EMC WHITE PAPER

EMC Unified Storage System Fundamentals for Performance and Availability

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

Published: October, 2011

EMC Unified Storage System Fundamentals for Performance and Availability

P/N h1094.5

Contents

Figures

Tables

How to Use This Guide

You can read this as a tutorial to gain a basic understanding of VNX storage-based performance and availability features. This paper:

- Presents an overview of the VNX storage system.
- Explains concepts that you need to understand when evaluating the VNX's performance and availability options.
- Answers many of the questions that new VNX users frequently ask.
- Supplies the background information you need to understand the *EMC Unified Best Practices for Performance and Availability: Common Platform and OE Block 31.0 — Applied Best Practices* white paper. (This is also referred to as the VNX Best Practices white paper.) If you are a new VNX user, this paper should be read before you read the VNX Best Practices paper.
- Provides a glossary at the end of this paper defining many EMC-specific terms.

If you use this paper as a reference, you need to decide whether you need help with your VNX's performance, availability, or both. Then find the appropriate sections and read the entire section.

Audience

This document is for VNX storage system administrators new to EMC storage systems. It is an introductory paper for those who need background on storage systems technology and features before implementing VNX block or file storage system best practices. An understanding of the basics of data center hosts (servers), networks, and IT concepts is assumed. The beginning of the document is general in nature, and detail is gradually increased. These general sections may be skipped by experienced readers.

A note on terminology: In this document, the term *drive* refers to both mechanical hard drives and *flash* drives. Flash drives are nonvolatile memory-based drives that are commonly referred to as solid state disks (SSDs) in the IT industry. Where the two drive types differ, they are discussed separately.

References

- EMC CLARiiON Storage Solutions: Microsoft Exchange 2007 — Best Practices
- EMC NetWorker PowerSnap and SnapView for CLARiiON
- EMC Unified Best Practices for Performance and Availability: Common Platform and OE Block 31.0 — Applied Best Practices
- EMC VNX Series — A Detailed Review

Questions, comments, or corrections about this document?

If you have questions , comments, or corrections, about this document, use the 'Feedback to Author'link next to the document title in Powerlink. You will be contacted shortly by the author or an EMC representative.

Chapter 1 Storage Environment

This chapter presents these topics:

EMC® VNX® storage systems reside in storage environments that include these components:

- VNX-series storage systems
- Hosts
- Storage area networks (SANs), network-attached storage (NAS), or converged networks

In a storage environment, one or more storage systems (like the VNX) are connected through SANs or a NAS to one or more hosts. These hosts interface with *clients*, which can be human end users or other computer systems. If a NAS is involved, the storage system may be directly attached to the Celerra/file client.

The VNX storage systems may be configured to be *block-level (FC, FCoE, iSCSI)*, *file-level* (NAS — NFS and CIFS) or unified storage systems (file *and* block). The ability to configure for storage is model-dependent, where higher-level models can be both and lower only one.

Block-level storage writes and reads blocks of data using *logical block addresses* (LBAs), which are translated into disk sector addresses on the drives. (See Chapter 8 on page 60 for details.) SAN storage environments use block-level storage to provide a higher level of performance compared to file-level storage. (See Figure 1 on page 3.)

The alternative architecture is a *file-level storage* system. Storage systems using file-level storage add a level of abstraction above the block-level access; the host's data is sent as file system extents, which must be mapped to logical disk blocks before they are stored on the hard drives. The term *network-attached storage* (NAS) refers to file-level storage.

Storage environments are usually complex. Each component has its own architecture that affects the performance and availability of itself *and* the entire storage environment.

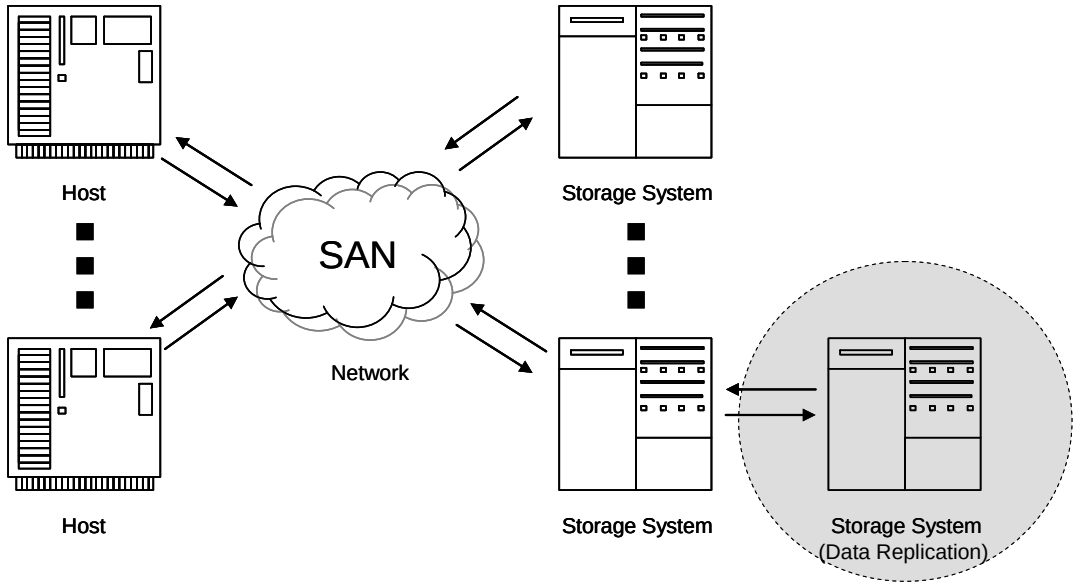

Figure 1 Components in a block storage environment

The key measurements of a storage environment are its:

- Performance — how long it takes a storage environment to retrieve data for end users.
- Availability — a measure of a storage environment's ability to retrieve data for users, especially when a component in the environment fails.
- Capacity — the amount of data that can be stored in a storage system.
- Flexibility — the capability to serve varied workloads.

Storage systems

Generally, storage systems are categorized as:

- Entry level
- Midrange
- Enterprise

Entry-level systems typically serve the needs of small businesses, or form part of a larger distributed storage environment. Entry-level systems host a maximum of about 100 drives. The performance and availability for entry-level performance varies greatly between systems. The EMC VNX5100 storage system is an example of entry-level storage. The EMC 3000-series VNXe line are all considered entry-level systems. The VNXe series is not covered in this paper.

Enterprise systems serve the central business needs of large corporations or public-sector organizations. Enterprise systems are designed and built for the highest possible performance and availability. Smaller enterprise storage systems may host about 500 drives, but 1,000 is more typical. Larger enterprise storage systems host more than 2,000 drives. The EMC Symmetrix® line and the largest model VNX7500

storage systems are examples of enterprise storage systems. The Symmetrix series is not covered in this paper.

Between the entry-level and enterprise systems is the midrange. The majority of the VNX series models are examples of midrange storage systems. (Figure 2 on page 4 shows the range of models.)

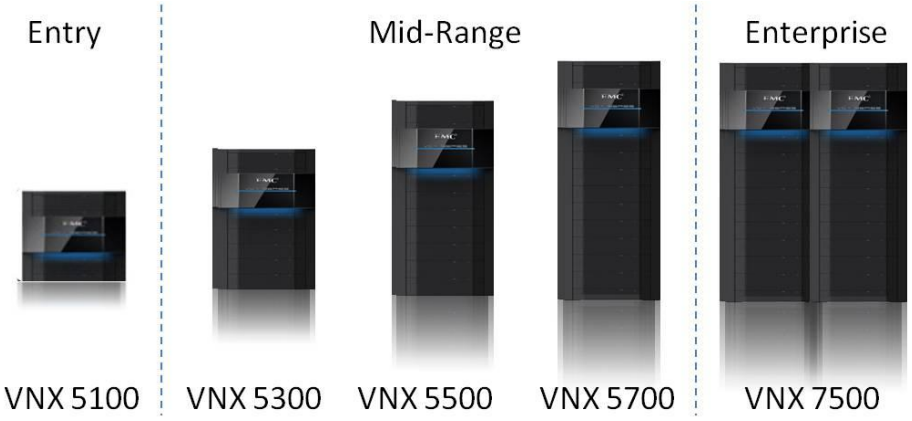

Figure 2 VNX-series model ranges

The midrange spans a large amount of storage system performance and capacity. The smallest midrange systems are similar to the largest entry-level systems in performance and capacity, although midrange systems have more features and better availability. The smaller midrange systems are also more *scalable*. Scalable means the performance and capacity of the system can easily be increased to meet growing needs. For example, in a VNX 5500 it is easy to increase the number of drives in the system from 150 to 225.

Midrange systems are also more *extensible* than entry-level systems, which means additional hardware and software features can easily be added to address new requirements. For example, the VNX can communicate over more than one protocol via Ethernet *and* Fibre Channel.

The category which a storage system falls into is somewhat elastic. The largest midrange systems can be configured to be similar in performance and capacity to the smaller enterprise systems. An entry-level system can be provisioned to have low-end mid-range capability. Large midrange systems typically have the advantage of being less expensive to purchase than enterprise systems.

General architecture

VNX storage systems like the one shown in Figure 3 on page 5 have a modular architecture. The main parts of the VNX are the:

- Front-end ports
- Storage processors — Consist of CPUs that manage the storage system's functions and memory,
- Mirrored storage system cache.
- Back end — I/O interface between the storage processors and the drives.

- Storage — Mechanical hard drives and flash drives used for mass storage.

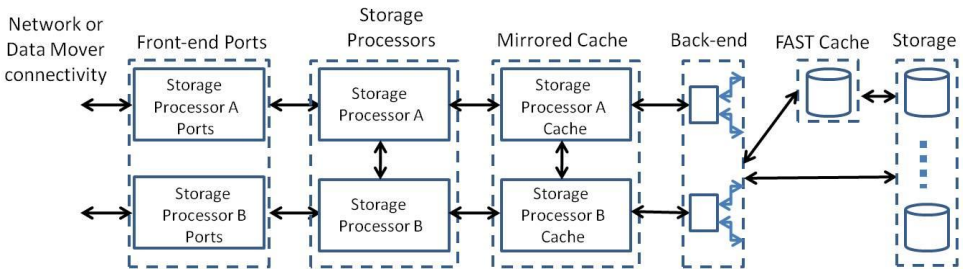

Figure 3 **VNX storage system block diagram**

Front-end ports consist of the I/O ports attaching the storage system to the hosts directly or through networked communications, like Ethernet or Fibre Channel. Front-end ports can also be directly connected to other storage systems. In the case of a NAS configuration, they would be connected to NAS File servers, sometimes called *X-Blades* or *Data Movers*. The number and interface type of front-end ports is model dependent. The higher-end models host a larger number of ports than the entry-level models.

Storage Processors are the custom-designed computers controlling the storage system. Each VNX system has two fully redundant storage processors. Sometimes storage processors are called *SP*s. The microprocessors within the storage processors are multicore. Multicore places more than one CPU on a microprocessor chip. It makes for efficient multiprocessor (multi-CPU) computer-based device designs. Clustering the CPU cores on a single processor chip speeds communications between the separate CPU cores by sharing local on-chip bus and memory resources and reducing the physical distance between them. A single-socketed chip also consumes less circuit board real estate and allows for consolidating power and cooling of the CPUs on the circuit board.

The mirrored storage processor cache optimizes reads and writes to storage by buffering them in high-speed storage processor memory as well as providing high availability. There are separate read and write caches. The read cache uses a predictive read-ahead technique to prefetch data from storage ahead of its use. The write cache buffers writes to level out I/O, combining writes and eliminating the need to rewrite data. The write function of the cache is fully redundant (mirrored) to support storage processor redundancy.

The back end is the I/O interface that connects the storage processors to the storage devices.

When we use the term storage, we are referring to an architecture consisting of a large number of storage devices grouped together to form a subsystem. Traditionally this has been made-up of *drives.* These drives have been mechanical hard drives. Newer semiconductor-based storage technology storage is also available. Hard drives may have several types of physical attachments (SATA, SAS, Fibre Channel), a large range of capacities, and several rotational speeds. These characteristics have several effects on performance and availability.

The Hosts

Hosts are computers running applications. Hosts can connect to VNX storage systems directly or over networks. Hosts are like the storage system — they have CPU, memory, and I/O resources to manage. Hosts range in performance and availability from high-performance *blade servers* with many individual servers in a single enclosure designed for efficiency, space conservation, and high availability, down to modest personal computer (PC) type desktop workstations. The way that hosts manage their CPU and I/O resources can have an effect on the storage system's performance and availability.

Hosts may operate singly, or be integrated into a *network* of hosts performing a related function. Networks of hosts are sometimes called *clusters*. A cluster is a group of host servers executing a common file system in distributed fashion and presenting their clients with a single software system. Microsoft Cluster System (MSCS) is one example of this architecture.

Hosts can also reside in *virtual machines* (VMs). VMs are software that emulates a hardware environment. Virtual machines allow for computer resources to be increased, decreased, partitioned, allocated, or deallocated as host computing needs change. VMs are supported by a *hypervisor* layer that manages each VM's access to the underlying hardware platform. VMware ESX™ Server and Microsoft's Hyper-V™ are two examples of VM-type Hypervisors. VMs can boost the usage rate of host hardware, but they can also pose performance challenges when multiple VMs compete for the same underlying platform and storage system resources.

Hosts execute *system* and *application* software, both of which affect performance and availability of the connected storage system.

System software includes operating systems (*OSs*), such as Microsoft Windows Server 2008, HP-UX, and Linux.

Applications include programs such as Microsoft Exchange Server, Oracle 11i, and IBM WebSphere. Applications generate most of the storage system's workload, although system software generates part of the workload during startup and maintenance. Applications can be deployed onto storage systems in several ways:

- Dedicated: A single application's data has sole use of a storage system This creates a single storage system workload
- Shared: More than one application's data is stored on the storage system. This creates multiple workloads.
- Optimized: Application data is distributed across more than one storage system. This creates single or multiple workloads, depending on the scale of the application.

Storage networks

The complexity of the network and its protocols has an effect on the storage system's performance and availability. Storage networks are typically either DAS, Fibre Channel iSCSI SANs, or NAS (see Chapter 2 on page 9). They can also be connected to more than one type of network at once.

Direct-attached-storage (DAS)

VNXs can also connect directly to hosts with no network devices between the storage system and the host. This is known as *direct-attached storage* (DAS). The advantages of DAS are its simple topology, and the dedicated communications channel between the host and storage; there is no network contention. The disadvantage is that the storage resource cannot be shared. The DAS connection is used primarily for local storage. Many smaller-capacity, entry-level storage systems are configured in this fashion. It can be used for SANs or NAS. The physical connection can be either Fibre Channel or Ethernet.

Storage area networks (SAN) and network-attached storage (NAS)

Fibre Channel has been the most common networking technology for block-level storage SANs. It has high throughput and low latency. It is also highly reliable. However, general purpose Ethernet networks are widely used at storage system deployment sites. Ethernet's large installed base, low cost, and relative simplicity make it an attractive choice for SAN usage.

FC SANs connected to the VNX use the *Fibre Channel* (FC) protocol, while Ethernet SANs use the iSCSI protocol.

File level storage NAS uses the NFS and CIFS protocol on top of TCP/IP and Ethernet.

The performance and availability of the SAN or NAS is highly dependent on the *network topology*. Network topology is the physical and logical arrangement of the network's elements. Network elements are the components that make up the network, such as high-capacity/ high-availability routers; fiber-optic or copper cabling; and Fibre Channel or Ethernet switches.

Converged networks

A *converged network* is a single network used to meet more than one data communications need. Instead of separate storage, LAN, and process communications networks, a single network is maintained. The network's fabric is chosen to meet its performance and bandwidth requirements, such as low-latency or high bandwidth. The converged network shares adapters, cabling, and switches between the users of the fabric.

If latency is not a consideration, Ethernet can be used to converge storage-related traffic instead of maintaining separate Fibre Channel block storage (SAN) and Ethernet file storage (NAS) networks. The Fibre Channel over Ethernet (FCoE) runs the FC protocol over Ethernet. With the introduction of the FCoE protocol, traditionally Fibre Channel SANs can be more easily migrated to Ethernet fabrics. The iSCSI traffic of the SAN and NAS traffic can share a single Ethernet network. This would be an example of convergence of SAN and NAS traffic onto a single network fabric.

Chapter 2 Common Protocols

This chapter presents these topics:

A protocol is a set of rules that governs the communications between computers and other devices on a network. Several protocols describe how to physically transmit data and how to control, manage, and interpret the data transmitted between hosts and the storage system.

Protocol stack

Protocols set standards. Protocols also have a hierarchical structure. They layer on top of each other. This is sometimes called a *stack*. A brief discussion of the protocols only needs to discuss the following three stack levels:

- Physical and ata link layer protocols — Describe the specification for a network's copper or optical fiber cables and the operation of network devices.
- Network layer protocols — Data passed over the physical layer is sent using one or more *network protocols* that specify how data is sent and received.
- Application layer protocols — Describe how the data is interpreted. For example, it may be application data, or it may be a command controlling the communication. On the VNX, these are read and write I/O requests.

For example, read and write I/O requests arrive at the storage system over Fibre Channel or Ethernet networks. Fibre Channel networks connect to Fibre Channel front-end ports. Ethernet networks connect to iSCSI and FCoE front-end ports. Read and write I/Os are SCSI protocol commands.

The iSCSI protocol is one of several networking protocols used to deliver the SCSI protocol commands. Figure 4 on page 10 shows the details of the iSCSI stack.

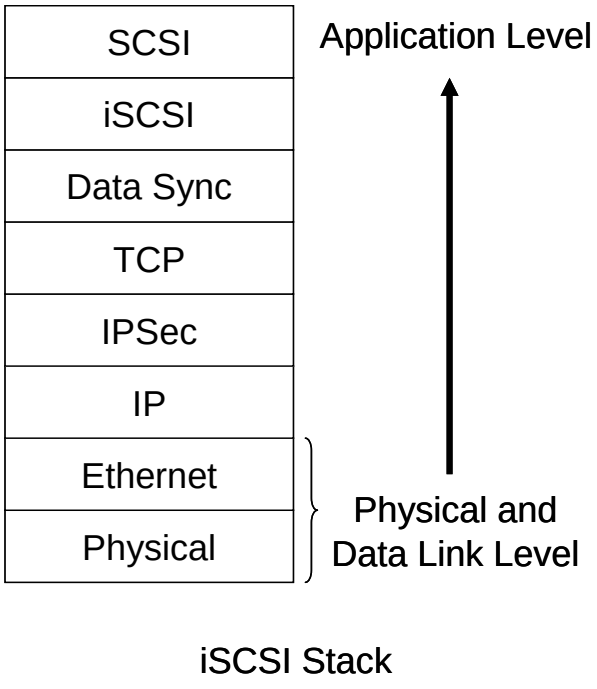

Figure 4 Detailed iSCSI stack

Physical and data link protocols

There are three main physical and data link protocols used to send and receive data to and from the storage system, and to send data within the storage system. They are:

- Fibre Channel
- Ethernet
- SAS

These protocols are the physical layer protocols of the networks that connect the hosts to the storage system and the VNX's storage processors to its drives.

As previously mentioned, the VNX's front-end ports are either Fibre Channel or Ethernet protocol. The VNX series storage processors also connect to the storage system's drives by more than one SAS interface in the back end.

Drives connected to storage system processor buses use SAS attachments. The SAS protocol is a point-to-point connection between the storage processor and the SAS storage device.

Fibre Channel protocol

The Fibre Channel protocol is standardized to the American National Standards Institute (ANSI) standard INCITS 387.

Fibre Channel is a layered protocol as described above. It is an *entire* stack. Each layer describes a different aspect of the protocol, from the physical transmission of bits on the cable to the interpretation of high-level commands and data. Note that it is common to refer to one layer or all the Fibre Channel stack's layers simply as Fibre Channel.

There are two important Fibre Channel stack layers to understand:

- FC-PH
- FC-4: Protocol Mapping layer

The Fibre Channel protocol specifies the bus interface that device controllers use to communicate with one another

FC-PH

The FC-PH is the physical layer that actually consists of four layers (FC-0 through FC-3). These layers constitute the Fibre Channel cabling, including the data link. Data links connect one location to another; digitally-encoded information is transferred over data links.

On the VNX, Fibre Channel can provide front-end communication with hosts. The FC protocol has operational speeds of 1 Gb/s, 2 Gb/s, 4 Gb/s, 8 Gb/s, and 10 Gb/s.

FC-4

The FC-4 is the network layer in which SCSI protocol messages are encapsulated in Fibre Channel protocol packets for delivery. This layer is discussed below.

Ethernet protocol

The Ethernet protocol is standardized as the Institute of Electrical and Electronics Engineers (IEEE) standard IEEE 802.3. Ethernet has a single physical layer.

Ethernet networks connect to the VNX's front-end Ethernet ports. These Ethernet ports are referred to as *iSCSI* ports after the intermediate network protocol used in workload communications. The Ethernet protocol has operational speeds of: 10 Mb/s, 100 Mb/s, 1 Gb/s, and 10 Gb/s. The 1 Gb/s speed is referred to as *GigE*. The 10 Gb/s speed is *10 GigE.*

SAS protocol

Serial Attached SCSI (SAS) is an ANSI standard (ANSI INCITS 376) for device attachment.

SAS is a block-level protocol. It provides a point-to-point serial interface in which device controllers may be directly linked to one another. In the VNX's it is a connection between the storage processors to mass storage drive controllers. SAS integrates the SCSI and SATA protocol. It combines the SCSI's device utility and reliability with SATA's serial architecture.

The SAS protocol has operational speeds of 3 Gb/s and 6 Gb/s.

Network protocols

There are several network protocols used in data communications to the VNX. They are:

- Fibre Channel
- TCP/IP
- FCoE
- iSCSI

These protocols describe how the data is transmitted and received by the storage system. They also control and manage the connection between the sender and receiver.

Note that network protocols can be sent on different physical protocols. For example, Fibre Channel protocol formatted data can be sent on a TCP/IP (see below) network.

Fibre channel protocol

This is the FC-4 layer Fibre Channel protocol. The Fibre Channel protocol carries embedded SCSI commands for reads and writes over Fibre Channel networks. It also includes control units. Control units are discrete messages that are separate from data-bearing messages. These messages manage the connection over which the data is passed.

TCP/IP protocol

TCP/IP is also referred to as the *Internet Protocol*. It is actually two standardized protocols that are "stacked" together. TCP (Transmission Control Protocol) is the higher-level protocol; it creates packets for data and messages that need to be transmitted over the Internet and ensures they arrive in order. IP is the lower-level protocol; it is responsible for adding addresses to the packets to make sure that the packets reach the proper destination. IP can also be used with other high-level protocols.

The Internet Engineering Task Force (IETF) standardized the TCP/IP protocols, and documents them in a set of documents called RFCs, or Request for Comment:

- IPv4 is described in RFC 791.
- IPv6 is defined in RFC 2460.
- TCP is described in RFC 793.

Like Fibre Channel, TCP/IP is a carrier protocol. It is used to encapsulate iSCSI protocol packets.

FCoE protocol

Ethernet can also be used to deliver Fibre Channel packets on SANs. Fibre Channel over Ethernet (FCoE) is an encapsulation of Fibre Channel packets for delivery over Ethernet-based SANs. This allows Fibre Channel to use high-bandwidth and long-distance Ethernet networks while preserving the Fibre Channel protocol.

iSCSI protocol

iSCSI is the Internet Small Computer System Interface protocol. It is used to carry block-level data over IP-based Ethernet networks. iSCSI is layered on top of the TCP/IP protocol. iSCSI is used to carry SCSI commands.

iSCSI is standardized in RFC 3720.

Note that the TCP/IP and iSCSI protocols are used on an Ethernet network at the same time. iSCSI would be used for SAN workload communications. TCP/IP would be used for other network communications. However it is recommended that SAN traffic be either physically or virtually separated (VLAN) from all network traffic.

Virtual LANs (VLANs)

A storage network VLAN partitions Ethernet networks to provide more resources.

VLAN operation is part of the TCP/IP suite of protocols, and is standardized under IEEE 802.1q.

VLANs create logical collision and broadcast domains. With the VLAN broadcast domains, all messages are not sent to all destinations. Traffic still physically coexists on the same cabling, hubs, switches, and routers. However, broadcast and multicast traffic is limited by the creation of the VLAN domains. The VLAN's bridging software defines which nodes are to be included in the broadcast.

As noted above, a dedicated network is recommended for iSCSI (storage) use. Physically isolating your storage environment's network traffic produces higher

network performance and greater reliability. However, sometimes a physically separate storage network is not practical, and the available Ethernet network may need to serve for both storage communications and the networking needs of the general users. If not using a dedicated Ethernet LAN, either separate the iSCSI traffic should onto its own physical LAN segments by routers that are not used for general LAN traffic, or use VLANs.

Restricting storage traffic to its own VLAN is not a proper security method. The confidentiality of the data cannot be protected by a VLAN alone. Proper security on a VLAN is possible only by implementing CHAP or iSCSI authentication.

CHAP protocol

Security is provided on iSCSI networks though the use of Challenge Handshake Authentication Protocol (CHAP). RFC 1994 defines CHAP.

CHAP is a protocol that is used to authenticate the peer of a connection. It is based on symmetrical encryption, which is a shared key between the peers. On a SAN, the peers are a host and the storage system. Plaintext is unencrypted text. With CHAP, both the host and storage system know the plaintext of the secret key. However the key is not sent over the network.

SNMP protocol

Simple Network Management Protocol (SNMP) is used for managing the network devices of a SAN or NAS. Its primary task is to allow a host to get statistics from any network node, like hosts, switches, and routers. A Management Information Base (MIB) is used as a primary resource in SNMP. Management data on the managed network is made available in the protocol as variables. The protocol can also modify and reconfigure the network through remote modification of these variables. The variables are organized in hierarchies. These hierarchies and variable metadata are described by the MIB. SNMP uses UDP as transport protocol in its communications because it has lower overhead, is lightweight, and is simple.

Network port aggregation

Network port aggregation enables more than one active Ethernet connection to the same switch to appear as a single link with a single MAC address. This increases the overall bandwidth of the connection.

There are two types of port aggregation, Link Aggregation Control Protocol (LACP), and Ethernet Channel. LACP is an open standard, available on all switches that support the IEEE 802.3ad standard. Ethernet Channel is proprietary to CISCO systems. It works on CISCO switches in conjunction with CISCO's EtherChannel™ standard.

Note that port aggregation is a logical connection. The physical connection to the host is limited to the physical Ethernet port's bandwidth.

On the aggregated port, data is distributed among the physical links by using MAC addresses. Both the storage system's port and the network switch require configuration changes to implement port aggregation.

File protocols (CIFS/NFS)

Common Interface File System (CIFS) and Network File System (NFS) are the primary network protocols used with NAS (see Figure 5 on page 15). Either protocol can be used on the clients supported, although, NFS is typically found on Linux clients and CIFS on MS Windows clients.

NFS is defined in RFC 1094. CIFS is an update of the legacy SAMBA (SMB) protocol, and is preferred over SMB. CIFS is a Microsoft Windows protocol.

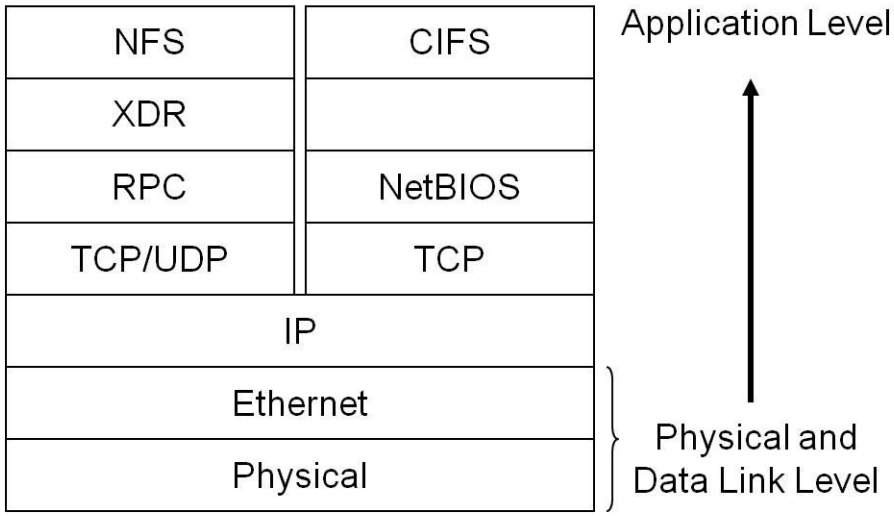

Figure 5 NFS /CIFS stack comparison

Note that NFS relies on remote procedure calls (RPC) and external data representation (XDR). RPC is a set of function calls used by a client program to call functions on a remote server program. XDR is a library of routines that translate data formats between processes. In addition, NFS can use either the TCP or the User Datagram Protocol (UDP). UDP is an unreliable connectionless protocol and is defined by RFC 768 and 1122. It is a datagram service. There is no guarantee that the data will reach its destination. UDP is meant to provide service with very little transmission overhead.

CIFS relies on the NETBIOS protocol. NetBIOS is a Microsoft protocol that controls the sessions between computers and maintains connections.

Block protocols (SCSI)

Small Computer System Interface (SCSI) is the application-layer protocol used to send commands to drives, including read and write I/Os. I/Os are sometimes referred to as *SCSI commands.* SCSI is the standard for connecting and transferring data between hosts and the VNX's LUNs that are made up of its drives.

SCSI commands may contain data or control information. They are sent by host applications and arrive embedded in either iSCSI protocol or Fibre Channel protocol message units. Figure 6 on page 16 shows the relationship between the major storage system's FC and iSCSI protocols.

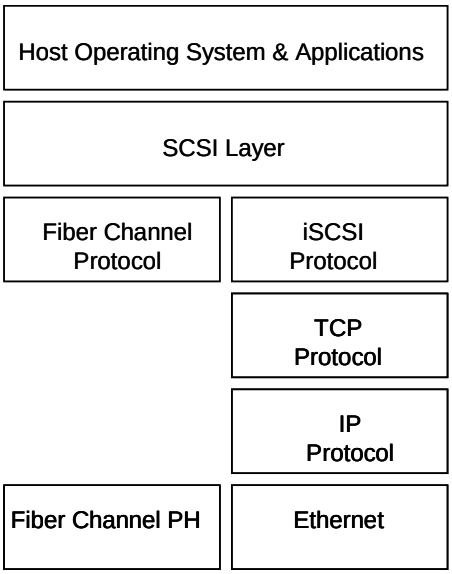

Figure 6 Block protocol summary iSCSI layer

Chapter 3 Performance and Availability Metrics

This chapter presents these topics:

Performance is the amount of work accomplished by a system compared to the time taken and resources used. The time and resources need to be measurable to judge performance. The definition of high performance is dependent on the production goal. The VNX-series storage systems achieve high performance in a number of operational profiles, from high bandwidth, high latency to high throughput, low latency.

Availability refers to the storage system's ability to provide users access to their applications and data even when a hardware fault is present (this is sometimes called a *degraded* state or mode). Midrange systems like VNX-series are classified as *highly available* because they continue to provide access to data with any single failure, although typically, operation in degraded mode is with some level of reduced performance.

Identifying the measurements appropriate to the resource is an important part of architecting or evaluating a storage system's performance and availability. The most common terms used to measure a system's performance and availability are:

- Bandwidth
- Throughput
- Response times
- Availability
- Recovery time objective (RTO)

The availability metric should not be confused with *general availability*, which is the combination of reliability and redundancy needed to avoid the non-functional system state.

Bandwidth

Bandwidth is the measurement of the amount of data that can be transferred along a channel per second. Storage-system bandwidths are measured in megabytes per second (MB/s) or gigabytes per second (GB/s). Note the use of the upper-case *B* indicating bytes. Communications network bandwidth is typically measured in bits per second (b/s).

Storage-system bandwidth usually describes sequential or large-block I/O performance, but it can describe any workload.

The storage system's *deliverable* bandwidth is directly affected by the bandwidth of the attach layer, which is also referred to as the *attach bandwidth*. Attach bandwidth is usually measured in megabits per second (Mb/s) or gigabits per second (Gb/s). Note the use of the lower-case *b* indicating bits.

It is important to remember that storage-system bandwidth is measured in *bytes* per second, and attach bandwidth is measured in *bits* per second. There are frequent

conversions between *attach* bandwidth and the *storage system drive's* bandwidths. In addition, there is a difference between the *maximum specified* bandwidth (called *wire rate* or *wire speed*) and the *actual* bandwidth realized.

Due to the effects of packet overhead, and the network protocols, data always travels at less than nominal bandwidth because of protocol overhead.

For example, keeping in mind that a byte is eight bits, and the overhead of protocols, the maximum gigabit Ethernet (GigE) speed of 1000 Mb/s is equivalent to about 119 MB/s.

Throughput

Throughput is the number of I/O operations that are processed per second (*IOPS*). *This measurement is independent of request size.* Individual components, such as hard drives, up to entire storage systems can be rated for their throughput performance. Throughput is the measurement typically used to describe random and small-block I/O performance.

