If all drives are on one back-end port, then the rebuild rate may be limited by backend-port bandwidth.

<table>
<tr><th rowspan="2">RAID Group</th><th>ASAP 10K RPM SAS RAID Group Rebuild Rate</th><th>ASAP 15K RPM SAS RAID Group Rebuild Rate</th><th>ASAP Flash SAS RAID Group Rebuild Rate</th></tr>
<tr><td>Rate (MB/s)</td><td>Rate (MB/s)</td><td>Rate (MB/s)</td></tr>
<tr><td>RAID 1/0 (4+4)</td><td rowspan="3">100</td><td rowspan="2">150</td><td>125</td></tr>
<tr><td>RAID 5 (4+1)</td><td>122</td></tr>
<tr><td>RAID 6 (6+2)</td><td>128</td><td>101</td></tr>
</table>

Table 13 ASAP mirrored and parity RAID 10K RPMSAS drive RAID group rebuild rates with no load, OE Block 31.0

The following table shows the hard drive type speed adjustment. Adjustment is applied to the 10K RPM rate found inTable 12 to arrive at the rate for different drive types. For example to convert the 10K RPM rate of the table to the much faster flash drive rate.

Drive rebuild speed adjustment	
Hard drive type	Rebuild speed multiplier
10k rpm SAS	1.0
15k rpm SAS	1.5
7.2k rpm NL-SAS	1.0
Flash drive	1.3
Flash drive (RAID 6)	1.0

Table 14 Drive rebuild speed adjustment, OE Block 31.0

When a hot spare is used for the rebuild, an additional equalize operation occurs when the faulty drive is replaced and the hot spare is copied to it. The equalization rate for all rebuild priorities with 10k rpm SAS drives is about the same as the rebuild rate. This rate is independent of other factors.

Basic rebuild time calculation

Use the following formula to estimate the time required to complete a rebuild.

- Time: Duration of rebuild
- Failed hard drive capacity: RAID group capacity utilization * hard drive size in GB
- Rebuild rate: If priority is ASAP, use the time listed in Table 13. Otherwise, use the value from Table 12.
- Hard drive type and speed adjustment: Speed adjustment is found in Table 14 Drive rebuild speed adjustment, OE Block 31.0. Use this data, if the ASAP data, if using economical rebuild priorities otherwise for ASAP use Table 13 ASAP mirrored and parity RAID 10K RPMSAS drive RAID group rebuild rates with no load, OE Block 31.0 which has the drive type built-in.
- Equalization Rate: Speed at which the hot spare is copied to replacement for a failed drive, use the Rebuild Rate for the RAID group.

Time = ((Failed hard drive capacity * Rebuild rate) * Drive type and Speed adjustment) + ((Failed hard drive capacity * Equalization rate) * Drive type and Speed adjustment))

Note the rebuild has two parts: rebuild and the equalization. Manual replacement of the failed hard drive must occur before equalization. After the rebuild the RAID group is running at full capability. The RAID group is no longer in a degraded status. The secondary equalization is an automated background process starting with the replacement of the failed drive.

Example calculation

How many hours will it take to rebuild a 600GB 15K RPM SAS drives in a 5-drive RAID 5 group at the ASAP priority?

Extracting the needed rates from the specified tables and converting the units results in the following table.

Failed Hard Drive Capacity (MB)	Rebuild Rate (MB/hour)	Hard Drive Type and Speed Adjustment	Equalization Rate (MB/hour)
614400	540000	1	540000

Table 15, Rebuild Example parameters

Applying the values to the example calculation results in:

2.28 hrs = (614400 MB * (1 hour / 540000 MB) * 1) + (614400 MB * (1 hour / 540000 MB) * 1)

Maintaining performance during an ASAP rebuild

The effect of an ASAP rebuild on application performance depends on the workload mix. There are three effects to consider:

- Rebuilding drive utilization
- Backend port bandwidth utilization
- Relative drive queue contention

All applications performing I/O with LUNs on the rebuilding RAID group will be slower. The rebuild will be sending up to eight I/Os at a time to each drive in a parity group. If the group is a mirror, it will send eight I/Os to the remaining mirror. Longer service times for the large rebuild reads and the longer queue will increase response time.

Port bandwidth utilization will affect all drives on the same port. As port utilization climbs, there is less bandwidth available for other processes. Please note that when all drives of a parity RAID group are on the same bus, depending on the RAID group size and drive type, a large percentage of the available port bandwidth may be consumed by an ASAP rebuild. The rebuild of large RAID groups can consume a significant portion of available port bandwidth. If bandwidth-sensitive applications are executing elsewhere on the storage system, RAID groups should previously have been distributed across all available ports. (See SAS back-end port balancing section.)

The contention for the drives between production and rebuild can be seen in the relative queues. Rebuild and equalize operations do not use the read or write cache subsystems, but they do send multiple, large I/Os at the same time. Host I/O behavior may depend on the source pattern or on cache usage. Sequential writes destage from cache with multiple threads per drive with large coalesced block sizes competing with the rebuild operation. Concurrent sequential writes and a rebuild slow each other about the same amount. Sequential reads do not generally generate a similar parallel thread load per drive. Furthermore, there is a reduction of sequentiality at the drive due to contending workloads. For example, sequential read operations like LUN backups run much slower during ASAP rebuilds than when an ASAP rebuild is not executing.

If you have an active sequential read workload on the rebuilding RAID group, the ASAP rebuild rate will be lower than the rates described in this section's tables, due to contention. Storage system performance during an ASAP rebuild can be maintained by increasing the prefetch variables to favor the host read. One option is to change **prefetch multiplier** to 32 and **segment multiplier** to 4. For example, a 64 KB sequential read stream will prefetch 2 MB at a time in 256 KB chunks, substantially increasing the read rate during rebuild. Of course, if other processes are accessing those drives in normal operation, they also will be affected by the bias towards sequential reads.

If the adverse performance effects of an ASAP rebuild cannot be tolerated by the workload, rebuild priority should be set to High, Medium, or Low. These rebuild rates are slower than ASAP, but even at the High setting production workloads are only competing with the rebuild for 10 percent of the time.

Additional information

Additional information on rebuilds may be found in the *EMC Unified Affect of Priorities on LUN Management Activities* available on Powerlink.

Disk power saving (Spin-down)

Disk-drive Spin-down conserves electrical power by *spinning down* drives in a RAID group when the RAID group's drives have not accessed for 30 minutes. Spun-down drives enter an *idle* state. In the idle state, the drives platters do not rotate, which results in saving electricity. A RAID group that is in the power-saving idle state for 30 minutes or longer uses 60 percent less electricity.

Its recommend to use Spin Down for storage systems supporting development, test, and training because these hosts tend to be idle at night. We also recommend Spin Down for storage systems that back up hosts.

Spin-down operation

When an I/O request is made to a LUN whose drives are in spin down to idle mode. The drives must *spin up* before the I/O request can be executed. A RAID group can be on idle state for any length of time. The storage system periodically verifies that idle RAID groups are ready for full-powered operation. RAID groups with drives failing the verification are automatically rebuilt.

Spin-down effect on performance

A host application will see an increased response time for the first I/O request to a LUN with RAID group(s) in idle. It takes less than two minutes for idle drives to spin up. Storage system administrator must plan for this delay when deciding use the Spin Down feature. In particular, consider the ability of the application to wait for I/O.

Spin-down restrictions for use

Spin-down is not a feature available with all VNX drive types. Select drives that have been qualified by EMC for this feature.

For example, all NL-SAS drives are qualified for Spin Down. Spin Down can be configured at either the storage system or the individual RAID group level. It is recommended that the storage system level be used. Using the storage-system level Spin Down will automatically put unbound drives and hot spares into idle. Storage system level setting for drive spindown should be enabled for the RAID Group level setting to be effective.

Spin Down is not supported for a RAID group if private LUNs are bound to the in the group. Generally this includes RAID groups provisioned for use with:

- Installed applications using the Reserved LUN Pool: MirrorView/A, MirrorView/S, SnapView, or SAN Copy
- Virtual Provisioning pool-based LUNs
- MetaLUNs

Spin down cannot be enabled for the system drives.

Additional information

Details on the disk-drive Spin Down feature can be found in the *An Introduction to EMC CLARiiON CX4 Disk-Drive Spin Down Technology* white paper available on Powerlink.

Chapter 4 Block Storage System Best Practices

Block storage system best practices advises on the logical storage objects and their affect on overall storage systems performance and availability. Logical storage objects include:

- RAID groups
- LUNs
- Virtual Provisioning pools
- MetaLUNs

A recommended introduction to the storage system logical storage objects can be found in the *EMC Unified Storage System Fundamentals* whitepaper available on Powerlink.

Performance

The following sections cover the storage system's logical storage object's performance Best Practices.

RAID groups

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

VNX storage systems support RAID levels 0, 1, 3, 5, 6, and 1/0. Refer to the *EMC Unified Storage System Fundamentals* white paper to learn how each RAID level delivers performance and availability.

RAID groups and storage systems

The number of RAID groups that can be created is storage system dependent. The RAID group is the primary logical storage object. It is important that the storage system can support the logical storage object's planned for its use. The table below shows the maximum number of RAID groups per storage system model.

VNX model	Maximum RAID Groups
VNX5100	75
VNX5300	125
VNX5500	250
VNX5700	500
VNX7500	1000

Table 16, Maximum RAID Groups per Storage System Model, OE Block 31.0

Private RAID groups

The Virtual Provisioning feature creates *Private* RAID groups in pool creation. The automatic creation of private RAID groups is included in the number of RAID groups shown in the table above. See the "Virtual Provisioning: Pools" section for details.

RAID groups and drives

There are a few restrictions on the number of drives that make-up a RAID group.

Drives per RAID group

The maximum number of drives in any RAID group is 16. The minimum number depends on the RAID level. The table below shows the minimum number of drives for the most common RAID levels.

Common RAID Levels Minimum and Maximum Drives per group		
RAID Level	Minimum Drives	Maximum Drives
RAID 1/0	2	16
RAID 5	3	
RAID 6	4	

Table 17 Minimum drives for common RAID levels

Generally the more drives in a RAID group the higher its performance. This is because the IOPS of the individual drives is additive for I/O operations that are striped within the RAID group, In addition, capacity utilization increases with increasing number of drives, since the ratio of data to parity increases with the number of drives added to the RAID Group. RAID groups should not be created with the minimum number of drives, unless there are mitigating circumstances – such as when a less-than-optimal number of drives remains and the storage administrator must put those drives into a pool or RAID Group.

RAID capacity utilization characteristics

The different RAID levels have different levels of capacity utilization. The relationship between parity RAID group type and RAID group size is important to understand when provisioning the storage system. The percentage of drive capacity in a RAID group dedicated to data protection decreases as the number of drives in the parity RAID group increases. Creating large RAID groups is the most efficient use of capacity.

For example, for RAID level 6, a 10-drive 8+2 RAID group has a better capacity utilization than an 8-drive 6+2.

With mirrored RAID groups, the storage capacity of the RAID group is half the total capacity of the drives in the group.

With parity RAID groups, the data-to-parity ratio in a RAID group is depends on the RAID level chosen. This ratio describes the number of drives capacity-wise within the group dedicated to parity and not used for data storage. The data-to-parity ratio depends on the number of drives in the RAID group. A low ratio of data-to-parity limits the utility of parity RAID groups with the minimum number of drives. For example, a four-drive RAID 6 group (2+2) is never recommended.

In RAID 5 and RAID 6 groups, no single drive is dedicated to parity. In these RAID levels, a portion of all drives is consumed by parity. That is, the parity metadata is spread across all the drives of the group. For parity RAID level 5 the drive-equivalent capacity dedicated to parity is one; for parity RAID level 6 the equivalent capacity is two.

RAID Group Capacity Utilization	
RAID Group Level and Size	Usable Storage
RAID 1/0 (4+4)	50%
RAID 5 (4+1)	80%
RAID 5 (6+1)	86%
RAID 6 (6+2)	75%

Table 18 Common RAID group capacity utilizations

In summary, some percentage of available drive capacity is used to maintain availability through data redundancy. Configuring the storage system's drives as parity-type RAID groups results in a higher percentage of the installed drive capacity available for user data storage than with mirror-type RAID groups. Note that setting the Virtual Provisioning pool RAID level sets the RAID level of the private RAID groups making-up the pool. The larger the parity RAID group is, the larger the usable *storage percentage* becomes. However, performance and availability, in addition to storage capacity, also need to be considered when provisioning RAID types.

RAID group performance and I/O block size

Best RAID group write performance results from 'the full stripe write'. This is the most efficient movement of data from cache to the storage devices. Stripe capacity or *stripe size* is calculated as the number of user drives in the RAID group multiplied by the stripe block size. The default RAID group stripe block size is 64KB.

Aligning the majority workload I/O block size with the RAID Group stripe size has a general performance benefit.

The table below shows the stripe size for the commonly used RAID Groups.

RAID Group Stripe Size	
RAID Group Level and Size	Stripe Size (KB)
RAID 1/0 (4+4)	256
RAID 5 (4+1)	
RAID 6 (6+2)	384

Table 19 Common RAID group stripe sizes

Aligning the RAID group stripe size with the most common I/O block size results in higher performance. Alignment matches the I/O block size exactly to the stripe size capacity-wise or as a multiple of the stripe size. The performance advantage is particularly true for RAID groups used for large-block (≥ 64KB) random workloads or when cache is disabled for the LUN. The "evenness" of the stripe size has little or no effect on small block random or sequential access. Note alignment is most effective when the I/O block size is not too varied.

The full stripe is by nature 'even'. The write utilizes all the drives in the RAID group at the same time. An odd-sized stripe with a large I/O block size results in the I/O wrapping to a single drive of the next stripe. This delays the I/O.

For example, on an 'odd-sized' RAID 1/0 group of six drives (3+3), the stripe is 192 KB (3 * 64 KB). If the majority I/O block size is 256 KB, it results in wrapping to the next stripe by one drive. This is less optimal for high bandwidth, large-block random I/O.

General RAID level performance characteristics

The different RAID levels have different performance depending on the type of RAID and the number of drives in the RAID group. Certain RAID types and RAID group sizes are more suitable to the I/O of particular workloads than others.

For example, the 10-drive RAID level 6 (8+2) group is particularly suited to the large-block sequential I/O pattern and high availability requirements of the backup workload. In the same way, the 8-drive RAID level 1/0 (4+4) is used for the small-block, random I/O pattern of On-line Transaction Processing (OLTP) workloads.

The following sections address the most commonly used RAID levels. For a complete discussion of all the available RAID levels see the *EMC Unified Storage System Fundamentals* whitepaper available on Powerlink.

When to use RAID 5

RAID 5 has excellent random read performance. Performance improves with increasing numbers of disks in the RAID group. Random write performance is slower than read due to the parity calculation. Sequential read performance is good. Sequential Write performance is good to excellent. Highest sequential write performance occurs when full stripe writes occur.

RAID 5 is the recommended RAID level for flash and SAS drives. It has the best ratio of usable to user capacity for parity-protected RAID groups. When manually provisioning RAID groups with RAID 5, the preferred RAID grouping is five drives (4+1). The default RAID level 5 Virtual Provisioning option is the five drive (4+1). This size offers the best compromise of capacity, performance, and availability for the largest number of workloads.

When to use RAID 6

RAID 6 has similar performance to RAID 5. Where RAID 6 suffers in comparison is in the requirement for the additional parity calculation. Random write performance (equal user data drive count) is slower with RAID 6 than RAID 5 due to the double parity calculation. It has excellent random read performance. Sequential read performance is good. Performance

improves with smaller stripe widths. Sequential write performance is fair to very good. Highest sequential write performance occurs when full stripe writes occur.

The default RAID level 6 Virtual Provisioning pool option is an eight-drive (6+2) group. This size offers a good compromise of capacity, performance, and availability for the largest number of workloads.

The use of RAID 6 is recommended with NL-SAS and drives ≥1 TB in capacity. In particular, when high-capacity capacity drives are used in Virtual Provisioning pools, they should be configured in RAID 6.

The ratio of data to parity is an important consideration when choosing a RAID 6 group size. There is a reduction in user capacity equivalent of two drives with each RAID 6 group. For example, a five-drive RAID 5 (4+1) group would need to migrate to a six-drive RAID 6 (4+2) group of equal capacity drives to have the same user data capacity.

RAID 6 vs. RAID 5 performance

In general the difference in performance between RAID 5 groups and RAID 6 groups is small. The difference is two additional I/Os (a read and a write) to the RAID group per write I/O operation.

With the same number of user drives, for random workloads, RAID 6 performs the same as RAID 5 for read I/Os. Random writes are different. The additional parity drive over a RAID 5 group increases the RAID 6 back-end workload by 33-percent for writes. This affects the performance of the RAID group as the number of drives in the RAID group increases. In addition, the overhead of RAID 6 may lead to cache filling earlier than might happen on RAID 5. However, as long as workload can be normally destaged from cache, avoiding forced flushing, RAID 5 and RAID 6 groups will have about the same random I/O host response time.

For sequential workloads with the same number of drives, read performance is nearly identical. Sequential writes to a RAID 6 group have about a 10-percent lower performance than a RAID 5 group with the same number of user drives.

From a host response time perspective, running fully cached, with a properly sized system the performance difference described above is marginal.

When to use RAID 1/0

RAID 1/0 receives a performance benefit from mirrored striping. It has very good random read and write performance. RAID 1/0 includes some optimization of reads that takes advantage of two drives with identical data. RAID 1/0 offers better small-block write performance over any parity RAID level because it does not need to calculate parity during write

RAID 1/0 has good sequential read and write performance. However, it is outperformed by RAID 5 and RAID 6 in sequential write workloads. This is because it needs to write sequential data twice on its drives. With full stripe writes, both RAID 5 and 6 can write data only once.

The default RAID level 1/0 Virtual Provisioning option is an eight-drive (4+4) group. This size offers a good compromise of capacity, performance, and availability for the largest number of workloads.

RAID 1/0 has the lowest capacity utilization. With a small number of drives, it does not suffer in comparison to parity RAID levels. However, as the number of drives in the group increases, the mirrored RAID 1/0 capacity utilization is outstripped by parity RAID groups.

RAID group symmetry

In general, drives should only be grouped with the same type of drive.

The best performance and capacity utilization occurs when RAID groups are made-up of drives of the *same* form factor, speed and capacity.

The result of grouping unlike drives together in RAID groups is that at best the RAID group will have inconsistent performance, at worst the entire group has performance based the lowest performing drive's operating characteristics.

For example, a 5-drive RAID 5 group made-up of four 15K rpm SAS drives and one 10K rpm SAS drive. The response time with I/Os to the 10K rpm drive will be slower than to its peer drives. The overall throughput of the RAID group will be lower than a RAID group made-up of entirely of 15K rpm drives.

Drives of the same type and speed, but of different form factor may be grouped together. However, be aware that there may be performance differences under different I/O workloads.

Groups created from different capacity drives result in all the peer drives being truncated in capacity to that of the lowest capacity drive. This results in a lower usable capacity for the RAID group than the sum of all the drives usable capacity.

For example, a 5-drive RAID 5 group made-up of five 600 GB (raw) SAS drives would have a raw user capacity of 2.4 TB (4 * 600 GB). If the RAID group were made-up of four 600 GB drives and one 300 GB (raw) SAS drive, the higher capacity drives would be truncated to the capacity of the lowest. This RAID group would have a raw user capacity of 1.2 TB (4 * 300 GB). Note the usable capacity of the RAID group is halved by the inclusion of the single lower capacity drive.

It is *strongly* recommended that all the drives in a RAID group be the same form factor, type, speed, and capacity. This results in the highest and most consistent level of performance and highest capacity utilization.

RAID group creation

There are different ways to group drives into RAID groups to increase performance and availability. The main factor is which type of RAID protection (parity or mirror) will be chosen for the drives.

Binding tools

The process of grouping together drives into RAID groups and presenting them to the operating environment is called *binding*. Binding can be performed though the Unisphere Graphical User Interface (GUI) or the Unisphere Command Line Interface (CLI).

Manual binding gives a greater degree of control over the process than the UI-based approach. However, manual binding is more complicated and requires additional knowledge about the hardware platform and its provisioning. Following the binding recommendations in this section applies to traditional LUNs only. It requires you to know how to use the Navisphere CLI; specifically with the `creatrg` and `bind` commands.

Additional Information

To learn about CLI commands refer to the *Unisphere Command Line Interface (CLI) Reference* available on EMC Powerlink.

Binding across backend ports

Binding of drives across back-end ports is discussed in the SAS back-end port balancing section.

Binding mechanical hard drives

Generally, for mechanical hard drives there is no real performance advantage for manually distributing the member drives of the most commonly used RAID group sizes across all the backend-ports. However, binding a small RAID group's drives to all be on the same back-end port and RAID groups with a large number of drives on separate backend ports has a slight availability advantage. The advantage is dependent on the RAID configuration, and in all cases the differences are slight.

Binding flash drives

Flash drives may receive a performance advantage by manually distributing the member drives of a RAID group across all available backend-ports.

For high IOPS workloads *only*, distributing the peer drives of the RAID group as widely as possible is recommended. As with mechanical hard drives, the availability considerations for binding a flash-drive based RAID group's are the same. However, very large flash-based RAID groups are rare.

Note, a FAST Cache's flash drives are a candidate for this type of provisioning when practical. FAST cache uses RAID level 1. Manual re-positioning of the physical flash drives may be needed to get the pairs of drives on separate abuses.

Binding parity RAID groups

The most common VNX parity RAID groups are RAID levels 5 and 6. Binding small parity RAID groups such as five-drive RAID level 5 (4+1) groups so each drive is in a separate back-end port does not help performance. However, there is a small increase in data availability in this approach with large RAID groups. Parity RAID groups of 10 drives or more benefit from binding across two backend ports, as this helps reduce rebuild times and the effect of rebuilds on drives sharing the backend SAS ports of the rebuilding RAID group.

For example, when you bind a 10-drive (8+2) RAID level 6 group, bind five drives to one backend port, and bind the remaining five drives onto the other backend-port. Note that using the CLI or manual re-positioning of drives within DAEs may be needed to get this distribution.

Binding mirrored RAID groups

The most common VNX mirrored RAID group is RAID 1/0. There is no performance advantage in binding a mechanical hard drive based RAID 1/0 group onto more than one backend port, but it is not harmful in any way.

Binding with DPE or DAE-OS drives

Only bind the system drives with drives on the DPE or DAE-OS, if possible. It is not required, but try to bind the drives in the DPE or DAE-OS together and not with drives outside the enclosure.

In the rare event of a total power-fail scenario, the standby power supply (SPS) supplies battery-backed power to the SPs and the enclosure (DPE or DAE-OS) containing the system drives. This allows the storage system to save the contents of the write cache to non-volatile storage.

However, the power to the enclosures not housing the system drives is not maintained. RAID groups split across the standby powered DPE or DAE-OS and an unpowered DAE may have unsynchronized I/Os as a result of a failure. Writes to the powered DPE or DAE-OS will complete, while writes to unpowered DAEs will have failed.

When the storage system reboots, LUNs with I/Os outstanding are checked, using the Background Verify (BV) process. This checks to make sure that there were no writes in progress during the power fail that may have resulted in partial completions. A rebuild is required to correct a failed BV.

To avoid a rebuild on boot, only bind drives in the DPE or DAE-OS to other drives within that enclosure. In addition, do not include drives from these enclosures in Virtual Provisioning pools, if practical. If DPE or DAE-OS drives must be bound with drives outside their enclosure the following guidelines apply:

- Do not split RAID 1 groups, including FAST Cache drives, across the DPE or DAE-OS enclosure and another DAE.
- For RAID 5, make sure at least two drives are outside the system drive enclosure.
- For RAID 6, make sure at least three drives are outside the system drive enclosure
- For RAID 1/0, drive pairs either exclusive to the DPE/DAE-OS or outside the DPE/DAE-OS i.e. do not split any RAID 1/0 pair..

The LUNs

Hosts see Logical Units (LUNs) as physical disks. LUNs are frequently referred to as: disks, or *volumes*, or *partitions* depending on the context. LUNs hide the organization and composition of

pools and RAID groups from hosts. LUNs are created to allocate capacity, ensure performance, and for information security. Note that information security is *not* data protection, but rather a . privacy feature.

As with the underlying RAID group and its drives, when provisioning a LUN on a storage system you need to consider the workload's primary IO type, capacity requirements, and the LUN's utilization.

The VNX series supports more than one type of LUN. The following LUN types are available on the VNX series:

- Virtual Provisioning pool LUNs
  - Thin LUNs
  - Thick LUNs
- Traditional LUNs
- MetaLUNs

Virtual Provisioning pool LUNs are logical storage objects that exist within Virtual Provisioning pools. They are a logical storage object within a logical storage object. Pool-based LUNs map address spaces into the capacity provided by an automatically managed pool of drives. This presents an application with the virtual capacity of the pool.

Traditional LUNs are manually allocated capacity from a single RAID group.

A MetaLUN is a managed grouping of traditional LUNs for the purpose of extended LUN capacity and to extend performance whilst maintaining availability.

Basic LUNs

This section discusses basic LUN parameters that may affect the choice of LUN type used for hosting user data.

Maximum Number of LUNs

The number of LUNs that can be created on the storage system is model dependent. In addition, there are limitations on the number of LUNs that can be created within a Virtual Provisioning pool or on a RAID group. The number of LUNs eventually needed may determine which model VNX storage system is needed, and the LUN type to provision with.

		VNX5100	VNX5300	VNX5500	VNX5700	VNX7500
Storage System	Maximum LUNs	512	2048	4096	4096	8192
Virtual Provisioning Pools	Maximum LUNs per Pool	512	512	1024	2048	2048
	Maximum LUNs all Pools					
Traditional LUNs	Maximum LUNs per RAID Group	256				
MetaLUNs	Maximum MetaLUNs per Storage System	256	512	512	1024	2048
	Maximum LUNs per MetaLUN	1				

Table 20 Maximum Host LUNs per LUN Type VNX O/S Block 31.0

For example, from the table, you can see that you can create a single Virtual Provisioning pool on a VNX5300 with 512 LUNs, or two pools each with 256 LUNs. Likewise, on the same

VNX5300, is possible to create 512 LUNs using Traditional LUNs on only two RAID groups. However, that model VNX can never have greater than 2048 LUNs.

Private LUNs

Private LUNs support user-related LUN data objects such as metaLUN components, and the Reserve LUN Pool. They area created by features and installed applications.

Private LUNs subtract from the total number of storage system LUNs available. Be aware that private LUN creation reduces the number of available user LUNs that can be created.

Maximum LUN capacity

When provisioning LUNs based on capacity *only*, all of the available LUN types can be provisioned with modest to moderate capacity LUNs. Moderate capacity LUNs are <4 TB (raw) in capacity.

In some cases the host O/S's ability to address a LUN's capacity will be a limiting factor in provisioning large and very large capacity LUNs. In general, O/Ss with 64-bit architectures are required. Refer to the host O/S file system information for specifics in addressing large LUNs.

However, the different LUN types do have different maximum capacities. When very large capacity LUNs or the capability for frequent LUN expansion is needed, use Virtual Provisioned LUNs or MetaLUN.

Maximum Virtual Provisioning LUN capacity

The capacity of the largest Virtual Provisioning pool LUN thick and thin for OE – Block 31.0 is 16 TB.

Maximum traditional LUN capacity

The capacity of the largest traditional LUN, is that of the largest, highest capacity drive RAID group.

A RAID group can be made-up of as many as 16 drives. The data protection scheme, parity or mirror reduces the maximum capacity of the drives. See the RAID capacity utilization characteristics section.

For example, a 28 TB raw capacity LUN can be created using a 16-drive RAID-level 6 (14+2) dedicated group using 2TB NL-SAS drives.

Maximum MetaLUN LUN capacity

MetaLUNs provide the largest capacity of all LUN types. Through a combination of striping and concatenating FLARE LUNs, very large MetaLUNs can be constructed. See the MetaLUNs section for details.

It is easily possible to create LUNs of Petabyte capacities by using MetaLUNs.

For example, assuming the maximum number of drives in a RAID group (16), and the maximum number of component LUNs that can make-up a MetaLUN (512), the largest MetaLUN that can be provisioned drive-wise could contain 8192 drives (16 drives * 512 component LUNs) This value exceeds the capacity of the largest VNX storage system the VNX7500, which can accommodate 1000 drives.

Binding LUNs

LUNs are bound after Virtual Provisioning pools or RAID groups are created. LUNs are available for use immediately after they are created.

However, the bind is not strictly complete until after all the bound storage has been prepared and verified. The preparation and verification occurs through background processing. Its affects may adversely affect initial performance. Depending on the LUN size, verification priority, and the

storage system's workload, the two steps (preparation and verification) can vary considerably in duration.

LUN preparation and verification

LUNs use *Fast Bind*. The Fast Bind immediately makes a LUN's capacity available for use, before it is completely initialized in a bind or the unused remainder of the new or the destination LUN is initialized in a migration.

Newly bound LUNs and new Virtual Provisioning pools have their drives zeroed. This is called *background zeroing*.

Background zeroing erases any data previously written to the drive. It provides confidentiality, and pre-conditions the drives for background verification. The background zeroing only occurs on drives that have previously been used. New drives from the factory are 'pre-zeroed'.

The zeroing occurs in the background. However this still allows the LUN to be immediately available. The NL-SAS drives have a lower zeroing rate than flash drives, which have the highest rate. The zeroing rate varies between 20 MB/s and 50 MB/s, depending on the drive type.

The complete zeroing of large-capacity LUNs or disks in new Virtual Provisioning pools can take several hours. This process may adversely affect storage system performance, particularly when many LUNs or maximum-size storage pools are created at the same time. Creating large numbers of LUNs or large pools without pre-zeroing should be avoided while a production workload is in progress, if possible.

The zero-ing process can be accelerated in the following ways:

- Use new drives from EMC.
- If the drives have been previously used, manually pre-zero drives before binding or pool creation.

New drives from EMC are pre-zeroed. New drives are automatically detected as being pre-zeroed, and are not "re-zeroed."

Manually zeroing drives ahead of binding decreases the length of time it takes for LUNs or pools to be completely available. Pre-zeroing is executed by the individual drives internally at a rate of about 100 MB/s. Pre-zeroing needs to be performed on the drives *before* they are assigned to RAID groups or pools, and requires use of the Navisphere CLI. The following commands can be used to pre-zero drives:

```
1. naviseccli zerodisk -messner <disk-id> <disk-id> <disk-id> start
2. naviseccli zerodisk -messner <disk-id> <disk-id> <disk-id> status
```

The first command "start" begins the drive zeroing. The second command "status" can be used to monitor the zeroing progress.

Verification involves a Background Verify (*BV*). A BV is a reading of the LUN's parity sectors and verification of their contents. A BV is executed by default after a bind. This default can be manually overridden in Unisphere to make the bind faster. A BV is also scheduled when a storage processor detects a difference between a stripe's parity and the hard drive's sector parity. This is an availability feature of the VNX.

Note that a LUN or a Virtual Provisioning pool is not completely initialized until a Background Verify has been completed. The bind dialog provides a "no initial verify" option for LUNs; if you select this option the bind dialog does not execute the Background Verify. This option is *not* available under Virtual Provisioning.

General LUN performance

The performance of LUNs depends on the performance of their underlying RAID groups. That their groups are composed of flash drives or mechanical hard drives, and how many drives is important to all three types of LUNs: Virtual Provisioning, Traditional, and MetaLUN.

The more drives the better

Creating LUNs on the largest practical RAID group or with Virtual provisioned LUNs the largest number of RAID Groups, is the easiest way to ensure high small-block random IO performance. This applies to both flash drives and mechanical hard drives.

Increasing the capacity of the LUN on its current RAID group(s) does not improve its performance.

Performance Domains

Create LUNs with application performance requirements in mind. *Performance domains* are keeping LUNs with complementary performance requirements together on the same underlying RAID groups. It also means keeping LUNs with conflicting performance requirements separated from each other.

This applies to both the logical and physical storage objects.

LUN provisioning by I/O type

When practical, separate LUNs performing different types of I/O onto different RAID groups. That is, separate LUNs doing mostly random I/O from LUNs doing mostly sequential I/O.

Virtual Provisioning pool LUNs

When handling both I/O type LUNs in a Virtual Provisioning pool, create a pool with as many drives as is practical. This spreads the I/O as widely as possible, making all I/O appear to be random I/O within the pool.

For small Virtual Provisioning pools needing a higher-level of performance, segregating pools by I/O type offers some advantage, particularly for the sequential I/O performance. This is due to the limited number of RAID groups over which the load can be spread.

Traditional LUNs

In a workload environment characterized by random I/O, it is prudent to distribute a workload's LUNs across as many RAID groups as is practical given the available drives and configured RAID groups.

In a workload characterized by sequential I/O, it is advantageous to distribute the workload's LUNs across as few RAID groups as possible to keep those RAID groups performing the same I/O type.

When more than one I/O type is handled by the storage system, the LUNs supporting the different I/O types should be kept as separate as possible. That is, if possible, do not put LUNs supporting workloads with mostly random I/O on the same RAID group(s) as LUNs supporting workloads with mostly sequential I/Os.

If high utilization LUNs from the same workload must be placed in the same RAID group together, place them next to each other on the RAID group. That is, without any intervening LUNs between them on the RAID group. This will minimize the drive seek distance (time) and get the highest performance between these highly utilized LUNs. The order of the LUNs created will determine the placement within the RAID group.

LUN provisioning by percentage utilization

Ideally, all the active RAID groups in the storage system should have about the same percentage of utilization. This would be the most efficient use of the storage system's resources. However, this rarely occurs.

At any time, some LUNs may be *hot LUN*s, and other LUNs are essentially idle. A hot LUN is a LUN serving a workload causing its underlying RAID group to have drive utilization significantly higher than the average for the RAID groups on the storage system.

A levelling of RAID group drive utilization across the storage system should be sought to get the best performance from the storage system. Levelling is performed by moving LUNs between

RAID groups or pools. Specifically, LUN migrations are used to move the LUNs. See the LUN Migration section.

Traditional LUNs

When more than one LUN shares a RAID group, try to achieve an average utilization by matching high-utilization with low-utilization LUNs or average-utilization with other average-utilization LUNs on RAID groups to achieve an overall average RAID group utilization for the LUNs on the storage system.

When the workload is primarily random, the averaging will be across as many RAID groups as is practical to meet capacity, availability, and performance requirements. When the workload is sequential, the averaging will be across as few as is practical to meet capacity, availability, and performance requirements.

LUN provisioning by temporal allocation

Another way to distribute LUNs is by when they are used (*temporal allocation*). It is likely that not all LUNs are constantly active; position LUNs together onto RAID groups or Virtual Provisioning pools that are active at different mutually exclusive times.

For example, LUNs supporting a business day workload from 8 A.M. to 8 P.M. will have their highest utilization during this period. They may have either low utilization or be idle for much of the time outside of this time period. To achieve an overall average utilization, put LUNs that are active at different times over a 24-hour period in the same RAID group or pool.

Unisphere Analyzer provides information on drive utilization to determine how to re-distribute the LUNs across the storage system. Information on how to use Unisphere Analyzer can be found in the Unisphere on-line HELP..

LUN Ownership

LUNs are managed and accessed by a single storage processor. This is called *LUN ownership*. By default LUN ownership is automatically assigned within Unisphere to storage processors in a round-robin fashion when a LUN is bound.

Ownership can be manually changed through Unisphere or the CLI.

It may be necessary to change a LUN's ownership to its peer storage processor for performance reasons. For example, this change may be needed to balance storage processor usage within the storage system.

Note that trespassing a Virtual Provisioning pool-based LUN will adversely affect its performance after the trespass. See Basic LUN Availability section.

LUN queues

Access to a LUN is mediated through a queuing mechanism. Each arriving I/O for a LUN takes a queue position. I/Os are taken off the queue and processed by the storage processor. The number of queue entries for a LUN at the SP front end port is called the LUN queue depth.

The maximum number of queue entries to a LUN depends on the number of user-data drives in the LUN. The larger the number of data drives in the RAID group the deeper the queue.

Virtual Provisioning pools and traditional LUNs have the same queue lengths. Although, with pools, the entries apply to the pool's private RAID groups and are fixed.

For example, a pool of RAID level 5 with default 5-drive (4+1) private RAID group based LUNs requires 88 concurrent requests for its queue to be full. The RAID 6 pool default private RAID group LUNs would be 116 and the RAID 1/0 pool 60.

Each pool LUN comes with its own set of queues. The more LUNs there are in the pool, the more queues there will be (both at the client, and at the SP). The greater number of queues allow for sending more concurrent I/O. Certain applications, such as Oracle databases are highly concurrent. Their performance improves under these circumstances.

If the queue depth is exceeded, the storage system returns a *queue full* (QFULL) status to the host in response to an I/O. The exact effect of a QFULL on a host is O/S dependent, but it has an adverse effect on performance.

QFULLs are rare. Host Bus Adapter (HBA) queue depth settings usually eliminate the chance of them being generated. Note that QFULL can also be triggered at the I/O port-level.

LUN Identification (ID)

A LUN is identified by either its LUN Name or its LUN ID.

The *LUN Name* is an identifier. A LUN can be assigned an identifying text string created by a user. Through Unisphere, the user can create a free form text field of up to 64-characters for host LUN identification. Unisphere only uses the LUN ID numbers (see below) that are assigned by the storage system; it does not use the LUN name which is changeable. There is no check for duplicate LUN names and no restriction on duplicate names.

LUNs are internally identified on the VNX by their *LUN ID*. The range of LUN IDs and the maximum number of LUNs are model dependent.

There are two types of LUN IDs: user and private. The available LUN ID numbers (both user and private) always exceed the maximum total of LUNs that can be created on a VNX. However, the sum of user and private LUN IDs in use cannot exceed the VNX model's maximum total of LUNs. (There will always be extra LUN IDs.)

User LUNs created through Unisphere are automatically assigned available user LUN IDs starting from 0 and incrementing by 1 for each LUN created. Users can also manually select an available unused user LUN ID number from the model's range of LUN IDs at the time of creation through Unisphere or the CLI.

Dual drive ownership

All drives on a VNX storage system are dual ported and can accept I/O from both storage processors at the same time. Dual ownership is having LUNs owned by both storage processors on a single RAID group. LUNs owned by both storage processors on a single RAID group can make balancing their utilization difficult.

Each storage processor operates somewhat independently when issuing requests to any drive in a RAID group. Dual ownership may subject the drives of a RAID group to deeper queue usage than RAID groups with LUNs from a single storage processor.

Deeper queue utilization is beneficial in increasing drive utilization. This is particularly an advantage for RAID groups provisioned with flash drives, which become very efficient when operating this way. However, with mechanical hard drives, this may also result in higher host response times than with single drive ownership. For the most deterministic performance with mechanical drives, single drive ownership is recommended.

Single ownership of a RAID group's drives by a storage processor is recommended for deterministic performance of traditional LUNs on mechanical hard drives. However, if maximum throughput is required through data distribution, RAID group drives can be configured with dual ownership.

Dual ownership is the default with Virtual Provisioning pools. For example, dual ownership is needed to get an even spread of the I/O load across a large pool for its efficient operation.

LUN migration

LUN migration is an important feature for use in storage system tuning. The LUN migration facility is used to change a:

- LUN's type
- LUN's RAID group topology
- Location within the storage system

- LUN's capacity

All the operations above are used to 'balance' or optimize the performance of LUNs within the storage system.

In addition, migrations can be used to decrease the amount of capacity used within a pool. LUN migrations sourced from traditional or Virtual Provisioning thick LUNs to a Virtual Provisioning pool thin LUN will allow for LBAs with zeros to be detected and removed. This feature is called *Space Reclamation*. LBAs with zeroes are empty or unused. This will likely result in less capacity being consumed from the pool than is allocated to the host.

Smaller capacity LUN to larger migrations

A LUN migration going from a smaller source LUN to a larger destination LUN is done internally in two steps. First, the existing capacity is migrated to the new destination. Then, the system initializes the new capacity by using its standard binding algorithm. The standard binding zeroes allocated space before usage. Note that LUNs sharing the RAID group or pool of the destination LUN are adversely affected by both the migration and the initialization.

Same capacity LUN migrations

When migrating a LUN to a destination on a new RAID group that may have different physical characteristics, you can guarantee the exact capacity of the source on the destination by specifying 'blocks' instead of GBs or MBs. The number of blocks in a LUN can be found in the original LUN's properties from the original LUN properties. This will ensure that the destination LUN is exactly the same capacity as the source LUN.

Larger capacity LUN to smaller migrations

See the LUN Shrink section. In addition, note that 'zero detection' occurs when the destination is a thin LUN.

LUN migrations with Virtual Provisioning pool thin LUNs

There will be a lowering in the rate of migration when the source or destination LUN is a Virtual Provisioning thin LUN.

It is difficult to determine the transfer rate when the source LUN is a thin LUN. It will be lower. The decrease in the rate depends on how sparsely the thin LUN is populated with user data, and how sequential in nature of the stored data is. A densely populated LUN with highly sequential data increases the transfer rate. Random data and sparsely populated LUNs decrease it.

The amount at which the rate will decrease when the destination LUN is a thin LUN is difficult to determine. When a thin LUN is the destination, the Space Reclamation can affect the rate.

Space Reclamation scans each migration buffer for zeros and if a group of zeros is found that is 8K or larger it is removed. This decreases the in-use capacity of the destination LUN. It also, splits the transfer into multiple I/Os which decrease the rate.

For example, assume a migration buffer contains 256KB of LUN data at the beginning followed by 576KB of 'Zeros' followed by a further 256KB of LUN data. That will causes three separate operations a 256K write, a 576K zero fill and another 256K write. If no zeros had been detected, the transfer would have been handled as a single write.

If there are no large sequential groups of zeroes on the source, assuming the source is a Virtual Provisioning thick or traditional LUN, then the performance is equivalent to a migration between Virtual Provisioning pool thick LUNs.

Migrations into FAST VP pool-based LUNs

When migrating into or between FAST VP pool-based LUNs, the initial allocation of the LUN and the allocation policy have an important effect on its performance and then capacity utilization.

Tiering policy setting (Highest, Auto, Lowest) will determine which tier within the pool the data of the source LUN will be first allocated to. Be sure to set the correct policy needed to ensure the expected starting performance for *all* the source LUN's data. As much capacity from the source LUN will be allocated as possible to the appropriate tier.