Throughput can be further divided into:

- Read IOPS: Read I/Os per second
- Write IOPS: Write I/Os per second
- Total IOPS: Sum of read and write I/Os per second

This distinction is made because of the different amounts of storage system resources consumed in servicing a read operation versus a write. For a storage system I/O, it is very common to see the total IOPS stated with the ratio of read-to-write IOPS. An example is "10,000 IOPS, 70 percent reads."

Bandwidth and throughput on a storage device

The relationship between bandwidth and throughput on a storage device is important to understand. This relationship is related to I/O size. That is, as I/O operations (measured in bytes per second) go up in block size, the quantity of data per unit time transferred likewise goes up. However, with constant bandwidth, as the I/O size goes up, the I/O rate (throughput) goes down. The relationship can also be looked at the other way: As the I/O operation goes up in block size, the bandwidth goes up.

Figure 7 on page 20 shows the relationship of throughput to bandwidth for a typical mechanical drive. On the left axis is throughput. On the right axis is bandwidth.

The red line is bandwidth with its MB/s axis of measurement on the right. The green line is throughput with its IOPS axis of measurement on the left.

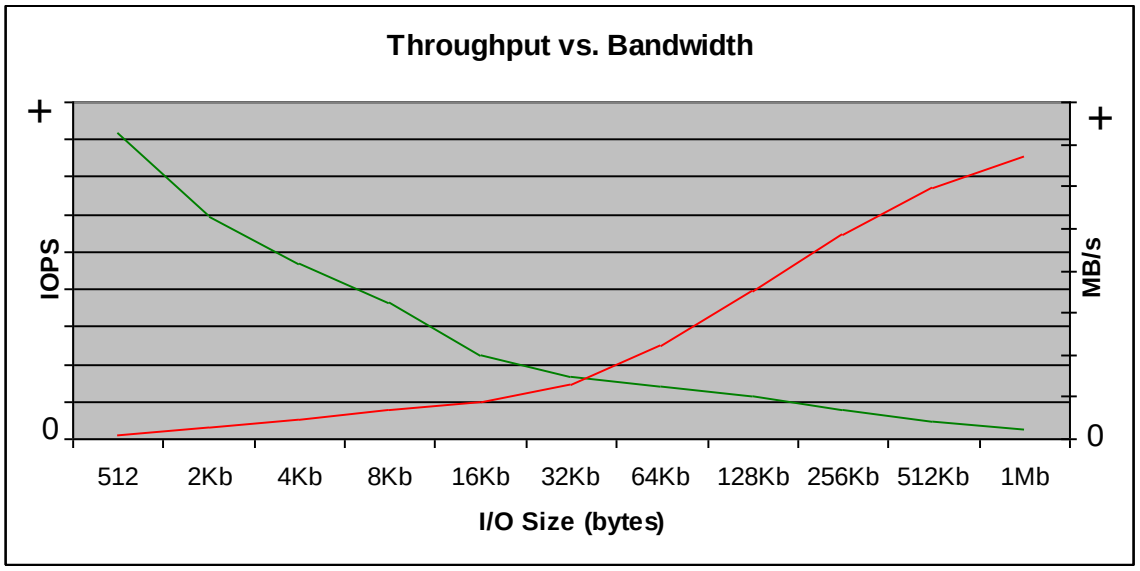

Figure 7 Mechanical hard drive throughput vs. bandwidth

From the graph, you can see that if the I/O size is a small 512 bytes, the bandwidth (red line) is low, but the throughput (green line) is high. Oppositely, as the I/O size increases, the bandwidth (red) goes up, and the throughput (green) goes down. This is shown most clearly by the far right side of the graph.

Note that any discussion of a device or interface being able to support "so many" GB/s of bandwidth, and "so many" IOPS of throughput is meaningless without knowing the size of the I/O. Knowing your I/O size is the first step in a bandwidth/throughput analysis.

Response time

Response time is a measurement of the elapsed time between the end of an inquiry or of a request for service and the beginning of the request complete. There are two types of response time discussed: user and I/O.

User response time is the "user experience" with the system, of which the storage system is a part. I/O response time is a storage-system-only metric measuring how long it takes to satisfy a host read or write request.

User response time

Users expect a uniform, high level of performance across a variety of operating conditions. The primary metric for user performance is *user response time*. Sometimes this is called *client-side response time* or *host response time*. User response time is the amount of time (typically milliseconds) it takes a system to respond to a request. This may be a query request, report generation, or a simple terminal screen refresh. Often there is a *Quality of Service* (QoS) agreement to quantify the performance promised to the user by the system's architects and administrators.

Each component in the system shown in Figure 8 on page 21 has its own response time ("T").

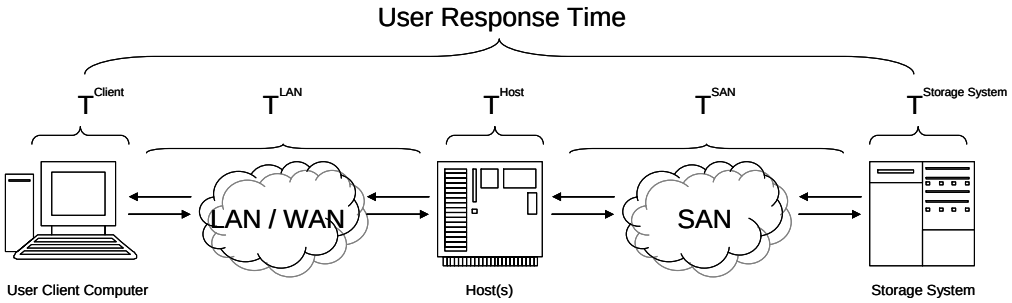

Figure 8 Conceptual view of user response time

User response time is a particularly high-level measurement. The user response time is the sum of the response times both for transmit and receive of all the subsystems within the system. These subsystems may include:

- The user's workstation (client)
- The network — local area network (LAN) or wide area network (WAN) — connecting the client computer to its host server
- One or more host servers
- The storage area network (SAN) connecting the servers to one or more storage systems
- The storage system that is the source or destination of the I/O

A single subsystem with a high response time can adversely affect overall user response time.

For example, it is possible for the storage system to have a low I/O response time, and the intervening LAN to have a high response time, due to network congestion. This results in an overall high user response time.

A high user response time is one of the first indications a bottleneck exists in a storage environment. However, it does not indicate where the bottleneck is located or its root cause.

I/O response time

I/O response time is a measurement of how long a single I/O takes to complete. It can apply to either a read or write from an individual drive such as a hard drive, or a read or write to a LUN made up of several drives. Always be clear on which object is being measured when stating response time.

I/O response time includes time spent waiting in queues as well as time spent servicing a request. Queues are temporary storage areas where I/Os wait for their turn at execution. Short service times and short queues result in short I/O response times. Long queues increase response time.

This measurement is typically in milliseconds (ms) although some devices can respond in microseconds (μs).

I/O response time and bandwidth

Servicing small-block-throughput oriented workloads is where low response times can be achieved very easily. Low response time with bandwidth workloads can be more difficult to achieve. The response time of a large block-size bandwidth oriented workloads may be of a different magnitude. This is simply because of the time to service each I/O.

Availability

Availability measures the percentage of time the system is able to return data when requested by the client. Degraded performance is not included in this metric.

For example, *five 9s availability* means 99.999 percent availability. This percentage translates into a total data-not-available time of about five minutes and fifteen seconds per year.

The following metrics are used in developing data recovery plans. They determine how often data needs to be backed-up and what kind of retention policies have to be in place to ensure meeting the SLAs that are driving these requirements.

RPO metric

Recovery point objective (RPO) refers to the amount of data at risk. RPO is determined by the amount of time between data protection events, and reflects the amount of data that potentially could be lost during a disaster recovery. This metric is an indication of the amount of data at risk of being lost. This metric may be described in intervals as short as seconds, or as long as hours.

RTO metric

The recovery time objective (RTO) is related to availability. RTO is the time required to return an application to a functioning state after a failure. The RTO metric determines the magnitude of a storage system's high-availability requirement. This metric may be described in intervals as short as milliseconds or as long as days

RPO/RTO examples

Assume that a relational database has an RPO of two hours and an RTO of five minutes.

To ensure that no more than two hours of user data is lost, SnapView snapshots every two hours would likely be sufficient, because snapshots support a short RTO.

Assume an RPO is 15 minutes and the RTO is 30 minutes for the database.

A different strategy is required in this case, because the RTO is much longer than the RPO. The recovery plan in this situation may require full database backups every 6 or

8 hours with transaction logs every 15-minutes. The longer-duration RTO is supported by recovery through “re-do” logging.

Chapter 4 Workload Profiles

This chapter presents these topics:

A *workload* is the number of work units assigned to a resource over a period of time.

Readers unfamiliar with storage objects may want to read the Storage Objects chapter before this one to familiarize themselves with some of the terms in this section.

The VNX's workload is the number of read and write requests it receives and transmits each second. The overall system's applications — how they are used, how they are configured, and when they are used — greatly affect the storage system's workload.

A storage system may contain the data for one or more applications. Storage systems hosting a single application are dedicated, and have one workload. Many storage systems store data for more than one application or class of applications. For example, a storage system may be hosting a NAS as well as serving applications doing block-level I/O. This results in two or more workloads. In addition, applications may change their operating profile over time as a result of the tasks they perform.

The overall workload is the sum of the individual application workloads. More than one workload can result in complementary or conflicting storage system resource usage. If the workloads use the storage system's resources in the same way and are mindful of the "resource budget", they are complimentary. Should the workloads use the storage system's resources very differently, or to the exclusion of each other, they would be conflicting workloads. Each application workload needs to be understood by itself before the overall system workload can be understood.

For example, a NAS system typically does small-block random I/O. Applications like backup do large-block sequential I/O. A small-block random and a large-block sequential workload sharing the same physical storage objects is an example of a conflicting workload.

Finally, there are applications whose data is distributed over more than one storage system. This case is treated in the same way as with multiple applications.

Describing the workload

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. This is because they consume storage system resources differently. Knowing the ratio of the reads to writes performed by the application directly affects which best practices to apply to your cache, RAID group, and LUN provisioning.

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 back end. 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. For example, creating redundant data or metadata for data protection is not required with a read. In addition, most storage devices perform reads faster than writes. Reads that find their data in the storage system's primary cache (a *cache hit*) or secondary cache consume fewer resources than those that do not. 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 level protection, and LUN provisioning to apply to your workload.

An application can have three types of I/O:

- Sequential
- Random
- Mixed

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

Random I/O has the source or destination of the I/O operation "scattered' throughout the address space of the storage device. This causes mechanical hard drives to seek. (See" Physical" on page 61.) Sequential I/O has the source or destination of the I/O in sequence. The I/O requests are either next to or very close to each other.

Small random I/Os use more storage system resources than large sequential I/Os. (See the next section.) Random I/O throughput is affected by many additional factors within the storage system. Applications that perform only sequential I/O have higher bandwidth than applications that perform random or mixed I/O. Working with workloads that include 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 next section 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 that this definition has changed over time, with larger-block sized I/O becoming more common. For the purposes of this paper, I/Os up to and including 16 KB are considered small, and I/Os of 32 KB and greater 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 online 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 IOPS of the physical storage device.

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 I/O

It is important to know the workloads applications' locality when planning 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 Pools (FAST VP) feature's tiered storage within Virtual 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. Locality applies to random-access workloads. Sequential workloads have no locality.

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 re-accessing of storage locations within a short period of time. A short duration is considered to be within seconds, hours 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 on the storage object. 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. In most of the cases, there is a high likelihood that addresses adjacent to recently accessed locations will also be accessed.

Secondary caching and automated tiering exploit locality of I/O requests to achieve higher throughput by ensuring that actively accessed data with high locality is on storage devices with the lowest response time. Actively accessed data is sometimes

called the *working data set*. These features are triggered by 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 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. Be aware that locality is a "statistical" distribution. The great majority of user data may be within a relatively constrained address-space. However, there are likely to be outliers — I/O to addresses outside this space. A 3- to 5-percent degree of locality is common, but 20 percent is easily possible. Certain data types, typically metadata, such as object-oriented database index tables, can have very high locality.

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. FAST caches and FAST VP tiers would be sized capacity-wise to meet the high locality user data making up the working set.

Note that there are applications with very low locality. For example, a workload with perfectly random I/O would have the lowest possible locality. Benchmarking applications, such as the public domain IOMeter™ I/O subsystem measurement and characterization tool are capable of generating perfectly random workloads.

VNX storage system administrators should confer with their storage and application architects on the locality of their applications.

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 resource utilization to apply to your workload.

I/O traffic to the storage system can be steady or sporadic. Sporadic 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 not totally predictable, large increase in activity. To manage spikes requires that 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, 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.

VNX storage system administrators should confer with their storage and application architects on the threading-model of their applications.

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, nonrepeatable reading (reading a previously changed value), and blocking. The higher the I/O concurrency is, the better the storage system's performance.

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

I/O characterization and workload

Following are descriptions of six common workloads to demonstrate a wide range of I/O characteristics:

- Online transaction processing (OLTP)
- Mail system
- File serving
- Decision support system (DSS)
- Backup to disk
- Rich media

OLTP workloads are found in database systems for transaction-oriented applications. Examples of these systems include sales ordering, inventory control, and electronic banking. This workload is random access, majority reads, small block, and throughput intensive.

Mail systems are used for messaging and collaboration services applications for all sized organizations. Mail systems are a popular application for storage systems. Microsoft Exchange Server is one example of a mail system. This workload is random access, mixed read and write, and throughput intensive.

File-serving workloads are found in storage systems used as a network-attached storage (*NAS*) device. This workload requires the storage system to maintain a high number of IOPS. This workload is random access, majority read, and throughput intensive.

A DSS workload is the storage system's hosting of a transactional database with complex queries made to a large number of large tables with different data types. Historically, this workload used to be sequential access; recent applications now have random access. In both cases the I/O is majority read and bandwidth intensive.

The backup-to-disk workload is a backup process that uses disk-based storage (instead of magnetic tape) for archival storage. The storage system copies data, databases, or servers to create an archival duplicate of the source material. Staging primary storage backups to secondary disk-based storage takes less time than staging to tape-based storage. Disk-based backups are also quicker if data recovery is required. The backup data on disks is already on line and available for a rapid data restore. This workload is large block, sequential access, majority write, and bandwidth intensive.

The rich media workload is the storage system's usage in multimedia streaming. Multimedia streaming is presenting video and sound to many end-users. This workload is sequential per stream access, majority read, and bandwidth intensive. It's possible for this workload to be mostly random when a large number of streams are being serviced.

Table 1 on page 32 summarizes the I/O patterns typically observed by these workloads.

Table 1 I/O characterization of workloads

Workload	I/O type		Access type		I/O size		I/O flow		Descriptive metric	
	Random	*Sequential*	*Reads*	*Writes*	*Small*	*Large*	*Steady*	*Bursty*	*Throughput*	*Bandwidth*
OLTP	X		X	X	X			X	X	
Mail system	X		X	X	X	X		X	X	
File serving	X		X	X	X	X		X	X	
DSS	X	X	X			X	X			X
Backup-to-disk		X		X		X	X			X
Rich media		X	X			X	X			X

Note that some workloads have both types selected in a category. This indicates a mixed usage or different access types depending on the infrastructure.

For example, in a mail-system workload, the I/O size is not markedly small or large block. More information would typically be included: a description to accurately describe a workload; for example an OLTP workload might be concisely described as, *"an application generating random I/O. The I/O creates a total of 20,000 IOPS of which 70 percent are reads. The I/O size is 16 KB."*

Knowing the workload

To implement best practices, you should understand your storage system's workload(s). This requires knowledge of the host applications. Please remember that *applying performance best practices has little effect when the workload's demands exceed the storage system's performance capabilities*.

Performance is an evaluated behavior. It is important to maintain historical records of system performance. Having performance metrics *before* applying any best practices to judge results saves considerable time and labor.

New workloads should be benchmarked in non-production environments before entering production. Contact your EMC sales representative to engage an EMC USPEED professional for the proper techniques for estimating and simulating new system performance before going live.

Finally, be aware of any changes in the workload or overall system configuration so that you can understand the change's effect on overall performance. It is prudent to regularly use Unisphere Analyzer to monitor and analyze performance. *Regular monitoring with Analyzer provides the baseline performance metrics for historical comparison. This information can give early warning to unplanned changes in performance.*

Chapter 5 Software

This chapter presents these topics:

The following chapter provides a brief overview of the VNX's software and several important features new users should become familiar with.

Operating environment

The operating environment (OE) controls the operation of the VNX series storage systems. VNX OE for Block manages all I/O functions of the storage system. OE Block is extensible, so it can supply various replication features depending on configuration.

VNX OE for File is the operating environment for the VNX NAS Head. The NAS head is also called the X-Blade or the Data Mover. OE File is also a part of the VNX NAS Control Station.

Unisphere

Unisphere is a set of applications used to configure, monitor, and manage VNX storage systems. Different Unisphere programs are useful in different storage environments.

Unisphere Manager

Unisphere Manager provides a browser-based interface for system management. Through Unisphere Manager, multiple EMC storage systems and hosts can be managed. Typical management tasks include:

- Provisioning (configuring) storage objects, including drives, RAID groups, LUNs, file systems, checkpoints, and so on
- Managing the cache
- Monitoring errors
- Provisioning virtual LUNs

Unisphere CLI

The Unisphere command-line interface (*CLI*) provides a command-line interface to the VNX. A small number of advanced tuning tasks are only possible through the *CLI*. The secure CLI (naviseccli) provides a secure credential-based interface enforcing user roles and adding an audit capability.

The CLI can also be used for *scripting*. CLI scripts are executable files containing Navisphere command statements. Scripts can be used to automate storage system management tasks.

Unisphere Analyzer

Unisphere Analyzer is the VNX performance analysis tool. The metrics it provides are an important part of performance and availability tuning. Analyzer allows a review (either real-time or off line) of the storage system's performance to identify performance bottlenecks. Analyzer also has a CLI interface.

Having recorded historical performance information on the storage system's workload is the first step in performance and availability tuning.

Replication layered applications

Replication layered applications, sometimes called *layered apps*, are storage system applications. They provide host-independent services important to overall system maintenance and availability. These applications are licensable software that must be purchased separately with the storage system.

Data replication is an *availability* function that operates independently of the host(s). Data replication layered apps include:

- MirrorView™
- RecoverPoint
- SAN Copy™
- SnapView™
- SnapSure™

The important difference in these data replication apps is local versus remote replication. Local replication creates a copy in the local storage system. Remote replication creates a copy on another, sometimes geographically separate, storage system.

Best practices for individual replication layered applications are found on Powerlink.

MirrorView

MirrorView/S and MirrorView/A are optional software applications supported by the VNX-series storage systems. Both provide remote replication of data,

MirrorView/S mirrors one or more LUNs *synchronously* (and consistently) from one storage system to another. The secondary data images remain in sync with the primary during normal operation.

MirrorView/A mirrors one or more LUNs *asynchronously* (and consistently) from one storage system to another. The secondary images are synchronized with the primary periodically. The synchronization interval is adjustable to meet the deployment's architecture. This is typically a distance consideration.

RecoverPoint

RecoverPoint is a data-protection application that provides local and remote data replication of LUNs. It works with any storage system. It runs on SAN network elements called RecoverPoint appliances and on hosts. Its redirection feature (called a splitter) can also be installed to run directly on the storage system.

SAN Copy

SAN Copy is a data migration application. It supports the bulk transfer of data between or within storage systems. Neither of these storage systems has to be an EMC-manufactured system. Using Open Replicator, data can be transferred from storage system to storage system without host involvement.

SnapView

SnapView is an application that allows the creation of either point-in-time copies of storage system data (called *snapshots*) or full, local mirrors (called *clones*). This feature is sometimes called *local replication*. These copies can be used for online backups or data replication. Copies can be accessed directly by other applications such as development testing. They can also be used to offload the backup activity from production hosts

SnapSure

EMC Celerra SnapSure enables the creation of point-in-time logical images of Production File Systems (PFSs). The point-in-time views of the PFS are called *checkpoints*. With checkpoints, individual files or entire file systems may be restored from online backup.

Chapter 6 VNX Physical Architecture

This chapter presents these topics:

The VNX-series has a rack-mounted, modular physical architecture, made up of two or more enclosures per rack. Each enclosure contains the hardware for performing one or more system functions. The VNX uses the following enclosure types:

- Storage processor enclosure (SPE) or disk processor enclosure (DPE)
- Standby power supply (SPS)
- Disk array enclosure (DAE)
- Data mover enclosure (X-Blade enclosure)
- Control Station

Different models, storage capacity, and usage storage systems are configured by combining the enclosures together into a configuration. Generally this includes, selecting the SPE or DPE for storage processor performance, number and model DAEs for storage capacity, and adding or omitting the file-related X-Blade and Control stations.

Enclosure measurements

A rack unit, or U, is an EIA standard unit of measure used to describe the height of IT equipment intended for mounting in equipment racks. It is 1.45 inches or 482.6 mm. VNX racks are a maximum 40U in height. A VNX series storage system may be one or more racks, depending on the number of drives installed.

The width of the enclosures conforms to the IT *19-inch rack* standard. The 19-inch (482.6 mm) standard is the most widely used for IT computing equipment.

All VNX enclosures follow the 19-inch racking standard in width. The different enclosure types come in different types, depending on their function and model. The enclosures also have different depths.

Storage processor enclosures

VNX storage processors are housed in storage processor enclosures. There are two types of enclosures for storage processors: storage processor enclosures (SPEs) and disk processor enclosures (DPEs). See Table 2 on page 38. Note that the DPE is a type of storage processor enclosure, one that also houses drives.

Note that an SPE is 2/3 the height of a DPE. However, an SPE requires one DAE (the O/S DAE) in every configuration. An SPE consumes more rack space than a DPE.

There is only one SPE or DPE per storage system.

Table 2 VNX storage processor enclosures by model

VNX model	Storage processor enclosure type	Height (U)
VNX5100	DPE	3
VNX5300		
VNX5500		
VNX5700	SPE	2
VNX7500		

Storage processor enclosure-based (SPE) systems

A SPE enclosure provides connectivity, cache, and cooling, but no drives. This design offers the most flexible deployment options.

Disk processor enclosure-based (DPE) systems

A DPE enclosure provides connectivity, cache, and a number of drives in the same enclosure. These systems offer the highest density of storage since there is no separate SPE consuming rack space.

Standby power

VNX series storage systems have redundant power supplies, redundant power distribution, and battery backup to ensure against internal and external power failures.

The following applies to Alternating Current (AC) supplied storage systems only. Direct current (DC) supplied systems are available, but not discussed. Contact your EMC representative for details on DC power configured storage systems.

The VNX employs *dual-shared power supplies*. VNX 5100, 5300, and 5500 have only two power supplies in the DPE. A single power supply can keep write cache enabled for normal operation. There are two cooling modules, one per SP. A single cooling module can keep write cache enabled and operation normal.

VNX 5700 and 7500 have four power supply/cooling modules per SPE. That is, for each enclosure in the cabinet, two power supplies share power distribution to it. When one power supply fails, the surviving power supply supports the enclosure.

The VNX cabinet comes equipped with power distribution units (PDUs). DAE-type enclosures in the cabinet plug directly into the PDUs to receive power. SPE/DPE don't plug directly into the PDUs. Each enclosure is connected to a PDU on the left and a PDU on the right. This ensures redundancy in the event of a power failure. If power fails on either side, the opposite side can maintain power to the entire system.

The standby power supply (SPS) is a power monitoring and battery backup unit (BBU). Power monitoring determines if data center power quality variations or disturbances may adversely affect the storage system's operation and trigger the battery backup function. The SPS provides write cache protection against brief external power fluctuations and for complete power loss.

Some VNX models can be configured for only one SPS. All models can be configured for two SPSs. Dual SPSs provided a higher degree of storage system availability than

single-SPS installations. The storage system requires only one functional SPS to enable write caching. Maintaining and enabling write cache while still guaranteeing data integrity is an important performance feature.

The SPS does *not* function like an uninterruptable power supply (UPS). It is not designed to and does *not* keep the storage system up and running for long periods in anticipation of power being restored. The SPS functions to protect cache and dump it to the vault when there is power loss. The vault resides on the first four drives of the storage system and can hold the contents of cache indefinitely without data loss in the cases of an extended outage. Saving of the cache can vary anywhere from several seconds to several minutes, depending on the model, cache capacity, and type of system drives.

The SPS is typically a 1U unit. On SPE-based VNX models configured with a DAE/OS containing more than 25-drives, the SPS is a 2U unit.

SPS battery power

SPSs use battery types with different *Amp-hour* ratings. (Amp hours is the measurement of battery capacity.) These batteries are sized to support the connected SP and storage system components required to maintain the write cache long enough for it to be stored to the vault to complete. This process is sometimes called *a dump* to the vault.

SPS readiness testing

The VNX's SPS system is tested for readiness in two ways: on storage system power up and periodically (weekly) during operation. Both tests verify configuration and correctness of power cabling and ensure that the SPS battery has a sufficient charge to support the connected SP and storage system components required to maintain the write cache long enough for the dump to the vault to complete. The weekly SPS test is not conducted if it will cause the write cache to be disabled. The time of the weekly test is configurable through Unisphere. The interval is not configurable.

VNX AC power failure behavior

The VNX series with its persistent cache continues operation though a wide range of failures. (See "System drives and write cache availability" on page 149.) The storage system's behavior in an AC power failure is dependent on the failure cause and the number of SPSs installed. The AC power failure scenarios are power lost to:

- Both SPSs
- One SPS in a dual SPS system
- Only SPS in a single SPS system
- The non-SPS-protected-SP in a single SPS system

AC power loss to SPSs: This is the most serious condition. An example of this is a complete data center AC power mains failure. When AC power is lost, all SPSs report "Battery Online" to the operating environment (OE). The OE regularly polls the SPS status for a short period to determine if the failure is an intermittent power loss or a complete outage. If all SPSs continue to report "Battery Online" through the polling period, the dump of the write cache to the storage system's vault is started. After the

write cache is written to the vault, the OE continues to check the SPS state. If "Battery Online" state persists, the DPE or SPs and DAE O/S are powered off. When AC power returns, the SPs and Bus 0 Enclosure 0 are automatically powered up and operation is restored.

AC power loss to one SPS in a dual SPS system: This condition may occur if one of the two data center AC mains providing power to the storage system fails. As in the full AC power loss scenario, the SPS reports "Battery Online" to the connected SP and the peer SP, which is still running on AC power from its SPS. In this case, the SPS with no AC power input is shut down by the OE to prevent unnecessary battery discharge. Both SPs continue operation powered from the power supplies connected to the SPS receiving AC power. The storage system's write cache is still enabled.

AC power loss to the only SPS in a single SPS system: The SPS reports "Battery Online" to its connected SP and the peer storage processor, which is still running on AC power. Note that the peer has no connected SPS. If the SPS reports "Battery Online" through the polling period, the write cache dump to the vault is started. After the write cache dump completes, the storage system continues to operate without write cache which may result in a reduced level of performance. This continues until AC power to the SPS is restored and the SPS battery is fully recharged.

AC power loss to the non-SPS-protected SP in a single SPS system: The SP reports AC power fail to the OE, but the storage system continues to operate on the power from the peer SP's power supplies. Write cache remains enabled. Note that the VNX 5100 and VNX 5300 with a single SPS installed is an exception. The power supplies for each SP are for its own SP. They are not redundant with the peer SP's unit(s). They are redundant to themselves through the second AC main.

Disk array enclosure

Disk array enclosures (DAEs) offer disk-drive expansion to the SPE or DPE. There are several types of DAEs. The VNX series DAEs hold up to 60 drives. In addition, two drive form factors: 2.5-inch and 3.5-inch are supported. The two form factors require different DAEs. All drive counts are not supported by both drive form factors.

The number of DAEs in a storage system is model dependent. DPE-based storage systems do not require any DAEs. SPE-based storage systems require at least one DAE.

DAE O/S

On non-DPE VNX storage systems (see Table 1 on page 32), the DAE housing the storage system's *system drives* is called the *DAE O/S*. The system drives are drives containing the operating environment's file system. Note that Bus 0 Enclosure 0 refers to the DAE housing the storage system's system drives, which includes the system drives.

DAEs and storage

DAEs provide redundant connections both to drives and storage processors. Drives (mechanical hard disks or flash) receive their I/O requests through *ports*. All drives

are dual-ported. Each drive in the DAE is connected to a SP's SAS back-end port. The dual-ported storage can receive I/O requests from either or both SPs at any time.

DAEs internal connections to drives and the storage system's back-end ports are SAS protocol-based. Any type drive: SAS, Flash, or NL-SAS may be hosted in a DAE (or DPE) with any other type drive.

For example, in Figure 9 on page 42, Bus 0 connects SP A and SP B to hard drive 0 in DAE0.

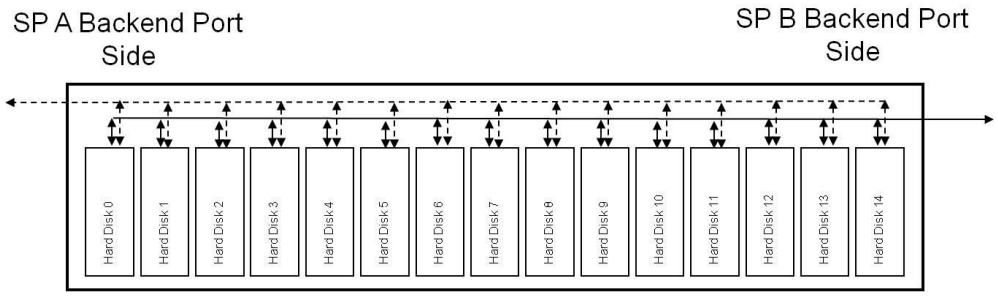

Figure 9 VNX 15-drive DAE with dual-ported storage connection to the SAS back-end ports

A VNX series DAE is supported by link controller cards (LCC). There are two LCCs per DAE. An LCC is the interface between the back-end buses, the DAE, and the drives. (See the Drives and DAEs section for additional information.)

X-Blade enclosure

The X-Blade is a network and storage interface device for file storage. Sometimes it is referred to as the *Data Mover* or *NAS head*.

Control Station enclosure

The Control Station is a management interface device for file storage.

Physical configuration example

Figure 10 on page 43 shows a front (left-side) and back view of a VNX5300, an entry-level storage system configured for a NAS workload.

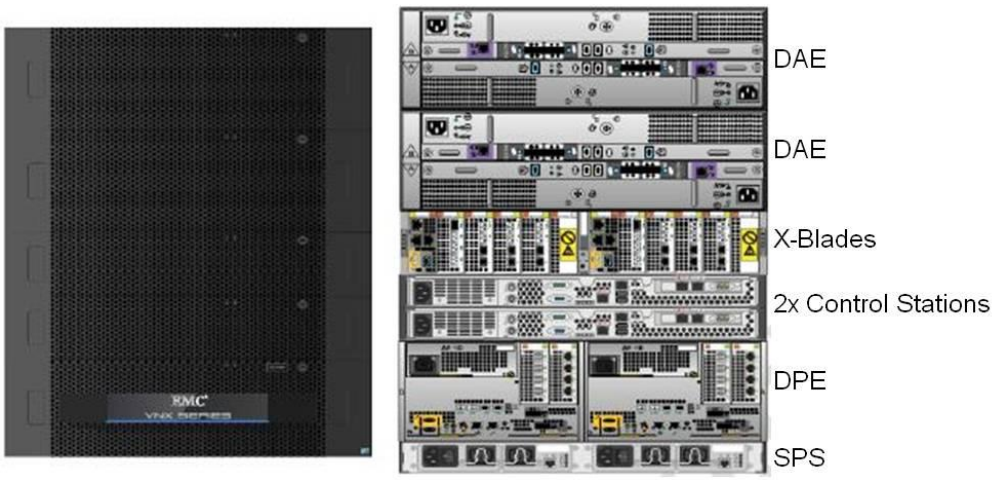

Figure 10 VNX5300 example configuration

The figure shows a VNX5300 with seven enclosures. This configuration totals 14U, or about 20 inches (52 cm) in height. From the bottom going upward, the enclosures are:

- Dual standby power supplies (SPS)
- DPE with dual storage processors
- 2x Control Stations
- Dual X-Blades
- 2x DAEs (3.5" drive form factor for 15-drives)

Note the modular architecture. The storage capacity of the system shown may be increased by adding additional DAEs. The maximum number of drives for a VNX5300 is 125. An additional five 15-drive DAEs can be added to this configuration. If a larger storage capacity or additional storage processor capability is needed, the DPE can be upgraded to a VNX5500 model DPE.

Hardware documentation

To become more familiar with VNX hardware, we recommend reading the *Introduction to the EMC VNX Series — A Detailed Review.* This document is downloadable from Powerlink.

Chapter 7 VNX Logical Architecture

This chapter presents these topics:

The following sections describe the logical architecture of the VNX storage system.