Once the migration is complete you can adjust the tiering policy to Highest, Auto, Lowest, or No movement for the appropriate promotion and demotion of the original source LUN's data between tiers to get the correct capacity allocation based on frequency of access. See the Virtual Provisioning pool-based LUNs section.

Low, Medium, High priority LUN migrations

LUN Migrations are a prioritized operation. Low, Medium, and High priorities have increasing effect on the performance of production workloads. However, they are below the resource utilization level where they would affect a production workload on an un-degraded or over utilized storage system. These economical priorities implement the transfer as very large block, timed, sequential transfers.

The following table shows the rule-of-thumb migration rate for SAS RAID-groups:

Priority	Rate (MB/s)
Low	1.4
Medium	13
High	44

Table 21 Low, Medium, and High migration rates, OE Block 31.0

The economical transfer rates are throttled by design to allow production workloads to continue executing without adverse performance effects during the migration process. The default rate is High.

The rates shown in this table combines both the Migration and Initialization processes that would occur for a smaller capacity to larger capacity LUN.

ASAP priority LUN migrations

ASAP LUN migrations with normal cache settings should be used with caution. They may have an adverse effect on system performance. EMC recommends that you execute at the High priority, unless migration time is critical.

The ASAP setting executes the migration I/O with a minimum of delay between I/Os. Working on the SP itself, the inter-I/O latency is very low. The result is akin to a high-performance host running a heavy workload against the source and destination hard drives. The workload has the characteristics of a large block sequential copy from the source LUN to the destination LUN.

The following table shows the ASAP "rule-of-thumb" migration rate for 10k rpm SAS and 15k rpm SAS drives with the Virtual Provisioning pool thick LUNs on the default RAID groups and the default write cache OFF settings.

LUN based on RAID group type	Migration Rate (MB/s)
RAID 5 (4+1)	85
RAID 6 (6+2)	
RAID 1/0 (4+4)	

Table 22 ASAP migration rate default rates for OE Block 31.0

When migrating between Virtual Provisioning pool-based LUNs, particularly when the source or destination is a thin LUN, decrease the ASAP migration rate.

The default ASAP migration settings avoid cache congestion. Up to two ASAP migrations at the same time, per storage processor are allowed. Avoid using ASAP for LUN migrations on busy systems.

Cached Migration

By default a LUN migration bypasses write cache. You can achieve an expedited migration by increasing the value for the destination LUN write-aside parameter with an ASAP priority. This enables write cache. If you change the write-aside value for the destination LUN to 3072 or greater, you can make the migration up to four-times faster.

LUN migration with write cache enabled is *not* recommended for production environments with an active workload. It may cause a forced cache flushing. Forced cache flushing has an adverse effect on overall storage system performance. However, cache enabled migrations may be helpful in reducing the time required for some maintenance operations.

Estimating a LUN migration's duration

Priority has the largest effect on the duration of a LUN migration. However, the configuration of the underlying RAID groups that support the source and destination LUNs also affects migration rate. The RAID types, drive types, number of drives, and speeds of hard disks effect on the migration rate. The largest and simplest factor affecting the ASAP migration rate is whether the underlying RAID groups are mirror or parity RAID groups, and the number of drives in the groups.

When estimating the duration of a smaller capacity to a larger capacity LUN, assume the migration rate and the initialization rate are identical.

Use the following formula to estimate the time required to complete a LUN migration.

- Time: Duration of LUN migration
- Source LUN Capacity: Size of source LUN in GB
- Migration Rate: Rate of copy from source LUN to destination LUN from Table 14 or Table 22 depending on the selected migration priority
- Destination LUN Capacity: Size of destination LUN in GB
- Initialization Rate: Speed at which new additional storage is initialized in MB/s (Table 22 for ASAP or else omit)

*Time = (Source LUN Capacity * (1/Migration Rate)) + ((Destination LUN Capacity – Source LUN Capacity) * (1/Initialization Rate))*

Example calculation

How many hours will it take to migrate a 400 GB LUN to an 800 GB LUN, at the ASAP priority?

Extracting the needed rates from the specified tables and converting the units results in the following table.

Source LUN Capacity (MB)	Destination LUN Capacity (MB)	Migration Rate (MB/hour)	Initialization Rate (MB/hour)
409600	819200	306000	306000

Table 23, Example LUN Migration Calculation Parameters

Applying the calculation using the parameters above results in:

2.68 hrs = (409600 MB * (1 hour / 306000 MB/Hr)) + ((819200 MB – 409600 MB) * (1 hour / 306000 MB/Hr))

LUN shrink

Traditional LUNs, metaLUNs, and pool-based LUNs may have their capacity reduced to reclaim unused storage capacity. This process is called *LUN shrink*. LUN shrink is a standard feature on VNX OE Block 31.0 and later.

LUN shrink is only supported by hosts with the Microsoft Server 2008 operating system and later.

LUN shrinking is a two-step process consisting of a host-volume shrink followed by a LUN-volume shrink.

Host volume shrink

Host-volume shrink is performed from the MS Server host though its Disk Administration function. LUN volume shrink is directed by the user on the host, and executed on the CLARiiON. Both steps may be performed while a workload is present. The LUN volume shrink step requires the "`DISKRAID.EXE`" application be installed on the host. It is performed automatically, after the first step. A LUN shrink requires several seconds (less than a minute) to perform.

A host file system defragmentation should be performed before a traditional or pool-based LUN shrink to consolidate the LUN's capacity. This should take place before the host-volume shrink. A file system defragmentation yields the largest amount of capacity from the traditional or thick LUN that can be shrunk.

Pool-based LUN considerations

A pool-LUN file system defragmentation may be needed for the host-side, when a substantial amount of the capacity is being shrunk. Generally, it is not recommended to do file system defragmentation of Virtual Provisioning pool-based LUNs. However, Pool-LUN capacity that was previously consumed, and host reported usage of LBAs need to align for the shrink to complete. Note defragging the pool-based LUN may have an adverse effect on cache and backend performance. Try to omit defragging. Perform the host-volume shrink without a prior defragmentation first, to see if acceptable results are yielded.

Fixed Files

Be aware that some system and application files may have fixed themselves to locations on the LUN when they are installed. They are not re-locatable by ordinary methods, like defragmentation. These files will prevent a shrink past them in the LUN map, despite there being unused capacity in front and behind them. It may be required to delete these files or disable Windows features that fix these files to the LUN.

RAID group defragmentation

Shrinking a FLARE LUN may leave the LUN's underlying RAID group in a fragmented state. You may need to perform a RAID group defragmentation to achieve the maximum RAID group capacity that is provided by a LUN shrink.

OE Block 31.0 does not support RAID 6 group defragmentation. This means that you may not be able to utilize the aggregated capacity on a RAID 6 group after shrinking RAID 6 LUNs.

Virtual Provisioning: Pools

The Virtual Provisioning feature provides storage utilization efficiency and *ease-of-use*. Pools require less manual administration and specialized knowledge than storage system's provisioned with traditional LUNs. Ease-of-use trades management efficiency for flexibility of provisioning. Best practices are implemented in the pool algorithm and this will apply to most of the VNX users.

Pools overview

A pool is a logical storage object. It is made-up of drives organized into private RAID groups. (See RAID groups and drives section.) LUNs are created within Virtual Provisioning pools.

A storage pool is overlaid onto private RAID groupings of drives. The pool automates storage allocation and data placement within the pool. In addition, provisioning of the pool is simplified through the graphical user interface, which algorithmically applies Best Practices when given a set of drives to either create or expand a pool's capacity.

With OE Block 31.0, a single RAID level of data protection applies to all the pool's private RAID groups. This rule applies to homogenous and heterogeneous pools too. The RAID type of a pool can be RAID types 5, 6, or 1/0. Use the general recommendations for RAID group provisioning for traditional LUNs when selecting the provisioning of the storage pool's RAID types.

Pools may be homogenous with a single level of performance, or heterogeneous. Heterogeneous pools are also called tiered pools, which use the FAST VP feature. A heterogeneous pool may also be created without FAST VP being enabled. However, automated data tiering would not be performed within the pool. This is *not* recommended unless the intent is to add the FAST VP enabler shortly after creating the pool.

In addition, Virtual Provisioning pools support the LUN Compression feature. Compression performs an algorithmic data compression of pool-based LUNs. LUN Compression is a basic feature included with all VNX Systems starting from the model VNX5300 and higher.

Additional information

An in-depth discussion of Virtual Provisioning can be found in the *EMC VNX Virtual Provisioning: Applied Technology* white paper, available on Powerlink.

High-level Virtual Provisioning pools recommendations

When creating Virtual Provisioning pools, if the goal is:

- Deterministic pool-based performance, create a homogeneous storage pool with the largest practical number of drives.
- Best pool performance for the most frequently used data – create a FAST VP pool with the appropriate capacity in tiers using the highest performance drives for the most frequently used data and capacity drives for the less frequently used data..

Homogeneous Pools

A homogeneous pool is provisioned with a single drive type: flash, SAS, or NL-SAS.

Homogeneous pools are the most straightforward Virtual Provisioning pools to configure. It is easier to quantify and predict performance when only a single drive type is present. The performance of a homogeneous pool is the most deterministic of the pool types.

Heterogeneous Pools with Fully Automated Storage Tiering for Virtual Pools (FAST VP)

FAST VP allows for data to be automatically tiered in pools made-up of more than one drive type.

Tiering allows for an economical provisioning of storage devices within a tier instead of an entire pool. The separate tiers are each provisioned with a different type of drive. Tiered storage creates separate domains within the pool based on performance. The feature's software algorithmically promotes and demotes user data between the tiers based on how frequently it is accessed. More frequently accessed data is moved to higher performance tiers. Infrequently accessed data is moved to modestly performing high-capacity tiers as needed to make room for frequently accessed data on the higher performance drives. Over time, the most highly accessed data resides on the fastest storage devices, and infrequently accessed data resides on economical and modestly performing bulk storage.

FAST VP is a separately licensable feature, available on all VNX Systems starting from the model VNX5300 and higher. With FAST VP, any or all pools can be tiered by virtue of having heterogeneous drives.

Selecting Pool-based storage

The following considerations which also apply to traditional LUN performance, should always be considered when implementing Virtual Provisioning pool-based storage

- *Drive Contention*: More than one LUN will be sharing the capacity of the drives making up a pool. When provisioning a pool, there is no manual control over data placement within a pool.
- *Host Contention*: More than one host will likely be engaging each storage processor. Both storage processors have equal and independent access to a pool. Unless separate pools are created there is no control over host access within the shared pool.
- *Application File system layout:* More than one application's file system can be hosted in a pool. The capacity and performance of a pool needs to be planned accommodate the one or more file systems it is storing.

Pools are designed for ease-of-use. The pool dialog algorithmically implements many Best Practices.

Number of Storage Pools

The number of storage pools per storage system is model-dependent. The table below shows the maximum number of pools per model.

VNX model	Maximum storage pools per storage system (Pools)
VNX5100	10
VNX5300	20
VNX5500	40
VNX5700	
VNX7500	60

Table 24 Virtual provisioning storage pools per storage system and LUNs per pool, OE Block 31.0

The number of pools created per storage system has no effect on performance. It is the number of LUNs per pool and their individual capacities that affect pool performance. That is, the creation and management of one pool has the same effect on performance as two, three, four, or the maximum number of pools. See Table 20 for the maximum LUNs per pool.

Drives per storage pool

A pool can be as small as a single RAID group or as large as all the available drives on the storage system (less the system drives).

Minimum and maximum number of drives

The minimum number of drives that can be used to create a pool is the smallest number of drives needed to create a pool's selected RAID-level: 5, 6, or 1/0. See Table 17 Minimum drives for common RAID levels.

The maximum number of drives in a storage pool is storage system model-dependent. It is the maximum number of drives for the model, less the four drives of the system drives. (Only entire drives may be provisioned in pools.) See FAST Cache

It is required that flash drives be provisioned as hot spares for FAST Cache drives. Hot sparing for FAST Cache works in a similar fashion to hot sparing for traditional LUNs made up of flash drives. See Hot spares section. However, the FAST Cache feature's RAID 1 provisioning affects the result.

If a FAST Cache Flash drive indicates potential failure, proactive hot sparing attempts to initiate a repair with a copy to an available flash drive hot spare before the actual failure. An outright failure results in a repair with a RAID group rebuild.

If a flash drive hot spare is not available, then FAST Cache goes into degraded mode with the failed drive. In degraded mode, the cache page cleaning algorithm increases the rate of cleaning and the FAST Cache is read-only.

A double failure within a FAST Cache RAID group may cause data loss. Note that double failures are extremely rare. Data loss will only occur if there are any dirty cache pages in the FAST cache at the moment both drives of the mirrored pair in the RAID group fail. It is possible that flash drives data can be recovered through a service diagnostics procedure.

System drives section.

For example the maximum sized pool on the VNX7500 would be 996 drives.

Practical minimum and maximum number of drives

It is helpful *not* to think of pools in 'number' of drives, but instead of the number of private RAID groups making-up the pool. It is the private RAID groups which define a pool's performance, capacity, and availability. See the Drive counts section below.

With Virtual Provisioning pools, the larger the number of RAID groups in the pool, the better the performance and capacity utilization. In most cases, it is not practical to create pools with a small number of RAID groups due to the relatively fixed overhead of storage capacity, memory and CPU utilization. In larger pools, the overhead is very small.

As a rule of thumb, pools should contain at least *four* private RAID groups. For example, assume a RAID-level 5 homogenous pool is made up of the default 5-drive (4+1) private RAID groups. The smallest practical pool would be made up of 20 drives (four private RAID groups, each RAID 5 4+1).

On the other hand, there are likewise practical considerations about the upper limit of pools. Larger pools may be created on the larger model VNXs. However, large pools require special consideration, particularly in maintaining consistent performance through pool expansions and with maintaining high availability. See Pool Availability sections. In some cases, it may be more practical to create several pools with fewer private RAID groups than to have only one or two very large pools.

Multiple-pool strategies

It is a recommended practice to segregate the storage system's pool-based LUNs into two or more pools when availability or performance may benefit from separation.

Greater availability may be achieved by keeping the number of private RAID groups that make up a pool from getting too large, Higher performance may be achieved by separating busy LUNs that may be causing contention within the pool. Additionally, FAST Cache may be more compatible with some pool LUNs and not others. Separate pools can be created for the purpose of segregating those LUNs. FAST Cache would then be enabled on the pool with the LUNs better suited for its use.

There are several strategies available for creating multiple pools. It is up to the user to determine how many pools meet their storage goals and business priorities.

Users can choose between the simplest approach of creating a single multi-purpose pool for all the storage system's pool-based LUNs, or on the other hand a more complex approach of creating distinct Virtual Provisioning pools according to performance or application requirements.

For example, the following describes possible ways to configure multiple pools. Most of these configurations require a more in-depth knowledge of the applications residing on the LUNs. Pools can be created to contain LUNs:

- Serving the Celerra NAS function
- Grouped together based on primary I/O type (random or sequential)
- Whose performance is augmented by the FAST Cache or FAST VP feature
- Servicing a single application, such as a database or mail server
- Optimized for bulk storage with the highest capacity utilization
- Optimized for the highest availability

Creating Virtual Provisioning pools

Homogenous pools are the most straight-forward and easiest to implement pool type. The same restrictions and recommendations for drive types within a RAID group apply to homogenous pools.

Ideally, all drives in a homogeneous pool have the same type, format, speed, and capacity drives. When all the pool's underlying RAID groups are the same, the most consistent performance and highest capacity utilization is achieved.

Different type, speed, form factor and capacity drives of the same type *may* be mixed in a homogenous pool. However, mixing may have an adverse effect on performance and capacity

utilization. Mixing drives with different performance characteristics within a homogenous pool is *not* recommended. See the RAID group symmetry section and RAID capacity utilization characteristics sections for more details.

Creating FAST VP pools

Tiered pools, or pools using the FAST Virtual Pools (FAST VP) feature, are made up of either two to three tiers. Each tier is based on a different drive type. Tiers are created to a first order to meet performance requirements, and secondly to meet capacity requirements.

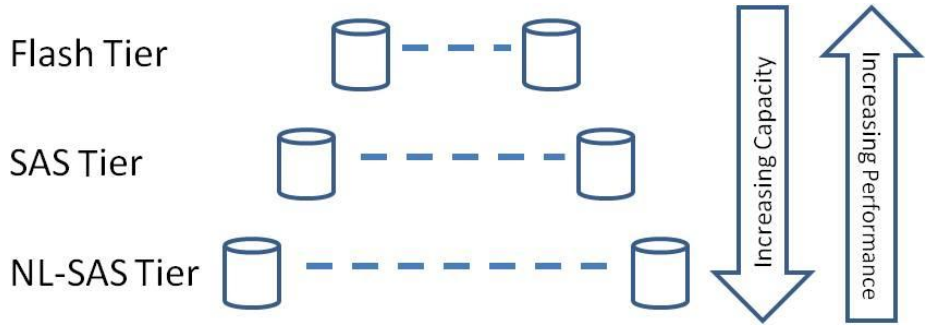

Figure 6, FAST VP Pool Conceptual Diagram

The three possible tier types are:

- Extreme Performance (Flash)
- Performance (SAS)
- Capacity (NL-SAS)

Each tier has the performance and capacity characteristics of its drive type. Drives of the same type, with different capacities, formats, and speeds, may be provisioned in the same tiers. However, this is *not* recommended.

Tier configurations

FAST VP pools can be made-up of either two or three tiers. Two tiered FAST VP pools can be made-up of any combination of the three drive types.

Three tiers: Flash, SAS, and NL-SAS drives.

Three tiers require all the supported drive types to be present in the pool. This tiering structure provides from the highest to moderate performance depending on the workload.

It is recommended that you consider using flash drives as a FAST Cache with a two-tiered mechanical drive provisioned pool before provisioning flash drives to create a third tier. See the FAST Cache section for details.

Two tiers: flash and SAS drives.

This configuration gives the highest performance when the tiers are provisioned with the VNX's highest-performing flash and 15K rpm SAS drives. Performance is reduced proportionately when using the 10K rpm version SAS drives.

It is recommend that you consider using flash drives as a FAST Cache with a homogeneous mechanical drive provisioned pool before provisioning flash drives in a two-tiered pool. The Flash drives can then be shared across the storage system as a global resource. See the FAST Cache section.

Two tiers: flash and NL-SAS drives.

This configuration gives from the highest to modest performance range. Performance may vary significantly depending on the capacity of the flash tier, its state, and the locality of the workload's I/O.

When configuring this way, the flash tier *must* have a large enough in capacity to contain the workload(s)' most actively accessed data. Configuring the Flash tier to be slightly larger than the working set's capacity is preferable. In this configuration, placement of new data or access to data that has found its way onto NL-SAS drives due to lower activity must be tolerated by the application until that data can be migrated to the higher tier.

It is recommend that you consider using a three tiered pool, or in some cases a homogeneous pool with a FAST Cache with before using flash drives in a two-tiered pool with NL-SAS drives. This particular configuration requires careful study and an in-depth knowledge of the workload's locality to setup correctly. The very large difference in the performance between the two tiers may result in a large difference in host response times, if the flash tier is under-provisioned, locality is wider than anticipated, or the pool is un-warmed. See the FAST Cache section.

Contact your EMC Sales representative to engage an EMC UPSPEED professional for assistance with setting-up this configuration.

Two tiers: SAS and NL-SAS drives.

This configuration gives from high to modest performance. It is the most economical capacity-wise. Performance is moderate when 10K rpm SAS drives are used in the SAS tier, and high with 15K rpm SAS drives. Expect modest performance for large amounts of infrequently used data stored on the large-capacity NL-SAS drives.

Two tiers with SAS and NL-SAS is a particularly well performing combination of FAST VP provisioning. Consider a 20 percent SAS with an 80 percent NL-SAS drive *capacity* allocation when unsure about the locality. Use the Highest pool allocation policy with this configuration.

Additional performance can be achieved when FAST Cache is enabled for this pool. As a Rule-of-Thumb, plan on approximately 5 percent of the pool's total capacity being replicated by FAST Cache capacity.

Expanding a homogeneous pool to a heterogeneous pool for Tiering

A Virtual Provisioning pool which was previously created as a homogenous pool, can later be expanded by adding another drive to support tiering. For example, a pool is initially created with NL-SAS drives. At a later date, after FAST VP has been installed on the system, the pool is expanded with some SAS drives. The SAS drives immediately become the pool's highest tier, and the most frequently accessed data in the pool would be moved to this newly added highest tier. Users can use the automated relocation scheduler, or manually invoke a relocation, for the data movement to occur.

Tiers and capacity

The capacity of the tier is the sum of the user data drives in the tier's private RAID groups. The capacity of a tier is typically the inverse of its performance. That is, a flash-drive-based tier has the fewest private RAID groups made-up of the smallest capacity drives. A NL-SAS tier has highest-capacity drives, although with modest performance.

For example, a three-tiered FAST VP pool may have a *capacity* distribution of as shown in the diagram below. This is one example and not representative of all loads.

Tiers and performance

The performance of a FAST VP tier is dependent on the workload's I/O characteristics and the tier's provisioning. Performance can range from very high to quite modest.

For example, the highest FAST VP performance is with a highly concurrent, small-block random read workload to a Flash drive-based tier. More modest performance results from a small-block random workload to an NL-SAS-based tier. This range of performance is very possible within a three tiered pool.