The VNX is logically divided into the following sections:

- Front-end ports
- Storage processors
- Read/write cache
- Back end
- FAST cache (optional secondary cache)
- Storage

Figure 11 on page 46 shows the relationship between these major logical components. The order of presentation generally follows this diagram, proceeding from left to right.

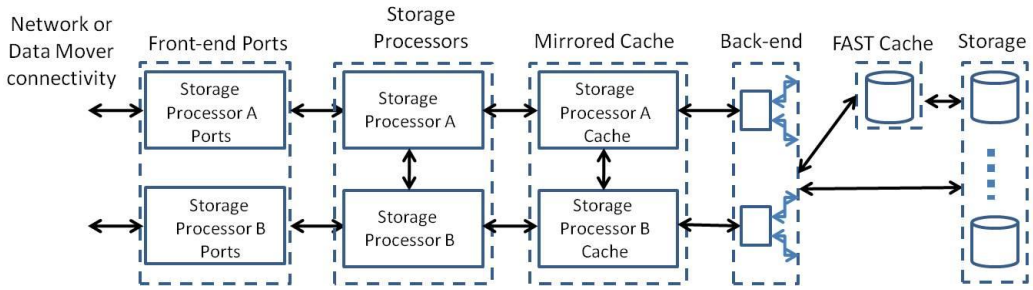

Figure 11 Storage system conceptual block diagram

Front-end ports

Each storage system has its front-end *ports*. The port refers to the physical cable connection to the network. The front-end ports may be directly connected to the data mover, directly connected to hosts, or connected to hosts through a network in a SAN environment.

VNX front-end ports

The VNX series may be configured with: UltraFlex™ I/O modules adapted to the following physical protocols:

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

iSCSI ports only support the iSCSI protocol despite utilizing a standard Ethernet port. Both the hardware model and the user interface also refer to them as iSCSI ports rather than Ethernet.

The number and type of ports standard to the storage system are model dependent. All VNX models have provision for adding additional ports, or ports that are not part of the basic configuration. The higher-end VNX models are capable of hosting a larger number of ports than the lower range models. Generally, the fewest number of

front-end ports needed to meet the workload's I/O performance and availability requirements should be used, leaving as many expansion slots free for future expansion as possible.

The number and type of ports required typically depends on the application's workload and host connectivity. In some cases the front-end port requirements may be dictated by the throughput or bandwidth of the dominant workload. Some of the front-end ports get reserved for MirrorView automatically.

I/O modules are always installed in pairs — one module in SP A and one module in SP B. Both SPs must have the same type of I/O modules in the same slots. Additional I/O modules can occupy any available slots.

Note that the Fibre Channel, iSCSI, and FCoE standards continue to evolve, particularly with regard to bandwidth. The modularity of the UltraFlex port allows for the retrofitting of higher-bandwidth connections as the standard develops.

Port location

Depending on the model of VNX, I/O ports may be located within the DPE or SPE. This information is also found in the product documentation.

Management and service ports

VNX series storage systems have two additional Ethernet ports per SP that are not part of the front end. These are the *management* and *service* ports. These ports are used exclusively for access and management of the SP through Unisphere. They do not support the iSCSI protocol, and cannot be used for workload data communications.

Auto-negotiation

If supported, a port's speed may automatically adjust to a lower bandwidth switch or DAS connections with a resulting loss of bandwidth. This automatic adjustment is called *auto-negotiation*. In auto-negotiation, network peers exchange their link-level protocol capabilities to provide compatible service. Auto-negotiation is performed by many front-end ports by default. Not all VNX front-end ports support multiple speeds and thus cannot support auto-negotiation.

For example, the 10 Gb/s iSCSI and 10 Gb/s FCoE ports do not auto-negotiate to lower speeds.

Fibre Channel ports

The Fibre Channel ports communicate either through a Fibre Channel SAN or to host(s), or directly to a host. Hosts must have a Fibre Channel HBA. These ports can also be directly connected to the Fibre Channel ports of other storage systems or the X-Blade.

The Fibre Channel front-end ports on the VNX series are 8 Gb/s optical-interface standard.

The standard Fibre Channel front-end ports on the VNX series use optical OM3 cables as their standard interface. These cables are sometimes referred to as M5E .

With optical cabling, the ”OM” in the cable type refers to the quality of the optical fiber. It is called the Bandwidth-distance product (BDP). This is measured in Megahertz/km. The higher the number, the more data that can be carried over a greater distance.

For example, OM4 can support 10 Gb/s over 550 meters, while OM3 can only support that rate over 300 meters.

Fibre Channel ports automatically adjust to the highest speed supported by all components. Lower bandwidth switches or direct connections are supported with a resulting loss of bandwidth. Not every protocol speed may be negotiated.

For example, 8 Gb/s Fibre Channel ports does not auto-negotiate down to 1Gb/s speed.

File and MirrorView support with Fibre Channel ports

On-board ports are reserved for Fibre Channel connection between the storage processors and the File X-Blades. In addition, MirrorView for ports data replication are likewise assigned by default. There is no sharing of port usage between MirrorView and X-Blade ports. These ports assigned automatically on the first power-up of the storage system. However their assignment can be manually overridden for all-block or no MirrorView workload storage systems.

iSCSI ports

The iSCSI ports communicate either through an Ethernet SAN to host(s), or directly to a host’s iSCSI HBA or network interface card (NIC). These ports can also be directly connected to the iSCSI ports of other storage systems.

The iSCSI ports communicate through a gigabit Ethernet (GigE) or 10 GigE LAN to hosts. The 1Gb/s iSCSI and 10 Gb/s iSCSI ports are on separate modules. Both the 1 Gb/s and 10 Gb/s iSCSI ports are an optional purchase.

The standard 1 Gb/s iSCSI front-end ports on the VNX series use the standard CAT 6 GigE-copper cabling to interface to the LAN or hosts. The 10 GigE iSCSI interface is either copper or optically cabled. The copper interface is active twin axial cabling, commonly called *Twinax*. Optical OM3 cables with LC connectors are the typical interface.

GigE ports automatically adjust to lower bandwidth switches or direct connections with a resulting loss of bandwidth. Some restrictions apply as to how low an iSCSI port can auto-negotiate downward.

FCoE ports

The FCoE ports communicate through an Ethernet SAN to host(s). The FCoE ports cannot be directly connected to a host or to the ports of other storage systems.

The FCoE ports are capable of 10 Gb/s communication only. A 10 Gb/s Converged Enhanced Ethernet (CEE), full duplex network is required to support lossless communication in a Fibre Channel over Ethernet (FCoE) environment.

A host must have a Converged network adapter (CNA) which is capable of FCoE protocol transmissions. The EMC Support Matrix identifies the current list of supported adapters along with (any) software requirements. An alternative is for the host to have a Fibre Channel HBA and to use a network switch to convert the FCoE transmissions to Fibre Channel.

The FCoE front-end ports on the VNX series use the same 10 GigE copper or optical cabling as the 10 GigE iSCSI port. The FCoE ports do not auto-negotiate downward from 10 GigE.

Front-end port performance

For small I/O size, the IOPS performance of Fibre Channel and iSCSI ports is similar through most of their performance envelope. However, Fibre Channel ports provide higher bandwidth than iSCSI ports. In addition, usage of iSCSI results in slightly higher storage processor CPU utilization.

The FCoE ports have similar performance to 4 Gb/s Fibre Channel.

The difference in IOPS performance between the port types depends on the protocol and how efficient the ports are at de-queuing I/O. Fibre Channel ports are more efficient at this than iSCSI.

However, unless the workload is bandwidth intensive or the I/O is large (> 64 KB), the performance differences between the Fibre Channel and iSCSI ports are minimal. Examples of bandwidth intensive workloads include rich media, backup-to-disk, and DSS.

Fan-in

Fan-in is how many hosts use a single port. The maximum number of host HBAs that can connect to a VNX storage system, called *initiators,* is model dependent. While certain models support greater than 256 initiators per SP, a single port cannot exceed 256 initiators. However, knowing how many hosts can attach to a port is an important consideration.

Initiators-per-port estimate

The initiator limit is not enforced at the per-port level. It is possible, although unlikely, to oversubscribe a port. Typically, a single host's I/O requirements are modest compared to a port's capabilities. Yet many hosts can be fanned into a single port.

The quick estimate of hosts into a port is performed as follows:

host port IOPS = port IOPS / host IOPS

host port bandwidth = port bandwidth / host bandwidth

For example, assume a type of production host requires 250 IOPS of throughput and 10 MB/s of bandwidth. Assume only one path per storage processor for this example.

A single 4 Gb/s Fibre Channel port can handle 360 MB/s of bandwidth, and 50k IOPS. A single GigE iSCSI port can handle 80 MB/s of bandwidth, and 10k IOPS.

How many of the described hosts can be fanned in to a single 4 Gb/s Fibre Channel port? How many servers can be fanned into a GigE iSCSI port?

- Server IOPS: 250
- Server bandwidth (MB/s): 10
- 1 GigE port IOPS: 10,000
- 1 GigE port bandwidth (MB/s): 80
- 4 Gb/s Fibre Channel IOPS: 50,000
- 4 Gb/s Fibre Channel bandwidth (MB/s): 360
- Fibre Channel hosts per port IOPS = 200 = 50,000 IOPS / 250 IOPS
- Fibre Channel hosts per port bandwidth = 36 = 360 MB/s / 10 MB/s
- iSCSI hosts per port IOPS = 40 = 10,000 IOPS / 250 IOPS
- iSCSI hosts per port bandwidth = 8 = 80 MB/s / 10 MB/s

In both cases, bandwidth is the limiting factor. In this example, 36 hosts can be attached to a Fibre Channel port, or eight hosts can be attached to an iSCSI port.

Front-end port queues

Front-end port queues hold I/O that is being handled or awaiting dispatch to the storage processor. The front-end port queue is managed by the storage processor owning the I/O port.

The number of outstanding I/Os per port has a direct effect on its performance. I/O to a front-end port originating from several host HBAs can be heavy and be very bursty. To accommodate the I/O, the front-end ports are a queued device.

Each arriving I/O takes up a queue position. I/Os are taken off the queue and processed by the storage processor. The number of queue entries ready for use is called the *port queue depth*. Fibre Channel, FCoE, and iSCSI ports have the same queue depth.

If I/Os arrive at a port faster than they can be handled by the storage processor, the port queue fills up. 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 depends on its O/S. However, the queue-full condition always adversely affects throughput.

Storage processors (SPs)

VNX's SPs are special-purpose processors optimized for I/O throughput. They have CPU, memory, and I/O resources to manage.

Every VNX has two SPs. The SPs are dual redundant. This means that either SP may take-up the workload of its peer, in the case of its peer's failure.

Operating Environment (O.E.) Block is the Storage Processors specialized operating system.

Each VNX SP has a single CPU. Each CPU has two or more CPU cores. Higher-model VNXs have a greater number of higher-speed cores than lower models. The greater number of cores allows the storage system to perform more functions at the same time. The higher the speed of the core determines how quickly it can execute those functions.

The SPs each have their own memory and I/O resources. Higher-model VNXs have a larger installed memory capacity than the lower models. In addition, SPs share memory resources with their peer SP. The most important use of memory on the storage system is the read/write cache. Cache is memory used by the storage system to reduce the response time of read and write requests from hosts.

Storage Processor CPU

The VNX's CPU resources depend on the model.

All VNX-series SPs are multiprocessor-based because SPs have *multi-core* CPUs. A *core* is a processor unit on a CPU chip. This is an efficient design for multi-processor (multi-CPU) computer-based device designs. Clustering the CPU cores on a single processor chip speeds communications between the separate CPU cores by sharing local on-chip bus and memory resources and reducing the physical distance between them. A single chip also consumes less circuit board real estate, and allows for consolidating power and cooling of the CPUs on the circuit board.

VNX entry-level and midrange level storage systems have fewer cores, but always at least two. Enterprise-level storage systems have a larger number of cores, at least four or more. The CPUs on different models have different clock speeds. The cores on a CPU all run at the same speed, although, the cores that are not actively in use can have their speed lowered to reduce their consumption of power and generation of heat. The higher the clock-speed of the CPU the more instructions it can execute per unit time.

There is a trade-off between number of cores and the speed of cores. If fewer simultaneous functions are being performed, fewer, faster cores are preferable. If a large number of simultaneous functions are being performed, more, but slower, cores may result in faster execution. Because entry-level VNX's have fewer cores per CPU than the higher models they are less extensible than the midrange models.

CPU performance

An important SP performance metric is SP utilization. This is a measure of the SP's CPU utilization.

Ideally, in a production environment with strict requirements for high availability and consistent performance, the SP CPU utilization should be around 50 percent. This way, if an SP failure triggers an SP failover, the peer SP can easily accommodate its own processing load and the failed-over load of its peer.

Note that the SP Utilization as reported through Unisphere is the average of all CPU core utilizations. It does not reflect individual core utilizations of the multicore CPUs.

Memory

Each SP has its own memory. All SP memory is error correction code (ECC). It performs error detecting and correcting to ensure the accuracy of its contents.

The SPs of the different VNX models have different amounts and speeds of installed memory. Entry-level storage systems have less and slower memory than higher-end models. Generally, the larger the memory capacity and the faster the memory of the SP, the more functions it can provide with a higher level of performance at the same time.

System memory is divided into SP memory and cache memory. Write cache memory is mirrored to the peer SP. Read cache is not. Figure 12 on page 52 shows a conceptual view of the storage system's memory.

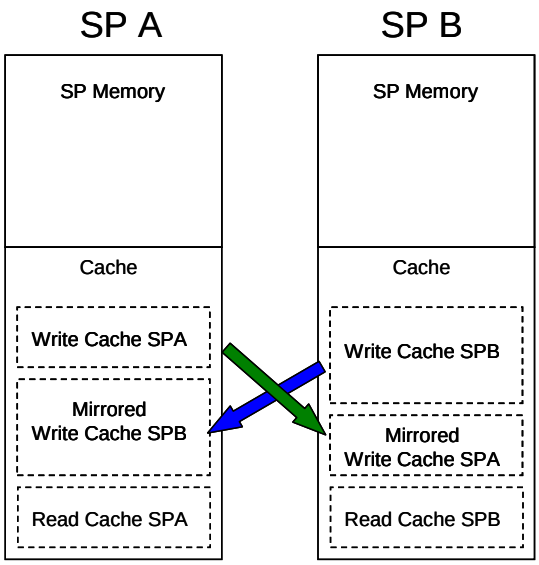

Figure 12 VNX memory: conceptual view

SP memory

A portion of a storage processor's memory is devoted to the operating environment (Block OE), the Unisphere GUI, features and installed applications. Features include FAST Cache, FAST VP, thin provisioning, and LUN compression. Installed applications include replication applications such as SnapView, MirrorView, and SAN Copy.

The amount of memory dedicated to SP memory depends on the model, the number of features, and the applications installed. The greater the number of features and installed applications, the larger the portion of the available system memory allocated to the SP memory.

SP cache

The storage system's read/write cache is a contiguous portion of the SP's memory that is separate from the memory used by the OE for systems software and applications.

The amount of cache memory available on a VNX is model dependent. Cache capacity scales within the series, entry-level models having less cache memory than higher models.

Cache itself is logically divided into two regions: *read cache* and *write cache*. Read cache is for data that is held in memory in anticipation of its being requested in a future read I/O. Write cache stores write request data waiting to be written to a drive.

The write cache partitioning applies to both SPs. Read cache can be partitioned independently for each SP. The great majority of all cache memory is allocated to write cache, with the remainder to read cache. This is because it is less likely for a read to find its I/O already in cache outside of workloads performing sequential read I/O. It is also important *not* to run out of write cache.

The capacity of the read and write caches can be adjusted to more closely accommodate specific workloads or to spikes in demand. However the total capacity of both the read and write caches cannot be greater than the model's total maximum SP cache capacity. The capacity of the cache is also dependent on the capacity of the SP memory and on which optional software applications are installed. For a user to re-allocate read or write cache memory, the caching function must be disabled. This results in a temporary decrease in storage system performance, until the cache is re-enabled.

In addition to an SP's own read and write cache, cache memory contains a mirror copy of its peer SP's write cache. This is an important availability feature. Note that the amount of memory useable as write cache at any time on an SP depends on the write cache usage of its peer SP, the amount of memory allocated to write cache is configured per system, and is the same for both SPs.

Cache pages

All cache is organized into pages. Incoming I/O is placed in the page. A page is the smallest unit of cache allocation. If the I/O is smaller than the page, more than one I/O may be made to share the same page when I/O has contiguous LBA addresses. This is an efficient use of the cache pages. As the I/O is executed (sent to the drives and acknowledged), its page is reused for the next I/O. A write cache page whose I/O has not yet been written to storage is called a *dirty page*.

Read cache operation

When a host read request is received, the storage system checks its read and write caches first. If the data not in either cache, it must be read from storage. After the data is read, it is held in the read cache for subsequent use.

Read requests that find their data in cache (a *cache hit*) consume the least amount of storage system resources; have the highest throughput, and the lowest response time. However, reads that are not found in cache (a *cache miss*) have a higher response time because the data has to be retrieved from a drive.

Write cache operation

The storage system's write cache is a mirrored write-back cache. This means for every write, the data is stored in cache and copied to the peer SP, and then the request is acknowledged to the host. In this process, write cache is duplicated *(mirrored*) between the storage system's SPs to ensure data protection through redundancy. In addition, requests are acknowledged before they are written to disk. Acknowledging the request from write cache decouples the response time of writes from disk speed and RAID effects, which improves performance.

The availability feature of mirroring write cache has a slight effect on performance. The mirroring delays the host I/O request's acknowledgment until it completes. The memory usable by an SP for write cache is also reduced by having to maintain a copy of its peer's cache. In bandwidth-intensive workloads, the mirroring process may limit overall storage system write bandwidth.

Back end ports

The VNX series implements a 6 Gb/s Serial Attached SCSI (SAS) back end. The SAS protocol is an evolution over the Fibre Channel protocol used in the back end of legacy EMC CLARiiON midrange storage systems.

The VNX's SAS expanders and links constitute the back end. Technically, the back end 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 minimal latency while still providing the same bandwidth for increasing workloads.

Figure 13 on page 55 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.

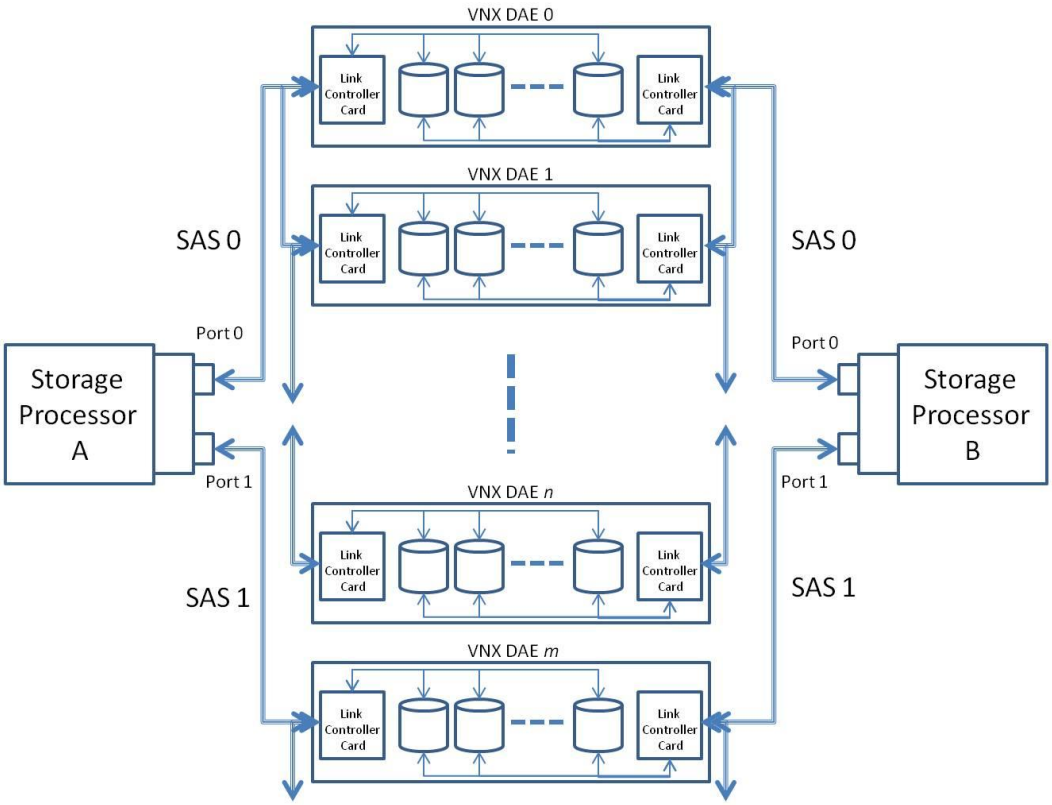

Figure 13 VNX back-end conceptual diagram

The maximum number of VNX enclosures (DPE, SPE, and DAEs) per port (shown in the Figure 13 as *n*) is 10. The maximum possible number of drives per port hosted by the enclosures is 250.

The addition of an enclosure with more drive slots than the bus or system model maximum is not supported.

For example, if there are already 240 drives on a bus and one more enclosure is added, it is not brought online.

a.k.a. “buses” and “loops”

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

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), as shown in Figure 14 on page 56. Ports support links to devices. A device is typically an SAS drive, but can also be another expander. Wide ports support multiple SAS links (sometimes called *lanes*).

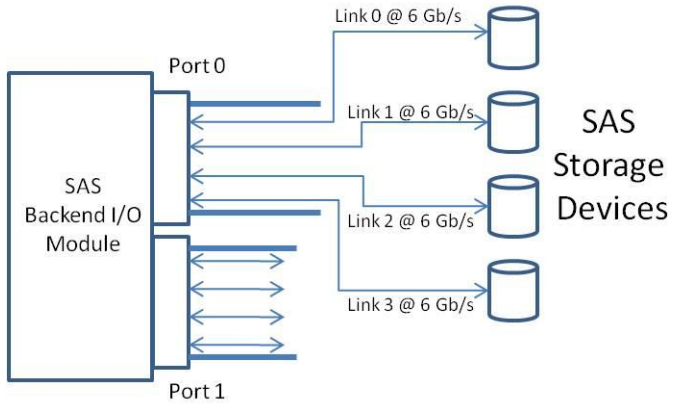

Figure 14 VNX SAS Back-end port conceptual diagram

Each VNX port is made up of four links. Each link is capable of 6 Gb/s. That is a nominal 2400 MB/s per port. However, there is about 2200 MB/s per aggregated port after taking into account arbitration delays and framing. Note that the practical link bandwidth is dependent on the storage device at the end of 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 workload and storage processor model. This architecture can produce 2.4 GB/s between expanders. It can reach about 750 MB/s on a link with an ideal workload.

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 drives at the same time.

Note that the VNX back-end port has more than twice the bandwidth of legacy EMC CLARiiON buses. In addition, there is increased parallelism in the back end. The older-technology CLARiiON bus was only capable of addressing a single drive at a time; the VNX back end can address four at the same time.

The wide ports also provide redundancy. Loss of any individual port lane degrades performance, but it 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 in the DAE, one connected to each SP. See Figure 13 on page 55. 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.

The utilization of its bandwidth has an effect on port performance. As the back-end port approaches its full transfer capacity, it is said to *saturate*. It is easier to saturate a back-end port with high bandwidth workload, but it is much more difficult to do with a high-throughput workload. The addition of more load does not scale linearly. The result is increased response times for drive access to the port.

FAST Cache

The VNX series supports an optional performance-enhancing, feature called FAST Cache. The FAST Cache is a pool of flash drives configured to function as a secondary I/O cache.

FAST Cache overview

FAST Cache reduces the host response time for both read and write I/O.

The FAST Cache is a combination of hardware and software. It is installed as an optional feature. The feature's software monitors the storage processors I/O activity for blocks of addresses that are being read or written multiple times over a specific interval from storage. When it detects this type of I/O activity, it promotes those blocks into the FAST Cache. Once a block has been promoted, FAST Cache handles the I/O to and from that block.

The FAST Cache is provisioned with, flash drives. Flash drives have a very low response time. (See" Physical" on page 61.) The FAST Cache's flash drives are located in DAEs or the DPE. However, conceptually, they are located "in front" of the drives typically considered to be mass storage, due to the function of the feature's software. Entire flash drives, allocated in pairs, are required for provisioning a FAST Cache.

When the activity to a block of addresses abates, a block of addresses may be *evicted* to make room within the cache for a block of addresses with a higher usage. In that case, the FAST Cache sends its cached writes to the main storage, typically mechanical hard drives.

FAST Cache reduces read as well as write activity to the back end. A FAST Cache can improve the performance thin LUNs within Virtual pools. It also allows the storage processor's write cache to flush faster, because it is flushing to high-speed flash drives. This allows the primary read/write cache to absorb a greater number of non-FAST Cache write I/Os. Overall system performance is improved.

FAST Cache can show benefits for the entire storage system and accelerate those workloads with a high locality of reference profile. However, the FAST Cache is a resource that must be managed. The use of FAST Cache should be limited to LUNs that benefit the most from its use.

The ideal I/O profile for FAST Cache is random I/O, particularly I/O with a high percentage of reads. Avoid FAST Caching LUNs with small-block sequential I/O. Logs are the typical example of a LUN receiving small-block sequential I/O.

The capacities of the FAST Cache are model dependent and range from 100 GB up to 2 TB. However, application data sets can be quite large in capacity and have a very wide range of locality. Be careful not to have too small a FAST Cache.

For example, a very large database may contain a 2 Petabyte data set. A low locality of just 5-percent would exceed the capacity of the largest FAST Cache.

The FAST Cache reduces the load on mechanical drives and as a result improves overall storage system performance where the workloads have good locality of reference and a compatible I/O profile and have been sized correctly for FAST Cache.

Storage

Storage is the mechanical hard drives and memory-based flash drives, along with their housing enclosures.

Storage enclosures

An UltraPoint™ DAE is the hardware enclosure providing the physical protection, organization, and connectivity to storage system drives. Likewise, a DPE provides the same drive housing function, but includes the storage processor hardware in the enclosure. (See" Disk" on page 41).

UltraPoint DAEs used in the VNX-series come in several different drive slot capacities and drive form factors. A *slot* is the opening in the DAE into which the drive is enclosed.

Dual-ported drives

All drives are dual ported. They have two independent SAS ports per drive designed to connect to separate back-end ports. Each drive port connects to a separate LCC in its DAE, which in turn is connected through a separate back-end port to an SP. (See Figure 14 on page 56.)

Drive and back-end port bandwidth

The back-end bus has more available bandwidth than a mechanical hard drive's maximum external transfer speed.

For example, a 6 Gb/s SAS back-end bus has greater than 2100 MB/s of bandwidth. The typical 15K rpm mechanical SAS hard drive has a sustained transfer rate of between 70 and 100 MB/s. (The burst rate is much higher, but still cannot saturate the link.)

However, more than one drive can be addressed at the same time. The differences in the available bandwidth between the bus and the drive are mediated through the SP's queuing mechanism for I/O commands and data sent to the drive.

Each drive has two I/O queues, one per storage processor, which is one per drive port. The queues are maintained in software by the SP. SPs send commands and data to a drive and the command is added to the queue. Drives execute the command and acknowledge it. When the command is acknowledged, it is removed from the queue. Only a limited number of commands are sent per drive per SP. This number is set by the number of possible queue entries. The number of possible queue entries is limited to ensure optimal performance, even though the drive may be capable of handling additional commands, albeit with a response time penalty. Commands are sent until the SP's drive queue fills. If a drive queue is full, commands are withheld by the storage processor until an enqueued command completes. Serialization rules are enforced by the storage processor issuing the commands to prevent ordering problems with the drive's internal I/O execution.

File logical devices

In addition to the block storage discussed earlier, the logical architecture of file storage includes X-Blades and Control Stations.

X-Blades

The X-Blades are independent file servers for transferring requested files to client computers.

Operating Environment File is the X-Blades specialized operating system. It is optimized for performing file operations between the NAS and the underlying VNX storage system. It supports the NFS and CIFS protocols.

The X-Blades are installed in redundant pairs, where one is the active primary, and the other is the standby. The standby becomes active in the event of a primary's failure.

Control Stations

Control Stations are dedicated computers for managing the NAS environment. Management includes installing, provisioning, and monitoring the X-Blades and the File environment.

Chapter 8 Storage Objects

This chapter presents these topics:

The VNX-series has two types of storage objects:

- *Physical* storage objects — Drives such as mechanical hard drives and flash drives.
- *Logical* storage objects — Storage pools, RAID groups, LUNs, and metaLUNs.

Physical storage objects

Both mechanical and semiconductor-based drives are used to persistently store data. Hosts read them to retrieve them, and write to them to create, modify, or delete them.

Basic mechanical hard drive terminology

A hard drive is a computerized device. A microprocessor-based *disk controller* controls the operation of the drive. Among other things, it positions the read/write head in response to I/O requests, executes performance optimizations, data protection functions, and diagnostics.

A hard drive contains one or more rotating magnetically coated *platters*. The platters are connected together by a *spindle* that is turned by the drive motor. Sometimes hard drives are also referred to as *spindles*. Data is read from or written onto the platters by *read/write heads*. A drive has as many read/write heads as it has platters. The read/write heads are mounted on an *actuator* arm that positions the read/write heads over the location on the platter to be read or written. The movement of the arm is referred to as a *stroke*. A *full stroke* is a move from the outermost track to the innermost track adjacent to the spindle.

A *track* is a concentric ring-like region around the platter's spindle on which data is stored. A track is divided into *sectors*. A sector is the smallest individually-addressable unit of data. Sometimes sectors are referred to as *blocks*. A drive sector typically contains 512 bytes of data. Many VNX configuration and tuning options are in *block-sized* units, which is 512 bytes. VNX drive sectors contain a total of 512 bytes of data and 8 additional bytes of metadata, totaling 520 bytes. This metadata is a proprietary data integrity feature of the storage system. *Logical block addressing* (LBA) is a mapping technique for addressing a sector independent of its physical track/sector location on the drive's platters. The lowest numbered LBAs are on the outermost tracks of the drive. The LBAs increase as they get closer to the innermost tracks.

Hard drives have physical *attachments* that use a specific *protocol*. The attachments are through the previously mentioned ports. All VNX drives are dual ported. Dual porting provides redundant connectivity. The VNX supports hard drives with the SAS protocol only.

Basic flash drive terminology

Flash drives have a simpler organization than mechanical hard drives. Flash drives are designed to emulate mechanical hard drives for compatibility with computer-based equipment.

Internally, a flash drive's hardware architecture can be divided into the following major areas:

- Attachment: Providing power and data to the flash drive
- Controller: Microcomputer managing drive functions
- Mass Storage: Nonvolatile memory storing data

Attachment

As with a mechanical hard drive, the attachment provides power and data to the flash drive. With the 6 Gb/sec SAS attach, 768 MB/sec is possible.

Controller

The controller is comprised of several components. It includes a microcontroller, RAM, and nonvolatile memory (NVRAM). The microcontroller manages all drive functions. It is important to understand that a flash drive is a computer-based device. The controller is typically a proprietary device, designed and fabricated to specification and licensed from a semiconductor manufacturer. It can have one or more embedded processor cores. In dual-core configurations, one core typically handles the I/O while the other manages the storage. The flash drive's controller implements many of the drive's features in software. This is an important difference from mechanical hard drives. With flash drives, as long as the controller's CPU is fast enough, the biggest influence on flash drive performance is the architecture of its software. In addition, this software can be easily changed to add features, such as on-board encryption or to improve performance.

The RAM is used in the management of data being read and written from the flash drive as a cache, and for the flash drives operational programs and data. Typical RAM on flash drives ranges from 32MB to 256MB. The controller cache enhances the overall performance of the flash drive. Flash memory writes more slowly than it reads. The cache uses fast dynamic RAM (DRAM) to minimize the number of writes and improve response time to the slower flash memory of the drive. Write coalescing is one of the techniques employed within the cache. This is the process of grouping write I/Os and writing them in a single internal operation, as opposed to many smaller-sized write operations. In addition to caching, the RAM contains mapping tables and access history. Mapping tables correlate the internal data structure of the flash drive to the host file system's data structure. Access history is used to increase drive longevity through *wear-leveling*. Wear-leveling is an internal housekeeping process which averages the number of writes to the drive across all the drive's memory locations. This process extends the life of memory by avoiding over-use of locations.

Flash drive cache memory is nonvolatile. If the drive loses power, drive cache contents are kept and the drive's contents are not corrupted. Currently, nonvolatile RAM solutions to protect flash drive caches are battery back-up and super capacitors. Both technologies provide enough electrical power in a power fail situation to allow time for saving the cache's contents to the drive's non-volatile storage.

Mass storage

The NVRAM is used to store the flash drive's operational software and data when power is off. Not all flash drives have separate NVRAM for their operational software. Some models store their programs and data to the drive's mass storage. The NVRAM is addressed by internal channels. The channels divide the addressable storage into independently addressable blocks of storage. The more channels a flash drive has, the higher its concurrency. The number of channels and the amount of capacity addressable by a channel is drive model dependent.

Management of flash storage

Externally, the smallest addressable unit of data for a flash drive is a sector. LBAs are emulated by the flash drive for compatibility. However, internally, the smallest unit of data that can be written is the *page*. A page is a multiple number of sectors: eight sectors (4 KB), 16 sectors (8 KB), or 32 sectors (16 KB), depending on the drive model. The ideal I/O block size for a flash drive is its native page size. Empty, unused pages can be written to directly. Finally, flash drive memory must first be erased before it can be written to. The unit of memory that is erased at a time is the block, which is a multiple number of pages.

Writing an entire block at once is the preferred operation. All the pages within a block must be addressed before a block is written. In addition, only blocks that have been erased may be written. A drive with a large amount of empty blocks has very high performance. A portion of the drive's overall capacity is kept reserved for this purpose and not user accessible. (This capacity is not part of the drive's nominal capacity.) While empty blocks exist, a new page of a block is used for every write. When all of a drive's empty blocks are consumed, its performance decreases. This is called the "re-write penalty" or "drive aging." Even a drive that outwardly shows available storage capacity may no longer have empty blocks — all blocks have a combination of valid data and pages that are invalid (due to deletion or overwrite).

Once a drive has no empty blocks, pages marked invalid are reused. Invalid pages are separated from in-use pages to create blocks with only in-use pages and new empty blocks with pages available for writing. The blocks with invalid pages are erased and become the target of all new writes. This separation involves a read-modify-write process.

To create new empty blocks internal to the drive, the contents of one or more blocks are read, in-use pages are packed onto blocks of wholly in-use pages. Packing the in-use pages together allows the creating of empty blocks from the surplus pages that were marked invalid. The empty blocks are erased and used for all new writes. These extra steps are performed by the drive's controller and take time. Every write request to the drive is slowed by the need to perform block rewrites. Reads may be delayed by previously queued writes requiring a rewrite, or by waiting for blocks that are locked while they are consolidated. The creation of new blocks is continuous. It occurs when needed, and when the drive is idle.

Categorizing hard drives

Hard drives are broadly categorized by their:

- Capacity
- Speed
- Buffer Size
- Form factor
- Type

For example, a drive might be referred to as a 600 GB, 15,000 rpm, 64 MB buffer, 3.5-inch, SAS hard drive.

The flash-type drive shares many of the categories of mechanical hard drives. However there are exceptions. For example, flash drives do have the delays associated with seeks.

Capacity

Hard drives are frequently referred to by their capacity. Typically, capacity is in terms of either nominal gigabytes or terabytes of storage. Some examples are 600 GB or 2 TB.

The raw capacity of a mechanical hard drive is determined by the diameter of the platter, the number of platters within the drive, and the platter's areal density.

The raw capacity of a flash drives is determined by the amount of user nonvolatile memory installed in it. Flash drives actually host a larger capacity than reported. This is the capacity reserved for internal housekeeping, such as the previously described management of blocks. The larger the reserved capacity the better a flash drive's performance is under a sustained write workload. Reserved capacities range between flash drives, depending on the make and model. They are typically between 20 and 10 percent non-volatile RAM on-board above the reported raw capacity.

Areal density (mechanical drives only)

Areal density is the number of bits that can be stored on an area of a mechanical hard drive's platter. With a high areal density, a large amount of capacity can be put on a single platter. Areal density is measured in Gb/in^2 or Gb/cm^2. A high areal density would be about 625 gigabits per square inch. With current technology, this allows for up to 1 TB to be stored on a single 3.5-inch platter.

Increasing areal density was necessary to increase drive capacity and performance while maintaining the form factor (3.5 or 2.5 inch) and decreasing drive complexity, power consumption, and cost per GB. With higher areal density and more sensitive read/write heads, fewer platters are needed to reach a given capacity. Fewer platters in turn reduce the number of actuators and drive heads. This makes the drives mechanically simpler. At the same time, it also facilitates improved performance by allowing the drive head to access more data over shorter physical distances on the platter's tracks.

Integration scaling

Integration scaling is the increase in the number of bits that can be stored on an NVRAM chip used in flash drives. As semiconductor processing technology advances,

individual chips are manufactured with greater memory capacities on chips with the same area. This results in greater capacity flash drives in the same or smaller form factor. In addition, as integration increases, there is typically an increase in NVRAM chip performance. This is due to a decrease in the time needed to access individual pages.

Why capacity quotes can vary

Unfortunately, there is more than one way of quoting storage capacity. It can be:

- Decimal or binary, that is, in *decimal base* (base 10) or *binary base* (base 2). This is explained in this section.
- Capacity *after* the disk is formatted; this it its *usable* capacity, *or user-data* capacity.
- Capacity *before* the disk formatted; this is *nominal* capacity, or *raw* capacity. Formatting the disk uses up some of this capacity. This is the quoted capacity when drives are purchased.

It is not uncommon for new users to wonder why their host's O/S reports that a 1 TB drive has a formatted capacity of 931 GB. The explanation is that hard-drive manufacturers report capacity in *decimal gigabytes*, while host O/Ss report capacity in *binary gigabytes*.

Binary vs. decimal capacity

A byte is always eight bits. However, a kilobyte can be *decimal* (1000 bytes) or *binary* (1024 bytes), depending on the source of the report. The difference between their capacity measurements increases as the unit of measure increases. Table 3 on page 65 shows the difference between the common capacity measurements.

Table 3 Decimal versus binary capacity

Unit of measure	Binary bytes	Decimal bytes	Percentage difference
Kilobyte	1024	1000	2
Megabyte	1,048,576	1,000,000	5
Gigabyte	1,073,741,824	1,000,000,000	8
Terabyte	1,099,511,627,776	1,000,000,000,000	10

Note that in terabyte capacities, there is a 10 percent difference in the values. A computer O/S using a *binary* gigabyte to measure its available capacity shows a hard drive having a much lower capacity than the manufactures listed capacity, which is listed in *decimal* gigabytes.

The Unisphere storage management application reports disk capacity in both binary gigabytes and decimal bytes to avoid confusion.

Drive formatting and OE private space

When a drive is formatted, its sector size is permanently set. The conventional standard is 512 bytes per sector. Drives on a VNX storage system do not use

conventional sectoring. An additional eight bytes per sector is used for storing sector-level data-protection information. These additional bytes are used to verify and maintain data integrity. This 520-byte sectoring of a drive reduces the usable capacity of a hard drive by less than two percent. However, this small reduction in capacity has a large positive effect on availability. In addition, each drive in a VNX also has a fixed amount of space reserved for VNX system information.

Host LUN formatted capacity

Partitioning and formatting reduce the available capacity of a LUN. (See "Logical Units (LUNs" on page 78.) This difference in capacity is due to *metadata*. Metadata is data about the data. It is used to support availability features such as data integrity, recovery, and information security (confidentiality). It also includes file system information

The operating environment (OE) reserves a small amount of disk capacity for tracking tables and logs. This accounts for about 64 MB per drive.

Different file systems have different amounts of metadata overhead. For example, Linux file systems such as ext2 and ext3 have a smaller amount of metadata than Microsoft's NTFS.

Raw LUN storage

Partitioning and formatting a LUN reduces the available capacity before any user data is written to the drive. However, some applications do not require formatted drives. A raw device is a LUN without a file system, so it does not lose capacity to file-system overhead. Certain applications, such as Oracle, manage data on raw devices, which not only eliminates file system overhead but also improves transactional throughput.

Speed

Rotational speed categorizes hard drives into performance classes. Rotational speeds are measured in revolutions per minute (rpm). The most common rotational speeds for hard drives are:

- 15,000 rpm
- 10,000 rpm
- 7,200 rpm
- 5,400 rpm

The speed at which a disk drive spins (for example 15,000 rpm) is specific to each model disk drive. It does not change. The speed of other drives in the same enclosure or of the back-end port has no effect on the drive speed.

The rotational speed can be abbreviated using the K-notation. For example, a 15,000 rpm hard drive can be described as being 15k rpm.

Note that rotational speed does *not* apply to flash-type drives.

Buffer size

Drives typically have an on-board buffer used by their microcontroller to optimize their operation. The buffer is a managed memory space allowing faster request response time than a disk access. Sometimes the buffer is called the *disk cache*.

A drive's *burst* speed is the transfer rate it can achieve through its buffer's RAM. This speed is not affected by the mechanical operation of the drive, or the somewhat slower operation of a flash drive's NVRAM compared with the buffer RAM.

Typical buffer sizes are 16 MB, 32 MB and 64 MB, where the larger capacities can provide greater drive performance.

The advantage of this buffer is in read buffering. Read buffering is performed independently by the drive. The drive can prefetch reads into the buffer by itself. A read miss at the storage-processor level may be a read hit within the hard drive's buffer. This improves overall performance, especially in file systems where there are gaps between sequentially arranged file sections on the drives.

Except on flash drives and specific ATA rebuild scenarios, a drive in a VNX does not use its independent buffer for writes or write caching. This is for data integrity reasons. There is no battery backup for the buffer of mechanical drives. In the event of a complete power failure or drive removal, the buffer's data contents may be lost.

Form factor

Form factor describes an IT industry standardized set of dimensions for a drive. A mechanical hard drive's *form factor* is determined by its platter size. The form factor quoted is the diameter of the hard drive's platters in inches. There are 3.5, 2.5, and 1.8-inch form factor drives commercially available. The 2.5-inch form factor is sometimes called *laptop format*.

For physical compatibility, flash drives have the same form factor as mechanical hard drives.

VNX storage system's support drives in both the 2.5 and 3.5-inch form factor.

Type

The VNX series support the following types of drives:

- SAS
- NL-SAS
- Flash

These drives have different characteristics that make them more or less appropriate for different workloads.

SAS hard drives

Serial attached SCSI (SAS) hard drives have the highest performance, and availability of all VNX mechanical drives. SAS hard drives currently use the 6 Gb/s SAS protocol attachment. This has a theoretical maximum bandwidth of 600 MB/s. SAS hard drives are enterprise-level reliable hard drives.

The highest performance SAS hard drives are 15k rpm. In addition the VNX supports a more economical and modest-performing 10k rpm drives.

Near-line SAS (NL-SAS) hard drives

The NL-SAS hard drives provide the most economical storage solutions capacity-wise on the VNX with modest performance.

NL-SAS drives use the 6 Gb/s SAS standard attachment. This has a maximum burst speed of about 600 MB/s. As their name suggests, these hard drives are near-line reliability-level drives. Near-line has a somewhat lower MTBF than enterprise rated storage.

NL-SAS drives are available in 7.2k rpm. These drives have the largest capacities available on the VNX-series. They are very economical for workloads requiring large amounts of storage.

Flash drives

Flash drives are semiconductor-based drives. They have no moving parts and have very high performance and reliability.

Metrics such as average service time for flash drives are extremely low when compared to mechanical hard drives. For example, a flash drive can perform a small-block random read in about 0.1 ms. Contrast this to the fastest SAS mechanical hard drive, which would take about 4 ms. A flash drive performs a read 40 times faster.

For compatibility, a flash drive emulates a mechanical hard drive to the operating environment. Flash drive storage uses the SAS protocol to attach directly to the back-end ports. Flash drives have enterprise-level reliability.

With the current flash technology, read I/O is significantly faster than write I/O. Random reads have a significant amount of storage system read cache-misses. This makes flash drives ideal for workloads with a high percentage of random reads. As the I/O block size increases, and with sequential I/O, the performance of flash drives decreases. In addition, setting-up multiple LUNs on flash-based RAID groups owned separately by both storage processors is a useful way to take advantage of the flash-drives concurrency. Flash drives can also be an alternative to the use of short-stroked mechanical drives.

In addition to having high performance, flash drives consume much less power than mechanical hard drives.

Special considerations

All SSDs, regardless of their manufacturer, including the VNX's flash drives, are subject to changes in performance as a result of their state and load.

Flash drives emulate mechanical hard drives; however their internal operation is much different. They are internally random-access storage devices. Operations common to mechanical hard drives may degrade a flash drive's performance by imposing linear addressing on them. Operations to avoid include:

- File system defragmentation

- Linux media format and check
- Linux

Disk power saving (spin down)

The VNX supports an electrical-power saving feature called spin down. Spin down allows a RAID group's mechanical hard drives to be configured to enter an electrical-power conserving state when not in use. This state is called *standby*. A RAID group enters standby when it does not receive I/O for a predetermined amount of time. Drives in standby are periodically tested to ensure their readiness. When a host I/O request is made to a LUN with drives in standby, the storage system pends the I/O while it brings the RAID group's drives to readiness, and then the read or write request is satisfied.

In addition, drives not configured into RAID groups, such as hot spares and unbound drives, can be configured to be kept in standby until needed. Note that only drives qualified by EMC specifically for spin down may be spun down. Unisphere detects and presents candidate spin down drives to the user from the storage system's provisioned drivers. Drives in Virtual Pools and the storage system's system drives cannot be spun-down.

Logical storage objects

Provisioning is the term used to describe the process of logically configuring the storage system to meet its workload's need for efficient resource usage, capacity, performance, and data security. Logical storage objects include RAID groups, LUNs, metaLUNs, and Virtual Pools.

RAID groups

Drives are collected together into related groups of a single RAID level, called a *RAID group* (Figure 15 on page 69). The number of drives in the RAID group and its RAID level define its availability, capacity, and performance. Understanding the capability of RAID groups is an important fundamental concept of the storage system.

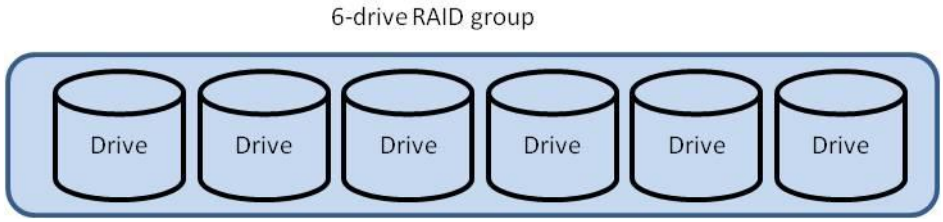

Figure 15 RAID group conceptual diagram

RAID

Redundant Array of Independent Disks *(RAID*) is the key technology for performance and availability. An understanding of RAID technology is needed to decide how many disks to use in a RAID group, understand the capacity they will provide, and estimate their performance in operational and degraded states.

RAID levels provide data protection through redundancy. Do not confuse redundancy with data integrity. While both are related to availability, they are different.

Storage redundancy is the ability of a storage system to continue providing access to data after a hardware failure. Redundancy is achieved through the duplication of drives, power supplies, multiple back-end port access to drives, and RAID protection schemes.

Redundancy is not a substitute for data backup or point-in-time data replication. Erroneous data, user-corrupted data, and data deletions are all redundantly stored on the drives. This "bad data" is still available (or not, in the case of a data deletion) in the event of a drive failure.

Data integrity is maintaining the accuracy of data while on or when going to or from the drive. Data integrity is handled on the storage system through its automatic error detection and correction mechanisms (see Chapter 11 on page 129). It is possible for drives to corrupt data without detection; for example through a media error on the drive's platter. On the VNX's drives, data integrity is maintained through the drive's 520-bit sectoring. Additional capacity per sector is allocated to metadata, which is used for error detection and the correction of many sector-level data errors. Error detection and correction are performed both as a continuous background process by the SNiiFFER service and on data access.

It is important to note that RAID implementations may differ between storage system vendors. A storage system administrator or architect should not make assumptions about a RAID level's operational characteristics based on experience with another vendor's storage system.

RAID levels

The VNX-series supports six RAID levels. Each RAID level has different capacity, performance, and availability characteristics. The two broad types of RAID are mirror and parity. A technique called striping applies to both RAID types. A striped non-redundant RAID type (RAID 0) and individual disk storage type are separately discussed.

Mirror RAIDs

Mirror-type RAIDs (Figure 16 on page 71) are used when random write performance and availability are more important than the cost of capacity. Mirror RAIDs create an exact copy of data on two or more drives to protect the data. The *primary* (drive) has the original copy of the data while the *mirror* (drive) has an exact copy of the data. The data is protected from the loss of either one or two drives in the RAID group.

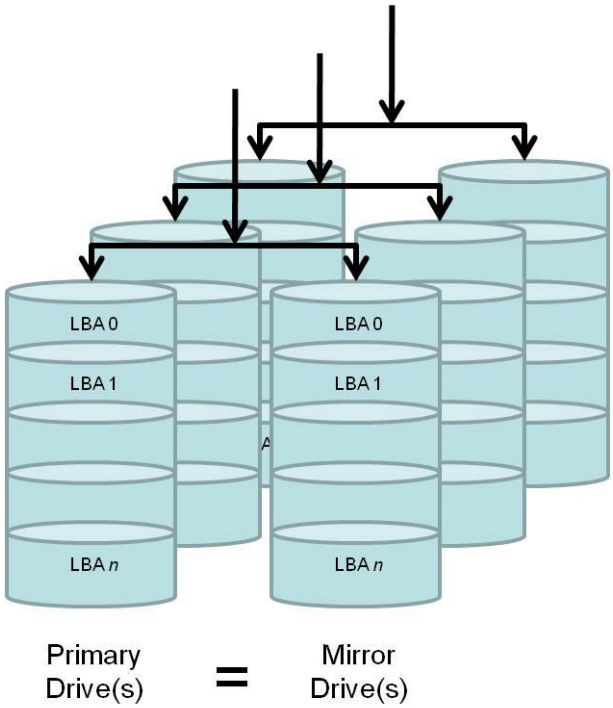

Figure 16 Mirror RAID conceptual view

Parity RAIDs

Parity is an algorithmic data protection technique where redundant data is stored to allow errors to be detected and possibly fixed. The parity metadata may be distributed across all drives, or be on a single drive. (See Figure 17 on page 71.) All parity RAID types are striped. A parity RAID stripes data across three or more disks. Depending on the RAID level, the data is protected from the loss of either one or two drives in the RAID.

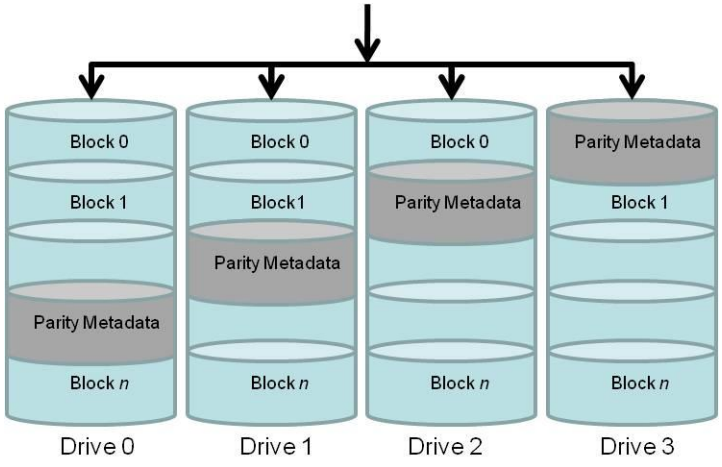

Figure 17 Parity RAID conceptual view

A striped RAID is a RAID level that provides the capacity and performance benefits of multiple drives, but no data protection.

Striping

Striping is the distribution of data across more than one drive to improve performance and increase available capacity. (See Figure 18 on page 72.)

Both bandwidth and throughput are increased through striping. This is because either a single large or multiple small I/Os can be performed in *parallel* to the disks of the RAID group. Parallelism involves a number of disks servicing a host's I/O request at the same time. More drives in a stripe can result in a higher degree of parallelism and a better transfer rate.

Likewise, capacity is increased by striping across multiple disks. Each additional drive added to the stripe adds to the total available storage capacity. This is also useful for data sets that require more capacity than a single drive offers.

The performance and capacity benefits of striping depend on the function of all the drives in the stripe. Higher-performance drives result in higher stripe performance. Higher-capacity drives provide a greater number of stripes.

The distribution of data across the stripe creates interdependency between the drives. This interdependency decreases reliability. If one drive in the stripe has a permanent data loss, all the data on all the devices is lost. The redundancy of data in parity and mirror-RAID levels mitigates this possibility. However, parity increases a RAID levels complexity, and in some cases decreases its performance. In general, striped RAID levels remain more vulnerable to data loss than mirror levels.

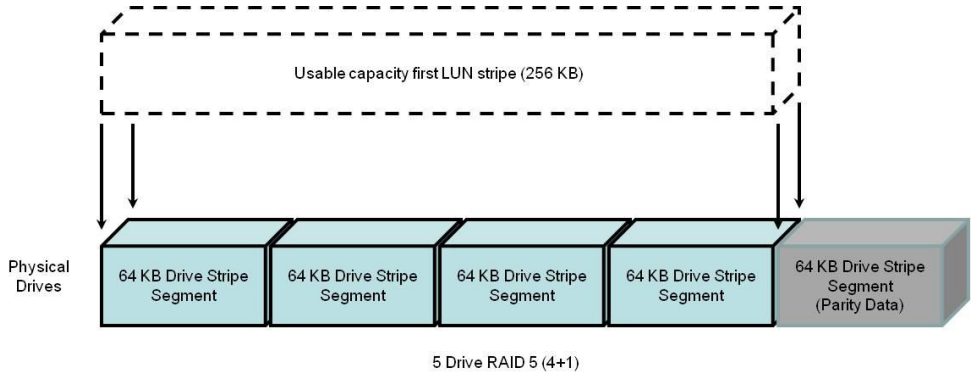

Figure 18 Parity RAID group stripe elements

Stripe elements

The stripe element is the amount of contiguous data stored on a single disk of the stripe. Stripe elements are measured in 512 byte blocks or in kilobytes (KB). The default stripe element size is 128 blocks, which is 64 KB. The stripe element size is *not* configurable through Unisphere. The default element size has been optimized for the operating environment. Attempts to "tune" it are likely to reduce performance.

Stripe size

The *stripe size* is the amount of user data in a RAID group stripe. This does not include drives used for parity or mirroring. The stripe is measured in KB. It is calculated by multiplying the number of stripe disks by the stripe element size.

For example, an eight-disk RAID 1/0 has a stripe width of four, with a stripe element size of 64 KB has a stripe size of 256 KB (4 * 64 KB). A five-disk RAID 5 (4+1) with a 64 KB stripe element size also has a stripe width of 256 KB drive.

Supported RAID levels

Table 4 on page 73 provides a brief summary of the RAID levels supported by the VNX-series.

Table 4 RAID levels summary description

RAID Level	Description	Minimum Disks
0	Striped unprotected data. Data is striped over a set of hard drives. Provides the same individual access features as the RAID 5 type, but does not have parity information. A single disk failure in the RAID group results in the data on the RAID group being lost.	3
1	Mirror-protected data. Provides data redundancy by mirroring its data onto another disk. This is data duplication. This RAID type provides high data availability at the greatest cost in disk space. This RAID level has a maximum size of two disks.	2
3	Striped, parity–protected data. Provides data redundancy using parity information that is stored on one physical disk in the RAID group.	3
5	Striped, distributed, parity-protected data. Provides data redundancy using distributed parity information stored on each drive in the RAID group.	3
6	Striped, distributed, double parity-protected data. Provides fault tolerance from two drive failures, meaning the storage system continues to operate with up to two failed drives. RAID 6 has the highest availability level.	4
1/0	Mirror-protected data striped for performance. Commonly called *RAID ten*, sometimes *RAID one-zero*. Provides high performance with high data redundancy but high cost of storage capacity.	2

The RAID level does *not* indicate the number of drives used to implement that RAID level. This is a common, mistaken assumption made by newcomers to the technology.

For example, a RAID 5 group does not necessarily have five drives in a stripe.

RAID group composition

Different drive types (flash, SAS, and NL-SAS) cannot be bound together in RAID groups. For example, flash drives cannot be bound into RAID groups with mechanical hard drives. Unisphere prevents the binding of unlike drive types.

Ideally, RAID groups should be created from the same speed (15k rpm, 10k rpm, and 7.2k rpm), and capacity drives. This is called RAID group *symmetry*. Having identical drives in a RAID group ensures the highest usable capacity, and consistent performance and availability.

If you mix drives of different capacities in a RAID group, the capacity of the smallest bound drive is used from each disk in the group to make up the RAID group's overall capacity. The full capacity of the RAID group's larger disks is unused.

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 private RAID group would have a raw user capacity of 1.2 TB (4 * 300 GB). Note that the usable capacity of the RAID group is halved by the inclusion of the single lower-capacity drive.

Likewise, mixing higher-rpm drives with lower rpm drives lowers the overall RAID group performance. I/O requests to the slower drive(s) take longer to complete than I/O to faster drives. This increases the average RAID group response time and the response time of multi-drive I/O.

For example, a 5-drive RAID 5 group is 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 are slower than to its peer drives. For random I/O, involving individual drives, the difference may not be noticeable. However, for I/O to the 10k rpm drive, and I/O involving full-stripe writes, the response is slower. The overall throughput of the RAID group is lower than a RAID group made-up of entirely of 15K rpm drives with the addition of the single slower drive.

RAID group symmetry also applies to Virtual pools. (See the Virtual Pools section.) For ideal performance and capacity utilization, all the drives in a homogeneous pool should be the same. In a FAST VP pool, all the drives in a tier should likewise be the same.

Minimum-sized RAID groups

Table 4 on page 73 gives the minimum drives for supported RAID levels.

The minimum RAID group size is two drives. Only mirror-type RAIDs can be configured with this minimum number of drives. Configuring a two-drive RAID 1/0 allows for additional drives to be added to the mirrored RAID group over time, in contrast with configuring a two-drive RAID 1 that has the same initial configuration, but does not allow adding additional drives directly. (LUN Migration or metaLUN could be used to augment its capacity.)

Virtual Provisioning pool default private RAID group's have different size minimums.

Number of drives

Depending on the RAID level's minimum number of required disks (see Table 4 on page 73), a group can contain as few as two and as many as 16 hard drives.

For example, the smallest, RAID Level 6 RAID group, would contain four hard drives, the largest 16 disks.

Stripe width

Stripe width is simply the number of drives in a stripe. It applies to distributed parity schemes (RAID 5 and RAID 6) and is only really useful in that context. The width

defines the number of drives that can be read from in parallel, assuming random I/O. So it is the number of drives in the stripe.

RAID group notation

RAID group notation uses two numbers in parentheses, separated by a plus (*+*) sign. For example: *(4+1)*. The sum of the two numbers is the total number of drives in the RAID group. The first number describes the number of drives in the group available for data storage. The second number is the number of drives or drive equivalents in capacity used for data redundancy (see "RAID group's and LUN capacity" on page 75).

The following are examples of possible RAID groups:

- A two-disk RAID 1 group: (1+1)
- A five-disk RAID 3 group: (4+1)
- A seven-disk RAID 5 group: (6+1)
- A ten-disk RAID 6 group: (8+2)
- A twelve-disk RAID 1/0 group: (6+6)

Note that RAID group notation does not distinguish between RAID types 5 and 3.

Create a RAID group

RAID groups are created using Unisphere or the command line interface (CLI), the storage systems management software, or automatically through a feature. The RAID group is the fundamental drive level of organization.

The maximum number of RAID groups creatable per storage system is model dependent. Entry-level storage systems have fewer RAID groups than higher models.

Binding is the name of the process for creating a RAID group using the CLI. Any drive of the same type may be bound into a RAID group. Drives being of the same type is enforced by the operating environment. The speed and capacity of the drives in a RAID group do not need to be the same, and is not enforced by the operating environment. However, for performance, capacity utilization, and maintenance reasons it is prudent to use the same speed and capacity together in RAID groups (see "Hard drive specifications" on page 98 for details).

Note that in some cases, to configure RAID groups for maximum performance, the Unisphere CLI may be required.

RAID group's and LUN capacity

The different RAID levels have different levels of capacity usage. When creating a RAID group, the eventual usable user data capacity of the LUNs hosted on the group depends on the RAID level chosen. (See "Logical Units (LUNs" on page 78.) The "Total capacity" reported by Unisphere for a RAID group represents all the allocable storage on the RAID group. It is net of all disk-level formatting, private space usage and RAID protection.

In the LUN properties, Unisphere reports "raw" capacity, which is net of disk-level format, but includes RAID protection. Many service level agreements are described

on the basis of raw capacity. Users can use this as a way of estimating total raw capacity used in the storage system.

Note that if an entire RAID group is allocated to a single LUN the "raw" capacity is less than the capacity listed by the drive vendor, as discussed in the previous section. This is due to the binary-to-decimal difference, the use of 520-byte sectors, and drive capacity used for metadata.

RAID group drive user or data capacity

The capacity available for data storage is called the *user capacity* or *data capacity*. The user capacity of drives placed in a RAID group is less than the raw capacity, due to the capacity consumed for data protection by both redundancy and data integrity mechanisms.

Mirror RAID

With mirrored RAID levels, the user storage capacity of the drives at the RAID level is half the capacity of the drives. For example, the capacity available for data storage of eight unformatted 600 GB hard drives (raw total of 4.8 TB) placed in RAID level 1/0 (4+4) together is an unformatted 2.1 TB.

Parity RAID

Generally, parity-type RAIDs are used when cost of capacity is a more important consideration than random I/O performance.

With parity RAID, the ratio of the capacity available for user data-to-data protection changes depending on the RAID level and the number of disks in the RAID group.

For parity RAID Levels 3 and 5, one drive's worth of capacity is required for parity. For RAID 6, two drives' worth is used for parity. In RAID 5 and RAID 6, no single drive is dedicated to parity. In these RAID levels, a portion of each drive in the group is consumed by parity. This mechanism is called *rotating parity*. In RAID 3 there is a single dedicated parity drive. This drive contains only parity data, and no user data.

Since the *number of drives' capacity used for parity is fixed,* for any size RAID group, the *percentage* of storage capacity dedicated to parity decreases as the number of drives in the parity-RAID group increases. This makes parity-RAID groups with a large number of drives very economical in cost per GB. A trade-off of capacity versus availability is that large RAID groups have a statistically greater chance of multi-drive failure and take longer to rebuild than smaller groups (see the next section).

For example, the raw capacity available for data storage of a RAID 5 (2+1) is 66 percent; 33 percent of total raw capacity dedicated to data protection and not available to store user data. By increasing this RAID group by two drives, to a RAID 5 (4+1), the percentage of the group's raw capacity dedicated to parity decreases to 20 percent.

Example RAID group capacities

Table 5 on page 77 shows the user capacity for the most commonly provisioned RAID groups of the VNX, using the available hard drives. Note that the user capacity takes

into account the 520-byte sectoring used to ensure data integrity (see "Capacity" on page 64), the RAID type's overhead used in redundancy, and metadata. The user capacity is in binary TB, *rounded* to 10 GB. Binary GB is the capacity reported by hosts. A LUN bound using a RAID group from the table reports that capacity to the host. A host file system formatting further reduces the available user capacity.

Table 5 Common RAID group useable capacities (TB)

RAID level	Drives	100 GB flash (TB)	200 GB flash (TB)	300 GB SAS (TB)	600 GB SAS (TB)	2 TB NL-SAS (TB)
RAID 1 (FAST Cache)	1+1	0.09	0.18			
RAID 1/0	3+3	0.27	0.54	0.79	1.57	5.37
	4+4 (pool default)	0.36	0.72	1.05	2.10	7.17
RAID 5	4+1 (pool default)					
	8+1	0.72	1.43	2.10	4.19	14.13
RAID 6	6+2 (pool default)	0.54	1.07	1.57	3.15	10.75
	8+2	0.72	1.43	2.10	4.19	14.33

RAID 0

RAID 0 is a special case. RAID 0 is the only striped RAID level. There is no data protection with RAID Level 0 to consume capacity or inhibit performance. With RAID Level 0, the raw storage capacity of the level's disks is the sum of the capacity of the raw drives. In addition, reads and writes to RAID 0 groups are faster than to either mirror or parity RAID levels. However, with no data protection, any drive failure within the group results in total data loss.

Disk

The VNX supports an individual disk (disk) storage type. The disk type functions just like a standard single drive. Individual disk is not a RAID storage type. It does not have the data protection provided by parity or mirroring of data. It does not have the performance or capacity benefits provided by striping. This type can be used for temporary directories that are not critically important.

Private RAID groups

The Virtual Provisioning feature creates *private* RAID groups in pool creation. These RAID groups are managed by the feature's software and are not user accessible. (See Figure 19 on page 78 and "Virtual Pools" on page 87.)

RAID group expansion

RAID group expansion is the process of adding additional drives to an existing RAID group. Unlike legacy EMC mid-range storage systems, the VNX series does not support RAID group expansion.