The IOPS and bandwidth of a tier can be estimated by calculating the value for the private RAID group's that make-up the tier. However, there will always be a degree of variability in host response time with a tiered pool independent of the performance of the tiers. The tiers are allocated capacity and performance based on locality. A certain portion of the data being

accessed will not be on the highest performing tier. This must be taken into account when developing SLAs based on the calculations.

Combined, the Extreme Performance and Performance tiers should be sized to handle all of the pool's performance needs. Lower-level tiers may have a reduced performance capability.

Always be aware that the host response time for data on the lowest-level tier will be higher than the data on higher-level tiers. The observed host response time is a weighted average of the tiers, where the weighting is dependent on the: locality of the workload's data, the promotion rate of user data, 'warmed state of the pool, and the provisioned capacity of the tiers.

Note that in some cases, it may be prudent to overprovision a tier capacity-wise to ensure either a larger (drive-wise) or a larger number of private RAID groups with subsequent higher IOPS within the tier. See Drive counts section below.

Pool Provisioning

Pools are created and expanded by selecting drives from within the provisioning dialog. It is important to remember that these drives are being used to create the basic RAID group storage object within the pool.

The user interface algorithmically implements many Best Practices. However, it can only work with the available resources, typically drives. The best pool performance and capacity utilization occurs when the pool's private RAID groups closely align with RAID group best practices. See the RAID groups section.

Ideal number of drives

The pool provisioning algorithm tries to create the largest number of default sized private RAID groups with the drives given when doing either a pool creation or expansion. The following recommendations have an important affect on pool capacity utilization and performance.

Any number of drives may be used to create or expand a pool, taking into account minimum restrictions. See Table 17 Minimum drives for common RAID levels.

When initially provisioning the pool, use the largest number of drives as is practical within the storage system's maximum limit. Remember: pool performance scales directly with the number of drives.

Pool Expansion

With OE Block 31.0, when a Virtual Provisioning pool is expanded, the existing pool data is *not* re-striped across the newly added drives.

The Virtual Provisioning feature initially gives preference for usage of newly added storage versus the original pool's storage. New pool storage is allocated from the expansion drives till the point where all the drives in the pool are equally utilized percentage of capacity-wise.

Expansions that double the size of the pool are ideal. They ensure that subsequent storage usage has the same performance as the initial pool configuration, assuming the same type and speed drives are used when expanding.

It should be understood, however, that this approach may result in the most recently written LUN data having lower performance than the original pool storage, because the new data is being written over a smaller number of drives than the initial pool configuration.

Drive counts

The practical number of drives to use in creating or expanding a pool depends on its RAID level protection.

If the pool's RAID level is:

- RAID 5: initial drive allocation and expansion should be *at least* five drives or a number of drives evenly divisible by five (5).

- RAID 6: initial allocations should be at least eight drives or a number of drives evenly divisible by eight (8).
- RAID 1/0: initial allocations should be at least eight drives or a number of drives evenly divisible by eight (8).

The goal is to create the largest number of private RAID groups with the same group configuration. See the RAID group symmetry and RAID capacity utilization characteristics sections. The user dialog:

1. Creates as many default sized private RAID Groups as possible.
2. If there are additional drives and it can create a non-default private RAID group out of those drives, it will,
3. If it cannot create a private RAID Group with the remaining drives, it will spread them over the RAID Groups that have already been created.

The following examples show practical provisioning. If you specify:

- 20-drives for a RAID 5 pool expansion of initial provisioning; Virtual Provisioning creates four 5-drive (4+1) RAID groups. This is optimal provisioning example.
- 18-drives for a RAID 5 pool; Virtual Provisioning creates three 5-drive (4+1) RAID groups and one 3-drive (2+1) RAID group. This provisioning is not optimal. The 2+1 has less native performance than its peers. It also has lower capacity utilization.
- 10-drives for a RAID 6 pool; Virtual Provisioning creates one 10-drive (8+2) RAID group. This is larger than the default pool RAID level 6 size of an eight drive (6+2). It occurred, because an additional RAID 6 group of any number of drives could not be created, given the number of drives. This size is acceptable and has some advantages, but subsequent private RAID groups ideally should have the same number of drives to maintain symmetry in performance.
  - The 8+2 has an I/O advantage with large-block sequential I/O than the default RAID 6 group.
  - The 8+2 has higher capacity utilization than the default RAID 6 group.
- 10-drives for a RAID 1/0 pool; Virtual Provisioning creates one 8-drive (4+4) and one 2-drive (1+1) RAID group. This is *not* optimal. Avoid private RAID groups of two drives being created. For RAID 1/0 pools, if the number of drives you specify in pool creation or expansion isn't divisible by eight, and if the remainder is 2, the recommendation is to add additional drives or remove two drives from the allocation.

Adding drives to a pool

The maximum number of drives that you can create or expand a pool with is model-dependent. The increments are shown in the table below.

VNX Model	Maximum pool drive incremental increases
VNX5100	20
VNX5300	40
VNX5500	80
VNX5700	120
VNX7500	180

Table 25 Pool drive increments for different VNX-series models, OE Block 31.0

To create a pool with a greater number of drives than the maximum pool drive increment, create the pool and then add increments of drives until the pool has the desired number of drives.

Try to keep the number of drives in the increments the same, while still following the guidance in the Drive counts section above.

The pool can be expanded again after a delay of one or two minutes. The amount of time between increments depends on the storage system model, the drives added, and number of drives in the expansion.

When more than one increment is required for the expansion, perform then expansion increments as close together in time as is practical. In addition, all expansions should be performed when the pool is under no or low write I/O intensive workload.

General Virtual Provisioning pool recommendations

The general recommendations for creating Virtual Provisioning pools are as follows:

- Create pools or pool tiers using storage devices that are the same type, form factor, speed, and size.
- For Ease-of-Use:
  - Create homogenous or single tiered FAST VP Pools with the largest, available number of drives.
- For Performance:
  - Use 15K rpm SAS and Flash drives for pools with thin LUNs or higher performance requirements.
  - For small and moderate sized pools, and pools requiring a higher-level of write performance, use the RAID 1/0 level data protection. It provides the highest user data capacity and write performance per number of pool drives. However, it does not have the higher capacity utilization of RAID 5.
- For Availability:
  - For large pools, and pools that require high availability, use RAID 6. If the pool is composed of exclusively or significantly of high capacity (>1 TB) NL-SAS drives—use of RAID 6 is *strongly* recommended.
  - The OE Block 31.0 revision of FAST Virtual Provisioning supports only a single RAID group level across all tiers. It is well-understood that using RAID 6 for all tiers may not be the most efficient use of available capacity. However, if the pool best practices are followed with regard to drive mixes, 80 percent or greater of the pool's total capacity will be on high-capacity NL-SAS drives. The recommendation to use RAID 6 when 1 TB or greater capacity NL-SAS drives populate the pool provides the highest level of data protection for the great majority of user data.

Virtual Provisioning pool-based LUNs

There are two types of LUNs supported by Virtual Pools, thin and thick.

Thin and thick LUNs overview

The VNX Virtual Provisioning supports thin and thick LUNs. You can provision pool LUNs using Unisphere or the CLI.

Thin LUNs present more storage to an application than is physically available. This is an efficient way to manage capacity.

Thick LUNs provide guaranteed allocation for LUNs within a storage pool, as well as more deterministic performance.

Both thin and thick LUNs may be provisioned within the same pool.

Pool LUNs' capacity can be non-disruptively and incrementally added to with no affect on the pool's provisioned LUNs.

Thin LUNs

Thin LUNs in a Virtual Provisioning pool share the pool's available storage. The capacity of the thin LUN that is visible to the host is independent of the available physical capacity of the pool. To a host, a thin LUN is indistinguishable from any other type of LUN.

Thin LUNs maximize ease-of-use and capacity utilization at some expense to performance. The capacity utilization of the thin LUNs will be much higher than thick LUNs, as the thin LUNs allocate storage only 'as needed'. When setting SLAs however, assume thin LUNs will have about two-thirds the performance of thick LUNs or traditional LUNs.

Thick LUNs

The capacity of a thick LUNs distributed equally across the drives of a Virtual Provisioning pool. The amount of physical capacity reserved for a thick LUN, is the same user capacity visible to the host O/S. A thick LUN ultimately uses slightly more capacity than the amount of user data written to it due to the metadata required to maintain it.

A Thick LUN's performance is comparable to the performance of a traditional LUN and is better than the performance of a thin LUN.

High-level pool-based LUN usage recommendations

When creating storage pools, if the goal is:

- Most efficient use of capacity - provision the pool with thin LUNs.
- Highest level of pool-based LUN performance - provision the pool using thick LUNs.

Pool-based LUN capacity utilization

A pool LUN is composed of both metadata and user data, both of which come from the storage pool. There is a fixed capacity overhead associated with each LUN created in the pool. Take into account the number of LUNs anticipated to be created, particularly with small Virtual Provisioning pools that have more limited capacity.

LUN metadata

The pool LUN's metadata subtracts from the pool's user data capacity. Plan ahead for metadata capacity usage when provisioning Virtual Provisioning pools. Additional capacity used for maintaining and operating the pool is at the LUN-level. Two pools with 10 LUNs each have the same pool capacity utilization as one pool with 20 LUNs. Calculations for estimating pool capacity are provided below.

With multi-terabyte pools the percentage of the pools capacity used for metadata shrinks to less than 1% and should not be a concern. With small capacity pools the percentage of capacity used by metadata can become a considerable proportion of the pool. Always, create pools with enough initial capacity to account for metadata usage and any initial user data for the planned number of LUNs. In addition, be generous in allocating capacity to a created thin LUN. This will ensure the highest percentage of pool capacity utilization.

Pool capacity utilization

All thin LUNs will consume a minimum of about 3 GB of pool capacity. This minimum capacity includes:

- About 1 GB of capacity for metadata
- An initial 1 GB of pool capacity for user data
- 1 GB of pre-fetched pool capacity

The prefetch of 1 GB of metadata remains about the same from the smallest though to the largest LUNs.

Additional metadata is allocated from the first 1 GB of user data as the thin LUN's user capacity increases. The initial size of the LUNs and ideally an estimate based on historical rate of growth are needed to complete any estimate the eventual pool capacity utilization.

Thick LUN capacity utilization

All thick LUNs will likewise consume additional pool capacity beyond the User Capacity selected. However, thick LUNs reserve their capacity immediately when they are bound. Upon creation the reported consumed capacity from the pool. To estimate the capacity consumed for a thick LUN follow the same rule for thin LUNs.

Pool capacity estimate

To estimate the capacity consumed for a Virtual Provisioning pool-based LUN follow this rule of thumb:

Consumed LUN capacity = (User Consumed Capacity * 1.02) + 3GB.

For example, assume a pool must be created to hold a 3 TB LUN. How much raw capacity is needed to provision the pool?

3136 GB = (3072 GB * 1.02) +3 GB

Extended Example

An extended example of provisioning may be helpful. This is included to show Virtual Provisioning pool capacity usage through LUN migration into a newly created pool.

Assume a small, homogenous, RAID level-5, Virtual Provisioning pool provisioned with 20x 600 GB SAS drives. The drives give 12 TB raw capacity. Further assume five LUNs are initially needed. Two LUNs are thin LUNs. Each Thin LUN is 2 TB. One of the thin LUNs will receive 1 TB of user data via a LUN migration. The other thin LUN will be initially un-used. The remaining three LUNs are thick LUNs with different capacities. The table below shows the LUN types and capacities for the example.

What will be the remaining capacity in the pool?

LUN Types and Capacities Extended Example		
LUN Name	Type	Available Capacity (GB)
LUN0	Thin	2048
LUN1	Thin	2048
LUN2	Thick	512
LUN3	Thick	1024
LUN4	Thick	2048

Table 26, Pool Capacity-- Extended Example

The procedure is as follows:

1. Calculate the formatted capacity of the pool.
2. Calculate the capacity of in-use pool LUNs
3. Calculate the remaining pool capacity.

The important part of this example to note is the difference between available capacity and consumed capacity. Thick LUNs have their capacity completely allocated. Thin LUNs only have their consumed capacity allocated.

Calculate formatted capacity

First, calculate the formatted capacity of the drives. The 20x 600 GB drives will be grouped into four 4+1 private RAID groups using default settings. Each formatted 600 GB SAS drive 4+1 private RAID group has 2147 GB of usable capacity after formatting.

8588 GB = 2147 GB * 4.

This is the pool's initial estimated capacity without LUNs.

Calculate In-use pool capacity

Next, calculate the in-use pool capacity of the populated thin and thick LUNs using the given formula.

Consumed capacity = (User Consumed Capacity * 1.02) + 3GB.

Note that that LUN0 is a thin LUN, which may have 2 TB allocated, but is only consuming 1 TB (1024 GB).

LUN0: 1047 GB = (1024 GB * 1.02) + 3 GB

The table below shows the result of the calculations

In use Pool Capacity—Extended Example	
LUN Name	In-use Capacity (GB)
LUN0	1047
LUN1	3
LUN2	525
LUN3	1047
LUN4	2092
Total In-use capacity:	4714

Table 27, In-use Pool Capacity-- Extended Example

Calculate the remaining pool capacity

The remaining pool capacity is the difference between the formatted capacity of the pool and the In-use capacity. The formatted capacity of the pool from the first step is 8588 GB. The In-use capacity is from the table above

3874 GB = 8588 GB - 4714

Analysis

From the calculation, you can see that the almost 4 TB of capacity remains for the thin LUNs to allocate or for later expansion of the thick LUNs in the initial pool. Note, that the thin LUNs are only using 1050 GB of their combined subscribed 4096 GB capacity. Together, they could consume an additional 3 TB of pool capacity, If they were to become fully populated before the

pool was expanded. This would leave about 800 GB of unsubscribed pool capacity for additional LUNs or LUN expansion.

Pool LUN configuration: Initial data placement

When a pool LUN is created, its placement mode policy must be set. The correct placement mode will determine how quickly optimal pool performance will be reached.

Homogenous pools

Homogenous pools have no placement policy. Pool LUN data in homogenous pool is spread over the entire pool.

For thin LUNs, initial loading of user data into a newly created thin LUN results in a very compact and efficient organization. Pool capacity is consecutively allocated until the initial LUN data is loaded. This initial loading through migration of a thin LUN results in a 'thick LUN-like' allocation of pool capacity for the thin LUN. Subsequent writes to multiple thin LUNs will result in a more characteristic thin LUN data placement.

Tiered pools

The tiering policy of a LUN will determine the initial allocation of its data across the tiers. The policy also sets up the basic algorithm by which data is periodically relocated through promotions and demotions between the tiers.

A LUN's tiering policy can be changed after initial allocation.

There are several options available for initially populating the tiers of a newly created FAST VP pool with user data. The data placement policy selected will affect how long it will take the tiered pool to reach its highest efficiency.

The data placement policy options are:

- Lowest
- Highest
- Auto
- No Movement

Recommended policy

The data placement option has a big effect on the efficiency of the tiers and the relocation process. It also determines how quickly a pool will warm-up, reaching its optimal performance.

The default for OE Block 31.0 is Auto. The recommended FAST VP placement policy for your highest priority data is *Highest Available Tier*.

Highest results in the highest performance in the shortest amount of time, because data will be placed in the highest tiers as capacity allows. In particular with newly created, or sparsely populated tiered pools this gives the best LUN performance.

FAST VP pool warm-up

After a FAST VP pool is created, it may take some time for the data to be matched to the tier that best corresponds to its activity level. The system will begin tracking the relative activity of all data in the pool, and ranking it for movement up or down accordingly. This ranking will also factor in the user-defined initial allocation and tiering policy. Data will be moved according to the user-defined relocation schedule and rate.

the initial performance and allocation of LUN data will likely be different from its operational performance and data allocation between the tiers . The FAST VP pool must first *warm-up.* Warm-up is the filling of the appropriate FAST VP tiers with the workload's working set. Depending on the policy, frequently accessed LUN data will be promoted to higher performing tiers and infrequently accessed data will either remain in-place or be demoted to more modestly performing tiers.

The efficient operation of FAST VP depends on locality. The higher the locality, the higher the efficiency of the tiering process. Accumulating statistics on data access takes time. When a LUN is first bound to a pool, no frequency of access information exists for its contents. Host performance and capacity utilization will depend on the initial data placement assumptions made when creating the LUN. FAST VP collects the data continuously and updates hourly to indicate how much data needs to move in the pool. This information is available through Unisphere. As statistical information on data access accumulates, the FAST VP algorithm will efficiently reallocate the LUN's contents among its tiers for optimal performance and capacity utilization. This relocation of data happens during the relocation window only. The time it takes for the highest-level tiers to become completely populated with the most frequently accessed data depends on locality in the workload.

The time required to warm-up the pool is affected by:

- Workload
- Initial data allocation policy
- Data Relocation Rate
- Data Relocation Schedule.

Another major dependency is the rate at which the data access of the pools contents changes. Expect daily, monthly, quarterly and yearly business cycles to affect the pool's distribution of data between the tiers.

For example, End-of-month, quarterly or year batch runs may cause infrequently accessed data in lower tiers to promoted to upper tiers. Take this into account when calculating the performance of these activities. The tiering policy may need to be changed before a known period of high activity to avoid unwanted changes to the pool.

Increasing the rate and schedule may decrease the duration of the warm-up. However, the effectiveness of setting high rates and frequent scheduling depends on how much access data has already been accumulated. Setting high rates with no historical data will not have any effect, and will likely adversely affect overall storage system performance. Note that manual relocations of a LUNs data is possible at any time. This may be used to move the entire contents of a LUN between tiers.

When the pool is warmed up, FAST VP will attempt to maintain each tier with a maximum of 90 percent allocation unless the pool capacity is already over 90 percent in use. If the pool's capacity is running at 95 percent in use, FAST tries to maintain the same level of capacity usage on the highest tier (for example, 95 percent).

FAST VP tiering effect on performance

The resource-intensive part of FAST VP tiering is the data relocation that happens in the relocation window of time configured on the system.

In general the default rate of medium is recommended to minimize the affect of data relocation on overall system performance. In addition, the window for relocation should be set to occur at 'off-peak' storage system activity times.

Thin LUN migrations

There are some special considerations with migrations to thin LUNs that need to be taken into account to get the most efficient capacity utilization.

Alerts

Thin LUNs use slightly more capacity than the amount of user data written to it due to the metadata required to reference the data.

Unlike a traditional LUN or a thick LUN, a thin LUN can run out of disk capacity if the pool to which it belongs runs out of physical capacity. This event is generates an unrecoverable write error to the application and data from that write operation will not be available for future reading.

By default, the storage system issues a warning alert to the storage systems log when 70 percent of the pool's available capacity has been consumed. When 80 percent of the space has been consumed, it issues a critical alert. The alert thresholds are customizable. As thin LUNs continue consuming the pool's space, both alerts continue to report the actual percentage of consumed space.

Note that a pool provisioned exclusively with thick LUNs will not generate alerts, it cannot run out of capacity.

Immediately expand the pool's capacity on the first alert.

NTFS Capacity reclamation

Files that have been "Deleted" on a NTFS file system aren't removed from disk. The storage system requires that the host-based file system handle the reallocation of a deleted file's capacity.

For NTFS-based hosts (typically Microsoft-based), before using LUN migration, you should use the Microsoft `sdelete` utility with the `-C` option. It goes out and replaces deleted data with zeroes. Most write-ups on Microsoft refer to it as a security utility because you're overwriting hidden data. In our case, we just need the space written in zeroes so that we can punch holes in the target thin LUN.

If you want to "redo" your initial migration, migrate it back to the RAID group, use sdelete, then migrate it back to thin. Otherwise, you can give it a try on your next migration.

Usage: `sdelete [-p passes] [-s] [-q] <file or directory>`

`sdelete [-p passes] [-z|-c] [drive letter]`

-c Zero free space (good for virtual disk optimization).

-p passes Specifies number of overwrite passes.

-s Recurse subdirectories.

-q Don't print errors (quiet).

-z Cleanse free space.

LUN compression

The LUN compression feature provides the option of compressing the data on the LUN to free-up storage capacity for infrequently accessed data.

LUN Compression Overview

Compression performs an algorithmic data compression of Virtual Provisioning pool-based LUNs. All compressed LUNs become thin LUNs. Additional information

See the whitepaper *EMC Data Compression: A Detailed Review* for additional information. This paper is available on Powerlink.

Compression candidate LUNs

The typical post-processing compression ratios for are typically 2:1. Compressibility is highly dependent on the contents of the data. For instance, text files tend to be highly compressible, whereas media files that have native compression, tend to have little to no additional compressibility.

Uncompressible LUNs

Private LUNs cannot be compressed. Examples of private LUNs include metaLUN components, Snapshot LUNs, or any LUN in the reserved LUN pool.

Host I/O response time is affected by the degree of compression. Highly compressible data stored on a compressed LUN will have a longer response time, as it must be decompressed.

Uncompressible or host-compressed stored data that does not need to be decompressed by the storage system will have a short I/O response time.

Settings and configurations

One, some or all of the LUNs in a Virtual Provisioning pool or a traditional LUN may be compressed.

Compression can be turned ON or OFF for a LUN. Turning it ON causes the entire LUN to be compressed and subsequent I/O to be compressed for writes and decompressed for reads.