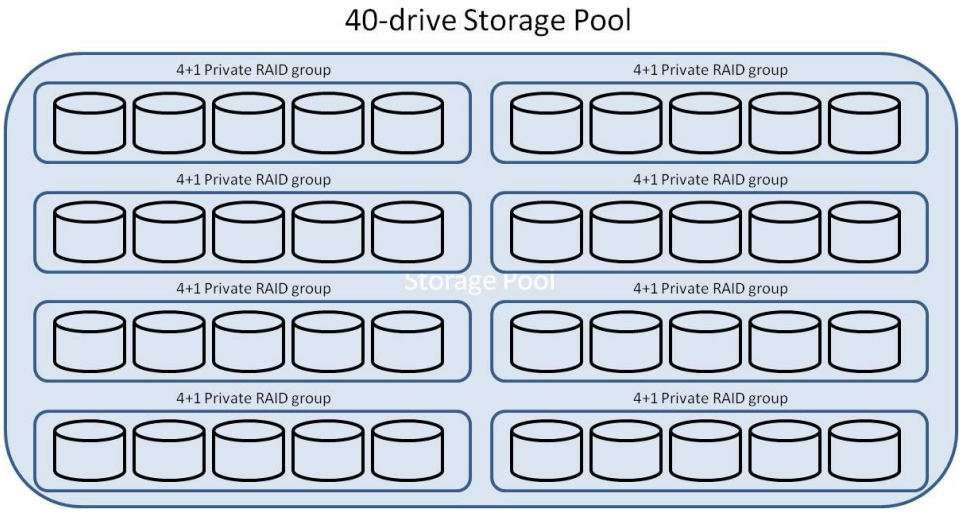

Figure 19 40-drive RAID 5 Storage Pool conceptual view

Logical Units (LUNs)

Logical Units (LUNs) are a logical storage construct overlaid onto RAID groups. Hosts see 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 RAID groups from hosts. LUNs are created to allocate capacity, ensure performance, and for information security. Note that information security is *not* data protection. Information security is a confidentiality feature.

User LUNs and private LUNs

There are two further types of LUNs, user and private. User LUNs are the LUNs visible to the host and on which user storage operations are performed.

Private LUNs support user-related or feature-created LUNs. Users have no control over the assignment of private LUN IDs.

Examples of private LUNs include metaLUN components, a FAST Cache's contents, and Reserve LUN Pool LUNs.

Private LUNs cannot belong to storage groups, and a server cannot perform I/O to them. The exception is the metaLUN. With the components of a metaLUN, the server is writing to those private components through the metaLUN.

RAID groups and LUNs

The data capacity of a RAID group can be partitioned into one or more LUNs. (See Figure 20 on page 79.) The maximum number of LUNs per RAID group is operating environment version dependent. For Operating Environment Block, revision 31 and later, the maximum number of LUNs per RAID group is 256.

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. Note that Virtual Provisioning pools and metaLUNs are implemented using private LUNs.

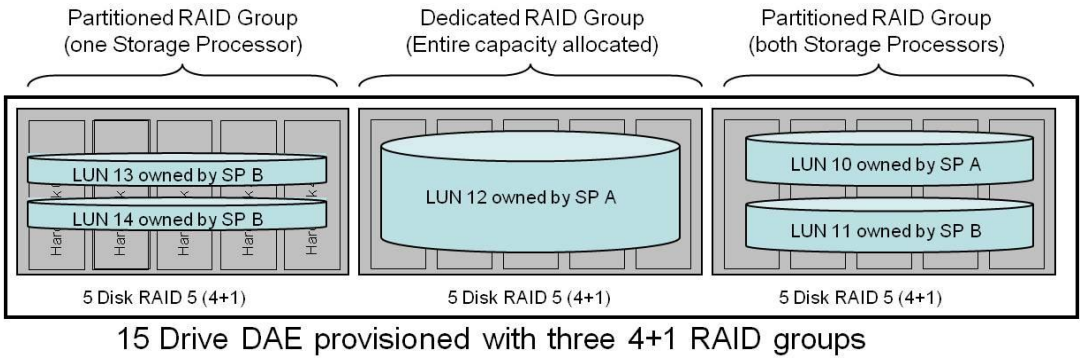

Figure 20 LUN conceptual diagram

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. The LUNS stripe across all the drives of the RAID group, while being "stacked" one on top of each other. 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.

All private RAID groups of Virtual pools are by definition partitioned RAID groups.

Binding LUNs

The process of creating a LUN is called *binding*. LUNs may be bound only after RAID groups are created.

LUNs may be created manually through Unisphere, or automatically by a feature. An automatically created LUN is sometimes called a *private LUN*; however, certain private LUNs may be created manually. It may also be referred to as a *reserved LUN*. Users have either a limited or no ability to manage private LUNs.

When a LUN is bound manually, users have limited control over the location of the LUN on the RAID group. The first LUN created on an unused RAID group always occupies capacity beginning with the lowest number LBAs. These LBAs occupy the outermost tracks of a mechanical hard drive. Subsequent LUNs created are created immediately after the capacity occupied by the previously created LUN. If unused capacity exists in between two LUNs, due to the deletion of a LUN, and the new LUN can fit within the gap, the new LUN is bound in that location. LUNs cannot be created

when there is not enough contiguous capacity in the RAID group to host them. RAID groups can be defragmented to make contiguous space and maximize their capacity utilization.

Binding process

LUNs are available for use immediately after they are created. The process called *FastBind* makes the LUN immediately available for use. 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.

The preparation and verification steps can vary considerably in duration. It depends on the LUN size, the verification priority, and the storage system's workload.

Preparation

During the preparation step, the storage allocated to the LUN is overwritten with binary zeroes. These zeroes erase any previous data from the storage and set up for the parity calculation. When zeroing is complete, parity and metadata is calculated for the LUN sectors. New systems received directly from EMC have storage "pre-zeroed." This eliminates this step for a drive's initial bind. In addition, drives can be manually pre-zeroed should the requirement exist through the use of the Unisphere CLI.

Verification

Verification involves a Background Verify (*BV*). A BV is a reading of the LUN's parity sectors and verification of their contents. This is an availability feature of the storage system. A BV is executed by default after a bind.

During the initialization of a new LUN, if the workload starts using the LUN before the BV finishes, there is an adverse performance effect. In addition, the BV of a large Virtual pool can have a long duration. This may lead to high utilization of the pool's resources until the BV completes.

The BV 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.

LUN ownership

LUNs are managed and accessed by a single storage processor. This is called *LUN ownership*. 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. It may be necessary to change a LUN's ownership to its peer storage processor for performance or availability reasons. Changing ownership is called a *trespass*. No data is moved in a trespass. The new owning storage processor simply begins managing the I/O for the trespassed LUN on its original RAID group.

For example, a trespass may be needed to balance storage processor usage within the storage system.

LUNs may also automatically be trespassed by the operating environment. This is an availability feature. Typically this is the result of a failure situation, where a storage processor loses connectivity to one or more of its LUNs. A LUN's original ownership may or may not be restored after the failure is corrected. (See "Active/passive ownership model" on page 148.)

LUN identification

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

LUN name

The *LUN name* is a user assigned identifier. A name can be any identifying text string. Through Unisphere, the user can create a free form text field of up to 64 characters for host LUN identification. There is no check for duplicate LUN names and no restriction on duplicate names.

LUNs are internally identified by their *LUN ID*. The entire range of CLARiiON LUN IDs is from zero to a model-dependent maximum. The range of LUN IDs, like the maximum number of LUNs per storage system is model dependent, the larger-model storage system being able to host a greater number of LUNs than the smaller.

Unisphere uses the LUN ID numbers (see the next section) that are assigned by the storage system; it does not use the LUN name, which is changeable. However, the LUN name is mapped to the LUN ID for user convenience.

User and private LUN IDs

Like there are user and private LUNs, 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 the storage system. However, the sum of user and private LUN IDs in use cannot exceed the storage system model's maximum total of LUNs. (There are always extra LUN IDs.)

User LUNs, created through Unisphere Manager, 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 within the model's range of LUN IDs at the time of creation.

Private LUNs are assigned LUN IDs from the highest available number in the LUN ID range at the time of creation. For example, the maximum number of LUNs on a VNX7500with OE Block 31.0 and later is 8192. The range of all possible LUN IDs (user and Private) that can be assigned is 0 through 16833 (a total of 16834) any of which are user-selectable, as long as it is available.

LUN expansion

Expansion is the process of adding capacity to a base LUN. However, a LUN cannot have a capacity larger than the maximum capacity of its Virtual Pool or its underlying RAID group's capacity. A metaLUN may, however, be created to contain the total capacity of the storage system, minus the capacity of the system drives.

Unbinding

Unbinding a LUN releases all the capacity it has bound on its component drives for reuse, resulting in the previously stored data being lost. Unbinding is performed through Unisphere when a LUN is deleted or a Virtual Pool is destroyed.

A user has the option to destroy a RAID group once all its LUNs are unbound.

RAID group defragmenting

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 fragmentation 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. The defragmentation physically moves the capacity consumed by LUNs in a RAID group to be adjacent to each other. This eliminates the gaps and recovers the unused capacity between the previously separated LUNs within a RAID group. When all the RAID group's in-use capacity is contiguous, there is potential for allowing creation of new, larger LUNs on the group.

Note that RAID group defragmentation does not apply to Virtual Pools. As of OE Block 31, RAID Level 6 RAID groups cannot be defragmented.

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.

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 available 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.

For example, a RAID 5 (4+1) based LUN requires 88 concurrent requests for its queue to be full. *Best Practices*[1] lists the calculation for determining LUN queue lengths. If

[1] *EMC Unified Best Practices for Performance and Availability: Common Platform and O.E.XX— Applied Best Practices* is available on Powerlink.

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. See "Front end" on page 148 for details.

Reserved LUN Pool (RLP)

The reserved LUN pool is a group of private LUNs associated with replication software, such as SnapView, Incremental SAN Copy, and MirrorView/A to store data or information required to complete a replication task.

The reserved LUN pool consists of one or more private LUNs. A LUN becomes private when it is added to the reserved LUN pool.

LUN types

The VNX supports several different types of LUNs. The types are:

- Virtual Pool LUNs
- Traditional LUNs
- MetaLUNs

All types of LUNs perform the same function. However there are differences in the way data is stored and how data storage can scale.

Virtual Pool LUNs

Two types of Virtual Pool LUNs are available — LUNs without the thin property, called *thick LUNs*, and LUNs with the thin property, called *thin LUNs*. LUNs in pools are often called *pool LUNs* to distinguish them from LUNs on RAID groups (RAID group LUNs, or traditional LUNs). Thin and thick LUNs may occupy the same pool, or a pool may be exclusively provisioned with one or the other type.

The following operations are available with pool LUNs:

- Expand the LUN's capacity
- Shrink the LUN's capacity (Windows Server 2008 hosts only)
- Compress the data on the LUN if the storage system has the Compression enabler installed
- Auto-tier the LUN if the storage system has the FAST VP enabler installed

The maximum capacity Virtual pool LUN with Block OE 31.0 is 16 TB.

Thick LUNs

The capacity of a thick LUN is distributed equally across the drives in the pool on which the LUN is created. Thick pool LUNs *reserve* their full capacity upon creation, but do not *allocate* that capacity to themselves from the pool's provisioned capacity. The reservation ensures that capacity is available when needed. The allocation is the commitment of that capacity to the LUN. When a write I/O is received within a given

LBA range of the LUN, a *slice* of the reserved capacity is allocated to the LUN. The provisioned capacity of a thick LUN is the same as the user capacity presented to the host. A thick LUN uses slightly more storage capacity than the amount of user data written to it, due to the need for metadata within the pool. Note that Virtual Pools provisioned exclusively with thick LUNs do not issue capacity utilization alerts. Unlike a thin LUN, a thick LUN can never run out of capacity.

Thin LUNs

Thin LUNs are supported for storage systems with the Thin Provisioning enabler installed. Thin LUNs provide for the most efficient use of storage capacity. A thin LUN shares the pool's available storage with other pool-based LUNs inside of itself. That is, the in-use capacity of one or more host LUNs is contained within the thin LUN.

The capacity of the thin LUN presented to the host is independent of the available physical storage in the pool. This prevents *over-provisioning*. Overprovisioning is the installation and commitment of physical storage capacity ahead of its use. The presentation of storage not physically available avoids overprovisioning of the hosts and the underutilization of the storage system's capacity. When a thin LUN eventually requires additional physical storage, capacity is nondisruptively and automatically added from the capacity of the pool. The pool has previously been provisioned with a margin of capacity to meet current storage needs.

For example, if a LUN needs 10 TB at the end of a year, but initially only needs 3 TB, initially provisioning the LUN with 10 TB would commit 7 TB of storage to a single LUN ahead of its need. This committed and unused storage may be more useful to other LUNs within the storage system. In this example, a thin LUN would present to the host 10 TB of capacity available for use, but only physically consume pool capacity for the initially needed 3 TB.

To the host, a thin LUN appears as a thick LUN or a traditional LUN. Unlike a thick LUN or a traditional LUN, a thin LUN can run out of capacity, if the pool to which it belongs runs out of disk space. Such an event is an unrecoverable write error, and data from the last write operation is not available. By default, the storage system issues a warning alert when 70% of the pool's capacity has been consumed; and, when 85% of the space has been consumed, it issues a critical alert. As thin LUNs consume the pool's provisioned capacity, alerts continue to report the actual percentage of consumed capacity. Like a thick LUN, a thin LUN uses slightly more capacity than the amount of user data written to it, due to the metadata needed for the pool.

Operational differences between Pool LUN types

Both thick and thin LUNs are allocated pool capacity through *slices*. When a user LUN is first accessed, a *slice* of the pool's capacity is allocated to the LUN. Slice allocations are automatically balanced across back-end SAS ports and the pool's private RAID groups.

For example, the 40-drive pool shown in Figure 21 on page 85 contains eight 4+1 private RAID groups. There are five user LUNs created in the pool (Blue, Orange, Green, etc.). Assume each user LUN has exactly three slices fully populated with user

data. Further assume that Blue and Orange are owned by SP A and the remaining by SP B.

One possible allocation of capacity based on the pool's algorithm is shown in the figure. (The blue stripes spanning the pool's RAID groups are the slices allocated to the Blue user LUN.) Fifteen total slices from five user LUNs are shown distributed over eight private RAID groups.

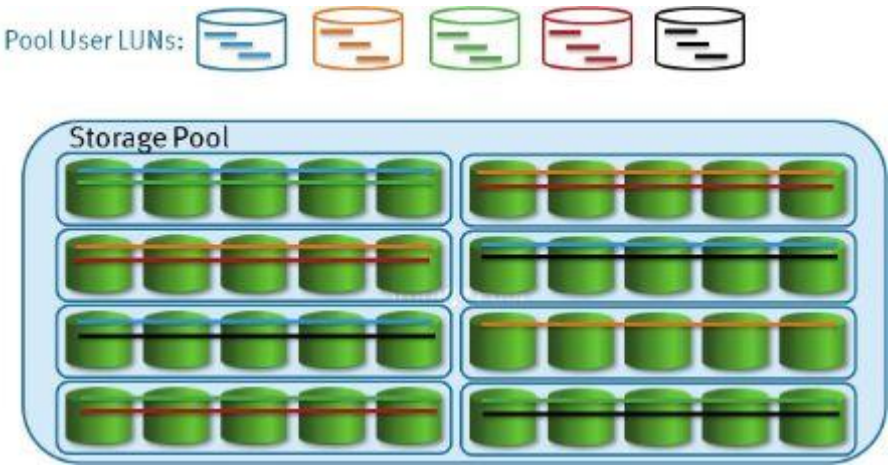

Figure 21 Capacity allocation with storage pools example

With thick LUNs, capacity is reserved, but not allocated to the LUN. A thick LUN can never run out of capacity. The reserved capacity prevents this from occurring. A thick LUN slice is a contiguous set of a billion logical block addresses (LBAs). The LUN's reserved capacity is decremented for the capacity allocated.

For example, with a newly bound thick LUN receiving a write I/O to LBA 1015, a slice of capacity for LBAs 1000 to 2000 would be allocated to the LUN. Should there be a subsequent write to LBA 1500, the capacity for that address has already been allocated to the LUN. There is no overhead in addressing the allocated capacity.

However a write to an LBA which is outside of a previously allocated range of capacity requires an additional slice to be allocated from the pool to the thick LUN.

For example, if the only two accesses to a thick LUN are to LBAs 1500 and 2600, this results in two 1 GB slices being allocated to the LUN. This is because the two addresses are further from each other than the 1 GB extent of a single slice.

With thin LUNs, capacity is neither reserved nor allocated when the thin LUN is bound. Like with the thick LUN, once writes are made to the LUN, the 1 GB slice gets allocated from the pool. However, this slice is not made up of contiguous LBAs. It is populated by individual addresses and groupings of addresses in a more random distribution.

From the previous example, if you accessed LBA 1500 and 2600 in the same LUN, you would still have both those operations happening in the same slice. A thin LUN gets a 1 GB slice from the pool and uses it for all accesses to the LUN until it needs more space, at which time it gets another 1 GB slice from the pool.

A perfectly random I/O workload to a thick LUN gets all its slices allocated from the pool in a short period of time. This is because the random LBAs would allocate all the LUN's capacity; as soon as there is a write to an LBA in any new 1 GB extant, the new slice gets allocated to the LUN. To achieve the same effect with a thin LUN, sequential I/O would need to be performed to every LUN address.

The allocated capacity of a thick LUN has contiguous addressing by nature. A thin LUN slice does not. There is always some pool overhead with both types of pool LUNs. The overhead is the result of the LBA mapping. This mapping allows for pool LUNs to more efficiently use capacity. However, there is a higher mapping overhead in case of thin LUNs.

The provisioned capacity and the allocated capacities of pool-based LUNs are reported through Unisphere. The Unisphere-reported capacities may not exactly match the capacity usage reported by the host's file system. This is because not all host file systems notify the storage system when capacity is deallocated. This causes the host to report less capacity allocated than the storage system's report. In addition, the granularity of the capacity allocation report between the host's file system and the storage system's Virtual Provisioning may be different. This can lead to under or over reporting of capacities, depending on the difference in the magnitude of the discrete capacity value reported.

In summary, from a performance point of view, thick LUNs perform better than thin LUNs, while thin LUNs provide a higher-capacity utilization.

Compressed LUNs

The LUN compression feature provides the option of compressing the data on any LUN to free up storage capacity. Compression performs an algorithmic data compression of thick and traditional LUNs. All compressed LUNs become thin LUNs.

VNX series storage systems with VNX OE 05.31 and both the Compression and Thin Provisioning enablers are needed for compressed LUNs.

Compression analyzes the data on a disk and applies algorithms that reduce the size of repetitive sequences of bits that are inherent in certain types of files. Unlike deduplication, compression does not depend on multiple copies of the same data. The storage system performs compression operations in the background, at the LUN-level, while continuing to service host I/O.

Compression of a thick LUN or traditional LUN migrates the LUN type to a thin LUN. The compressed LUN remains a thin LUN even after you decompress it. Like a thin LUN, the LUN's host-visible capacity is equal to the size that was specified when the LUN was bound.

The compression and decompression process is a storage processor CPU utilization intensive activity. It may adversely affect overall storage system performance. The compression feature should be applied to user data that is not changing. This is sometimes called *static* data. Compression should not be used for data that is subject to random I/O, and in particular random write I/O.

LUN Compression is a basic feature included with all VNX Systems starting from the model VNX5300 and higher.

Virtual Pools

A Virtual Pool is a logical storage object. It is made-up of drives organized into private RAID groups. (See Figure 22 on page 87 and "Logical storage objects" on page 69.) Pool LUNs are created within Virtual pools.

A pool is overlaid onto one or more private RAID groupings of drives. The number of drives and the number of Virtual pools supported is model dependent. The higher models support a larger number of pools than the lower-end models. A pool may be made-up of any number of drives. The limit to the number of drives in a pool is the maximum supported by the storage system model, less the system drives.

The pool feature's software automates storage provisioning, capacity allocation, and data placement within the pool. In addition, provisioning of the pool is simplified through the pool feature's graphical user interface, which algorithmically applies best practices when given a set of drives to either create or expand a pool's capacity.

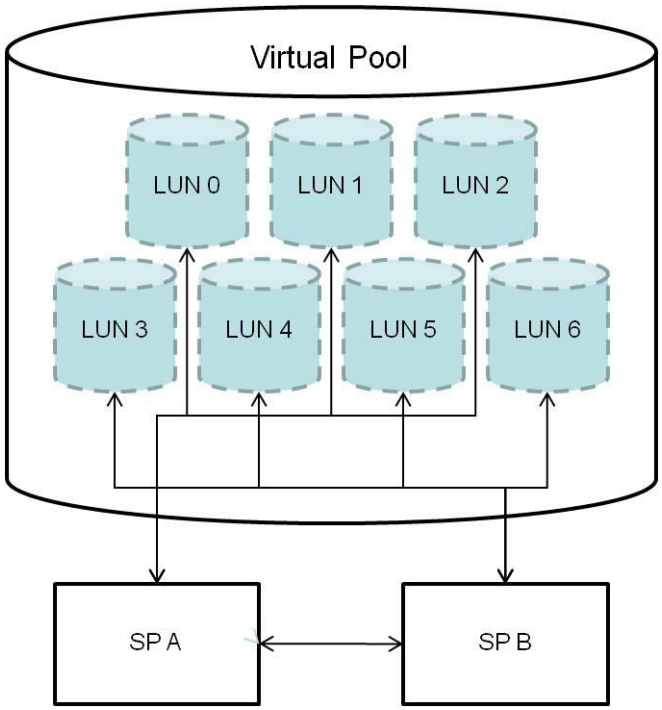

Figure 22 Virtual Pool conceptual diagram

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.

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 5, 6, or 1/0.

High-level pool usage

When creating Virtual Provisioning pools, the workload needs to be understood to balance the needed capacity with the deliverable performance. In general, if the goal is:

- Highly 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.

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 to accommodate the file systems it is storing.

Pools are designed for ease of use. The pool dialog algorithmically implements many best practices.

Balancing capacity and performance

Like all provisioning decisions, the capacity and the performance requirements of the pool's workload need to be understood.

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. This overhead is in addition to the user data capacity requirement. The initial capacity of the pool is slightly larger than the sum of its LUN's user data capacity. To accurately estimate the initial capacity of the pool needed, you need to know the number of LUNs anticipated to be created, their type (thick or thin), and their capacity.

Pool-based LUN performance utilization

A pool LUN is managed by the storage pool software. In addition, the LUN shares the storage resources of the pool with other LUNs. These both impose a small

performance overhead on the LUN. A margin of performance, typically additional IOPS, needs to be maintained within the pool's performance budget to account for the overhead of the pool and possible contention between LUNs for the shared storage resources.

Pools and workloads

The following workload combinations of performance and capacity apply to virtual pools:

- High capacity/High performance
- High capacity/Modest performance
- Low capacity/Modest performance
- Low capacity/High performance

High capacity and performance is the ideal workload combination for a pool. The higher the capacity of the pool, the greater the number of drives it requires to satisfy the capacity requirement. The greater the number of drives, the higher the pool's performance. High-capacity, high-performance SAS drives would match this type of pool. Likewise, a larger number of more modestly performing SAS drives could be applied. The larger number of SAS drives would result in a greater margin of pool performance being available. However, the performance of the individual LUNs needs to be understood to ensure there is not contention over pool resources.

High capacity and modest performance is easily achieved within a pool. A small number of very high capacity, modestly performing NL-SAS drives would match this type of pool. Conversely, a larger number of more modest capacity SAS drives may be used. The larger number of SAS drives would provide a greater margin of performance within the pool.

Low capacity and modest performance can be achieved with almost any pool provisioning. Either a small number of SAS drives or a smaller number of NL-SAS drives can be applied.

Low capacity and high performance is a problematic combination. This requirement may result in either an overprovisioning of capacity to achieve the performance requirement, or an under provisioning of IOPS in trying to more closely meet the capacity requirement. If practical, high-performance flash drives may be used to meet the workload needs of this pool workload.

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 and maintain. It is easier to quantify and predict performance when only a single drive type is present.

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.

Traditional LUNs

Traditional LUNs, illustrated in Figure 23 on page 90, are the legacy implementation of LUNs directly on RAID groups. They support the following:

- Celerra file system
- Legacy CLARiiON applications (e.g., supporting scripts, and software)
- Access to lower levels of logical-to-physical data mapping

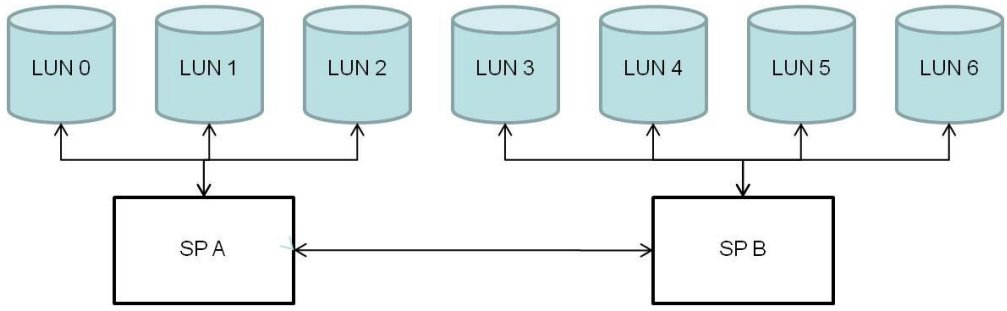

Figure 23 Traditional LUNs conceptual view

The maximum-capacity traditional LUN is dependent on the RAID group's drives and RAID level. All RAID levels supported by the VNX are available. A maximum of 16 drives can make up a RAID group.

For example, a 15-drive, 2 TB NL-SAS RAID level 5tTraditional LUN using the entire capacity of the RAID group would have a user capacity of almost 27 TB. Note that it may not be prudent from an availability standpoint to provision a LUN in this fashion.

The capacity of a traditional LUN is distributed equally across the drives of the RAID group to which the LUN is bound. The LUN can be bound to all or part of the RAID group's usable capacity. The provisioned capacity of a traditional LUN is the same as the user capacity presented to the host.

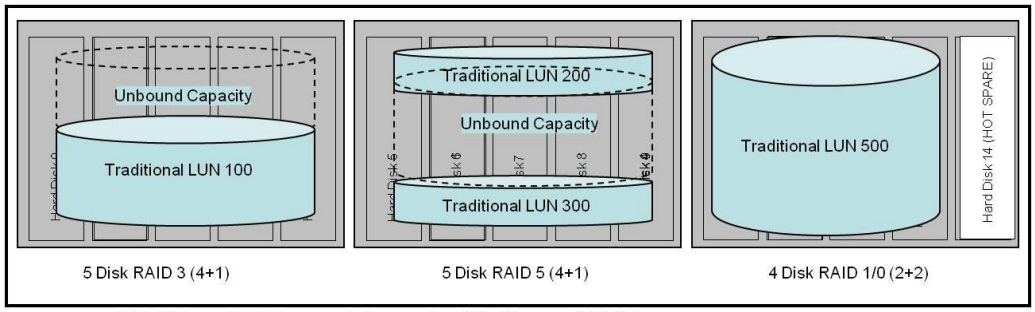

Figure 24 Relationship of LUNs to physical drives

MetaLUNs

MetaLUNs (Figure 25 on page 91) are LUNs created from two or more traditional LUNs. MetaLUNs are one solution to LUNs requiring capacity that exceeds the capacity of a single RAID group or LUNs with a greater maximum capacity than available within a Virtual Pool.

The maximum metaLUN capacity supported is the capacity of all the drives on the storage system, less the system drives.

MetaLUNs also provide a path for later *growing* of a base LUN's capacity without the host adding an additional separate LUN. This reduces the number of LUNs needing to be managed on the host and the storage system. In addition, creating metaLUNs provides a way to increasing LUN performance through the addition of drives.

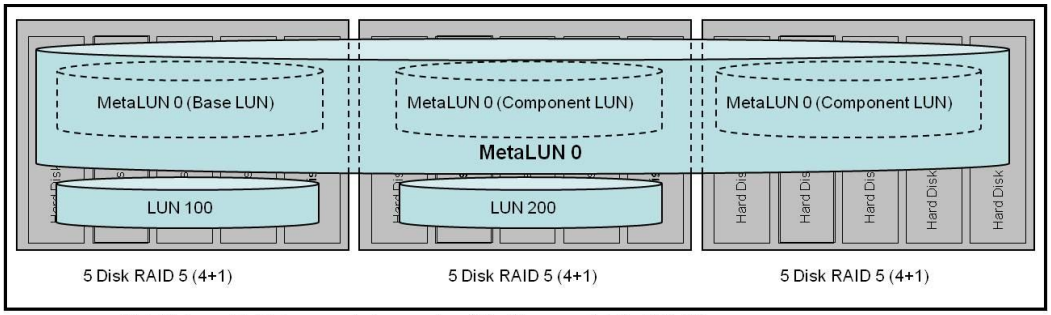

Figure 25 MetaLUN conceptual view

A metaLUN is made up of a *base* LUN and *component* LUNs. *A metaLUN is made up of two or more components up to a maximum of 16 components.* The base LUN is a special component LUN that provides the metaLUN's identity for addressing purposes. A metaLUN's capacity is the combined capacities of the base LUN and all the component LUNs that make it up.

MetaLUN expansion

Expansion is the process of adding capacity to a base LUN or an existing metaLUN. It is possible to create a metaLUN that includes all the drives on the storage system, less the system drives. However, this may not be practical, due to host addressing limitations, provisioning restrictions, and availability concerns.

Additional LUNs can be added to a metaLUN to further increase its capacity. A maximum of 32 FLARE LUNs can be included in a metaLUN component. During metaLUN expansion, the base LUN (base component's) data is always accessible. The additional capacity is not available until the expansion completes. MetaLUN expansion can be either *concatenated* (Figure 26 on page 92) or *striped* (Figure 27 on page 92).

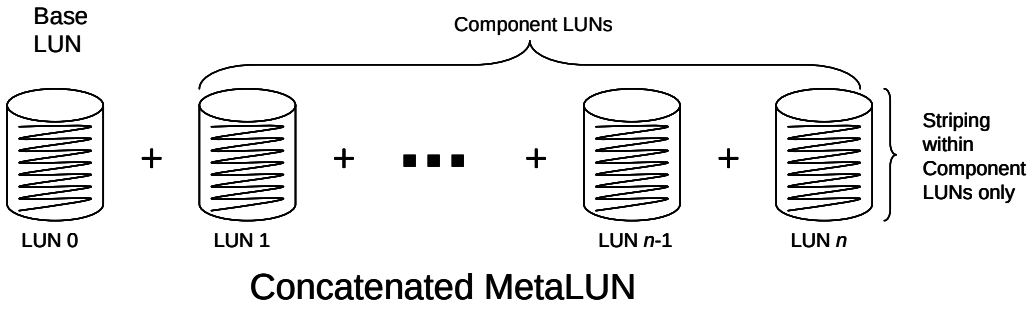

Figure 26 Concatenated metaLUN

Concatenated expansion simply adds additional capacity to the base entity. This base entity could be a base LUN or a striped metaLUN group in a hybrid metaLUN construct. Concatenated component LUNs do not have to be the same capacity or the same RAID type as the base LUN. When concatenating LUNs, all component LUNs must have the same level of protection. The LUNs being concatenated must all be unprotected (like RAID 0), or all N+1 protected (like RAID 5 or RAID 3), or all N+2 protected (like RAID 6). That is, RAID 6 LUNs can only be concatenated with RAID 6 LUNs. Concatenated expansion is very quick but may not result in a performance benefit from the additional drives.

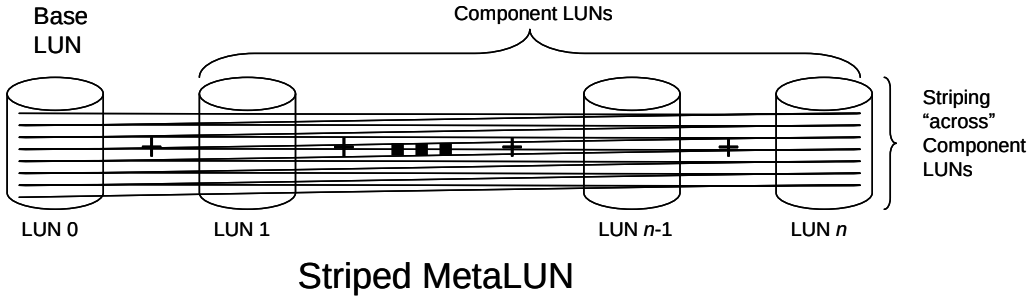

Figure 27 Striped metaLUN

Striped expansion restripes the base LUN's data across the base LUN and component LUNs being added. One or more component LUNs can be added at a time (Figure 28 on page 93). In striped expansion, all LUNs must be the same capacity and have the same underlying RAID group level. In addition, all LUN RAID group disks must be the same disk type. Performance is improved due to the increased number of drives being striped (see "Striping performance" on page 104). Striped expansion is the VNX metaLUN default.

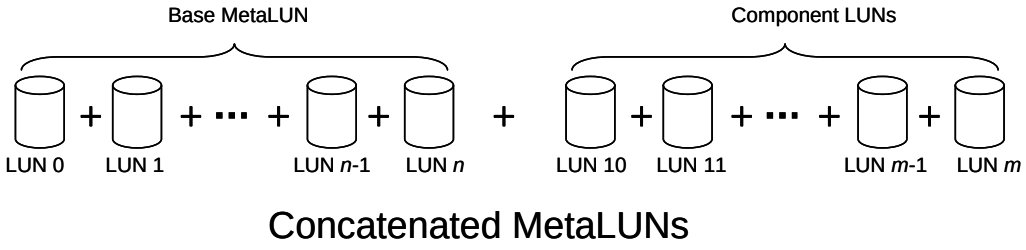

Figure 28 Concatenated components (metaLUNs)

All metaLUN have a base component, called component 0. MetaLUNs can be expanded by concatenating additional LUNs onto the base component to make very large volumes. This creates a hierarchy of components.