Compression is a background process, although read I/Os will decompress a portion of a source LUN immediately. The number of compressions taking place at the same time is model-dependent. Table 28 Maximum Compression operations by model shows how many can be active at the same time. The number of migrations represents the number of concurrent initial compression operations for non-thin LUNs. Initial compression of thin LUNs is bound to the maximum LUN compression per SP.

	Maximum LUN Compressions Per Storage Processor	Maximum LUN Migrations per Storage System
VNX5100	N/A	8
VNX5300	5	8
VNX5500	5	16
VNX5700	8	24
VNX7500	10	24

Table 28 Maximum Compression operations by model, VNX OE Block 31.0

Writes to a compressed LUN cause that area to be decompressed and written as usual, then it is recompressed as a background process. The background process may be prioritized to avoid an adverse effect on overall storage system performance. The priorities are: High, Medium, and Low, where High is the default.

If the maximum concurrent compression operations are active at the same time, High-prioritized decompression can adversely affect storage system performance. High utilization can continue for some time after compression completes. It is recommended to pause compression at the system-level should response time critical workloads be active.

Compression can be turned OFF for a LUN. This will cause the LUN to "decompress." Decompressing a LUN depends on the size of the LUN. .

Compressed LUN performance characteristics

Compression should only be used for archival data that is infrequently accessed. Accesses to a compressed LUN may have significantly higher response time than normal accesses to a Virtual Provisioning pool-based LUN. The duration of this response time is dependent on the size and type of the I/O, and the the extent to which the LUN has been compressed. Note that at the host level, with a fully operational write cache, delays for writes to compressed LUNs are mitigated.

In addition, the effects of I/O type and compressibility are cumulative; small-block random writes of highly compressible data have the highest response time, while small-block random reads of data that is not compressible have the lowest response time.

Storage administrators should take into account the longer response times of compressed LUNs when setting timeout thresholds.

In addition, there is an overall storage system performance penalty for decompressions. Large numbers of read I/Os to compressed LUNs, in addition to having a lower response time, have a small effect on storage processor performance. Write I/Os have a higher response time, and may

have a larger adverse effect on storage system performance. If a significant amount of I/O is anticipated with an already compressed LUN, you should perform a *LUN migration* ahead of the I/O to quickly decompress the LUN to service the I/O. This assumes available capacity for the compressed LUN's decompressed capacity.

Traditional LUNs

Traditional LUNs are a logical construct overlaid directly onto RAID groups.

RAID groups and LUNs

The data capacity of a RAID group can be partitioned into one or more traditional LUNs. The maximum number of LUNs per RAID group is shown in Table 20 Maximum Host LUNs per LUN Type VNX O/S Block 31.0. A LUN can be of any size capacity-wise from one block to the maximum capacity of the RAID group. A LUN's capacity is taken equally from all disks of the underlying RAID group.

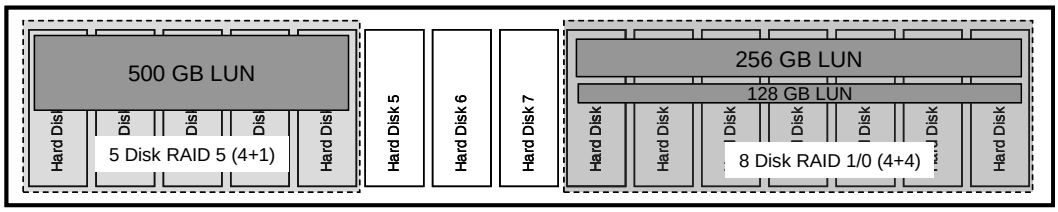

Figure 7. LUN conceptual diagram

Dedicated versus partitioned RAID groups

A RAID group with only one LUN, taking up all the available user capacity, is a *dedicated* RAID group. A RAID group having more than one LUN is a *partitioned* RAID group.

Due to the large capacities of individual drives, a single RAID group can have a very large capacity. LUNs can be used to partition a RAID group's available capacity into smaller parts. The capacity of most RAID groups is shared this way. LUNs on a partitioned RAID group consist of contiguous sets of RAID stripes. Sequential reads and writes and I/O operations using large I/O sizes cause all the drives of the underlying RAID group to work in parallel. This is a very efficient use of the storage system's resources resulting in high bandwidth.

Try to limit the LUNs per RAID group to the smallest possible number. This avoids possible *linked contention* and *drive contention*. Linked contention is when I/O to different LUNs forces mechanical hard drive based RAID group's drive heads to into large movements going from LUN to LUN. Drive contention is when more than one I/O stream needs to access the same drive at the same time. The recommendation comes because as the number of LUNs sharing a RAID group increases it becomes more difficult to predict or determine if the I/O is complimentary or is contentious without time consuming analysis and debug.

In a partitioned RAID group, a RAID group shared by the peer storage processors, other LUNs will share the underlying RAID group's IOPS and bandwidth. That is, if separate applications use a LUN created on the same RAID group, their combined usage needs may exceed the RAID group's performance capabilities.

LUN expansion

Expansion is the process of adding capacity to a base LUN. However, a LUN cannot have a capacity larger then its underlying RAID group's capacity.

Unbinding

Unbinding a LUN releases all its component drives for reuse resulting in the previously stored data being lost.

Defragmenting RAID groups and LUNs

Defragmentation is a process that applies to RAID groups and traditional LUNs only. Defragmentation of RAID groups and traditional LUNs may improve performance.

RAID group defragmentation

A partitioned RAID group may become fragmented when one of its LUNs is unbound. LUN fragmentation occurs when gaps of unused capacity exist between LUNs in a RAID group. This leaves less contiguous space in the RAID group for new LUNs or for expanding existing LUNs. In addition, this gap or gaps increases the seek distance between LUNs within the RAID group. Through Unisphere, a RAID group can be defragmented to compress the gaps and recover the unused capacity within a RAID group. This makes all the RAID group's in-use capacity contiguous allowing for creation of new, larger LUNs.

Note that RAID group defragmentation is *not* file system defragmentation. (They are frequently confused.) RAID group defragmentation has *no* effect on the positioning of application data within a LUN.

In addition, as of VNX OE Block 31.0 , RAID 6 groups cannot be defragmented.

File-system fragmentation

When the percentage of storage capacity utilization is high, file system defragmentation of traditional LUNs on mechanical hard drive based RAID groups can improve performance.

Note, EMC does *not* recommend file system defragmenting of:

- Virtual Provisioning pool-based LUNs
- Traditional LUNs bound on flash drive provisioned RAID groups
- Any LUNs that are using the FAST Cache feature

See the Host file-system fragmentation section.

MetaLUNs

A MetaLUN is two or more joined traditional LUNs acting in concert, presented to a host as a single LUN. MetaLUNs permit traditional LUNs to be scaled upward when the performance or capacity requirement exceeds that available from the maximum number of drives in a LUN bound to a single RAID group. They also allow users to provision to:

- Expand the capacity of existing traditional LUNs while maintaining performance
- Create traditional LUNs with very large capacity
- Create higher-performance traditional LUNs
- Move the LUN from one type of drives to another type
- Change the RAID level protection of the LUN

MetaLUN Overview

MetaLUNs are a VNX standard feature available on all models.

The simplest use of MetaLUNs is to expand an existing traditional LUN's capacity while maintaining its level of performance. This is done by joining a newly created LUN to an existing LUN that no longer has any available capacity or needs. The new "larger" LUN appears to a host to be identical to the original, except now with more capacity.

When creating a MetaLUN, a *base* LUN is joined to one or more *component* LUNs. The base LUN and component LUNs must originally be traditional LUNs. A MetaLUN can be scaled in a step-wise process by joining additional LUNs to it to make a larger MetaLUN. A simple MetaLUN may be made up of wholly component LUNs. A complex MetaLUN may be made up of MetaLUN components, where each MetaLUN component is one or more component LUNs.

MetaLUNs may be homogenous or heterogeneous in organization. That is, the traditional LUNs joined into a MetaLUN may have the same underlying RAID group organization or they may

come from RAID groups with different RAID levels with compatible redundancy schemes, and varying numbers of drives.

For example, a very large capacity traditional LUN using available 2 TB NL-SAS drives and a 12-drive RAID 6 would have a usable capacity of about 18.3 TB. Using MetaLUNs allows for two such LUNs to be joined together to create a larger 36.6 TB host LUN. More LUNs can be joined to make even larger capacity MetaLUNs.

The maximum capacity of a MetaLUN is limited by the number of drives available on the VNX storage system. Another benefit of creating LUNs with more drives is increased performance. Using the correct techniques, MetaLUN usage can increase the throughput (IOPS) and bandwidth of an existing LUN or create LUNs with higher IOPS and bandwidth than may be available with a single traditional LUN.

The initial provisioning and maintaining of MetaLUNs is *in addition to* traditional LUN creation and maintenance. Understanding the fundamentals of traditional LUNs and their underlying RAID group's organization is important if you wish to make use of the full potential of MetaLUNs.

Additional Information

An in-depth discussion of metaLUNs, including how to create them, can be found in the *EMC CLARiiON MetaLUNs - A Detailed Review* white paper available on Powerlink.

Availability

The following sections cover the storage system's logical storage object's availability Best Practices.

RAID groups

Each RAID level has its own data protection characteristics. For specific workloads, a particular RAID level can offer clear availability advantages over others.

RAID Group data redundancy

VNX storage systems support RAID levels 0, 1, 3, 5, 6, and 1/0. Refer to the *EMC Unified Storage System Fundamentals* white paper to learn how each RAID level delivers availability.

When provisioning RAID groups hosting either traditional LUNs or Virtual Provisioning pool-based LUNs with NL-SAS drives having capacities of 1 TB or larger, it is *strongly* recommended that RAID level 6 be used. All other drive types may use either RAID level 5 or 1/0.

Basic LUN Availability

This section discusses basic LUN parameters that may affect the choice of LUN type used for hosting user data.

Availability of LUNs is based on the underlying availability provided by the LUN's RAID groups, as well as the availability of the storage processors, caches, and back end. (The section on RAID groups will help you understand the availability implications of creating different types and capacity RAID groups.) However, in general the distribution of the workload's data across LUNs on more than one RAID group can increase the overall system availability. In addition, it can increase performance by reducing the time required to execute a rebuild.

When possible, place recovery data, such as clones and log files, on LUNs supported by RAID groups that do not also support the application's LUNs. Also keep backups on separate RAID groups from application data. When more than one instance of an application exists, separate log LUNs from their instance's application and application data. Placing an application's recovery data on LUNs that are not on the same RAID group as the application's LUNs speeds up the application recovery process. This is because the recovery data is immediately available and not affected by the delay of, for example, a drive rebuild.

Note that extending this to Virtual Provisioning pools, will require two or more pools be created.

Pool Availability

A fault domain refers to data availability. A Virtual Provisioning pool is made up of one or more private RAID groups. A pool's fault domain is a single pool private RAID group. That is, a pool's fault domain encompasses all RAID groups in the pool, which means that any failure to any one RAID group affects all pool contents.

Generally, availability considerations applying to pools are the same that applied to provisioning with traditional LUNs. Common considerations include:

- MTBF of underlying storage devices
- RAID level data protection
- Number of RAID groups
- Rebuild Time and other MTTR functions

Underlying storage devices

Device level availability should be carefully considered. Flash and SAS drives have the highest availability of all VNX storage devices. Flash drives which have no moving parts and lower power consumption than mechanical hard drives have higher availability than mechanical hard drives. For the highest availability, provision pools with either Flash or SAS drives. A RAID 6 level of protection is strongly recommended when NL-SAS is present, especially in large, virtual pools

RAID-level data protection

All the LUNs bound within a Virtual Provisioning pool will have data loss from a complete failure of a pool RAID group. The larger the number of private RAID groups within the pool, the bigger the effect of a failure.

It is important to choose a level of protection for the pool in-line with the value of the pool's contents.

Three levels of data protection are available for pools. They are:

- RAID 5 has good data availability. If one drive of a private RAID group fails, no data is lost. RAID 5 is appropriate for small to moderate sized pools. It may also be used in small to large pools provisioned exclusively with SAS and flash drives which have high availability.
- RAID 6 provides the highest data availability. With RAID 6, up to two drives may fail in a private RAID group and result in no data loss. Note this is true double-disk failure protection. RAID 6 is appropriate for any size pool, including the largest possible.
- RAID 1/0 has high data availability. A single disk failure in a private RAID group results in no data loss. Multiple disk failures within a RAID group *may* be survived. However, a primary and its mirror cannot fail together, or data will be lost. Note, this is *not* double-disk failure protection. RAID 1//0 is appropriate for small to moderate sized pools.

A user needs to determine whether the priority is: availability, performance, or capacity utilization.

If the priority is availability, RAID 6 is the recommendation.

If it is capacity utilization or performance, and you believe they have sound policies and procedures for data protection in place (backups, hot spares, etc.), pursuing a RAID level 5 or 1/0 provisioning of FAST pools is likewise a sound decision.

Number of RAID groups

A fault domain refers to data availability. A Virtual Provisioning pool is made-up of one or more private RAID groups. A pool's fault domain is a single pool private RAID group. That is, the availability of a pool is the availability of any single private RAID group. Unless RAID 6 is the pool's level of protection, avoid creating pools with a very large number of RAID groups

Rebuild Time and other MTTR functions

A failure in a pool-based architecture may affect a greater number of LUNs than in a traditional LUN architecture. Quickly restoring RAID groups from degraded mode to normal operation becomes important for the overall operation of the storage system.

Always have hot spares of the appropriate type available. The action of proactive hot sparing will reduce the adverse performance effect a Rebuild would have on backend performance. In addition, always replace failed drives as quickly as possible to maintain the number of available hot spares.

Chapter 5 Storage System Sizing and Performance Planning

This chapter is included to show how to estimate the provisioning of a storage system for a workload using Virtual Provisioning. This chapter is *not* a substitute for the software tools available to EMC Sales and Technical Professionals providing sales support.

Introduction

The section provides information on creating a Rough-Order-of Magnitude (ROM) estimate of a Virtual Provisioning pool based on a hypothetical workload. A ROM results in an *approximate* number of storage devices and a *candidate* storage system to handle a workload.

Workload

Understanding the workload is always the first thing you need to know for storage system configuration. The workload's requirements can broadly be defined as capacity and performance. It follows there is a two-step process for sizing a storage system. It consists first of calculating the right number of drives for capacity, and then calculating the number of drives for performance. The performance calculation itself consists of two sub-steps: calculating the correct number of drives for IOPs or bandwidth, and then the right model storage system to support the drive's performance.

In addition, future growth should be planned for. It is important to have enough storage capacity and performance to satisfy the workload's peak and near-future requirements.

The following sections show how to provision a storage system with a single OE Block 31.0 FAST VP pool. The example is general enough to be applied to traditional LUNs, homogeneous Virtual Provisioning pools, FAST VP pools with different configurations.

The Capacity

First determine the RAID type of the pool and drive-group size. This calculation affects capacity in parity RAID types.

Once the number of drives needed to meet capacity needs is known, the performance calculation can be made.

System drives

The first four drives in a VNX storage system are the system drives, and must be included in any estimate. System drives may not be used as part of a Virtual Provisioning pool. . In traditional LUN based provisioning, a portion of these drives capacity may be used in the workload.

Actual drive capacity

Accessible capacity may vary because some operating systems use binary numbering systems for reported capacity. Drive manufacturers consider a gigabyte to be 1,000,000,000 bytes (base 10 gigabyte). A computer O/S may use base 2 (binary) and a binary gigabyte is 1,073,741,824 bytes.

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

For example, a 600 GB drive has a formatted data capacity of about 537 GB.

Capacity Utilization versus RAID protection

The use of parity or mirroring to protect data from drive failure also reduces usable storage. Mirroring always requires that 50 percent of total storage capacity be used to ensure data protection. This is called *protection capacity overhead.*

For example, an 8-drive RAID 1/0 (4+4) has a protection capacity overhead of 50-percent. This is the default private RAID group created for RAID 1/0 Virtual Provisioning pools. Four of the eight drives are available for storage; the remaining four are mirror-copies providing data protection. Assume this RAID 1/0 group is composed of formatted 600 GB (raw capacity) 15k rpm SAS drives. The usable capacity of this RAID group is about 2147 GB. Note the difference between the actual usable capacity and the approximately 4800 (8*600 GB) 'raw' capacity that might be assumed, if formatting and mirroring are not taken into account.

The percentage of parity RAID space given to protection capacity overhead is determined by the number of drives in the group and the type of parity RAID used. RAID-level 5 has a single drive capacity equivalent of overhead out of total drives. RAID 6 has a two-drive equivalent overhead.

For example, a five-drive RAID 5 (4+1) group has a 20 percent overhead. It has the equivalent of four drives available for storage and the overhead of one drive for parity. Assume this RAID 5 group is composed of formatted 600 GB (raw capacity) 15k rpm SAS drives. Taking into account formatting and parity, this would result in a total usable capacity for this RAID group of about 2147 GB.

Performance

Performance planning or forecasting is a technique that requires an understanding of the storage environment. Information such as the threading model of the workload, the type of I/O (random or sequential), the I/O size, locality, and the type of drive all affect the final performance observed. The *EMC Unified Storage System Fundamentals* white paper contains detailed information on the factors affecting performance.

Rule-of-thumb approach

Estimates of RAID group response time (for each drive), bandwidth, and throughput need to account for the I/O type (random, sequential, or mixed), the I/O size, and the threading model in use.

To simplify this performance estimation, a rule-of-thumb approach is used for throughput (IOPS per private RAID group) and bandwidth (MB/s per private RAID group). Use the guideline values provided in this estimate. These numbers are intentionally conservative and result in a simplistic measure.

Small-block random I/O

Small-block random I/O, like those used in database applications and office automation systems, typically require throughput with an average response time of 20 ms or less. At an average drive-queue depth of one or two, assume the following per drive throughput rates:

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 one-third fewer IOPS per drive.

In cases of random I/O sizes greater than 16 KB, there will be a steady reduction in the IOPS rate for mechanical drives. The rate for flash drives is more abrupt. The IOPS rate for Flash drives should be reduced for large-block random I/O, from 8 KB and every time you double the I/O size to 64 KB reduce IOPS by about 30 percent.

In cases of well-behaved sequential access, the rate may be well double the listed IOPS for SAS drives, even for large I/O sizes.

When architecting for optimal response time, limit the drive throughput to about 70 percent of the throughput values shown in Table 9 Small block random I/O performance by drive type Optimal throughput can be achieved by relaxing response time and queue depth ceilings. If a response time greater than 50 ms and a drive queue depth of eight or more is allowable, the table's drive throughput can be increased by 50 percent more IOPS per drive.

Large-block random I/O

For random requests 64 KB and greater, drive behavior is usually measured in bandwidth (MB/s) rather than IOPS. As the block size increases, so does the per-drive bandwidth. At a queue depth of one, assume the following per drive bandwidth rates:

Drive type	64 KB (MB/s)	≥512 KB (MB/s)
15K rpm SAS	12.0	32.0
10K rpm SAS	10.0	24.0
7.2K rpm NL-SAS	6.0	24.0
Flash drive	100	100

Table 29 Large block random bandwidth by drive type, OE Block 31.0

Note the number of threads has a big effect on bandwidth. With a five-drive 15K rpm SAS RAID 5 (4+1) LUN with a single thread continuously reading a random 64 KB pattern, the per-drive queue depth is only 0.2 and the 8 MB/s bandwidth applies to the sum of the spindle activity. In contrast, a 16-thread 64 KB random read pattern can achieve about 60 MB/s.

The number of drives that can be driven concurrently at the shown rates will be limited by the available back-end bandwidth of the storage system. However, the back-end bandwidth of the VNX is large in comparison to legacy storage systems. The SAS backend-port can easily exceed loads of 700 MB/s without danger of saturation.

Sequential I/O

For 64 KB and greater block sizes running single thread sequential I/O, RAID group striping makes bandwidth independent of the drive type. Use 30 MB/s per drive as a conservative design estimate.

Depending upon your environment, drive bandwidth can be improved considerably though tuning. For instance, for block sizes between 32-64 KB, by using a prefetch multiplier of 16, and a segment multiplier of 16, a five-drive RAID 5 (4+1) can achieve 50 MB/s per drive.

Sending fewer, larger I/Os to the storage system's back-end improves sequential I/O performance. When more than one RAID group stripe is read or written, each drive in the group gets a single large I/O. This results in the most efficient use of the back-end port and drives.

This is particularly true if NL-SAS drives are the destination of the I/O. The best sequential write I/O performance with any RAID stripe size occurs when the default write cache page size of 16 KB and the RAID group's stripe size are evenly divisible. This allows for up to 2 MB to be sent to the drives from the cache in one operation of writing multiple RAID group stripes. For

example, with a 16 KB cache page, a RAID 5 (4+1) with its 256 KB stripe size has eight stripes written in one operation.

Mixed random and sequential I/O

In mixed loads, the pure sequential bandwidth is significantly reduced due to the head movement of the random load, and the random IOPS are minimally reduced due to the additional sequential IOPS.

The sequential stream bandwidth can be approximated using the values in**Table 29** and the random load can be approximated by using 50 percent of its listed IOPS. Aggressive prefetch settings (prefetch multiplier 16, segment multiplier 16) improve the sequential bandwidth at the expense of the random IOPS. Increasing the random load queue depth increases its IOPS at the expense of the sequential stream bandwidth.

Effect due to LUN and I/O type

The type of LUN also has an effect on I/O performance – such as whether the LUN is a thick or thin LUN, or even a traditional LUN. The I/O type must also be factored in.

Effect of FAST Cache

The use of a FAST Cache can augment storage system performance for workloads that can leverage it. An estimate of the FAST Cache hit rate is needed. The hit rate directly reduces the estimate of host IOPS being handled by the LUN. The estimate should be based on a warmed-up cache. Note that initially the cache is un-warmed and the hit rate shall be lower.

Storage System 'Sweet Spot'

For the most efficient throughput performance, the optimal mechanical drive count for the storage system is shown in the table below**..** Note this is the number of drives where the storage system *begins* to become efficient; it is not a maximum, although in some cases the sweet spot is the maximum drive count for the model. Efficiency scales upward from this point.

Use the maximum drive count for the storage system when considering entirely NL-SAS drive installations. Divide the required number of drives by the drive counts in the table to determine the type and number of storage systems to deploy.

High-performance storage systems require a more complex analysis. They are not covered in this example. Consult with an EMC USPEED professional for information regarding high-performance and high-availability storage system configurations.

VNX Model	Mechanical Drives Random I/O	Mechanical Drives Sequential I/O (Reads)*
VNX5100	75	60
VNX5300	125	80
VNX5500	250	120
VNX5700	500	140
VNX7500	1000	300

*Note that mechanical drive write performance is somewhat higher use these intentionally conservative values for ROM estimates.

Table 30 'Sweet-Spot' by model, VNX OE Block 31.0

Performance estimate procedure

The steps required to perform a ROM performance estimate are as follows:

1. Determine the workload.

2. Determine the I/O drive load.
3. Determine the number of drives required for Performance.
4. Determine the number of drives required for Capacity.
5. Analysis

The steps need to be executed in sequence; the output of the previous step is the input to the next step.

Determining the workload

This is often one of the most difficult parts of the estimation. Many people do not know what the existing loads are, let alone load for proposed systems. Yet it is crucial for you to make a forecast as accurately as possible. An estimate *must* be made.

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

Determine the I/O drive load

This step requires the use of Table 9 Small block random I/O performance by drive type Note the IOPS values in the table are *drive IOPS*. To determine the number of drive IOPS implied by a host I/O load, adjust as follows for parity or mirroring operations:

- **Parity RAID 5:** Drive IOPS = Read IOPS + 4*Write IOPS
- **Parity RAID 6:** Drive IOPS = Read IOPS + 6*Write IOPS
- **Mirrored RAID 1/0:** Drive IOPS = Read IOPS + 2*Write IOPS

An example the default private RAID group of a RAID 1/0 pool is a (4+4). Assume a homogenous pool with six private RAID groups. For simplicity, a single LUN is bound to the pool. Further assume the I/O mix is 50 percent random reads and 50 percent random writes with a total host IOPS of 10,000:

IOPS = (0.5 * 10,000 + 2 * (0.5 * 10,000))

IOPS = 15,000

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

- **Parity RAID 5:** Drive MB/s = Read MB/s + Write MB/s * (1 + (1/ (number of user data drives in group)))
- **Parity RAID 6:** Drive MB/s = Read MB/s + Write MB/s * (1 + (2/ (number of user data drives in group)))
- **Mirrored RAID 1/0:** Drive MB/s = Read MB/s + Write MB/s * 2

For example, the default private RAID group of a RAID 5 pool is 5-drive 4+1 (four user data drives in group). Assume the read load is 100 MB/s, and write load is 50 MB/s:

Drive MB/s = 100 MB/s + 40 MB/s * (1 + (1/4))

Drive MB/s = 150 MB/s

Determine the number of drives required for Performance

Make both a performance calculation to determine the number of drives in the storage system.

Divide the total IOPS (or bandwidth) by the per-drive IOPS value provided in Table 9 for small-block random I/O and **Table 29** for large-block random I/O.

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 16 KB (up to 32 KB), but less than 64 KB, increase the drive count by 20 percent. Random I/O with a block size greater than 64 KB must address bandwidth limits as well. This is best done with the assistance of an EMC USPEED professional.

Determine the number of drives required for Capacity

Calculate the number of drives required to meet the storage capacity requirement.

Typically, the number of drives needed to meet the required capacity is fewer than the number needed for performance.

Remember, the formatted capacity of a drive is smaller than its raw capacity. Add the capacity required for a Virtual provisioning pool to maintain the pool's file system. This is the pool's *metadata overhead*.

Furthermore, the system drives require four drives, and it is prudent to add one hot spare drive per 30 drives (rounded to the nearest integer) to the drive count. Do not include the system drives and hot spare drives into the performance calculation when calculating the operational performance.

Analysis

Ideally, the number of drives needed for the proposed I/O load is the same as the number of drives needed to satisfy the storage capacity requirement. Use the *larger* number of drives from the performance and storage capacity estimates for the storage environment.

Total performance drives

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

For example, if an application was previously calculated to execute 4,000 IOPS, the I/O is 16 KB random requests, and the hard drives specified for the group are 15K RPM SAS drives (see Table 9 Small block random I/O performance by drive type):

Total Approximate Drives = 4,000 / 180 + 0 + ((4,000 / 180) / 30) + 5

Total Approximate Drives = 28

Calculate the 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 supplying performance, capacity, and value to the client.

Select the storage system whose drive count best fits the client's requirement. Divide the number of drives by the maximum drive counts for VNX Table 31 to determine the number of storage systems necessary to service the required drives effectively.

For best IOPS performance, the optimal SAS drive count for the storage system is shown inTable 38. Use the maximum drive count for the storage system when considering entirely NL-SAS drive installations. Divide the required number of drives by the drive counts in Table 38 to determine the type and number of storage systems to deploy. High-performance storage systems require a more complex analysis. They are not covered in this simple example. Contact your EMC Sales representative to engage an EMC USEED professional for information regarding high-performance and high-availability storage system configuration.

Storage systems

Storage Systems = Approximate Drives / Storage System Drive Maximum Count

For example, if a high-performance storage system requires approximately 135 drives:

Number of Storage Systems = 135 / 250
Number of Storage Systems = 1 (VNX5500 storage solution)
Resolving performance and capacity needs

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

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

Sizing example: homogenous pool

The following example uses the procedure described above to create a Virtual Provisioning pool solution with a homogeneous pool for a hypothetical workload.

Step 1: Determine the workload

Having an accurate description of the workload is the first, most important step.

The following example includes the minimum workload requirements needed to perform a sizing estimate. Any error or ambiguity in the workload adversely affects the sizing calculations and may result in a system unsuitable for the client's needs.

Workload Information

A client has a storage requirement for a storage environment. The application is expected to service the LUNs shown in the table below. Moderate performance with ease-of-use are expected of the storage environment. Thick LUNs in a Virtual Provisioning pool are the favoured solution.

LUN Name	Capacity (GB)	Host IOPS	Read/Write Ratio
Alpha	5000	800	4:1
Bravo	1000	1250	2:1
Charlie	500	800	2:1
Delta	100	400	Write Only
Foxtrot	50	1000	Read Only
Total Capacity:	6650		

Table 31, Workload Component LUNs Capacity and Performance Requirements

Metadata Overhead

A Virtual Provisioning pool-based solution increases the capacity of the user LUNs.

Increase the needed capacity of the pool based on the number of LUNs and the metadata over head of the pool to ensure that the initial capacity of the pool can be met.

For example:

LUN Alpha: 5103 GB = (5000 GB * 1.02) + 3 GB)

LUN Name	User LUN Capacity (GB)	Pool Capacity (GB)
Alpha	5000	5103
Bravo	1000	1023
Charlie	500	513
Delta	100	105
Foxtrot	50	54
Total Capacity:	6650	6798

Table 32, LUN Capacities Adjusted for Metadata Overhead

For this workload, the metadata overhead of the provisioned LUNs is about an additional 150 GB. Candidate minimum pool capacities should be about 6800 GB.

Step 2: Determine the I/O drive load

Calculate the IOPS for the three RAID level types available for pools: RAID 5, RAID 6, and RAID 1/0.

For example for LUN Alpha the drive load is calculated as follows:

RAID 5 (4+1): 0.8 * 800 + 4 * 0.2*800 = 1280 drive IOPS

RAID 6 (4+2): 0.8 * 800 + 6 * 0.2*800 = 1600 drive IOPS

RAID 1/0 (4+4): 0.8 * 800 + 2 * 0.2*800 = 960 drive IOPS

LUN Name	RAID 5 Pool IOPS	RAID 6 Pool IOPS	RAID 1/0 Pool IOPS
Alpha	1280	1600	960
Bravo	2500	3333	1667
Charlie	1600	2133	1067
Delta	1600	2400	800
Foxtrot	1000	1000	1000
Total Disk IOPS	7980	10467	5493

Table 33, Workload Homogenous pool Drive I/O Loads

The RAID level that satisfies the workload with the fewest number of IOPS is generally the best choice. From an IOPS perspective, RAID 1/0 would be the choice in this step. RAID 1/0 has the lowest per drive I/O workload.

Step 3: Determine the number of drives required for Performance

Calculate the number of hard drives for each RAID type needed.

Rule of Thumb I/O performance and capacity by drive	IOPS	Formatted Capacity (GB)
300 GB SAS 15K rpm	180	268
300 GB SAS 10K rpm	150	
2 TB NL-SAS 7.2K rpm	90	1640
200 GB Flash drive*	3500	183

*VNX-series only.

Table 34, Example Rule-of-Thumb Drive Characteristics

For example, assuming 15,000 rpm SAS drives with hot spares and system drives included in the total. The performance capacity is the IOPS divided by the drive IOPS. Use the Total disk IOPS from Table 38and divide it by the IOPS per drive type from Table 34. Add Hot Spares and system drives to the total.

RAID 5: 7980/180 + ((7980/180)/30) +4 = 51 drives total

RAID 6: 10468/180 + ((10468/180)/30) + 4 = 65 drives total

RAID 1/0: 5493/180 + ((5493/180)/30) + 4 = 37 drives total

Note that NL-SAS and Flash drives cannot be used as system drives for the VNX with OE Block 31.0. SAS drives would need to be substituted for the system drives.

Drive type	RAID 5 Pool (Drives)	RAID 6 Pool (Drives)	RAID 1/0 Pool (Drives)
SAS 15K rpm	51	65	37
SAS 10K rpm	60	77	43
NL-SAS 7.2K rpm	96	125	69
Flash drive	8	8	7

Table 35, Workload Total Number of Drives IOPS, System, & Hot Spares

Typically, the provisioning using the fewest number of drives is the best choice. From a number of drives perspective, RAID 1/0 would be the choice in this case. RAID 1/0 provides the needed IOPS with the fewest number of drives.

Step 4: Determine the pool capacity

The number of drives needed to achieve performance needs to be resolved with the available number of drives needed to meet the storage capacity requirement. This calculation is performed using data drives only. Do not include vault and hot spare drives in the calculation. Use the formatted capacity from Table 33, Workload Homogenous pool Drive I/O Loads multiplied by the number of data drives in the performance estimate. Reduce the capacity of the data drives based on their level of data protection.

Assume the 15K rpm 300 GB SAS drives with a formatted capacity of 268 GB.

RAID 5 (4+1): 4/5 * (7980 IOPS/180 IOPS) * 268 GB = 9505 GB

RAID 6 (6+2): 6/8 * (10467 IOPS/180 IOPS) * 268 GB = 11688GB

RAID 1/0 (4+4): ½ * (5493 IOPS/180 IOPS) * 268 GB = 4089 GB

Drive type	RAID 5 Pool (GB)	RAID 6 Pool (GB)	RAID 1/0 Pool (GB)
300 GB SAS 15K rpm	9505	11688	4089
300 GB SAS 10K rpm	11406	14025	4907
2 TB NL-SAS 7.2K rpm	116331	143044	50050
200 GB Flash drive	334	410	144

Table 36, Workload Number of Drives Pool Capacity

Typically the capacity related number of drives that is closest to the required pool capacity (Table 32, LUN Capacities Adjusted for Metadata Overhead) is the best choice. From a capacity perspective accounting for performance, RAID 5 would be the choice in this substep. RAID 5 provides the needed capacity with the fewest number of drives.

Note that it is strongly recommended that pools containing NL-SAS drives, particularly pools with a large number of private RAID groups be implemented as RAID 6 storage.

Users concerned with high availability storage solutions should include this consideration in their provisioning plans.

The number of drives in the pool needs to be adjusted to have symmetrical RAID groups within the pool. That is, all the private RAID groups in the pool need to have the same number of drives. Typically, this is the default private RAID group size for the pool's level-of-protection.

For example, the RAID 5 Pool with 300 GB SAS 15K RPM drives results in 44 (7960 IOPS/ 180 IOPS) drives needed for performance. This is not an even multiple of the 5-drive (4+1) private RAID groups created by Virtual Provisioning. An additional drive is added to create an evenly divisible 45 drive total. That results in an even nine (4+1) private RAID groups for the pool.

The previous steps result in the table below.

Drive type	RAID 5 Pool (Drives)	RAID 6 Pool (Drives)	RAID 1/0 Pool (Drives)
300 GB SAS 15K rpm Data Drives	45 (9x (4+1))	56 (7x (6+2))	
300 GB SAS 15K rpm Hot Spares	1	1	
2 TB NL-SAS 7.2K rpm Data Drives			64 (8x (4+4))
2 TB NL-SAS 7.2K rpm Hot Spares			2
300 GB SAS 15K rpm System Drives	4	4	4
Total Drives:	50	61	70

Table 37, Workload Drives: Homogenous Pool SAS or NL-SAS w/ Thick LUNs

Step 5: Determine the number and type of storage systems

This table is used to compare the homogeneous pool options toward making a storage system selection.

Pool RAID type	Drive Load (IOPS)	Storage Capacity (GB)	Total Drives
RAID 5 w/ 300 GB 15K rpm SAS	7980	9648	50
RAID 6 w/ 300 GB 15K rpm SAS	10467	11256	61
RAID 1/0 w/ 2 TB NL-SAS	5493	52480	70

Table 38 Workload Homogeneous Pool Sizing estimate calculation results

Determine which VNX model most closely matches the calculated performance capacity, storage capacity, total drives, and stated requirements. Use ***Note that** mechanical drive write performance is somewhat higher use these intentionally conservative values for ROM estimates.

Table 30 'Sweet-Spot' by model, VNX OE Block 31.0. From the table above, the largest pool is about 60 drives. This number of drives is easily hosted by the VNX5100 model.

Step 6: Analysis

Generally, the storage solution using the fewest number of drives that meets the performance and capacity requirements is the best solution.

In this example, a VNX5100 with RAID 5 would be a good candidate. This reasonable solution would be 45 drives for data, one hot spare, and four drives for the system drives. Should the pool expand, or additional pools and workloads be added, a larger model VNX would be needed.

Sizing example: FAST VP pool

The following example uses the procedure described above to create a Virtual Provisioning pool using the FAST VP feature to create solution for a hypothetical workload.

To create tiered pools, FAST Virtual Provisioning must be enabled. FAST VP is an optionally licensable product.

Considering Locality

To leverage the FAST VP feature, locality of reference information is needed. The locality information is used to estimate the division of drive capacity and I/O load between the tiers.

Be aware that the quality of the locality information within the workload is important. By adopting a tiered solution, you are introducing some variability in the host response time. This variability is based on the difference in performance between two or more types of storage devices. Locality is a statement on the statistical distribution of your most frequently accessed capacity to infrequently accessed capacity. The most frequently accessed data goes on the highest performing storage devices. Occasionally, less frequently accessed data will be read or written. This I/O will have a longer host response time than more frequently accessed data located on higher performing storage devices. The accuracy of your locality data determines how variable the host response time shall be. The more accurate it is, the lower the average host response time.

Consult with your storage architects and application analysts to ensure the locality information's accuracy. A misstep in the allocation may result in a tier without enough capacity, throughput, or bandwidth to service the workload.

Step 1: Determine the workload

The following example uses the data from the previous example with extensions.

Assume that a two-tiered FAST VP pool is the favoured solution. Assume the tiers will be SAS drive and NL-SAS drive based.

Either the 600 GB or 300 GB capacity SAS drives may be used. In addition, either 15K RPM or 10K RPM SAS drives may be selected. The 300 GB 15K RPM SAS drives are used in the example. Although, the other capacity or speed drive can easily be substituted into the calculations.

LUN Name	Capacity (GB)	Host IOPS	Read/Write Ratio	Percent Locality
Alpha	5000	800	4:1	10
Bravo	1000	1250	2:1	5
Charlie	500	800	2:1	5
Delta	100	400	Write Only	30
Foxtrot	50	1000	Read Only	100
Pool Metadata	148			
Total Capacity:	6798			

Table 39, Workload Component LUNs Capacity, Performance, and Locality Requirements

Step 2: Determine the required tier Capacity of the top tier

Calculate the minimum capacity and performance (IOPS) for the highest performing tier. Do not include the Pool Metadata in this calculation. In this case the 'Top Tier', will be the 15K rpm SAS drives, while the 'Bottom Tier' will be the NL-SAS drives.

For example, LUN Alpha estimated capacity usage of top tier:

5000 GB * 0.1 = 500 GB

The calculations result in the table below.

LUN Name	Top Tier Capacity (GB)	Bottom Tier Capacity (GB)
Alpha	500	4500
Bravo	50	950
Charlie	25	475
Delta	30	70
Foxtrot	50	0
Tier Totals:	655	5995

Table 40, Workload 2-Tier FAST VP Pool Tier Capacity

The total capacity of the upper most tier is modest. About 10-percent (665 GB / 6650 GB) of the pool is most frequently used.

Both a RAID 5 and a RAID 1/0 RAID group have a user capacity of about 1072 GB (4 * 268 GB). A RAID 6 group has a user capacity of about 1608 GB (6 * 268 GB). Capacity-wise, a single 300 GB SAS drive RAID group in the highest tier will handle the capacity.

Both the RAID 5 and 1/0 NL-SAS RAID groups have a capacity of 6560 GB. The RAID 6 group has a capacity of 9840 GB. For the Bottom tier, a single NL-SAS RAID group of any RAID level will handle the remaining capacity of the pool.

Step 3: Determine the required tier I/O drive load of the top tier

The Pool performance next needs to be allocated. These calculations were previously performed and are captured in Table 33, Workload Homogenous pool Drive I/O Loads. However, the performance needs to be adjusted to account for the locality.

Top tier performance

The most conservative approach is to assume the greater portion of the performance will be handled by the Top tier. Note that not all of the Host IOPS need be handled by the highest tier. The lower tier has a margin of performance which can be productive too.

The RAID 5 calculations are as follows using Table 33, Workload Homogenous pool Drive I/O Loads and Table 39, Workload Component LUNs Capacity, Performance, and Locality Requirements's locality data. Note how the locality allocates the majority of IOPS to the top tier.

Alpha LUN: 1280 IOPS * .9 = 1152 IOPS

Bravo LUN: 2488 IOPS * .95 = 2375 IOPS

Charlie LUN: 1592 IOPS * .95 = 1520 IOPS

Delta LUN: 1600 IOPS * .7 = 1120 IOPS

Foxtrot LUN: 1000 IOPS * 1.0 = 1000 IOPS

The table below summarizes the calculations.

LUN Name	Top Tier Disk IOPS RAID 5	Top Tier Disk IOPS RAID 6	Top Tier Disk IOPS RAID 1/0
Alpha	1152	1440	864
Bravo	2375	3167	1583
Charlie	1520	2027	1013
Delta	1120	1680	560
Foxtrot	1000	1000	1000
Tier Totals:	7167	9313	5021

Table 41, Workload 2-Tier FAST VP Pool Top Tier Performance

Bottom tier performance

A check should be performed to ensure the bottom tier will be able to sustain the IOPS load that it may occasionally be required of it.

Using the fully loaded IOPs for a homogenous pool from Table 33, Workload Homogenous pool Drive I/O Loads, subtract the FAST VP pool Top tier to get an estimate of the IOPS load on the bottom tier.

Pool IOPS Estimate	RAID 5 Pool	RAID 6 Pool	RAID 1/0 Pool
Homogenous Pool IOPS	7980	10467	5493
Top Tier FAST VP Pool IOPS	7167	9313	5021
Estimate of Bottom Tier IOPS	812	1152	473

Table 42, FAST VP Pool Bottom Tier IOPS Estimate

A RAID 5 and 1/0 NL-SAS default Virtual Provisioning private RAID group have about 360 IOPS. A RAID 6 group has 450 IOPS. Note that a single NL-SAS RAID group in the bottom tier for the three RAID levels have low IOPS. This will result in high host response times for data on this tier.

Performance drive estimate

Using the ROT data from *VNX-series only.

Table 34, Example Rule-of-Thumb Drive Characteristics, calculates the number of drives needed for each of the RAID levels to satisfy the Top tier's performance requirements. The performance capacity is the IOPS divided by the drive IOPS.

RAID 5 (4+1): 7167/180 = 40 drives total

RAID 6 (6+2): 9313/180 = 52drives total

RAID 1/0 (4+4): 5021/180 = 28drives total

Provision the bottom tier with the NL-SAS RAID group. Round the number drives to create default private RAID groups, adding hot spares and system drives results in the following drive mix.

The table below summarizes the calculations.