Unbinding

Unbinding a metaLUN releases and unbinds all its component LUNs, destroying the component LUNs, resulting in the previously stored data being lost.

Storage groups

A storage group is an access control mechanism for LUNs. It segregates groups of LUNs from access by specific hosts.

When you configure a storage group, you identify a set of LUNs that are only used by one or more hosts. The storage system then enforces access to the LUNs from the host. The LUNs are presented only to the hosts in the storage group, and the hosts can see only the LUNs in the group.

A large number of LUNs may be assigned to a storage group, and a storage system may have many storage groups. However, the number of LUNs per storage group and the number of storage groups per storage system is model dependent. The *Best Practices* document provides current maximum configurations.

Physical provisioning example

The following shows an example of storage system provisioning. The object of the example is to show a technique for the mapping of logical to physical storage objects.

All workloads are different. There are a number of ways to provision a storage system. This is just *one* way. It is an illustrative example with the intent of showing a technique for planning the provisioning of a storage system. It is not intended to be comprehensive, or definitive.

The physical provisioning of a storage system is dependent on the following:

- Workload capacity requirements
- Availability and performance objectives
- Storage system model
- Type and quantity of enclosures
- Type and quantity of drives

- Features installed

Given all these, there are a very large number of possible provisioning combinations.

Assume the example's storage system has more than one workload, and that a detailed analysis results in the following provisioning recommendation. Briefly, the workloads are:

- High-availability tiered storage with a RAID 6 FAST VP Pool
- "Bulk, low-cost storage" with RAID 5 traditional LUNs on high-capacity drives
- High-performance, lowest write latency storage with RAID 1/0 Traditional LUNs

The FAST Cache feature, system drives, and availability-related hot spares are also included.

An explanation of how to conduct this analysis is found in EMC Unified Best Practices for Performance and Availability: Common Platform and O.E. Block XX—Applied Best Practices (VNX Best Practices). This document is available on Powerlink.

Error! Reference source not found. on page **Error! Bookmark not defined.** summarizes the logical and physical storage objects meeting the example's workload requirements. RAID group configurations appear in parentheses within the cells.

Table 6 Physical provisioning example workload

Drive type	FAST Cache (drives)	RAID 5 traditional LUNs (drives)	RAID 6 FAST VP Pool (drives)	RAID 1/0 traditional LUNs (drives)	Total drives by type
100 GB Flash	4 (2x (1+1))				5
100 GB Flash Hot Spares	1				
300 GB SAS 10K rpm Data Drives			16 (2x (6+2))	16 (2x (6+2))	37
300 GB SAS 10K rpm Hot Spares			1		
300 GB SAS 10K rpm System Drives	4				
2 TB NL-SAS 7.2K rpm Data Drives		5 (1x(4+1))	24 (3x (6+2))		30
2 TB NL-SAS 7.2K rpm Hot Spares		1			
Workload drives (less hot spares and system drives):	4	5	40	16	72

See "FAST Cache" on page 57 for an explanation of FAST Cache usage and configuration. See "Global hot sparing" on page 150 for an explanation of hot spare usage and configuration. See "System drives and write cache availability" on page 149 for an explanation of the system drives. A discussion of the workload configurations can be found in "LUN types" on page 83.

For the purposes of this example, assume a VNX5300 with the storage hardware configuration shown in **Error! Reference source not found.** on page **Error! Bookmark not defined.**.

Table 7 Provisioning example hardware configuration

Provisioning example VNX5300 configuration			
	Type	**Model**	**Quantity**
Enclosures	DPE	DPE7	1
	DAE	DAE6S	4
Total enclosures:			5

Note that in this example, only enclosures housing drives are shown. An actual configuration would include additional enclosures. All the drive housing enclosures

are 3.5-inch, 15-drive enclosures. They total 75 drive slots for the storage system. As a result, all the drives must be 3.5-inch form factor. Enclosures with more drive slots and different drive form factors are available. (See "Storage processor enclosures" on page 38 and "Disk array enclosure" on page 41.) In addition, the VNX5300 is a dual back-end SAS port storage system, with buses 0 and 1.

Figure 29 on page 96 shows a technique for visually exploring drive layouts on enclosures. In the table, "HS" is the abbreviation for hot spare. Rows are drive-housing enclosures and columns are drive slots.

The layout should include matching capacity, performance, and availability objectives with the logical and physical storage object's properties.

For example, in the figure, the flash drives are vertically provisioned to distribute their I/O across all the available back-end SAS ports. For highest availability, the primary and mirror of each RAID 1 pair should be on separate back-end ports. See "Back-end bus performance measurement" on page 126 and *VNX Best Practices* for a discussion of this provisioning technique.

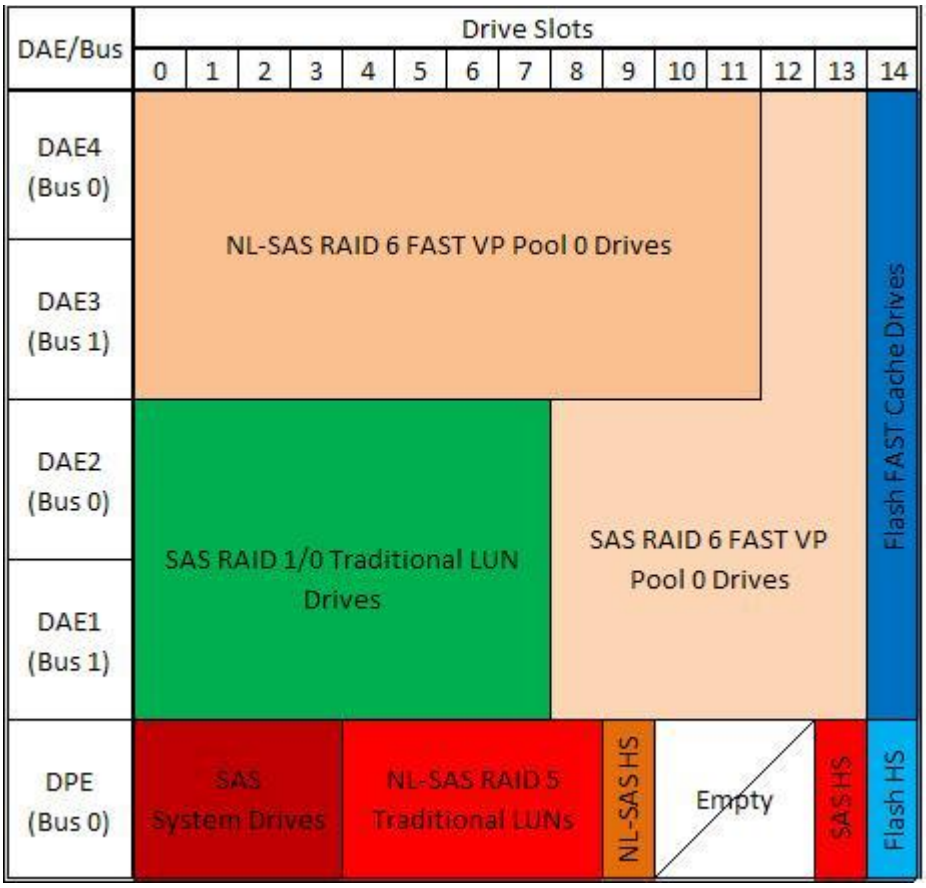

Figure 29 Example physical provisioning for a workload — drive layout

Chapter 9 Storage Object Performance

This chapter presents these topics:

Drive performance

Drives are the main physical storage device. The VNX can host both mechanical hard drives and nonvolatile memory-based drives.

Mechanical hard drives are the traditional storage. Hard drive performance is a product of spinning platter rpms, seek times, and burst-transfer rates. They have moderate to very high capacity.

The memory-based drives, known in the IT industry as *SSD*s, are also available on the VNX series. On the VNX, these drives are referred to as *flash drives*. They have higher availability and performance in comparison to traditional hard drives. In part, their higher availability comes from having no moving parts to wear out. The drives also do not seek. They are random access devices, so I/Os are executed very quickly. However, with current technology these drives have only modest capacity. They execute reads much more quickly than writes. Moreover, they are not as efficient at performing sequential I/O as they are random I/O. However, their performance of both random and sequential I/O is faster than mechanical hard drives.

Hard drive specifications

There are three important specifications for describing mechanical hard drive performance:

- Seek time — the time it takes to move read/write heads between tracks.
- Rotational latency — the time it takes the platter to move beneath the drive's read/write head.
- Transfer rate — the bandwidth of a hard drive.

Understanding the relationship between these specifications helps to classify a hard drive's performance. These values are important in determining two important hard drive performance metrics: throughput and response time.

Seek time

Seek time is the time it takes to move read/write heads between tracks. Seek time is measured in milliseconds. Seek times vary from drive to drive.

The average seek time is a frequently used metric; although both read and write seek times may also be specified. A typical seek time for a 15k rpm SAS hard drive is 3.5 ms to read and 4.0 ms to write, resulting in an average seek time of 3.8 ms. Seek time can also be measured track-to-track, or across the radius of the platter, a *full-stroke*.

Minimizing the amount of time a hard drive spends seeking improves performance. I/O type affects the amount of seeking a drive performs. Sequential I/O has the lowest seek time, because the read/write heads track continuously on the platters. Little or no seeking is performed once the drive heads have been set. Random I/O has longer seek times, because the read/write heads are constantly being positioned and repositioned on new locations on the platters with each new I/O.

Latency

The important factor relating to hard drive performance is *rotational latency.* Latency is the time it takes the platter to move beneath the drives read/write head. Latency

is measured in milliseconds. Higher speed drives, where speed is measured in rpm, provide noticeably higher overall random-access throughput and slightly shorter response times than slower drives. Table 8 on page 102 shows the relationship between average latency and rotational speeds.

Transfer rate

At the drive level, transfer rate is the bandwidth of a hard drive. Transfer rate is measured in MB/s. Transfer rate is further divided into internal and external rates.

The internal rate is how fast data can actually be read and written to the disk's platters. Transfer rate is higher on outer tracks than on inner tracks. The tracks have a greater number of sectors for the same linear distance on the outer tracks.

For example, for sequential bandwidth applications using a 3.5-inch 15k rpm SAS hard drive, a typical internal transfer rate is 50 MB/s for inner tracks to 100 MB/s for outer tracks. (See "Short stroking" on page 112.)

External transfer rate is the rate at which data can be transferred between the drive's attachment connector and the HBA or NIC. Multiple drives of a RAID group share this bandwidth. Note that burst transfer rates given by manufacturers are not seen with a storage system where a shared back-end port is in use. The manufacturer burst rates assume an internal connection, which allows direct access to system memory for the drive. With storage system's bus implementation, the actual transfer rate is determined more by the back-end transfer protocol, arbitration times, and capacity of the back-end port.

Hard drive queues

A hard drive executes one I/O at a time. If more than one request for an I/O read or write is sent to a disk, the requests are temporarily held in a disk queue for later execution

Drive queues allow for efficient use of the drive. The queuing sets up for constant usage. A drive queue depth of at least two is needed for a drive to work efficiently. In addition, there is potential for optimizations to be performed on the queue's entries that can improve I/O performance. Un-optimized drives execute requests stored on the queue in the order in which they were received.

When a disk reaches a queue size between 6 (for slower drives) and 20 (for faster drives) it is approaching its performance limit. This can occur because the bandwidth of the drive attachment typically exceeds the drive transfer rate. Some optimizations can be performed on queue entries as the queue lengthens. However, overall the disk queue depth is a large factor in drive response time.

Any time the disk spends seeking or waiting for a sector to spin under its head is wasted time, because no data is being transferred. When data on successive sectors of a disk are accessed, no time is wasted seeking. The access pattern of reading or writing successive sectors on disk is called *sequential access,* and the service time for sequential access is very low**.** The service time for a sequential read can be under 1 ms, so a much higher queue can be handled than for a random read, which has a service time of 4 to 6 ms.

The hard drive's disk controller may perform operations to improve response time. These optimizations try to maximize bandwidth by optimizing I/O to take advantage of the sequential nature of disk drive operation.

Command tag queuing (CTQ) on SAS drives uses optimizations such as the *elevator algorithm*, to improve random I/O performance. CTQ reorders I/O requests in the disk's queue to minimize the need to reposition the read/write head, resulting in sequential access. Enterprise-class drives perform these optimizations more efficiently than other classes of drives.

Calculating hard drive performance

There are four basic drive performance metrics used to compare drives. These metrics are calculated from the drive's specifications. The metrics are:

- Average service time
- Average response time
- Throughput
- Bandwidth

Average service time describes the average time it takes for a hard drive to serve up data. Average response time describes how much data can be served up to the host application. Bandwidth and throughput are two different descriptions of the quantity of data that can be read or written. They are dependent on the workload's I/O characteristics.

Average service time

Average service time measures the average time a disk can execute a single I/O. Service time is measured in milliseconds. The general formula for average service time is:

average service time = average seek time + average latency + transfer time

For practical purposes, the transfer time is zero on the VNX for small I/Os.

For example, to calculate the average service time for a 10k rpm, 2.5-inch 300 GB SAS hard drive with a read seek time of 3.6 ms and a write seek time of 4.2 ms:

- Write seek time: 4.2 ms
- Read seek time: 3.2 ms
- Average latency: 3.0 ms (from Table 8 on page 102)
- Transfer time: 0 ms

average service time = 6.7 ms = ((4.2 ms + 3.2 ms) / 2) + 3.0 ms + 0 ms

Note that seek time is a large component of average service time. In workloads characterized by random I/O, which is seek intensive, this component of the measurement provides an important comparison of drive performance. In workloads characterized by sequential I/O, where seeks are minimized, the differences in the average latency values are more significant when comparing drive performance.

Average response time

Average response time is the duration from when a request is queued to the moment the disk executes the request. The simplified calculation of average response time is as follows:

response time = (queue depth+ 1) * average service time

For example, the response time of a drive with an average response time of six milliseconds, and a queue averaging six entries:

- Queue depth: 6
- Service time: 6 ms

response time = 42 ms. = (6+1) * 6 ms

Throughput and bandwidth

Throughput and bandwidth are two measures of how much data a hard drive can read and write under a specific I/O regime.

The throughput of a drive is measured in IOPS. The number of I/Os per second (IOPS) a drive is capable of is determined by the seek time and the average latency. The drive's IOPS are affected by the ratio of reads to writes in the workload.

The general formula for calculating random I/O IOPS is:

IOPS = (1 IO /(average seek time + average latency)) * 1024 ms/sec.

Bandwidth is the quantity of data that can transfer in a second with the supported number of IOPS. Bandwidth is given by multiplying the I/O size times the IOPS.

The general formula for calculating bandwidth with I/O size measured in KB is:

bandwidth = (I/O size / 1024 KB/MB) * IOPS

For example, the throughput and bandwidth for a 3.5-inch, 7.2k rpm 2 TB NL-SAS hard drive with a 9.5 ms write seek time and an 8.5 ms read seek time in a workload with a 70:30 mix of read to write operations of 8 KB I/Os is:

- Write seek time: 9.5 ms
- Write ratio: 0.3
- Read seek time: 8.5 ms
- Read ratio: 0.7
- Average latency: 4.2 ms (from Table 8 on page 102)
- I/O size: 8 KB

throughput = 78.8 IOPS = (1/((9.5 ms * 0.3)+(8.5 ms* 0.7)+4.2 ms))*1024 ms/sec

bandwidth = 0.6 MB/s = (8.0 KB / 1024 KB/MB) * 78.8 IOPS

A common calculation error

Network and bus speed calculations can easily be in error. The correct rate is not always obvious. Network speed is always in decimal notation. Capacity notation is in binary. For example, a gigabit Ethernet is 1000 Mb, not 1024 Mb.

MHz is always 1,000,000 per second, while MB/s is 1,048,576 bytes/s. 1 MB/s (megabyte per second) equals 8 Mb/s (megabits per second). To calculate the bandwidth, multiply the bus width (in bytes) by the decimal speed, then divide by the decimal unit.

For example, a 64-bit bus (eight bytes) running at 100 MHz would have a theoretical bandwidth of:

763 MB/s ((8*100,000,000)/1,048,576) *not* 800 MB/s

Hard drive speed and performance

The rotational speed of a mechanical hard drive has a large effect on performance. The speed of high-rpm hard drives affects the I/O latency and the efficiency of storage system cache usage, as shown in Table 8 on page 102.

Table 8 Spindle rpm to latency relationship

Spindle rpm	Average latency (ms)
5,400	5.6
7,200	4.2
10,000	3.0
15,000	2.0

The faster rotational speeds result in less latency with the I/Os. High-rpm drives have a direct positive effect on random-read performance and a lesser effect on random writes.

Sequential reads and writes do not benefit as highly as random I/O from the faster rotational speed. However, they do increase the sequential read/write bandwidth over hard drives.

The rotational speed of a drive affects the rate of storage system cache flushing and filling. During cache flushing, data must be quickly written from the write cache to the drives (see "Write cache management" on page 119). Faster drives, particularly flash drives, are valuable under conditions when cache is being bypassed or is disabled. Likewise random reads that typically miss in cache occur faster. The benefit is a higher overall I/O rate for the storage system.

Hard drive capacity utilization and performance

Mechanical drive capacity has an effect on performance that is related to seek time.

Larger-capacity drives of a given type and speed may offer better performance than smaller capacity drives. This is because on a large capacity drive, with the same amount of data, the track-to-track seek distance is less, because there is more data per track. Contiguous data on a large-capacity drive occupies fewer tracks that are

closer together. This takes less time to access. On a smaller-capacity drive, that same amount of data occupies a greater number of tracks. At worst, the smaller drive may have to seek over the entire radius of the platter.

On a flash drive, there is also a relationship between capacity and performance. For access, drive capacity is distributed equally across a drive's channels. That is, there is the same number of NVRAM chips per channel. Channel contention can lower drive performance. Generally, the higher the number of channels and the fewer the number of NVRAM chips per channel, the higher the flash drive's performance.

In addition, the percentage of drive capacity reserved for internal housekeeping affects performance. The higher the percentage of the drive's overall capacity dedicated to internal housekeeping, the higher the performance.

Drive performance comparison

Table 9 on page 103 summarizes the characteristics of mechanical hard drives available with the VNX series.

Table 9 Drive performance factors, VNX

	Flash	SAS		NL-SAS
Rotational speed (rpm)	N/A	15k	10k	7.2k
Attachment speed (Gb/s)	6			
Typical average seek time (ms)*	0.03	3.1	3.7	9.0

*Read/write for flash drive

Figure 30 on page 104 shows the relative performance of a representative sample of the different types and speeds of hard drives on the basis of service time.

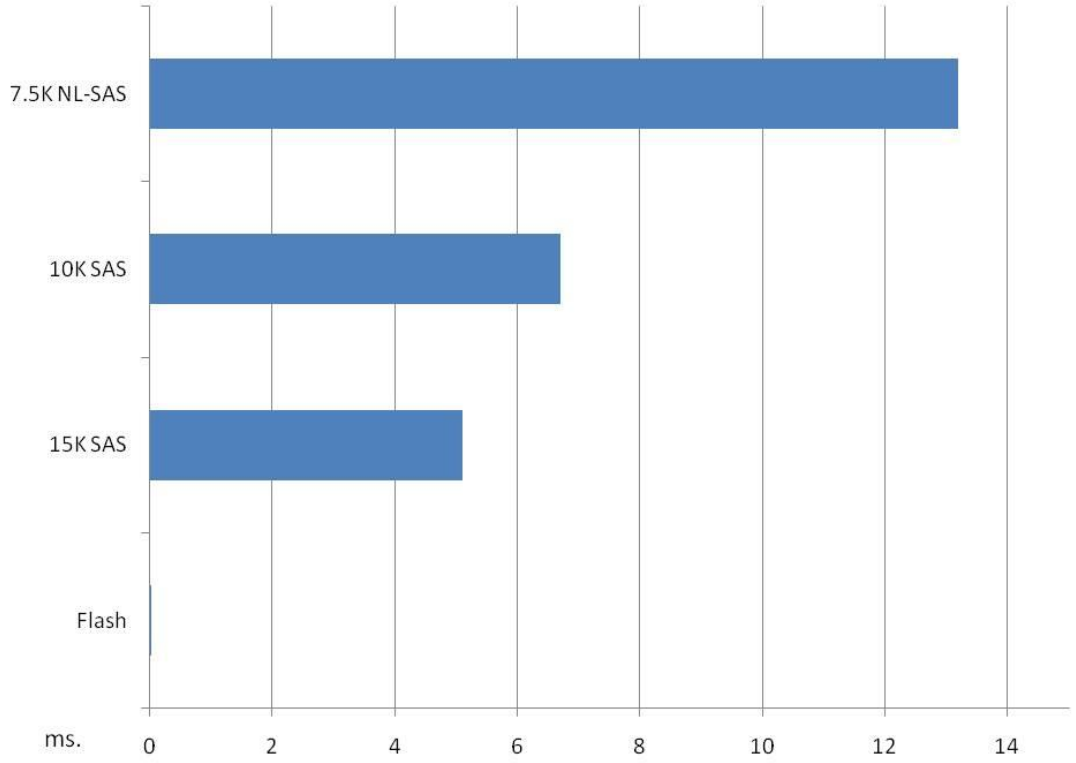

Figure 30 Drive typical service time comparison

From the figure, you can see that a hard drive's rotational speed has the largest effect on its performance. Note that flash drive service time is barely visible on the millisecond time axis of the figure. The very low read and write times of the flash drives give them the best performance. The 15k rpm SAS drives have the lowest service time of mechanical hard drives. The 10k rpm SAS drives provide an intermediate level of performance. The lower rpm NL-SAS drives have more modest performance. *Note that the difference in service times is an important consideration in random workloads, but less so in workloads characterized by sequential I/O.*

RAID group performance

The different RAID levels have different performance capabilities.

Striping is responsible for the largest part of a RAID level's performance. The remaining differences in RAID-level performance are only apparent under specific conditions involving the storage system's cache.

Striping performance

In I/O to a striped RAID level, the host I/O request is broken down into one or more I/O requests to the individual disks of the striped-RAID group based on I/O size and location. These I/Os are sometimes called *back- end I/O* or *disk I/O*. The characteristics of how a RAID implementation executes back-end I/O have a measureable effect on performance.

RAID 1/0, 0, and all parity RAID levels are striped.

Generally, a striped RAID level has better *read* performance than an unstriped RAID level. This is because it uses less back-end bandwidth. Striped parity-type RAIDs can deliver higher sequential write bandwidth than mirror RAID types.

Disk striping affects the amount of data that each disk transfers before seeking for the next I/O.

Striping has two beneficial effects: distribution of random I/O over more than one drive, and large-block parallel access.

In general, for any given workload characterized by random I/O, more drives mean better performance. Getting increased performance with sequential I/O is more complex. There is a dependency on the implementation details of the storage system's RAID operation that must be considered.

High bandwidth, especially in random, large-block workloads, is dependent on the I/O size at the drive. RAID groups with fewer drives deliver higher *per drive* bandwidth than groups with a large number of drives. This is because larger I/Os can be sent to fewer drives. The highest bandwidth may be achieved by using a larger number of RAID groups with a smaller number of drives per group rather than one large group. This also benefits by reducing contending threads. This is because there are fewer LUNs on a small RAID group for a given LUN size.

Stripe element size

The ideal stripe element size maximizes the amount of data each drive transfers in a single I/O request. The VNX's stripe element size is fixed at a value that has been highly optimized for use with its hardware and software architecture. The stripe element size is 64 KB.

Stripe size

Stripe size is the number of drives in the stripe times the stripe element size.

Disk crossings (alignment)

I/Os that cannot be accommodated by a single stripe element cross to a second disk. These are called *disk crossing I/Os*. (See Figure 31 on page 106.) Disk crossings adversely affect mechanical drive-based storage only. However, disk crossings are not necessarily a sign of a problem. It is normal for I/O larger than the stripe element size to cross drives. In the case of I/O smaller than the stripe element size, disk crossing I/Os may have a modest adverse effect on performance.

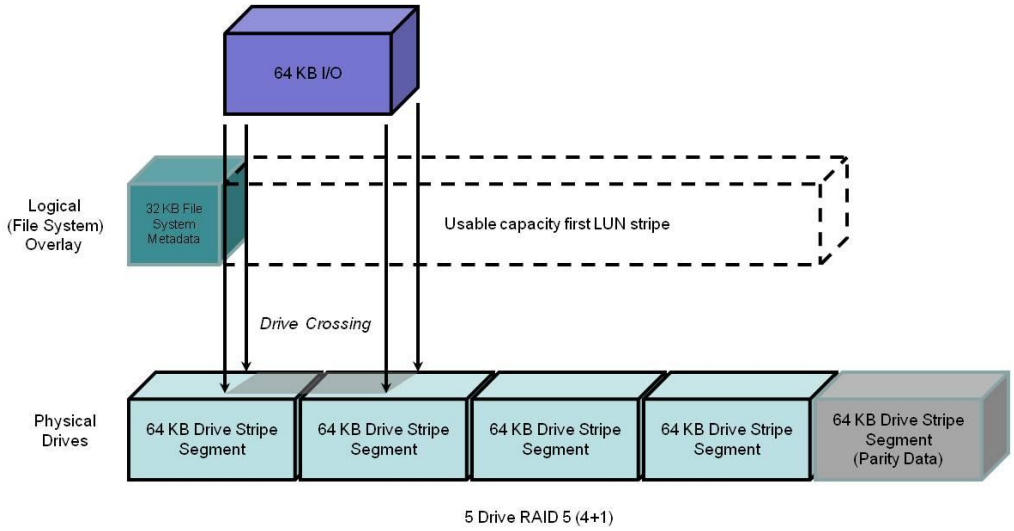

Figure 31 Disk crossing, unaligned LUN

In a disk crossing, an I/O is split across two drives instead of more efficiently occurring on only one. The splitting of the I/O across two drives increases the load on the drives and the back end. This results in longer queues for a given I/O rate. Longer queues increase response times for any I/O a drive receives. Even if the disk operations are buffered by cache, the effect can be detrimental, as it is still adding to disk queues, which affect all disk-related performance. Sequential large-block I/O benefits the most from alignment; random small-block I/O the least. Random reads, which by nature require disk access, are also affected, both directly (waiting for two drives to return data) and indirectly (making the disks busier than they need to be).

It is important to note that with its default 64 KB stripe element size, any I/O larger than 64 KB on a VNX storage system must involve a disk crossing.

For example, a 128 KB I/O block size must cross two drives.

The percentage of additional I/Os being caused by a particular I/O block size can be calculated as follows:

percentage disk crossings = (I/O block size / stripe element size) * 100

For example, with an 8 KB I/O, the expected percentage of disk crossings is about 13 percent ((8 KB / 64 KB) * 100). Note that this percentage is unavoidable, and should be considered ”normal” behavior.

Alignment ensures that I/O more neatly fits into stripe elements. This makes write I/Os more efficient. File system metadata can cause disk crossings through misalignment. While an aligned file system would quickly service the I/O with a single drive, a misaligned file system requires two for some portion of the I/Os.

LUNs can be aligned using file system utilities. (See Figure 32 on page 107.) Alignment is performed after the LUN is bound and presented to the host. The alignment process can only be performed on a LUN with no data. Alignment cannot be performed on previously partitioned LUNs.

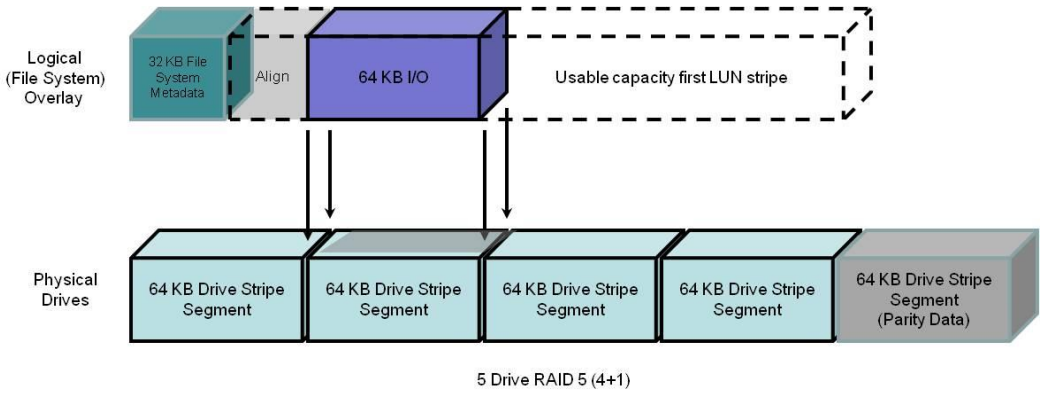

Figure 32 Aligned LUN

MS Windows 2008 Server volumes are automatically aligned by the host. Other O/Ss, such as Linux, may require a manual alignment. Note that Virtual pool thick LUNs and traditional LUNs may benefit from alignment. Virtual pool thin LUN do not benefit from alignment.

Full-stripe writes

The storage system in its normal cached operation optimizes its drive's I/O to perform full-stripe writes when possible. A full stripe write is sometimes called an *MR3 Write*. In a full-stripe write, either the I/O incoming is large enough to fill the stripe and aligned, or the storage processor accumulates write I/Os in write cache. Depending on the locality of the data, smaller I/O writes can be coalesced into fewer larger writes. Individual write requests can be sequenced until an entire stripe is cached before being written. This makes for an efficient use of the back-end ports and the drives.

Note that if uncached I/O is performed, such as when cache bypass is in effect, if the I/O completely fills the stripe or is an integer multiple of the stripe width, and is aligned, a full-stripe write is performed. However, when uncached, it is difficult to control and ensure a full-stripe write. For this reason, it is hard to get consistent high bandwidth with the cache disabled.

Cached performance

Under normal, fully cached operating conditions, the performance of the different RAID levels is essentially the same. The storage system's large physical cache and sophisticated caching mechanism result in optimally executed I/O performance. This caching results in a leveling of any performance differences between the RAID levels.

A large difference is seen when write cache is full and flushing occurs (see "Memory" on page 52). Generally, RAID 1/0 RAID groups have a modest to moderate advantage in flushing random writes from the write cache. This advantage also exists in normal operation, but the advantage is not pronounced. RAID 3, RAID 5, and RAID 6 groups have an advantage flushing large/sequential writes from the cache.

Uncached performance

Under circumstances of uncached write I/O, or the rare disabled write cache operation, the RAID level has a performance effect.

Uncached I/O can occur when a cache is disabled by the user. Caching may be disabled on certain LUNS to give preference for cache usage to other LUNs. I/O also bypasses write cache based on the LUN write-aside setting. LUNs made up of flash drives may or may not have write cache disabled. A disabled cache can also occur when the storage system is in a degraded mode. The high-availability cache of the VNX series also makes this state very uncommon.

See "SP cache" on page 52 for information on uncached I/O and disabled cache conditions.

RAID level performance differences

This section describes the relative performance differences between the RAID levels based on I/O type and access type. *Note that these differences would only be noticed under uncached, or cache full "forced flushing" I/O conditions*

Drives can be chosen from any DAE or DPE location on the storage system to be included in a RAID group. All the disks in the RAID group must be the same type of disk (NL-SAS, SAS, or flash drive).

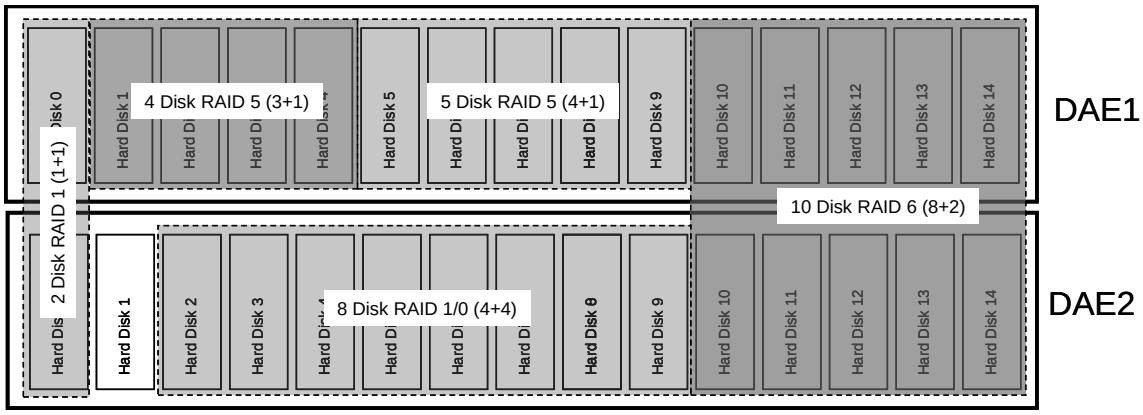

Figure 33 Conceptual RAID group binding on two 15-drive DAEs

Figure 33 on page 108 shows a conceptual view of several RAID groups created on two DAEs. Note that RAID groups can span DAEs.

RAID level performance: parity versus mirror

For individual I/Os, there is no read performance difference between parity and mirror RAID levels. There is a write performance difference.

CPU utilization

Computation of parity uses storage processor CPU resources. Mirror RAID levels do not have to compute parity. Without the parity computation, mirror RAID usage results in slightly lower storage processor utilization.

Back-end I/O operations

Read and write requests to LUNs are different from read and write I/O operations on the storage system's back end. A single LUN read or write request results in one or

more read or write I/O operations on the back-end port(s) and to the drives of a LUN's RAID group.

A read request generates a single back-end read or more than one, if the size of the read is larger than the stripe element size.

I/O operation to storage.

A single write request can generate one or more I/O operations. The number of I/Os generated by write requests on the storage system's back-end is the main difference in RAID level performance.

The number of back-end write I/Os depends on the RAID level and the operation of the cache. A full-stripe write to a parity RAID level results in one back-end write I/O operation; the stripe would be written, one for the data to each drive in the group and one to the parity disk in the group (see "Full-stripe writes" on page 107). A full-stripe write to a mirror RAID is also a back-end write to each stripe, one to the primary stripe and another to the mirroring stripe. Because there are more drives to service a stripe of any arbitrary size, more data in is sent to the back end. Note that for the same number of user drives, a mirror RAID group consumes more back-end port bandwidth than a parity RAID level group under this circumstance.

Random I/O is different. Assuming an I/O is smaller than a stripe element, mirror RAID writes generate two back-end write I/Os — one to each drive of the mirror's pair. A random write request to a parity RAID generates two read and two write I/O operations.

For example, an uncached, RAID 5, random, 8K write request:

1. Reads destination data sector(s) from the drive.
2. Reads parity from drive holding parity for this stripe (new parity is calculated from this data).
3. Writes a new sector including data to destination drive.
4. Writes (new) parity to the drive holding parity for this stripe.

Mirror RAID levels (RAID 1 and RAID 1/0) perform better with small-block and random writes. For random writes, parity RAID levels (RAID 3, RAID 5, and RAID 6) require the storage processor to read, calculate, and write a parity sector for each write operation. One parity calculation is needed for RAID 5 and RAID 3. Two parity sector calculations are needed for RAID 6. The time and resources needed to perform this calculation adversely affect parity RAID level performance during random writes.

Parity RAID levels perform better with large-block and sequential writes. With sequential writes, when full-stripe writes are possible, parity RAIDs performing fewer writes perform better than mirror RAIDs. This is because less redundant data is written: a single parity element as opposed to the two complete stripes of the mirror level RAID.

Mirror RAID levels (RAID 1 and RAID 1/0)

RAID 1

RAID 1 offers the best bandwidth on a per-drive basis. Random write performance is better than most RAID levels but limited by the maximum RAID group level size of two drives. Sequential read performance is about the same as a single drive.

In RAID 1, a read is one back-end operation and a write is two.

RAID 1/0

RAID 1/0 receives a performance benefit from striping. RAID 1/0 includes some optimization of reads that takes advantage of two drives with identical data. It has very good random read and write performance. It also has good sequential read and write performance.

A RAID 1/0 group can contain between 2 and 16 drives. A two-drive RAID 1/0 has identical performance to a RAID 1 two-drive RAID group.

In RAID 1/0, a read is one back-end operation and a write is two.

Parity RAID levels (RAID 3, 5, and 6)

RAID 3

RAID 3 has good random read performance due to striping, but not as good as RAID 5. There are no reads from parity drive. Random write performance suffers due to parity computation, and the chance of bottlenecking at the single dedicated parity disk. Sequential read performance is likewise good due to striping. Sequential write performance is excellent, as RAID 3 has the shortest, simplest code path of all parity RAID types.

In RAID 3, a read is one back-end operation, and a write is one for a full stripe and up to four for less than a stripe.

RAID 5

RAID 5 has excellent random read performance. Performance improves with increasing numbers of disks in the RAID group. Random write performance is fair, due to the parity calculation. RAID 5 random write performance is better than RAID 3, because there is no dedicated parity disk to bottleneck. Sequential read performance is good. Sequential write performance is good.

In RAID 5, a read is one back-end operation, and a write is one for a cached full stripe and up to four for a single small I/O less than a stripe.

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. It has the lowest random-write performance (equal user data drive count) of any RAID level. It has excellent random read performance. Sequential read performance is good. Performance improves with smaller stripe widths. Sequential write performance is fair.

In RAID 6, a read is one back-end operation, and a write is one for a cached full stripe and six for a single small I/O less than a stripe.

RAID 0

RAID 0 is a special case. It offers all the performance advantages of a striped RAID. It also does not have the computational overhead of parity calculation. However, there is no data protection built into RAID 0.

RAID 0 has very good random read and random write performance, particularly with large stripes. Sequential reads and writes are also very good.

In RAID 0, a read is one back-end operation, and a write is one for a full stripe and one for less than a stripe.

RAID group performance calculation

The throughput or bandwidth performance of a RAID group depends on the provisioning of the RAID group and the I/O it is performing.

Throughput estimate

The available back-end IOPS and bandwidth of a RAID group depend on the RAID group's:

- Number of drives
- IOPS or MB/s capability per drive model

A gross performance estimate can be made by multiplying the IOPS or bandwidth times the number of hard drives in the RAID group.

For example, assume a single 15k rpm SAS hard drive is capable of about 180 IOPS. For a 5-drive RAID 5 (4+1), user data is spread across all five drives. This RAID group has a potential throughput of 900 IOPS (180*5).

For a mirror-type RAID, during read, either stripe (primary or mirror) may be read. Reads are optimized to use the stripe with the fastest access time. During writes, both the primary and mirror are written. This requires that the IOPS be multiplied by two. However, the net effect is that only the IOPS of the primary stripe need be used in the calculation.

You should *not* do final performance planning on the basis of this type of calculation. It is not possible to accurately calculate the RAID group's performance without knowing the characteristics of its I/O.

Throughput calculation

I/O characteristics have a significant effect on RAID group performance. A more valuable and accurate estimate of a RAID group's bandwidth, throughput, and response time need to account for the:

- I/O type
- Access type
- I/O size

The I/O type (random or sequential), the ratio of reads to writes; and the size of the I/O all affect the performance (see "I/O characterization and workload" on page 31 for details).

The detailed method for RAID group performance calculation using RAID group characteristics and I/O types is given in the *EMC Unified Best Practices for Performance and Availability: Common Platform and Block OE XX — Applied Best Practices* white paper along with the information to perform these calculations.

LUN performance

The performance of LUNs depends on the performance of the underlying RAID groups.

Creating Virtual Pool LUNs in pools with the largest practical number of drives, or traditional RAID groups with the largest practical number of drives, is the easiest way to ensure high -performance LUNs for small random IOPS — more drives give more performance. Increasing the capacity of the LUN within its current Virtual pool or traditional RAID group does not improve its performance.

In a Virtual pool with more than one thick or thin LUN, or a traditional LUN on a partitioned RAID group, LUNs share the underlying RAID group's IOPS and bandwidth. That is, if separate applications each use a LUN created on the same RAID group, the combined needs of the applications may exceed the RAID group's performance capabilities.

Contention is when the I/O of two or more LUNs bound to the same RAID group interferes with each others' performance. Avoid *linked contention* and *drive contention*. Linked contention is when I/O to different LUNs forces the RAID group's drive heads 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.

It is recommended that you limit the number of LUNs per pool or per RAID group, because as the number of LUNs increases, it becomes more difficult to predict or determine if the I/O is complementary or contentious without time-consuming analysis and debug.

It is important to remember that more drives result in better performance than fewer drives.

Short stroking

Short stroking is a provisioning technique for increasing the throughput of a LUN on a RAID group made up of mechanical hard drives.

Conventional hard drives incur additional time to service I/O each time the read/write heads are required to move for a seek. Short stroking reduces these head repositioning delays by limiting the number of tracks being used on the drive. The outermost tracks are used on a drive first. The outermost tracks have more sectors per unit length than inner tracks. By limiting the number of tracks, we reduce service time for each I/O. That directly affects the response time of individual disks and of the RAID group as a whole. However, limiting the number of accessible tracks to achieve this performance gain reduces the usable capacity of the RAID group.

Small-block random I/O workloads with many threads benefit the most from short stroking, large-block sequential workloads benefit the least. A short-stroked RAID group is set up by creating a single traditional LUN on the RAID group. This positions the LUN on the *outermost tracks (fast tracks)* of the RAID group's drives. The smaller the capacity of the short-stroking LUN in relation to the RAID group's overall capacity, the greater the performance increase.

Ideally, only a single traditional LUN should be bound to a RAID group when short stroking, to eliminate the chance of LUN contention. Note that short stroking is not applicable to Virtual pools. In addition, the advent of flash drives has made the technique of short stroking obsolescent, except in performance benchmarking.

Chapter 10 Storage System Performance

This chapter presents these topics:

Optimal storage system performance is achieved by balancing the workload between the storage system's two storage processors. The workload balancing relates to the number of LUNs, the hosts allocated to each storage processor, and the I/O those hosts are directing toward those LUNs. Balancing the workload requires an understanding of the following related subsystems within the storage system:

- Front end — Consists of the front-end I/O ports that attach the storage system to the hosts.
- Storage processors (SPs) — Contain the storage system's CPUs and memory. Memory includes the important storage-system cache.
- Back end — Contains the I/O ports that connect the storage processors to the devices making up the LUNs.

The goal is to avoid any *bottlenecks* in any of these subsystems. A bottleneck is a condition that may constrict the flow of data from drives to the front-end ports.

Percentage utilization

Percentage utilization is a measurement of how busy a resource is. It is used in determining a system's performance capability. High utilization is not necessarily bad. Logical devices such as LUNs do not suffer physical limitations. Physical devices, such as storage processors, front-end ports, drives, and back-end ports do. For them, getting to high utilization may be a requirement for maximum throughput or IOPS. Conversely, if the goal is low host response time, seek low utilization. This is because with physical devices, high utilization is accompanied by longer queues and the resulting higher response times.

Utilization is an average over a period of time. At any point, a device is either busy or idle. The measurement is calculated by taking the time the device is busy, divided by total elapsed time.

For example, if a hard drive is receiving and servicing I/O requests for 300 ms during a 1-second interval, then its utilization is 300 ms / 1000 ms = 0.30, or 30 percent utilization.

A device or system receiving enough requests to never be idle has a utilization of 100 percent. One hundred percent utilization of a physical resource is not desirable. Unnoticed oversubscription becomes possible at that utilization. Oversubscription of a resource may not be easily detected, because the metrics become ambiguous.

Note the difference in determining the utilization of logical versus physical storage objects. The performance of physical storage objects is easily quantifiable. The hardware provides accurate metrics for calculations. The creation of logical storage objects that include physical storage devices introduce many dependencies, some of them obvious, others not.

For example, if a physical storage object is already at 100-percent utilization and it receives 50 percent more requests, its utilization is *still* 100 percent. This is despite the resource being heavily oversubscribed. Although, depending on the workload, a logical storage object such as a LUN at 100 percent *may* still have some available performance, depending on the physical drives supporting it.

Utilization reserve

From a planning perspective, systems with very high usage may be at the limits of their performance. This leaves them without any performance margins, should they enter degraded mode or see spikes in their workload. Systems with low utilizations have more overhead for bursts or to make up for failures.

It is prudent for utilizations of redundant resources *not* to be greater than 70 percent at the same time, for long periods of time. A reserve of 50 percent would be the conservative ideal. This is needed to support operation in degraded mode. In the event of a complete or partial failure, there needs to be enough performance margin to completely accommodate the failed resources load without a significant performance degradation.

A spike is a sudden and unpredictable large increase in activity such as I/O or CPU utilization. To manage spikes requires that a margin of storage system performance resources be held in reserve. Otherwise, user response times may suffer if spikes occur during busy periods. The reserve includes uncommitted SP utilization capacity, I/O bandwidth, and storage capacity. This reserve should be sized to handle the "worst case" demand of the spike.

Storage system resource utilization

There are three physical resources whose percent utilization is particularly important to monitor on the storage system. All are highly workload dependent. They break down into CPU, memory, and I/O utilization for the storage system. The resources are:

- Storage processor
- Cache
- Storage

Storage processor utilization measures how busy the storage processor's CPU is. Storage processor utilization affects how many tasks the storage processor may perform and how quickly it executes them.

Cache utilization is a memory utilization metric. Cache utilization directly affects the storage system's ability to perform I/O in a timely fashion.

Storage utilization includes back-end bus and storage-device utilization. Back-end bus utilization cannot be directly measured without analysis tools; it must be inferred from disk utilization. Disk utilization of greater than 70 percent for long periods is an indication of a performance problem.

Storage processor

For high performance *and* availability, storage system design should target storage processor utilization to be about 50 percent. However, CPU utilization may be allowed to reach *up to* 70 percent for either or both storage processors for short intervals. It is not prudent for *both* storage processors 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 storage processor failure, neither storage processor

has enough margin to completely accommodate its peer's load without a significant performance degradation.

For example, if both storage processors have utilizations of more than 70 percent, in the unlikely event of a storage processor failover, neither storage processor would have enough margin to take up its peer's load. In that event's wholesale LUN trespass, the peer SP's assumption of the additional load may result in unacceptably higher host response times.

With the advent of multicore CPUs, it is important to understand the storage processor utilization metric. In a multicore CPU, each of the cores is somewhat independent in performing functions. There may be different utilizations for each of the CPU's cores. The utilization of individual cores may be raised by the execution of individual features, such as parity calculation, LUN compression/decompression, and FAST VP relocations. The average CPU utilization of a multicore CPU may not indicate the true state the storage processor.

For example, in a four-core storage processor CPU, utilizations for each of the cores may be 10 percent, 100 percent, 50 percent, and 20 percent. This would result in an average storage processor utilization of 45 percent. At first look, storage processor utilization is well within limits. However, one of the cores is at 100 percent. It can perform no additional functions. Its full utilization is a potential bottleneck, which is not indicated by the metric.

Read/write cache

Read cache optimizations

With sequential read requests, the chances of a cache hit can be greatly increased through a technique called *prefetch* (this is also referred to as *read-ahead*). Here, when sequential access is detected by the SP, the read cache is filled ahead of actual host requests in anticipation of their being requested later.

Because of read-ahead, workloads characterized by sequential I/O can have a high percentage of cache hits. This greatly improves their response times. Workloads characterized by random I/O do not benefit from this technique.

Read cache hit ratio

The read cache hit rate (RCHR) is the percentage of read I/O requests found in the read cache. It is one of the metrics for determining if a read I/O workload is random or sequential.

If a workload's read I/Os are purely random, the RCHR can be up to 5 percent. If it is higher, there is probably either a mix of random and sequential I/O, or a multithreaded sequential workload being measured. There is also the chance that the same addresses are being repeatedly accessed or that there is high locality in the I/O.

For example, the metadata on the first 64 KB of a file system volume is a set of addresses that typically receives repeated access.

For purely sequential I/O, the metric's value depends on the number of threads performing the I/O. With single-threaded sequential, the RCHR should be about 100

percent, but not less than 80 percent. This is an example of the prefetch working at peak efficiency.

Multiple-stream multithreaded sequential access can reduce the effectiveness of the prefetch, which lowers the RCHR. Multiple-stream multithreaded sequential I/O causes many read accesses to many different regions of a RAID group's drives. This slows the prefetch process lowering the RCHR. Modifying the prefetch parameters may increase an RCHR in the lower end of the range. However, an in-depth read cache usage analysis is needed to determine the cache parameter settings for a higher RCHR. The final possible reason for a low RCHR value is that the workload is not truly sequential. The larger the random I/O component to the workload, the lower the value of RCHR.

Flushing overview

Flushing is writing a write-cache page out to storage. The caching algorithms flush the pages in order to keep adequate margins of cache to handle bursty events. Ideally, the number and frequency of write I/Os from hosts are in balance with the storage system's ability to flush data to storage. This type of balance is a result of good performance planning.

Write cache optimizations

Caching of write I/Os is a more efficient use of memory than caching of read I/Os.

Sequential writes when detected are optimized. Several smaller I/O writes can be *coalesced* into fewer larger writes to storage. Ideally, they can be coalesced into one or more full-stripe writes. In addition, writes to the same location when found in cache are superseded with the last write. This is a very efficient use of the storage system's back-end resources.

Backfill is a write cache optimization where data is read from storage so that a more efficient write I/O size can be achieved.

For example, in a RAID 5 (4+1) RAID group-based LUN, if a 192 KB write is pending, the adjacent 64 KB from the destination LUN may be read to achieve a more ideal 256 KB full-stripe write.

The storage system also has the option to write data directly to disk. The technique is called *write-aside*. It may also be referred to as *write-through*. The write-aside can be adjusted at the LUN level though Unisphere.

Write-aside allows large I/Os to bypass the cache when cache is enabled. The I/O goes directly to the storage devices. This prevents very large write I/Os from consuming too many cache pages. Losing these pages would adversely affect the overall workload's write-cache bandwidth. The bypassing I/O would have somewhat higher response times, being uncached. However, this is somewhat offset by the efficiency with which large I/Os can be handled.

Write cache management

Properly sizing cache to support its provisioned number of drives and ports is an important tuning task. In this tuning, the capacity of the primary cache is not as

important as the storage system's ability to perform orderly flushing of write cache pages, and to perform prefetches to read cache.

Flushing details

There are three types of flushing:

- Idle flush
- High-water flush
- Forced flush

Idle flushing is the execution of the pending I/Os to idle LUNs. A LUN is idle when it has no I/O for two seconds or longer. Idle flushing is a regularly occurring background process. As LUNs are found to go idle, their pages are flushed from cache to make room for the I/O of active LUNs. If no LUNs are idle or if a LUN has previously been flushed down to no dirty pages, there is no idle flushing.

High-water flushing activates when the percentage of dirty pages reaches a preset limit. This limit is called the *high watermark*. The high watermark is typically set to maintain a reserve capacity in the cache, which is intended to handle bursts of writes. When high water flushing is triggered, the SP increases the cache flushing rate to increase the flush rate over that of idle flushing. SP performance is minimally affected by high water flushing. Note that a percentage of 90-percent dirty pages for a sustained period is an indication of a performance problem, which may be corrected by adjusting the high watermark.

The write cache attempts to always have memory ready for incoming writes. Forced flushing is avoided, but it occasionally occurs even in a well-designed system. When a write-I/O request is received and cache is already full, a forced flush is triggered to write the pages of the destination LUN receiving the current request and to clear pages for later requests. The new write request receives space on a newly cleared page. Forced flushing can be triggered many times in succession until the cache can provide enough pages for incoming write requests. Forced flushing occurs occasionally when a single LUN runs out of its available pages. On a well-tuned storage system, forced flushing for all LUNs should occur only rarely.

A forced flush means the host is waiting for physical disk operations and full RAID operations, so host response time goes up. During any heavy flushing storage system, performance is adversely affected, as the writing of the cache's contents to the drives contend with other I/O requests received.

The use of flash-based storage, as in FAST cache and flash-based LUNs and 15k rpm mechanical hard drives, minimizes the effect of forced flushing. The lower service time of these devices allows for more pages to be flushed in a shorter period of time.

Cache enable/disable

There are certain conditions where caching may not be desirable or needed. Under these conditions, cache can be disabled.

Cache is enabled or disabled at two levels: storage processor and LUN. The storage processor is the higher level. If cache is disabled at the storage processor, there is no

caching at the LUN level. If cache is enabled at the storage processor level, individual LUNs may have their caches enabled or disabled.

The VNX series has a very robust caching. It operates fully cached through most of its operational usage. Storage processor caching is automatically disabled only if the following conditions are not met:

- At least one SPS is present and fully charged (write cache disabled only).
- At least one power supply is present and functional in the DPE/SPE. (Some models can take three PS faults). In SPE models, both DAE O/S power supplies must be functional.

The storage system's caching is controlled through Unisphere.

LUN caching

Caching is typically configured at the LUN level. Write cache pages are dynamically allocated from the available pages, based on the volume of I/O a LUN is experiencing. Busy LUNs are allocated the most pages; LUNs with no activity have no pages allocated. When there are many busy LUNs, the caching algorithms moderate how many pages a LUN receives.

LUNs can have any combination of read and write cache enabled or disabled. By default, a LUN's caching is enabled.

Flash-based LUNs are different. The caching of these LUNs is operating environment version dependent. For OE block 31.0 They have cache disabled as a default.

LUN caching is typically manually disabled to preserve cache pages when there are conditions that might tax the storage processor's cache. One example is when running multiple applications that generate large I/Os. Another example is to disable cache before running multiple full SAN Copy sessions. (SAN Copy is a bulk data replication layered application; see "Replication layered applications" on page 35.)

Cache performance

The storage system's cache is highly configurable. For most workloads, the default settings perform very well. However, for atypical workloads, or to achieve the highest levels of performance, cache can be tuned through the adjustment of a just a few parameters:

- Cache page size (global)
- Prefetch (local)
- Watermarks (global)
- Write-aside size (local)

Note that some parameters are local and are apply to single LUNs and global apply to all LUNs.

Cache page size

Cache page size applies to both read and write cache. Cache pages have a fixed size in KB. They may be 2, 4, 8, or 16 KB. Both read and write cache use the same page sizes. An 8 KB page size is the default.

Ideally, the cache page size is the same as the most common I/O write request size received by the storage system. For systems performing mostly large I/O or sequential I/O, a larger cache page size improves overall system performance. However, where I/O request sizes are mixed, the default size is the best.

Prefetch

Prefetch applies only to read cache. The storage system offers constant and variable prefetch for read I/O requests.

Constant prefetch is useful for tightly controlling the amount of data prefetched to avoid filling read cache. With constant prefetch, the same number of KBs is prefetched ahead of read I/O requests when sequential access is detected. The caching algorithms flush the pages and maintain the efficiency of the storage system, along with keeping adequate margins of cache to guard against bursty behavior. Constant prefetch is most efficient when I/O sizes are uniform.

Variable prefetch works using a combination of the read I/O request size, a segment multiplier, and a prefetch multiplier. Variable prefetch is the default.

- Read I/O request size: Is the size in KB of the read request from the host.
- Prefetch multiplier: Is used to calculate the total size of the prefetch.
- Segment multiplier: Determines the size of I/Os to the back end.

If the read request size is smaller than a cache page, the cache page size is used. The prefetch multiplier is used to calculate (prefetch multiplier * read request size) the total amount to prefetch in this request; the segment multiplier (segment multiplier * read request size) is then used to calculate the back-end I/O size that fulfills the total request.

If the segment multiplier and the prefetch multiplier are equal, a single request to the back-end results. If the segment multiplier is smaller than the prefetch multiplier, multiple requests to storage result. The segment multiplier is never larger than the prefetch multiplier.

For example, assume:

- Cache page size: 4 KB
- Read I/O request size: 8 KB
- Prefetch multiplier: 4
- Segment multiplier: 2

The read I/O request size is larger than the cache page size. The request size is used. The total amount prefetched is 32 KB (4 * 8 KB). This 32 KB is read from storage in two requests of 16 KB (2 * 8 KB) each.

For small I/O sizes, setting segment multiplier and prefetch multiplier to the same large number improves performance, because the back end can work more efficiently. This setting is particularly efficient during sequential reads when the read cache is large enough to hold the result. Setting them the same for large I/Os can cause prefetching to consume valuable back-end bandwidth. In production environments, setting the segment multiplier to one-half or one-quarter of the prefetch multiplier breaks up the prefetch sequence, allowing other I/Os on the storage system to execute. A smaller segment multiplier results in lower read throughput.

LUN prefetch settings

There are limiting parameters that can be applied to prefetch to prevent excessive storage requests that may adversely affect overall performance. These settings are made on a per-LUN basis. They are:

- Maximum prefetch (blocks)
- Prefetch disable size (blocks)
- Prefetch idle count.

Maximum prefetch setting is the limit, in blocks, for data requested. This is the maximum number of blocks to bring into read cache ahead of actual requests. For example, setting the maximum prefetch to 4096 restricts prefetch to 2 MB (512 bytes * 4096).

The prefetch disable size prevents I/Os of the disable size or larger from triggering a prefetch. This causes I/Os of this size and larger to be read directly from the disk uncached.

Prefetch idle count limits the execution of prefetch for a very busy LUN. The prefetch idle count is also used in cache flushing; changing this parameter should be done only at the prompting of EMC service personnel.

Watermarks

Watermarks are used to manage write cache flushing. The storage system has two watermarks: high and low. These parameters work together to manage the flushing conditions. Watermarks only apply when write cache is activated.

Cache capacity above the high watermark represents a reserve of pages to handle bursts of write I/Os. Watermark flushing starts when the high watermark is exceeded and continues executing until the low watermark is reached. Watermark flushing stops until the high watermark is reached again.

The margin of cache above the high watermark is set to contain bursts of write I/O and to prevent forced flushing. The low watermark sets a minimum amount of write data to maintain in cache. It is also the point at which the storage system stops high water or forced flushing. 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. This level can drop when a system is not busy due to idle flushing.

For block usage, the storage system's default low watermark is 60 percent. The default high watermark is 80 percent. Storage system defaults for file usage may differ.

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 see watermark flushing. The flushing activity is highest when the high watermark is passed. Intense flushing increases LBA sorting, coalescing, and concurrency in storage, but it may have an adverse effect on read I/Os. The smaller the difference between the watermarks, the more constant the flushing activity. A lower intensity of flushing permits other I/Os (particularly reads) to execute.

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

Figure 34 on page 124 shows how high and low watermarks work together to manage the write cache:

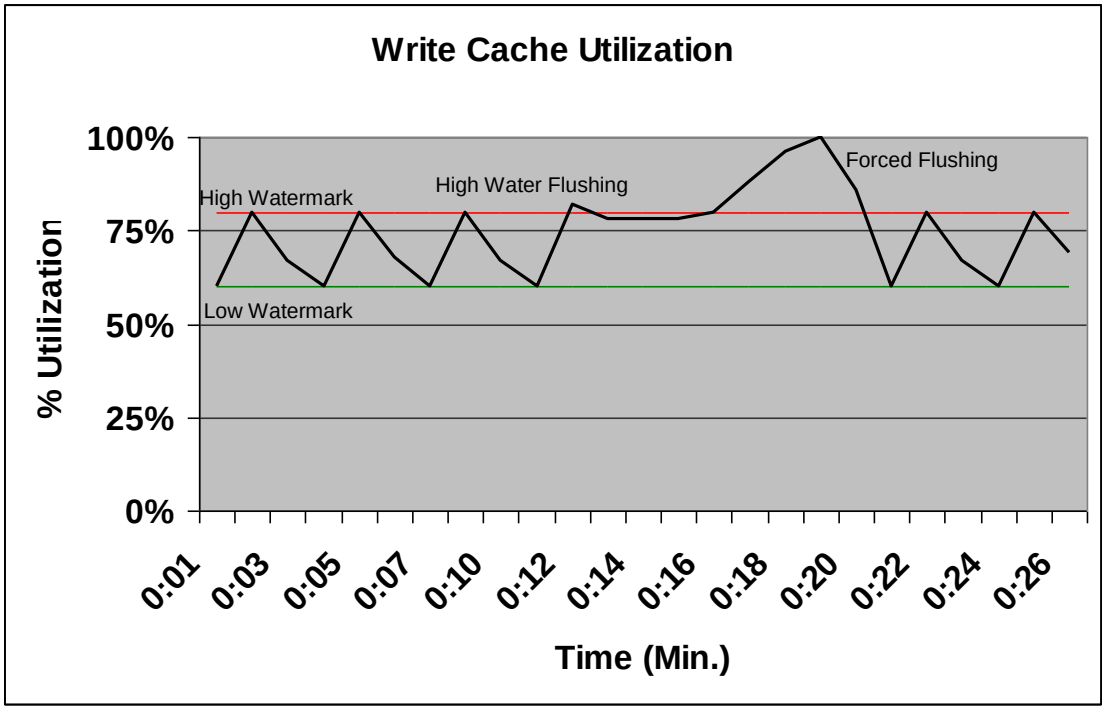

Figure 34 Write cache utilization

In Figure 34 on page 124, the black line is cache utilization, or the percentage of dirty pages. The workload's activity is shown to fill cache to the high watermark (red line). High-water flushing then flushes the cache down to the low watermark (green line), and cache is allowed to fill again. No flushing occurs until the high watermark is again exceeded.

Should the load increase (in the figure, this is the 12-minute mark), steady cache usage can occur as the workload's usage of cache pages and the flushing rate come to equilibrium. If there is a burst of I/O that fills cache and exceeds high water

flushing's ability to write pages out to storage, forced flushing occurs. (This is shown at the 17-minute mark.) The combination of forced flushing and high-water flushing make new pages ready for use. At the end of the burst, forced flushing and high-water flushing drive down cache usage to the low watermark.

The goal of setting watermarks is to avoid forced flushes while maximizing write cache hits. Forced flushes of greater than 20/sec for long periods are an indication of a performance problem. Using the watermarks, cache can be tuned to increase response time while maintaining a reserve of cache pages to match the workload's bursts, and system maintenance.

Write-aside size

The write-aside size parameter sets the largest I/O that is cached. Note that I/O needs to *exceed* this parameter to bypass cache. Write-aside is set on a per-LUN basis. Changing the write-aside is only available through the CLI.

Large writes primarily affect write cache bandwidth due, to write cache mirroring. The size configured for write-aside should be set large enough to avoid this bottleneck.

The write-aside size parameter is numerated in blocks. For example, 4096 blocks results in a write aside of 2 MB (512 bytes * 4096).

The default write-aside is 1 MB. That is, 1 MB write requests are cached, and write requests greater than 1 MB are *written through* (written directly to storage).

Setting the write-aside higher than the I/O sent by a high-bandwidth write application may cause cache to fill quickly, causing forced flushes. Setting the write-aside lower increases response time for I/Os larger than the write-aside size. If the target LUN is a parity RAID type, I/O bypassing the cache should be large enough to fill the parity stripe so that parity operations do not result.

Shared RAID groups

Drives are a shared resource between storage processors. A RAID group may contain one or more LUNs. The LUNs hosted by the RAID group may be owned by either storage processor. Both storage processors can own separate LUNs on a RAID group at the same time.

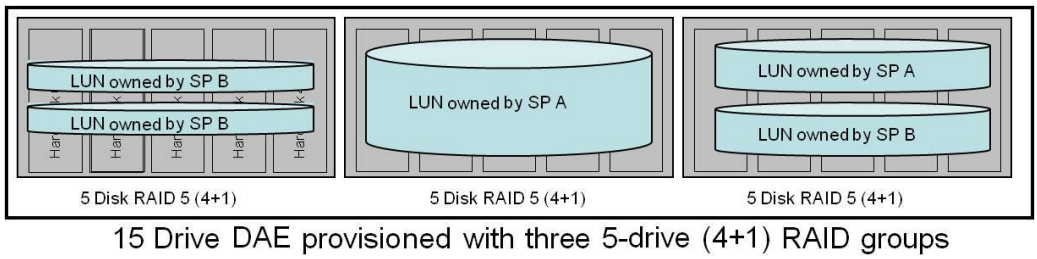

Figure 35 Shared RAID group drives

The simplest organization is to have sole usage of a RAID group by a single storage processor (shown in the center of Figure 35 on page 125). That is, a RAID group's LUNs are owned by a single SP. Sharing RAID groups between storage processors

allows for longer queues at the drive. A single storage processor is limited to the number of requests it can send to the drives. Dual-storage processor access to the drives is one way to get concurrency and achieve the maximum throughput rates from the drives. The tradeoff of longer queues at the disk is a somewhat longer response times.

Virtual pools inherently have LUNs sharing their private RAID groups.

Back-end bus performance measurement

The most important performance metric for the back end is the bandwidth from the drives. (Bandwidth is more likely to be exceeded than the limits for IOPS.) The bandwidth being received from the drives implies the bandwidth usage of the back-end port.

A bottleneck of the back end can be diagnosed from the bandwidth being received from the hard drives. Using Unisphere Analyzer, this information is found in the RAID group *total bandwidth* metric**.** Note that it is possible for a RAID group to span two or more back-end ports. This type of provisioning complicates the analysis.

In general, the I/O needs of the entire back end should be spread across all of its ports as evenly as is practical. Unsphere by default distributes RAID groups across back-end ports as evenly as possible. In round-robin order, each RAID group is provisioned on a separate port, with all member drives on the same bus. Be aware that this automatic distribution does not take into account the creation of LUNs and the I/O that is directed toward those LUNs.

Performance effect of buses

Horizontal provisioning is the practice of placing all of a RAID group's drives on a *single* back-end port. Most commonly, this is achieved by creating a RAID group fully contained within a single DAE, although spanning DAEs on the same back-end port has the same result.