Drive type	RAID 5 Pool (Drives)	RAID 6 Pool (Drives)	RAID 1/0 Pool (Drives)
300 GB SAS 15K rpm Data Drives	40 (8x (4+1))	56 (7x (6+2))	32 (4x (4+4))
300 GB SAS 15K rpm Hot Spares	1	1	1
2 TB NL-SAS 7.2K rpm Data Drives	5 (1x (4+1))	8 (1x (6+2))	8 (1x(4+4))
2 TB NL-SAS 7.2K rpm Hot Spares	1	1	1
300 GB SAS 15K rpm System Drives	4	4	4
Total Drives:	51	70	46

Table 43, Workload Drives: FAST VP SAS/NL-SAS Pool w/ Thick LUNs

Step 5: Analysis

This table is used to compare the FAST VP Pool options toward making a storage system selection.

Pool storage capacity is calculated by summing the capacity of the usable capacity of the data drives in the tiers. Using the data from Rule of Thumb I/O performance and capacity by drive table as follows:

RAID 5: ((4/5) * 40 drives * 268 GB/drive) + ((4/5) * 5 drives * 1640 GB/drive) = 15136 GB

RAID 6: ((6/8) * 56 drives *268 GB/drive) + ((6/8) * 8 drives * 1640 GB/drive) = 21096 GB

RAID 1/0: ((1/2) * 32 drives *268 GB/drive) + ((1/2) * 8 drives * 1640 GB/drive) = 10848 GB

The table below summarizes the capacity, performance calculations and drive estimates.

FAST VP Pool RAID type	Top Tier Drive Load (IOPS)	Pool Storage Capacity (GB)	Total Drives
RAID 5	7167	15136	51
RAID 6	9313	21096	70
RAID 1/0	5021	10848	46

Table 44 Workload FAST VP Sizing estimate calculation results

Generally, the storage solution using the fewest number of drives that meets the performance and capacity requirements is the best solution.

Use ***Note that** mechanical drive write performance is somewhat higher use these intentionally conservative values for ROM estimates.

Table 30 'Sweet-Spot' by model, VNX OE Block 31.0 to determine which VNX model most closely matches the calculated performance capacity, storage capacity, total drives, and stated requirements.

In this example, a VNX5300 with RAID 1/0 would be a good candidate. This reasonable solution would be 40 drives for data, two hot spares, and four drives for the system drives. Should the pool expand, or additional pools and workloads be added, a larger model VNX would be needed.

Caveats

The number of drives in the choice between using 15K rpm SAS drives and NL-SAS drives in RAID 5 versus in RAID 1/0 is close. The RAID 1/0 FAST VP pool solution most closely fits the requirements and uses fewer drives. In addition, the bottom tier's performance is compatible with the expected load.

However, the RAID 5 FAST VP pool solution has more capacity and IOPs in the top tier. Careful consideration should be applied to the reliability of the locality information (Table 39, Workload Component LUNs Capacity, Performance, and Locality Requirements). Choosing the RAID 5 FAST VP pool with its additional IOPs may be more prudent than the more economical RAID 1/0 FAST VP pool.

Sizing example: FAST VP pool with FAST Cache

The following example uses the procedure above to size a workload for a two tiered pool supplemented by a FAST Cache.

To create a FAST Cache, it must be enabled. FAST Cache is an optionally licensable product.

The Hit Rate

The FAST Cache works to reduce the load on the storage devices. It directly reduces the IOPS serviced by mechanical hard drives. There is the potential to reduce the number of drives in a homogenous pool, or 'thin', reduce the number of drives, in the higher performing tiers of a

FAST VP pool through its use. It may also be used to reduce the total number of drives in a traditional LUN.

The FAST Cache feature leverages the same locality of reference information that is needed for a FAST VP pool estimate. Estimate the capacity of the FAST Cache based on the ratio of the high locality capacity to total pool capacity. This ratio is the estimated 'cache hit rate'.

For the purposes of ROM estimates, do *not* assume greater than a 50-percent hit rate.

Step 1: Determine the workload

The following example uses the data from the previous FAST VP example with extensions.

Size the FAST VP Pool

Assume that a two-tiered FAST VP pool is the favoured solution. The tiers will be 300 GB 15K RPM SAS drive and NL-SAS drive based. Use the workload information found in Table 39, Workload Component LUNs Capacity, Performance, and Locality Requirements.

Use the tier-to-tier capacity allocation found in Table 40, Workload 2-Tier FAST VP Pool Tier Capacity. The table shows that 655 GB of data has high locality. This is and is the candidate capacity for a FAST Cache usage.

Size the FAST Cache

FAST Cache is implemented as a RAID 1. See "Rule of Thumb I/O performance and capacity by drive for capacities. A four drive 200 GB flash drive FAST Cache results in a FAST Cache with about 366 GB. The cache works with the high locality capacity assumed to be found in the Top tier. That would result in a 56-percent (366 GB/655 GB) potential for I/O to be handled by the FAST Cache. For the purpose of the calculation, a more conservative 50-percent hit rate will be assumed.

Note that the VNX5100 does not support the 4x 200 GB flash drives in a FAST Cache configuration, although the VNX5300 does.

Adjust the host IOPS served by mechanical drives by reducing them by the FAST Cache hit rate percentage. The table below summarizes the calculations performed.

LUN Name	Capacity (GB)	Host IOPS	Read/Write Ratio	Percent Locality	Host IOPs after FAST Cache
Alpha	5000	800	4:1	10	400
Bravo	1000	1250	2:1	5	625
Charlie	500	800	2:1	5	400
Delta	100	400	Write Only	30	400
Foxtrot	50	1000	Read Only	100	500
Pool Metadata	148				
Total Capacity:	6798				

Table 45, FAST VP Pool w/ FAST Cache Workload Component LUNs Capacity, Performance, and Locality Requirements

FAST Cache applies to all the LUNs in a pool. Note that the Write Only LUN Delta is unaffected by the FAST Cache. The FAST Cache is of no benefit for its I/O type. Best Practice is to omit this type of LUN from the pool. To simplify the example, it's full host IOPS are credited to the workload..

Step 2: Determine the required tier Capacity of the top tier

The tier capacity remains the same as in the previous example. See Table 40, Workload 2-Tier FAST VP Pool Tier Capacity.

Step 3: Determine the required tier I/O drive load of the top tier

Host IOPS to the Top Tier has been halved by the addition of the configured FAST Cache.

Top tier performance

The table below summarizes the calculations performed in Step 3 for previous examples using the Host Disk IOPS after FAST Cache column from Table 45.

LUN Name	Top Tier Disk IOPS RAID 5	Top Tier Disk IOPS RAID 6	Top Tier Disk IOPS RAID 1/0
Alpha	576	720	432
Bravo	1188	1583	792
Charlie	760	1013	507
Delta	1120	1680	560
Foxtrot	500	500	500
Tier Totals:	4144	5497	2790

Table 46, Workload 2-Tier FAST VP Pool Top Tier Performance

Bottom tier performance

A check should be performed to ensure the bottom tier will be able to sustain the IOPS Load.

Calculate the Bottom Tier IOPS by halving the IOPS to account for FAST Cache as in Table 42, FAST VP Pool Bottom Tier IOPS Estimate.

Pool IOPS Estimate	RAID 5 Pool	RAID 6 Pool	RAID 1/0 Pool
Estimate of Bottom Tier IOPS	323	468	178

Table 47, FAST VP Pool Bottom Tier IOPS Estimate

Note that a single NL-SAS RAID group in the bottom tier for all the pools should have enough IOPS.

Performance drive estimate

Using the ROT data from Table 46, Workload 2-Tier FAST VP Pool Top Tier Performance calculate the number of drives needed for each of the RAID levels to satisfy the pool's performance and capacity requirements.

Round the number of drives to create default private RAID groups, adding hot spares and system drives results into the drive mix.

The table below summarizes the calculations.

Drive type	RAID 5 Pool (Drives)	RAID 6 Pool (Drives)	RAID 1/0 Pool (Drives)
200 GB flash FAST Cache Drives	4 (2x (1+1))	4 (2x (1+1))	4 (2x (1+1))
200 GB flash Hot Spares	1	1	1
300 GB SAS 15K rpm Data Drives	25 (5x (4+1))	32 (4x (6+2))	16 (2x (4+4))
300 GB SAS 15K rpm Hot Spares	1	1	1
2 TB NL-SAS 7.2K rpm Data Drives	5 (1x (4+1))	8 (1x (6+2))	8 (1x(4+4))
2 TB NL-SAS 7.2K rpm Hot Spares	1	1	1
300 GB SAS 15K rpm System Drives	4	4	4
Total Drives:	41	51	35

Table 48, Workload Drives: FAST VP w/FAST Cache SAS/NL-SAS Pool w/ Thick LUNs

Step 5: Analysis

This table is used to compare the FAST VP Pool with FAST Cache options toward making a storage system selection.

The FAST Cache Load IOPS is calculated by subtracting the reduced top tier IOPS (Table 46, Workload 2-Tier FAST VP Pool Top Tier Performance) from the total top tier IOPS (Table 41, Workload 2-Tier FAST VP Pool Top Tier Performance).

For example:

RAID 5: 7167 IOPS – 4144 IOPS = 3024 IOPS

FAST VP w/ FAST Cache Pool RAID type	FAST Cache Load (IOPS)	Top Tier Drive Load (IOPS)	Pool Storage Capacity (GB)	Total Drives
RAID 5	3024	4144	11920	41
RAID 6	3817	5497	16272	51
RAID 1/0	2230	2790	8704	35

Table 49 Workload FAST VP w/ FAST Cache Sizing estimate calculation results

Generally, the storage solution using the fewest number of drives that meets the performance and capacity requirements is the best solution.

Use ***Note that** mechanical drive write performance is somewhat higher use these intentionally conservative values for ROM estimates.

Table 30 'Sweet-Spot' by model, VNX OE Block 31.0 to determine which VNX model most closely matches the calculated performance capacity, storage capacity, total drives, and stated requirements.

In this example, a VNX5300 with RAID 1/0 would be a good candidate. This choice is driven by the need to support the FAST Cache configuration. This solution would be 24 drives for data, a four flash drive FAST Cache, three hot spares, and four system drives. Should the pool expand, or additional pools and workloads be added, a larger model VNX would be needed.

Caveats

Note how close the choice is between using drives in RAID 5 versus in RAID 1/0. The number of drives is very similar. A careful consideration should be applied to the reliability of the locality and the assumption of the FAST Cache hit rate. An error in these assumptions could result in too few pool IOPS to support the load with the needed host response time.

The RAID 1/0 solution is the better choice economically. However, the RAID 5 FAST VP w/ FAST Cache pool solution uses the fewest drives. In addition, it has almost 150-percent more IOPS and 35-percent more capacity than the RAID 1/0 pool. These IOPS may be needed if an error is made in either the locality or assumption of the hit rate.

Pool sizing summary and conclusion

The previous examples show how to estimate the number of drives and the model storage system for a modest size Virtual Provisioning pool using a:

- Homogenous pool using mechanical hard drives
- 2-Tier FAST VP pool with mechanical hard drives
- 2-Tier FAST VP pool with mechanical hard drives supplemented by FAST Cache

Each example used the same workload. However, the application of the feature, its parameters, and the storage resources used changed the solution. Note that the storage system selection remained the same throughout, a VNX 5100, until the FAST Cache usage required a VNX5300.

The following table summarizes the solutions of the examples given:

Virtual Provisioning Pool Provisioning Summary	Top Tier Drive Load (IOPS)	Storage Capacity (GB)	Total Drives
RAID 5 Homogenous pool w/ 300 GB 15K rpm SAS drives	7980	9648	50
RAID 1/0 FAST VP pool w/ 300 GB 15K rpm SAS & NL-SAS drives	5021	10848	46
RAID 1/0 FAST VP pool & FAST Cache w/ 300 GB 15K rpm SAS & NL-SAS drives	2790	8704	35

Table 50 Virtual Provisioning Pool Combined Sizing Examples Results

In each case, with the use of Virtual Provisioning pool-based storage, as a feature was added, the number of drives needed in the pool decreased. This was due to the use of different drive types within or in addition to the pool meeting the workload's requirements.

For instance, the final example, FAST VP pool & FAST Cache uses three drive types, while the first example, Homogenous pool uses only one.

Homogeneous Pools

Homogeneous pools, or a single-tiered FAST VP pool is the most robust solution. This construct provides the highest and most deterministic performance.

Homogeneous Virtual Provisioning pools are a basic feature of VNX Block OE.

If you are unsure of your workload's locality information, use a homogenous pool. Note that a single tiered FAST VP pool can later be expanded into a multi-tiered pool, as the uncertainty with the workload's requirement's decreases through use of the pool.

Tiered Pools

A multi-tiered FAST VP pool is the most economical drive-wise provisioning. However, those economies are based on knowledge of the workload, particularly assumptions on its locality.

FAST VP is an optional licensed feature of VNX Block OE.

If you have confidence in your workload's performance and locality information use a FAST VP multi-tiered pool for the most economical usage of storage devices.

FAST Cache

A FAST Cache secondary cache may be used to reduce the I/O load on mechanical hard drives. It can be used with either homogeneous or tiered Virtual Provisioning pools. It may also be used with traditional LUNs. Note that a FAST Cache cannot be used with either pool tiers or traditional LUNs made-up of flash drives. Its successful usage, like the FAST VP feature depends on how well suited your workload it is, and your knowledge of the workload's locality.

FAST Cache is an optional licensed feature of VNX Block OE.

Note that not all workloads will benefit from a FAST Cache. If you are unsure of your workload's ability to leverage the FAST Cache use a conservative hit rate.

Workload

It is important to understand that success in provisioning is dependent upon the quality of the workload information. The correctness of the host IOPS and read/to write mix is just the first step. Successfully leveraging Virtual Provisioning requires further information about your data. The correctness of the locality information becomes crucial in the FAST VP and FAST Cache allocation of resources.

Conclusion

The included examples show how to estimate the number of drives needed to estimate the provisioning of a virtual provisioning pool by calculating the number of drives needed for capacity and performance requirements. . In addition, an example of FAST Cache provisioning and its affect on a pool's provisioning has been shown. The examples and the included informational tables may be used to create a ROM estimate of your Virtual provisioning pool-based storage.

For example, the 2-Tier FAST VP pool with mechanical hard drives example could easily be changed to use flash drives in the Top Tier.

The use of the Virtual Provisioning feature (Homogeneous Pools), FAST VP, and FAST Cache can create economies in storage provisioning given accurate workload information.

The advantage of applying the Virtual Provisioning features and the potential savings can be evaluated ahead of time by using the described methodology. You should be able to make a ROM estimate based on your workload of a Virtual Provisioning storage solution.

EMC USPEED personnel have software tools to determine a more exact storage system provisioning. These tools take into account the numerous factors affecting final capacity and performance of VNX storage systems. In addition, they can assist in provisioning high-performance and high availability configurations. Contact your EMC Sales Representative for information engaging a USPEED professional for assistance.

.

Glossary of Terms

"When I use a word, it means just what I choose it to mean -- neither more nor less."

-- Humpty Dumpty, "Through the Looking Glass (And What Alice Found There)" (1871) by Lewis Carroll

10 GbE— 10 Gigabit per second Ethernet protocol.

ABQL — Average busy queue length (per drive).

Active data — Working data set being addressed by an application.

Active-active — Redundant components are active and operational.

Active-passive — Redundant components are ready and in a standby operational mode.

AFR — Annual failure rate.

ALUA — Asymmetric logical unit access protocol.

Allocated Capacity — Total physical capacity currently assigned to pool-based LUNs

American National Standards Institute — An internationally recognized standards organization.

ANSI — American National Standards Institute.

Application software — A program or related group of programs performing a function.

Array — Storage system.

Asymmetric Logical Unit Access — An industry-standard multipathing protocol.

Attachment — Drive hardware connector or interface protocol. On a CLARiiON it can be Fibre Channel, SAS, or SATA.

Authentication — Verifying the identity of a communication to ensure its stated origin.

Authorization — Determining if a request is authorized to access a resource.

Automatic Volume Management — A VNX File feature providing for automated file system creation.

Available capacity — Capacity in a thin LUN pool that is not allocated to thin LUNs. .

Availability — Continued operation of a computer-based system after suffering a failure or fault.

AVM— Automatic Volume Management. .

Back-end — A logical division of the VNX's architecture from SP to the back-end bus(es) and drives.

Back-end bus — The VNX's SAS back-end ports that connect the storage processors to the drives.

Back-end I/O — I/O between the storage processors and the drives over the back-end buses.

Background Verify — Automated reading of the LUN's parity sectors and verification of their contents by the CLARiiON for fault prevention.

Backup — Copying data to a second, typically lower performance, drive as a precaution against the original drive failing.

Bandwidth — A measure of storage-system performance, as measured in megabytes per second (MB/s).

BBU — Battery Backup Unit.

Best practice — A specific type of professional or management activity that contributes to the optimal execution of a process.

and that may employ one or more tools and techniques..

Bind — To combine drives into a RAID group.

Bit — The smallest unit of data. It has a single binary value, either 0 or 1.

Block — The smallest addressable unit on a hard drive; it contains 512 bytes of data.

Bottleneck — A resource in a process that is operating at maximum capacity; the bottleneck causes the whole process to slow down.

Buffering — Holding data in a temporary area until other devices or processes are ready to receive and process the data; this is done to optimize data flow.

BURA — Backup, Recovery, and Archiving data storage security domain.

Bursty — When, over time, the volume of I/O is highly variable, or regularly variable with well-defined peaks.

Bus — An internal channel in a computerized system that carries data between devices or components.

Busy hour — The hour of the day in which the most I/O occurs.

BV — Background Verify.

Byte — Eight computer bits.

Cache — Memory used by the storage system to buffer read and write data and to insulate the host from drive access times.

Cache hit — A cache hit occurs when data written from or requested by the host is found in the cache, avoiding a wait for a drive request.

Cache miss — Data requested by the host is not found in the cache so a drive request is required.

Cache page size — The capacity of a single cache page.

CAS — Content Addressable Storage object-based storage as implemented by EMC Centera®.

Celerra— Name of the legacy file storage system.

Celerra Network Server — Oofficial name of the Celerra product, referred to both the and NS-* product lines.

CBFS — Common Block File System.

CHAP — Challenge Handshake Authentication Protocol.

CIFS — Common Internet File System.

CLI — Command Line Interface.

Client — Part of the client/server architecture, the client is a user computer or application that communicates with a host.

Client/Server — A network architecture between consumers of services (Clients) and providers (Hosts).

Clone — An exact copy of a source LUN.

CMI — Configuration Management Interface.

Coalescing — Grouping smaller cached I/Os into a larger I/O before it is sent to the drives.

Command line interface — An interface that allows a user use text-based commands to communicate with an application or operating system.

Common Block File System — File system of CLARiiON pool-based LUNs.

Common Internet File System — Microsoft Windows file sharing protocol.

Component — A constituent part, element, or piece of a complex whole.

Concurrent I/O — When more than one I/O request is active at the same time on a shared resource.

Configuration Management Interface — Used for peer to peer storage processor communications.

Consumed capacity — Total of capacity in use or reserved by a pool-based LUNs in a pool.

Control — The technique for comparing actual performance with planned performance, analyzing variances, assessing trends to effect process improvements, evaluating possible alternatives, and recommending appropriate corrective action as needed.

Core — A processor unit co-resident on a CPU chip.

Core-switch — Large switch or director with hundreds of ports located in the middle of a SAN's architecture.

CPU — Central Processing Unit.

Criteria — Standards, rules, or tests on which a judgment or decision can be based, or by which a product, service, result, or process can be evaluated.

DAE — Drive array enclosure.

DART — Data Access in Real Time.

DAS — Direct attached storage.

Data — Information processed or stored by a computer.

Data Access in Real Time. — Legacy Celerra operating environment.

Data center — A facility used to house computer systems, storage systems, and associated components.

Data link — Digital communications connection of one location to another.

Data mining application — A database application that analyzes the contents of databases for the purpose of finding patterns, trends, and relationships within the data.

Data Mover — A VNX enclosure component running VNX Operating Environment File.

Data warehouse — A collection of related databases supporting the DSS function.

DBMS—Data Base Management System.

Decision support system — A database application, used in the "data mining" activity.

Degraded mode — When continuing an operation after a failure involves a possible loss of performance.

Departmental system — A storage system supporting the needs of a single department within a business organization.

Destage — Movement of data from cache to drives.

Dirty page — A cache page not yet written to storage.

Disk array enclosure — The rack-mounted enclosure containing a maximum of 15 CLARiiON drives.

Disk controller — The microprocessor-based electronics that control a hard drive.

Disk crossing — An I/O whose address and size cause it to access more than one stripe element in a disk, resulting in two back-end I/Os instead of one.

Disk processor enclosure — The cabinet that contains the CLARiiON storage processor and drives.

Disk volume — VNX file system physical storage unit as exported from the storage system.

DLU — Direct LUN, also known as a Virtual Provisioning Pool Thick LUN.**DPE** — Disk processor enclosure.

DR — Disaster recovery.

Drive — A hardware component from which you can read and write data. Typically a hard drive, but also an Flash drive.

Dump — Copying the contents of the cache to the vault.

Edge switch —A Fibre Channel switch located on the perimeter of a core-edge configured SAN.

EFD — Enterprise Flash Drive.

Enterprise Flash Drive — EMC term for SSD-type drive.

Enterprise system — A storage system supporting the needs of an entire business organization.

Environment — A computer's hardware platform, system software, and applications.

Equalization — Copying data from a hot spare to drive that is replacing a failed RAID group's drive.

ESM — EMC Support Matrix.

ESX — VMware enterprise-level server virtualization product.

Estimate — The quantitative assessment of a likely amount or outcome. Usually applied to cost, resource usage, and durations.

Ethernet — A technology for high-speed bandwidth connectivity over local area networks. The IEEE 802.3 standard.

Failure — A malfunction of hardware component(s) in a system.

Fail back — Restoring the original data path after correcting a failure.

Fail over — Using an alternate path because the original path fails.

Failure mode — The cause of a failure, or the event that starts a process that results in a failure.

Fan-in — Attaching numerous hosts to a few storage-system ports.

Fan-out — Attaching a few storage-system ports to many hosts.

FAQ — Frequently Asked Question.

FAST — Fully Automated Storage Tiering.

FAST Cache — Secondary I/O cache composed of Flash drives.

Fault — An error in the operation of a software program.

Fault tolerance — The ability of a system to continue operating after a hardware or software failure.

FC — Fibre Channel.

FCoE — Fibre Channel traffic over Ethernet.

FCP — SCSI Fibre Channel Protocol.

Fibre Channel — A serial data transfer protocol: ANSI X3T11 Fibre Channel standard.

File — A collection of data.

File Mapping Protocol — A component of the File MPFS stack.

File storage pool — A grouping of disk volumes used to allocate available storage to VNX file-based storage systems.

File system — The system an OS uses to organize and manage computer files.

Filer — NAS fileserver accessing shared storage using a file-sharing protocol.

FLARE — Fibre Logic Array Runtime Environment. Legacy CLARiiON's operating system name.

Flash drive — Solid state disk storage device.

FLU —FLARE LUN, now known as a Traditional LUN.

Flush — Writing the data in the write cache to the drives.

FMP — File Mapping Protocol.

Forced flush — The high priority writing of data to drives to clear a full write cache.

Front end — A logical division of the storage systems architecture, including the communications ports from hosts to the SP.

GB — Gigabyte.

GB/s — Gigabytes per second.

Gb/s — Gigabits per second.

GbE— Gigabit Ethernet (1 Gb/s Ethernet).

GHz — Gigahertz.

Gigabit — One thousand million bits.

Gigabyte — One billion bytes or one thousand megabytes.

Gigahertz — One billion times per second (1,000,000,000 Hz).

GigE — 1 Gb/s Ethernet.

GMT — Greenwich Mean Time.

Graphical user interface — Interface that allows you to communicate with a software program using visual objects on a monitor.

GUI — Graphical user Interface.

HA — High Availability or Highly Available.

HBA — Host Bus Adapter; A device acting as a bridge between the host system bus and the storage system.

Head crash — A catastrophic hard drive failure where a read/write head makes physical contact with a platter.

Hertz — Once per second.

Highly available — A system able to provide access to data when the system has a single fault.

Host — A server accessing a storage system over a network.

Hot spare — A spare drive that the storage system can use to automatically replace a failed drive.

HP-UX — A proprietary Hewlett-Packard Corporation version of the UNIX OS.

HVAC — Heating, Ventilation, and Air Conditioning.

Hz — Hertz.

IEEE — Institute of Electrical and Electronics Engineers.

IETF — Internet Engineering Task Force

iFCP — Protocol allowing FC devices to use an IP network as a fabric switching infrastructure.

ICA — Image Copy Application.

IETF — Internet Engineering Task Force.

IEEE — Institute of Electrical and Electronics Engineers.

iFCP — Protocol allowing FC devices usage of an IP network as a fabric switching infrastructure.

IHAC — I Have A Customer.

Initiators — iSCSI clients.

Input — Any item or action, whether internal or external to a process that is required by a process before that process proceeds. May be an output from a predecessor process.

Institute of Electrical and Electronics Engineers — An international standards organization.

International Organization for Standardization — International organization that maintain standards.

Internet Engineering Task Force — International organization standardizing the TCP/IP suite of protocols.

Internet Protocol — A protocol used with TCP to transfer data over Ethernet networks.

IOPS — Input/Output operations Per Second.

IP — Internet Protocol.

IPSec — Internet Protocol Security.

IPv4 — Internet Protocol version 4.

IPv6 — Internet Protocol version 6.

iSCSI — Internet SCSI protocol. A standard for sending SCSI commands to drives on storage systems.

ISL — Interswitch Link. Connects two or more switches in a network.

ISO — International Organization for Standardization.

IT — Information Technology. Also, the department that manages a computer's computer systems.

JBOD — Just a Bunch Of Disks.

KB — Kilobyte.

Kb — Kilobit.

Kb/s — Kilobits per sec.

KB/s — Kilobytes per sec.

Kilobits — One thousand bits.

Kilobyte — One thousand bytes.

LAN — Local Area Network.

Large-block — I/O operations with capacities greater than 64 KB.

Layered Apps — Layered Applications.

Layered Applications — CLARiiON and VNX installed application software.

LBA — Logical Block Address.

LCC — Link Control Card.

Legacy system — An older storage system that does not have the latest hardware and software.

Link — A connection or data path between computer-based devices or devices within a computer.

Link aggregation — Combining separate links that have similar characteristics and the same source and destination into a single virtual link.

Linux — Any of several hardware independent open-systems operating system environments.

Little's Law — The long-term average number of users in a stable system *L* is equal to the long-term average arrival rate, λ, multiplied by the long-term average time a user spends within the system, *W*; or expressed algebraically: $L = \lambda W$.

Load balancing — The even distribution of the data or processing across the available resources.

Local Area Network — A computer network extending over a small geographical area.

Locality — Proximity of LBAs being used by an application within mass storage.

Log — A document or file used to record and describe or denote selected items identified during execution of a process or activity.

Logical Block Address — A mapping of a drive sector into a SCSI block address.

Logical unit number— A SCSI protocol entity, to which I/O operations are addressed.

Logical volume manager — A host-based storage virtualization application such as Microsoft Logical Disk Manager.

Loop— SP A and SP B's shared connection to the same numbered back-end bus.

Lower director— SP A to SP B direct communications bus connection.

LUN — Logical Unit Number.

LVM — Logical Volume Manager.

Maximum Transmission Unit — The largest size packet or frame, specified in bytes, that can be sent in a packet- or frame-based network such as an iSCSI SAN.

MAN — Metropolitan Area Network.

MB — Megabyte.

MB/s — Megabytes per second.

Mb — Megabit.

Mb/s — Megabits per second.

MCM — MPFS Configuration Manager

Mean time between failure — The average amount of time that a device or system goes without experiencing a failure.

Mean time to data loss — A statistical estimate of when a failure will occur that causes a RAID group to lose data.

Mean time to repair— An estimate of the time required to repair a failure.

Media — The magnetic surface of a hard drive's platter used for storing data.

Megabit — One million bits.

Megabyte — One million bytes or one thousand kilobytes.

Megahertz — A million cycles per second (1,000,000 Hz).

Memory Model — Description of how threads interact through memory.

Metadata — Any data used to describe or characterize other data.

MetaLUN — A LUN object built by striping or concatenating multiple LUN objects.

Methodology — A system of practices, techniques, procedures, and rules used by those who work in a discipline.

MHz — Megahertz.

Mirror — A replica of existing data.

MirrorView — CLARiiON and VNX disaster recovery application

MPFS — Multi-Protocol File System

MPFS Configuration Manager — File tool for automatically configuring the MPFS protocol on a client.

MPIO — Microsoft Multi-Path I/O

MR3 Write — The action the VNX RAID engine performs when an entire RAID stripe is collected in the cache and written at one time.

MTBF — Mean Time Between Failures.

MTTDL — Mean Time To Data Loss.

MTTR — Mean Time To Repair.

MTU — Maximum transmission unit

Multipath — The provision for more than one host I/O paths between LUNs.

Multi-Protocol File System — EMC alternative NAS protocol to NFS or CIFS.

Multithread — Concurrent I/O threads.

Name-server — A process translating between symbolic and network addresses, including Fibre Channel and IP.

NAS — Network Attached Storage.

Native Command Queuing — A drive-based I/O execution optimization technique.

Navisphere — CLARiiON's resource management system software for FLARE revisions up to 30.0.

Navisphere Analyzer — CLARiiON's performance analysis system software.

NCQ — Native Command Queuing

Network — Two or more computers or computer-based systems linked together.

Network element — A device used in implementing a network. Typically refers to a switch or router.

Network file system — A UNIX/Linux file sharing protocol.

Network interface card — A host component that connects it to an Ethernet network.

NDU — Non-Disruptive Update

NFS—Network File System

NIC — Network Interface Card.

Nondisruptive update — Upgrading system software while applications are running with minimal effect on the applications' performance.

Non-optimal path — The ALUA failed-over I/O path from host to LUN

OS — Operating System.

OLTP — OnLine Transaction Processing system.

Online transaction processing — Multiuser systems supported by one or more databases handling many small read and write operations.

OOB — Out of Band.

Operating environment — Operating System.

Operating system — The software on a computer that controls applications and resource management.

Optimal path — The normal operations I/O path from host to LUN

Output — A process, result, or service generated by a process. May be an input to a successor process.

Oversubscribed Capacity — Thin LUN configured capacity exceeding provisioned pool capacity.

Ownership — SP management of LUN I/O.

Page — Cache unit of allocation.

Parallel ATA — Disk I/O protocol used on legacy CLARiiONs.

PATA — Parallel ATA disk I/O protocol.

Petabyte — One quadrillion bytes or one thousand terabytes.

PB — Petabyte.

PC — Personal Computer

PCI— Peripheral Component Interface.

PCI Express — A bus protocol used by computer-based systems.

PCIe— PCI Express.

PCI-X— Extended-PCI.

PDU — Power Distribution Unit.

Percentage utilization — A measurement of how much of a resource is used.

Platform — The hardware, systems software, and applications software supporting a system, for example DSS or OLTP.

Platter — A component of a hard drive; it is the circular disk on which the magnetic data are stored.

Pool — A grouping of drives managed by the CLARiiON Virtual Provisioning feature.

Pool LUN — A LUN provisioned on a Virtual Provisioning pool.

Port — An interface device between a storage system and other computers and devices. Also, an interface between a drive and a bus.

Power distribution unit — A CLARiiON component connecting data center electrical power trunks to the storage system.

Powerlink — EMC's password-protected extranet for customers and partners.

PowerPath — EMC host-based multipathing application.

Prefetch — A caching method by which some number of blocks beyond the current read are read and cached in the expectation future use.

Preventative action — Documented direction to perform an activity that can increase availability and decrease the risk of a failure.

Private LUN — A LUN managed by FLARE and not addressable by a host.

Process — A set interrelated actions and activities performed to achieve a specified set of products, results, or services.

Procedure — A series of steps followed in a regular definitive order to accomplish a task.

Protocol — A specification for device communication.

PSM — Persistent Storage Manager.

QA — Quality Assurance.

Quality — The degree to which a set of inherent characteristics fulfils requirements.

Quality Assurance — The department, or policies and procedures for verifying promised performance and availability.

QFULL — Queue Full.

QoR — Quality of Result.

QoS — Quality of Service.

Quality of result — A term used in evaluating technological processes or implementations.

Quality of service Agreement — A defined, promised level of performance in a system or network.

Queue full — An iSCSI protocol signal sent to hosts indicating a port or LUN queue cannot accept an entry.

RAID — Redundant Array of Independent Disks.

RAID group — A logical association of between two to 16 drives with the same RAID level.

RAID level — An organization of drives providing fault tolerance along with increases in capacity and performance.

Random I/O — I/O written to locations widely distributed across the file system or partition.

Raw drive — A hard drive without a file system.

RDBMS — Relational Database Management System.

Read Cache — Cache memory dedicated to improving read I/O.

Read/write head — Component of a hard drive that records information onto the platter or read information from it.

Read-ahead — See "prefetch."

Rebuild — The reconstruction of a failed drives data from through either parity or mirroring.

Recovery time objective — The estimated amount of time to restore a system to full operation after a fault or failure.

Redundancy — The ability of the storage system to continue servicing data access after a failure through the use of a backup component or data protection mechanism.

Relational database management system — A database typically hosted on a storage system, for example, Oracle, DB2, Sybase and SQL Server.

Reliability — The probability of a product performing its intended function under specific conditions for a given period of time.

Request for information — A type of procurement document whereby the buyer requests a potential seller to provide information related to a product, service or seller capability.

Request for proposal — A type of procurement document used to request proposals from prospective sellers of products or services.

Request for quotation — A type of procurement document used to request price quotations from prospective sellers of common or standard products or services. Sometimes used in place of request for proposal.

Request size — In a file system, the size of the block actually read from the drive.

Request size — In a file system, the size

Requirement — A condition or capability that must be met or possessed by a system, product, service, result, or component to satisfy a contract, standard, specification, or other formally imposed documents.

Reserved LUN — See Private LUN.

Reserved LUN Pool — A grouping of LUNs supporting installed applications, such as MirrorView and SnapView.

Resource — Skilled human (specific disciplines either individually or in crews or teams), equipment (hardware or software), services, supplies, commodities, materiel, budgets (including durations), or funds.

Response time — A measure of performance including cumulative time for an I/O completion as measured from the

host.

RFC — Request for comments.

RFI — Request for information.

RFP — Request for proposal.

RFQ — Request for quotation.

Rich media — A workload that allows for active participation by the recipient. Sometimes called interactive media.

Risk — An uncertain event or condition that, if it occurs, has a positive or negative effect on objectives.

RLP — Reserved LUN Pool.

ROM —Rough Order of Magnitude.

ROT — Rule-of-Thumb.

Rotational latency — The time required for a disk drive to rotate the desired sector under the read head.

Rough Order of Magnitude — An estimate accurate to within an order of magnitude.

Rpm — Revolutions per minute.

RPQ — Request for product qualifier.

RTO — Recovery time objective.

Rule-of-Thumb— A metric or calculation used to create an estimate.

RV — Rotational vibration.

SAN — Storage area network.

SAN Copy — CLARiiON storage system to storage system copy application.

SAP — The enterprise resource planning application produced by the software company, SAP AG.

SAS—Serial attached SCSI.

SATA—Serial ATA disk I/O protocol.

Saturation — The condition in which a storage system resource is loaded to the point where adding more I/O dramatically increases the system response time but does not result in additional throughput.

SCSI — Small computer system interface.

Sector — The smallest addressable unit on a hard drive; a sector contains 512 bytes of data.

Sequential I/O — A set of I/O requests whose pattern of address and size result in serial access of a complete region of data in monotonically increasing addresses.

Serial ATA – A disk I/O attachment used on CLARiiONs.

Serial Attached SCSI - A point-to-point serial protocol for moving data to drives.

Service Time — The interval it takes a drive or resource to perform a single I/O.

Shelf — DAE.

Short stroking — A LUN performance optimization technique of only using a portion of a RAID group.

Skill — Ability to use knowledge, a developed attitude, and/or a capability to effectively and readily execute or perform an activity.

SLA — Service Level Agreement. A contract between a service provider and a customer providing a measurable level of service or access to a resource.

SLIC — Small I/O Card.

Slice Volume — On VNX for file, a region of a volume used to create smaller units of storage. **Small Computer System Interface** — Set of standards for physically connecting and transferring data between hosts and drives.

Small block — I/O operations up to 16 KB.

Small I/O card — The generic name for the CX4 UltraFlex I/O modules, either Fibre Channel or iSCSI.

SnapView — CLARiiON and VNX point-in-time copy application.

Snapshot — Backup copy of how a LUN looks at a particular point in time.

Solid State Disk — A drive using non-volatile semiconductor memory for data storage.

SP — Storage processor.

SPE — Storage Processor Enclosure.

Specification — A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system, component, product, result, or service. Often, the procedures for determining whether these provisions have been satisfied are included in the specification.

Spike — A sudden, sharp, and significant increase in load on the storage system.

Spin down —Setting inactive hard drives into a low-power "sleep" mode.

Spindle — A component of a hard drive; it is the axel platters are mounted on. Also, spindle sometimes refers to a hard drive.

SPS —Standby Power System.

SSD — Solid State Disk.

Stack — Layered protocols.

Standard — A document established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context.

Storage area network — A network specifically designed and built for sharing drives.

Storage array — A storage system.

Storage density — A measure of the quantity of information in GBs that can be stored in a given volume of a storage system.

Storage object — A logical construct or physical device supporting both read and write data accesses.

Storage pool — A logical construct of drives supporting both read and write data accesses.

Storage processor — A logical division of the CLARiiONs architecture including the CPUs and memory.

Storage processor enclosure — Physical rack mounted cabinet containing CLARiiON Storage Processor. This enclosure contains no drives.

Storage system — A system containing multiple hard drives, cache and intelligence for the secure and economical storage of information and applications.

Storage template — A predefined set of parameters for configuring VNX file storage drives.

Stripe crossing — If a back-end I/O is not contained in an entire stripe, a stripe crossing occurs because the I/O takes more than one stripe

Stripe element — Capacity allocated to a single device of a stripe.

Stripe size — The usable capacity of a RAID group stripe.

Stripe Volume — On VNX for file, an arrangement of volumes that appear as a single volume.

Stripe width — The number of hard drives in a RAID group stripe.

Stripe — Distributing sequential chunks of storage across many drives in a RAID group.

Stroke — Movement of a hard drives' read/write head across the platter.

Subscribed Capacity — Total capacity configured for thin LUNs in the pool.

Switch — A Layer 2 device providing dedicated bandwidth between ports and switching functions between storage network devices.

System — An integrated set of regularly interacting or interdependent components created to accomplish a defined objective, with defined and maintained relationships among its components, and the whole producing or operating better than the simple sum of its components. Systems may be based on either physical or logical processes, or more commonly a combination of both.

System software — Operating system and applications used for a computer's management.

TB — Terabyte.

TCP — Transmission Control Protocol: a protocol used with IP to transmit and receive data on Ethernet networks.

TCP/IP — The pair of communications protocols used for the Internet and other similar networks

TCP/IP offload engine — A coprocessor-based host component that connects it to an Ethernet network.

Technique — A defined systematic procedure employed by a human, hardware, or software resource to perform an activity producing a product, result or deliver a service, and that may employ one or more tools.

Terabyte — One trillion bytes or one thousand gigabytes.

Thin friendly — Applications and file systems that do not pre-allocated capacity during installation or initiation.

Thin LUN — Logical storage unit whose capacity may be less than host's viewable capacity.

Thread — An independent I/O request that may execute in parallel with other requests.

Threshold — A cost, time, quality, technical, or resource value used as a parameter, and which may be included in product specifications. Crossing the threshold should trigger some action.

Throughput — A measure of performance of I/Os over time; usually measured as I/Os per second (IOPS).

TLU — Thin Provisioning LUN.

TOE —TCP/IP Offload Engine.

Topology — How parts of a system component, subsystem, or system are arranged and internally related.

Track — A ring-like region of a hard drive platter on which data is organized and stored.

Tray — DAE.

Trespass — A multipathing host initiated change in SP LUN ownership as a result of a failure or command.

UER — Unrecoverable Error Rate

UNIX — Any of several open system operating system environments.

Unisphere —VNX's resource management system software for FLARE revisions 30.0 and later.

Unrecoverable error rate — Bit error reliability metric for hard drives.

UPS — Uninterruptible Power Supply.

User — An individual or organization who owns or operates a storage product or application software.

User Capacity — Total storage capacity of a physical or logical storage object available to a host.

Vault — Special area on drives of DAE0 for storage of CLARiiON system files and cache dumps.

Variance — A quantifiable deviation, departure, or divergence away from a known baseline or expected value.

Verification — The technique of evaluating a component or product at the end of a phase or project to assure or confirm it satisfies the conditions imposed.

Virtual machine — A software application emulating a server hardware environment.

Virtual Provisioning — Explicit mapping of logical address spaces to arbitrary physical addresses. For example,

presenting an application with more capacity than is physically allocated (pool-based storage).

VLAN — Virtual Local Area Network.

VLAN Tagging — Mechanism for segregating VLANs.

VM — Virtual Machine.

VMware — EMC's family of virtual machine applications.

Volume — LUN.

Volume profile — The definition for a standard method of building a large section of file-based storage from a set of disk volumes.

WAN —Wide Area Network

Warm-up — Interval during which active data is promoted into FAST Cache.

Watermark — A cache utilization set point.

WCA —Write Cache Availability.

Wide area network — A computer network extending over a large, possibly global, geographical area.

Windows — Any of several proprietary Microsoft operating system environments.

Wintel — Industry term used to describe computers based on an Intel hardware architecture and a Microsoft Windows operating system.

Wire Rate — The maximum bandwidth for data transmission on the hardware without any protocol or software overhead. Also known as "Wire Speed."

Workload — The characteristics of a pattern of I/O requests presented to the storage system to perform a set of application tasks, including amount of I/O, address pattern, read-to-write ratio, concurrency, and burstiness.

World Wide Name — A unique identifier in a Fibre Channel or SAS storage network.

WORM — Write Once Read Many.

Write Cache — Cache memory dedicated to improving host write I/O response time by providing quick acknowledgement to the host while destaging data to disks later in the background.

Write-aside — Bypass of the write cache, where the RAID engine dispatches a write immediately to the disks.

Write-aside Size —The largest request size, in blocks, written to cached for a particular LUN.

WWN — World Wide Name.

Appendix A Best Practices Summary Index