The default allocation method in Unisphere is horizontal provisioning. Unisphere by default evenly distributes RAID groups across back-end ports. In round-robin order Unisphere provisions each RAID group to be on a separate port, with all member drives on the same port.

Vertical provisioning is a *multiport* RAID group provisioning strategy. A RAID group's drives are positioned on two or more back-end ports. This always puts the group's drives within more than one DAE, with each of these DAEs on separate back-end ports.

There can be a modest performance advantage with multibus provisioning. However, with the built-in parallelism of the VNX-series SAS back end, the performance increase is less than with legacy CLARiiON storage systems. However, large RAID groups ($\geqq 8$ drives), RAID 1/0 groups, and flash-based RAID groups with high-throughput workloads may benefit from being distributed over two or more back-end ports.

Multiport provisioning is advisable when:

- Bandwidth requirements result in the need for careful load distribution.

- Bursty host I/O is diagnosed to cause back-end port bottlenecks.
- Write cache destaging of a RAID group's LUNs is diagnosed to interfere with host reads due to port saturation.
- Large-block I/O requests from a particular RAID group are diagnosed to interfere with host I/O response time.
- Flash drives are provisioned servicing high-throughput workloads.
- Large ($\geqq 6$ drives) FAST Caches are provisioned.

Be aware, that multiport provisioning requires more maintenance discipline and planning to set up and maintain than does single-port provisioning.

Chapter 11 Availability

This chapter presents these topics:

Availability in the VNX is achieved through a combination of high-reliability components, and redundancy of components and data.

Reliability

Reliability is the ability of the storage system to run for long periods, and under operational stress, without suffering hardware failure or software faults. The VNX-series architecture contains many features designed to ensure a high level of reliability. High-quality components, assembly, design verification, and quality testing back up this design. The storage systems' systemic reliability is rated at *five 9s* (99.999 percent) uptime based on hundreds of thousands of hours of field experience. The exact reliability measurements depend on the model and components used in its provisioning.

The storage system is designed and built to protect user data, with full data path protection, throughout the storage system. This data path protection prevents unreported *data miscompare* errors. The UltraPoint hard drive array enclosures (DAEs) include a point-to-point design within the drive enclosures. The block operating environment monitors low-level diagnostics at the drive level that include fault detection, isolation, and error correction. RAID protection is supplemented with global and proactive hot sparing, sector-based checksum of all user data, and automatic background verification of the drive media.

Redundancy

Redundancy is the ability of the system to continue operating during a failure, without the host losing access to data, while maintaining data integrity. Redundancy is achieved through the duplication of critical components of the storage system and through provisioning choices such as RAID type and front-end connectivity.

In systems built with redundant parts and components, failure events are statistically independent. The occurrence of one failure makes it neither more nor less probable for another failure to occur. For the user to lose access to data, two or more interdependent components need to fail at the same time. The probability of two statistically independent failures occurring is calculated by multiplying the probability of one failure occurring by the probability of the other failure occurring. In other words, given a low probability of failure, the likelihood of two failures occurring at the same time is very small. This dramatically increases the overall up time.

Active/passive architecture

The storage system implements an *active/passive* architecture on a *per-LUN* basis from the hosts. Active/passive means there is a single operational interface active from a storage processor to a LUN. Note that this is a logical interface. Both storage processors are typically active for their owned LUNs. Peer storage processors can physically share RAID groups. In a failover (trespass), the passive storage processor assumes temporary ownership of its peer's LUNs. This is often hidden from the host by an internal redirection within the storage system.

However, multiple connections between hosts or networks and storage system front-end ports are encouraged, to achieve availability in the case of a single path or storage processor failure.

Multipathing

Multipathing allows for the *failover* from a failed *I/O path* to an alternate path when more than one path to a LUN on a storage system exists. An I/O path starts at the host bus adapters (HBA) on the host, and includes all the components and connections in-between all the way to a storage system. Hardware components on the I/O path include cables, switches, front-end ports, storage processors, back-end buses, and drives. There are also layers of software on this path, including hardware drivers, device firmware, and applications running on the host. Any component on the path can affect the availability of storage presented to the host.

The original I/O path is typically the optimal performing path. The failover-availability feature may have an adverse effect on performance. For example, the original path's performance may benefit from special tuning or optimal physical routing through the network or the storage system.

Host-based application support

Path failover requires a host-based application.

EMC's PowerPath® is a host-based application adding multipathing as well as several other performance and availability features. For example, a performance-benefiting feature of PowerPath is that, with multiple active paths to a LUN, I/O load can be balanced across the storage processor's front-end ports and the host's HBAs to increase bandwidth and avoid bottlenecks. An availability feature of PowerPath is auto-failback. *Failback* restores the storage system to its original I/O pathway configuration once it detects a failed I/O path is fixed.

In addition, there are multipathing applications native to the O/S, such as Microsoft Multipath I/O (MPIO®), and HP-UX Physical Volume Links (PV Links®) providing similar functionality. The latest versions of MPIO and PV Links support failback.

ALUA (Asymmetric Logical Unit Access)

The storage system supports the industry-standardized Asymmetric Logical Unit Access (*ALUA*) protocol. ALUA provides path management by permitting I/O to stream to either or both storage processors to reach a LUN. ALUA can reduce the effect of some front- and back-end failures to the host. Internally, if a component failure makes a LUN unreachable, ALUA shields the attached hosts from failing-over and trespassing LUNs. It does this by routing I/O through the peer storage processor to the storage processor owning the I/O's LUN. Externally, should a network failure occur, I/O can reach its source through the peer's ports, if they still have connectivity. All attached hosts receive this benefit whether or not they are set to use ALUA on the front-end ports.

The original path is the *optimal path*. The path failed over to is the *non-optimal path*. (See Figure 36 on page 132.) ALUA does not perform automatic failback. A quick return to optimal path usage should always be made. The non-optimal path's I/O

routing uses both storage processors for a single I/O. This lengthens host response time and may adversely affect storage system performance.

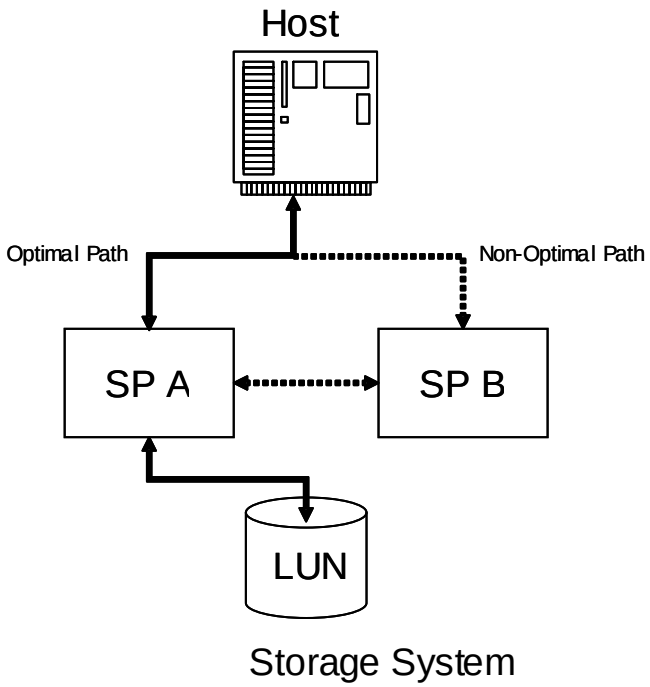

Figure 36 Optimal and non-optimal I/O paths

ALUA must be supported by the host's O/S. ALUA is supported by PowerPath, as well as by MPIO and PV Links.

Measuring reliability and availability

Reliability is the period of time that a device can perform its intended function upon demand. Reliability is calculated based on mean time between failures (*MTBF*) and mean time to product replacement.

Expected availability is presented as the percentage of time the system is able to access or transfer any desired block of data along with the number of downtime minutes per year. The reliability metrics for individual components are used to calculate the availability metric mean time to data loss (MTTDL).

Reliability metrics

MTBF

MTBF is a basic measurement of reliability. The MTBF metric is typically not well understood. Assumptions about it can lead to decisions that adversely affect availability.

MTBF is the period of time that passes before a component or system fails. Manufacturers use this measurement, and its value is usually a "best case." Typically, this period is given in hours.

For example, a disk manufacture's MTBF for a modern 15k rpm SAS hard drive is 2 million hours.

AFR

Annualized failure rate (AFR) is an alternative metric to MTBF. AFR is the probably number of failures per year for a given population. The AFR is a single number. It can be used to determine sparing policy.

For example, a disk manufacture's AFR for a modern 15k rpm SAS hard drive is 0.44 percent.

MTBPR

Expected reliability data is presented as the mean time between parts replacement (MTBPR) in hours, which reflects how often, on average, a system requires a part replacement. It is not prudent to wait for devices to fail. A disk's failure is usually predicted (due to a rate of media errors, for example) and the product is replaced before it fails.

Availability metrics

MTTR

Related to MTBF is MTTR (mean time to repair). This is a *maintainability* metric, and it is important to availability in that it defines how long a component or subsystem is in a degraded state. It is measured in hours. It is the amount of time needed to restore a system to functionality once a failure is identified. MTTR typically is repair time made with the assumption the failure is already correctly identified.

In the context of the storage system, the most common MTTR is the time it takes for a rebuild to complete. A rebuild occurs in the event of a RAID group drive failure. During the time it takes to complete a rebuild, a RAID group is degraded. The MTTR is an example of and how a rebuild affects availability.

Chapter 12 Storage Object Availability

This chapter presents these topics:

RAID was developed to provide data redundancy. Data redundancy is a technique that, in the case of a drive failure, provides that data can be reconstructed. On top of RAID, vendors such as EMC have developed other techniques to reduce the effect of media errors and data-transmission errors within the storage system. Lost or corrupted data can be detected and if possible reconstructed. Note that on the storage system the granularity of error detection and reconstruction is typically at the disk-sector level. That is, a sector is the smallest amount of data in which an error can be detected, and reconstructed.

Parity and mirror RAID levels differ in how they handle data redundancy to provide availability.

Physical storage object availability

The drives are designed and manufactured for reliability and high availability.

Hard drive failure modes

Hard drives are highly reliable storage with mean time between failures (MTBFs) of hundreds of thousands, if not millions of hours. However, a hard drive is a complex mechanical device. Drive failures increase with the age of the drive. Mechanical drives eventually wear out. They can also be damaged by shock, vibration, and excessive heat. Flash drives are more reliable, because they have no moving parts. However, they also wear out, having their own unique failure profile.

A hard drive has several ways it can fail. The broad categories of failure are sometimes called *failure modes*. Following are the most common failure modes:

- Mechanical
- Media
- Electronic

Mechanical

Mechanical parts failures are usually related to the spindle motor or its bearings, including the read/write head actuator. An example of a mechanical failure is the spindle motor not being able to reach its full operational speed during a spin-up. Excessive heat or vibration is one cause of mechanical failures. Old age is another.

Read/write head assembly failures can be mechanical or electronic failures. The infamous *head-crash*, where the read/write head scores the magnetic media of the platter, is a mechanical failure.

Rotational vibration (RV) can contribute to failures in mechanical hard drives. RV is measured in radians/sec^2. Vibration can come from inside the storage system, for example from neighboring hard drives and system fans. It can also come from outside the storage system, for example from nearby data center heavy equipment such as heating ventilation and air conditioning (HVAC) units. In addition, random and bursty workloads can cause drive vibration through rapid movement of the drive head.

Some drives in multidrive environments contain on-board optimizations including Rotational Vibration Feed Forward (RVFF) to attempt handling mechanical errors at

the drive level. RVFF detects and compensates for the ambient vibrations that may cause errors in positioning the drive heads.

Vibration of the hard drive can cause both mechanical and media failures. For example, vibration can increase seek times by delaying the time a hard drive head takes to settle, or cause the drive actuator to vibrate off track resulting in media errors being reported.

Mechanical failures are the most common failure mode. They are also a predicable error indicated by decreasing performance and increasing errors. Note that flash drives do not experience this type of failure.

Media

Media failures occur due to read or write errors to a sector. A media failure can also relate to a read/write head failure on a mechanical hard drive. On flash drives it can be failure to a memory page, block, or blocks.

A media failure is indicated by a read, write, or verification error occurring in a drive's sectors or pages. When a read error occurs, the sector is remapped. In remapping, the sector or page's data is transferred to one of a set of sectors for this purpose. As both mechanical and flash drives age, the number of media errors increases. A large number of remapped sectors or pages is a predictor of a drive failure.

Electronic

Electronic failures occur in the drive's circuit board when a discreet electronic component fails. There is also a possibility of a software error in the drive controller. A failure of the read/write head may be an electronic or mechanical failure. Heat and vibration are common causes of electronics failure. Old age is another. This failure mode is uncommon. It is also unpredictable.

Hard drive reliability classifications

Enterprise is the highest level of drive reliability. These drives are engineered and manufactured to provide reliable operation at 100% duty cycle over long periods of time. *Near-line* is a lesser, but still very high, standard of reliability. The lowest level is *consumer* or *desktop reliability*. Desktop hard drives are the products available at retail outlets. Desktop drives may be found in hosts, but never in EMC storage systems.

There are many differences between enterprise and near-line hard drives. For example, enterprise drives are typically dual ported rather than single ported, they have faster speed motors (15k rpm vs. 7.2k rpm), their media is "characterized" to verify its magnetic properties, and they have greater structural resistance to vibration. Enterprise drives also likely have dual controllers: one for platter/head tracking and another for bus arbitration and data dequeue/enqueue and buffering.

The most important difference is drive reliability is terms of the drive error rates. This metric is reported in terms of their unrecoverable error rate (UER). Table 10 on page 138 shows typical manufacturer-reported mechanical hard drive UERs.

Table 10 Manufacturer-reported mechanical hard drive UERs

Typical manufacturer reported UER values		
Drive reliability rating	**UER (bits)**	**Bytes**
Consumer	<1:10^14	<12.5 TB
Near-line	1:10^15	125 TB
Enterprise	1:10^16	1.25 PB

For example, the table shows that, with a near-line drive, an error will be observed about once for every 125 TB read from the drive. A 600 GB enterprise drive would require about 2,500 full-drive reads before the statistical probability of having a single error.

It is important to understand that a UER is a statistical measurement. That is, there is a certain likelihood that a particular near-line-rated drive will never have a UER. Likewise there is the low probability that an enterprise-rated drive will have a UER after 125 TB of reads.

Self-Monitoring, Analysis, and Reporting Technology (SMART)

Self-Monitoring, Analysis, and Reporting Technology (SMART) is an industry-standardized, drive-based failure detection and monitoring system.

SMART is a built-in availability feature of drives used on the VNX. SMART monitoring is performed independently by the drive's on-board disk controller. It uses the drive's onboard sensors and the manufacture's diagnostic algorithms to continuously monitor and evaluate drive health.

Should certain attributes (such as read error rates or the number of uncorrectable sectors) exceed preset manufacturer thresholds, a drive's SMART feature informs the storage system's system software of impending drive failure. Unfortunately, SMART cannot predict all possible failures. Generally, SMART is very efficient at detecting potential mechanical and media failures. However a large component of both mechanical and flash drives is electronic. Electronic failures are not as easily predictable as other failure types.

The storage system uses this information to initiate proactive sparing. See "Global hot sparing" on page 150.

RAID availability differences

All RAID levels except RAID 0 protect against single drive failures. Data protection is reduced when disk failures occur. Most RAID types are considered degraded when one drive in a RAID group fails. Certain RAID levels provide some level of protection against more than one drive failure. For example, a RAID 1/0 group can survive the failure of one drive from each primary or mirror pair. A RAID 6 group can survive two drive failures and still provide access to data. Data protection is restored upon completion of a rebuild operation, usually to a hot spare.

Mirror RAID groups can survive multiple disk failures, so long as they are in separate primary/mirror pairs, but RAID 6 can survive any two disk failures, making it the most highly available RAID type.

Mirror RAID level availability

Supported mirror RAID levels are:

- RAID 1
- RAID 1/0

With mirrors, two copies of data are stored on different drives. As data is written to a drive, the storage system automatically writes a second copy to a separate drive. The write is considered complete only after the data is safely written to both drives. Data can be read from either drive. The use of a full data copy for protection means that usable capacity of any mirror group is half the gross available capacity.

For mirrored data protection, should a drive fail or an error be detected, the lost or corrupted data is reconstructed from the copy on the surviving peer. The simple drive-to-drive copying of data during a rebuild affects access to the drive involved in the copy but does not affect other drives on the same back-end port.

For highest availability, position the primary and the mirror of each disk pair on separate back-end ports. Note that the order in which the drives are added to the RAID group is the order in which they are bound, with the first of the pair being the primary and the second the mirror.

Mirroring is not a substitute for data backup or point-in-time data replication. Erroneous information, user-corrupted data, and data deletions are all copied to both drives of the mirrored pair.

RAID 1

RAID 1 is single-disk mirroring. RAID 1 has good data availability. RAID 1 is the FAST Cache RAID level.

RAID 1/0

RAID 1/0 is the mirroring of disk stripes. RAID 1/0 has excellent data availability. A single disk failure results in no data loss. Multiple disk failures 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.

For example, in a four-disk RAID 1/0 (2+2), consisting of two primary disks 0 and 1, and their mirrors 0′ and 1′. Both 0 and 1 may fail, preserving the data on the mirrors. Likewise, 0′ and 1′ may fail, preserving the data on the primaries. However, neither 0 and 0′ together, or 1 and 1′ together can fail and have the data survive.

Parity RAID level availability

Supported parity RAID levels are:

- RAID 3
- RAID 5
- RAID 6

Parity is a data protection function. Parity RAID levels use an algorithmic method of error detection and reconstruction to maintain the integrity of data. The parity

technique creates information called parity data and the need to store parity data in the RAID group. The parity data is a form of metadata. This decreases the useable data capacity of the group. However maintaining the parity data uses less capacity than is used for mirroring. Parity data requires at most two drive's capacity (for RAID 6) per RAID group compared with mirroring, which is an entire copy of the user data.

All parity RAID levels are striped. In each stripe, one stripe element contains the parity data. Should a drive fail or an error be detected within the media, the parity elements and data elements from the working drives in the RAID group are used to reconstruct the contents of the failed drive.

The rebuild of disks from distributed parity information may adversely affect access to other drives on the same bus. The degree depends on the number of drives in the group and the rebuild rate. Rebuilds are a prioritized operation. At the priority of ASAP, rebuilds run at a high rate, and a large RAID group can use all a back-end bus's bandwidth.

RAID 3

RAID 3 has good data availability. In RAID 3 all parity data is located on a single dedicated parity drive. If one drive of the group fails, no data is lost.

RAID 5

RAID 5 has good data availability. If one drive of the group fails, no data is lost. RAID 5 parity data is distributed throughout the drives of the group. Parity consumes the equivalent of one disk of the group's capacity.

RAID 6

RAID 6 provides the highest data availability on the storage system. RAID 6 is a dual-distributed parity-RAID level. Two parity elements are computed for each stripe. A single or any two drives may fail in the group and result in no data loss. Note this is true double-disk failure protection. RAID 6 parity data is distributed throughout the drives of the group. Parity consumes the equivalent of two disks of the group's capacity.

Unprotected RAID levels (RAID 0)

RAID 0 is a special case. There is no data protection built-into RAID 0. It offers only the performance and capacity advantages of a striped RAID. We do not recommend storing any information of business value on RAID 0 groups.

RAID group operation in degraded mode

A failed drive in a RAID group sets the RAID group into degraded mode. With a RAID group in degraded mode, no user data is lost. However both performance and availability are reduced.

Performance in degraded mode

How RAID performance is affected by a failed drive depends on the RAID type.

Mirror RAID levels

Performance of a mirror RAID-level group is unaffected by a single failed drive. Reads and writes continue to the RAID group, with the surviving peer of the failed drive handling the load.

Parity RAID levels

Generally, performance of a parity RAID-level group decreases with a failed drive.

The failure of a drive in a parity RAID group causes a fraction of the RAID group's data, both parity metadata, and user data to become unavailable.

For example, if the RAID group of the failed drive is a 5-drive (4+1), 20 percent of the group's content, both metadata and user data becomes unavailable.

Write performance is unaffected by the missing drive. The data is written to the group without parity being calculated in a RAID Level 0 fashion. This actually results in better write performance for the group than under normal operation.

Read performance decreases on average. The performance decrease depends on whether the individual read is to a location on the failed drive. Reads to user data that were located on the failed drive involve regenerating the data through a parity calculation. This is resource intensive, with all the surviving drives in the group needing to be read, and storage processor CPU resources dedicated to the regenerating parity calculation. This results in an increased read host response time. If the failed drive contains the frequently read metadata for a volume, typically located on the first drive of the RAID group, performance is particularly adversely affected. Reads of user data on locations on surviving drives in the group do not affect overall storage system performance. They are simply read from their drive and transmitted.

The fewer the number of drives in the RAID group, the more likely a read request to the failed drive will be, and the lower the group's performance.

Note that if the failed drive is the dedicated parity drive of a RAID Level-3 group, performance is unaffected. Also that a single drive failure in a RAID Level 6 group has the same effect on performance as a single drive failure in a RAID Level 5 group. A double drive failure has an increased adverse effect on performance. The double parity calculation is more CPU intensive.

Availability in degraded mode

Availability is decreased in degraded mode. All RAID levels (except RAID 0) can survive a single drive failure without a resulting loss of data. Some RAID groups can survive two drive failures. However, the potential of data loss increases as the ability of the RAID group to provide data redundancy is removed.

For example, if the RAID group of the failed drive is a 5-drive (4+1) any additional drive failure within the group is unrecoverable and would result in data loss.

It is very important to repair failures as quickly as possible. Quick repairs decrease the potential for data loss. The probability of drive failures is complex. However, the failure of one drive in a storage system increases the likelihood of another drive failure. Conversely, the more time that passes since the last failure means the time

until the next failure increases. This relationship has an important effect on the rebuild process.

RAID group rebuilds

The reconstruction of a RAID-group drive's data in the event of a failure is called a *rebuild.* Sometimes this is referred to as *data recovery*. It takes time to recover from a drive error or failure. While a RAID group is rebuilding it is vulnerable to data loss. In addition, there may be some performance loss.

During a rebuild, the data in a RAID group's LUNs is recovered (see "LUNs" on page 78). A rebuild replaces the failed drive of a LUN's RAID group with an operational drive. With drives in parity RAID groups, data from the failing drive is rebuilt from the group's parity onto the replacing drive. With drives in mirror RAID groups, data is copied from the healthy peer of the failed drive onto the replacing drive.

Rebuild time is an important availability consideration. The time it takes to rebuild is affected by several things. The most influential factors are:

- Drive capacity in use by LUNs
- Drive type (SAS, NL-SAS, flash) and speed
- RAID type and size of the RAID group (parity groups only)
- Workload
- Priority

Large drives with a lot of their capacity in use take longer to rebuild than smaller drives; there is more data to rebuild.

Faster, higher performance drives such as flash drives rebuild more quickly than slower drives such as NL-SAS drives. Higher rpm drives rebuild more quickly than lower rpm drives of the same type.

The RAID level chosen can also extend or shorten the process. Parity RAID types must read parity information from all drives in the group and compute the parity to rebuild; the larger the number of drives in the RAID group, the more data that must be read. RAID 5 requires a single parity calculation per stripe, RAID 6 needs two. The additional parity calculation of requires more time and resources.

The background workload, and its requirement for storage system resources, can shorten or extend the process. If the RAID group being rebuilt is being read and written from, while at the same time rebuilding, the rebuild is extended.

Finally, we need to consider *priority*. Rebuilds can take place at four possible priorities: As Soon As Possible (ASAP), High, Medium, and Low. ASAP executes a rebuild very quickly, but uses a lot of storage system resources. This reprioritization may adversely affect workload performance. High, Medium, and Low priorities are economical in their usage of resources. High priority, naturally, completes in a shorter interval than Low. The default rebuild priority is High.

Rebuild time can become a major consideration when provisioning. During the rebuilding process, system performance is usually degraded. In addition, the data is vulnerable to additional drive failures during rebuild. It is prudent to make

provisioning choices that help rebuilds complete as quickly as possible while maximizing data protection.

The resources most affected by a rebuild are the back end's buses and the drives. CPU utilization is affected to a lesser extent. An ASAP rebuild noticeably slows host response times at the drive level, and in a parity rebuild, enough data may be put on the bus to reduce host bandwidth on that bus *even to RAID groups not rebuilding.*

The effect of a rebuild on the back-end ports can be minimized by using more than one port to support a parity RAID group. Spreading a parity RAID group's drives across additional ports likewise spreads the rebuild's I/O across them. This can result in faster rebuild times, making shorter the effect of the rebuild on the performance of application-generated workloads.

LUN availability

FLARE uses a process called Verify to check the integrity of a storage system's LUNs. The Verify process checks and repairs when possible the entire contents of a LUN location by location by:

- Detecting disk locations with data and parity inconsistencies
- Correcting locations with errors using redundant data before the LUN enters a degraded state and data cannot be reconstructed
- Remapping the location of corrected data
- Reporting uncorrectable disk locations

Location errors

Errors found in the LUN are either correctable by Verify or uncorrectable. Errors in a location are corrected from a RAID group's redundant data.

Any uncorrectable errors found by the Verify process are an indication of location(s) on the LUN that are unreadable by a host system. An example of an uncorrectable error location would be when a LUN's RAID group is operating in degraded mode with a single failed drive, and during that degraded period, one of the remaining drives encounters a media error. (Note that for RAID Level 6 this would be a correctable error.) The storage system's Verify process cannot determine if an uncorrectable error location on a LUN is in use by the host or not. However, an uncorrectable error notification should be treated as a serious event that may involve data loss.

LUN verification

There are two types of Verify performed on a LUN. The two LUN verification processes are:

- SNIFF (SNiiFFER): Executes continuously and cyclically on all storage system LUNs at a very low rate as a background process. This is simply a media check; it asks the drive to read a series of blocks. No consistency checks of parity or data are done.

- Background Verify (BV): Runs only when triggered by a failure or when requested by the user and executes at a higher rate. BV does a data-consistency check.

SNiiFFER

SNiiFFER is not intended to verify the state of LUNs. Instead, it provides a background media checking mechanism that operates without any significant effect on host I/O performance. Its benefit is that drive errors can be proactively detected before a host requests the information.

When a LUN is bound, SNIFF is enabled at a very low priority. SNIFF does not affect back-end bus bandwidth utilization. It requests that a 512 KB region on each disk in the LUN is read, about every second. Whenever a LUN is idle, SNIFF increases its verification rate, which reduces the overall time required to complete a full pass. The number of drives in the LUN does not matter. The SNIFF I/Os are performed in parallel on every drive in the LUN. Because it is executing at a very low rate, depending on the capacity of the LUN, it may take several days to completely verify a single LUN.

Background Verify (BV)

BV is used to immediately check the consistency of a LUN when there is a concern over the validity or integrity of its data.

A BV is an operation that can be completed in hours. The duration of a BV depends on such factors as priority, host I/O loading, LUN size, RAID group type, and drive type.

BV is a prioritized operation. The priorities are Low, Medium, High, and ASAP. The duration of BVs can be decreased by increasing its priority to either High or ASAP. Note that this consumes more storage system I/O resources, primarily disk utilization and back-end bus bandwidth. This may affect host performance. The default setting for BV is Medium.

A BV can be scheduled either automatically or manually. A BV automatically starts after a trespass from a faulted SP to its peer SP. BV can be started manually through Unisphere when there are concerns about the validity of a LUN and a more current LUN status than is provided by the SNiiFFER process is needed.

Virtual Pool availability

Generally, availability considerations applying to pools are the same that apply 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 Virtual Pools with either flash or SAS drives. A RAID 6 level of protection is prudent when NL-SAS are used in very large, homogeneous pools

RAID-level data protection

All the LUNs bound within a Virtual Provisioning pool 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:

- 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 that this is true of 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 the priority: availability, performance, or capacity utilization. If the priority is availability, RAID 6 should be provisioned. 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, pools with a very large number of RAID groups have lower availability than pools with a moderate to modest 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 more important for the overall operation of the storage system than with traditional provisioning.

Always have hot spares of the appropriate type available. The action of proactive hot sparing reduces the adverse performance effect a rebuild would have on back-end

performance. In addition, always replace failed drives as quickly as possible to maintain the number of available hot spares and decrease the interval a RAID group is in degraded mode.

Reliability

Reliability is the ability of the storage system to run for long periods, and under operational stress, without suffering hardware failure or software faults. The storage systems architecture contains many features designed to ensure a very high level of reliability. High-quality components, assembly, design verification, and quality testing back up this design. The storage systems' systemic reliability is rated at *five 9s* (99.999 percent) uptime based on hundreds of thousands of hours of field experience. The exact reliability measurements depend on the model and components used.

The storage system is designed and built to protect data, with full data path protection, throughout the storage system. This data path protection prevents unreported *data miscompare* errors. The UltraPoint hard drive array enclosures (DAEs and DPEs) include a point-to-point design within the drive enclosures. The block operating environment monitors low-level diagnostics at the drive level that include fault detection, isolation, and error correction. RAID protection is supplemented with global and proactive hot sparing, sector-based checksum of all user data, and automatic background verification of the disk media.

Chapter 13 VNX Availability

This chapter presents these topics:

Front end

All VNX series models come standard with multiple iSCSI and Fibre Channel front-end ports on each storage processor. Connections to iSCSI and Fibre Channel hosts are supported at the same time.

High availability requires multiple connections. In a highly available configuration, I/O connections from a host should be made to more than one port on a storage processor. In addition, an I/O connection between the host and the peer storage processor should be set up.

Configurations allowing the host to connect to more than one of the storage system's front-end ports are a technique referred to as *multipathing*. These additional ports may be used to provide primary access to one LUN and alternate access to another. The existence of alternate paths enables a front-end port to front-end port redirection on the storage system in the event of a path or port failure. *Failover* is the use of an alternate path in the event of a failure. *Failback* is the return to the original path, following the correction of the path's failure.

Typically a failover is the result of a networking error. However, it is possible for failover to occur as a result of a storage system component failure. Failover is initiated on the storage system by the host-based PowerPath, or other host-based multipathing applications (see "Multipathing" on page 131 for details). In failover, the operational port begins servicing host I/O requests.

Along with redundant network configuration, when provisioning it is prudent to create connections to at least two front-end ports on each storage processor. In addition, at least one or more further connections should be made to a front-end port of the peer SP on the storage system. This configuration allows port-to-port failover on the owning SP, in addition to preparing for a SP failover.

Storage processor

The storage system has two highly-reliable storage processors. When properly configured, the storage processors provide a redundant operating environment for storage access.

Active/passive ownership model

For workload balancing, the storage system's LUNs are distributed between the storage processors. This results in storage processors efficiently working in parallel to service the workload.

LUNs are assigned to only one storage processor at a time. This is considered *active ownership*. Either storage processor can access a LUN on the storage system's back end. However, I/O is not directed to the non-owning storage processor. The existing, but inactive paths to unowned LUNs are *passively owned* by the peer storage processor. Note that this requires the storage processors to mirror each other's write cache (see "Memory" on page 52).

If there is a network failure of all I/O to one storage processor, or the unlikely failure of a storage processor, LUNs are *trespassed* to their peer storage processor. Trespass changes the ownership a LUN from passive to active. Trespass is initiated

transparently by PowerPath or other host-based multipathing applications (see "Multipathing" on page 131). The surviving storage processor assumes ownership and begins servicing host I/O requests as soon as cache page associations with their new LUNs are completed.

Note that for high availability, a single storage processor needs enough CPU resources to handle the workload of the entire storage system.

Back end

The back end is composed of multiple disk enclosures with redundant ports and storage configured to ensure data security.

System drives and write cache availability

The first four hard drives (0 through 3) in the DPE or the DAE O/S have some capacity reserved for the storage system's operation. These drives are referred to as the system drives. They contain the vault. The vault is capacity reserved for the write cache (so it can be quickly written to this region during certain failures), the storage system's operating environment's system files, the Persistent Storage Manager (PSM), and the operating environment's configuration database.

The storage system boots off of all the system drives. There is actually considerably more private data on the system drives than just the vault, which is where write cache is destaged to in case of a power failure or certain component failures. There are several private LUNs on these drives including LUNs for:

- Operating system for both SPs in RAID 1
- Persistent Storage Manager (PSM)
- Operating environment database in RAID 1 triple mirror
- The vault in RAID 5

This organization of the drives provides protection from two drive failures.

System drives can be used just like any other drives on the system. However, system drives have less usable capacity than data drives, due to the reserved capacity. In addition, their usage by the storage system can affect the response time for application data stored there. See the *Best Practices* papers for system drive provisioning performance planning advice. Note that system drives cannot be used as part of a Virtual pool. Pool provisioning requires the entire capacity of a drive to be made available.

The O/S, PSM, operating environment database, or the vault is not rebuilt to a hot spare in the case of a single drive failure in any of these private LUNs. Only user LUNs created in the systems drive's user space are rebuilt to a hot spare; the private LUNs are rebuilt when the faulted drive is replaced. Note that the read/write cache is disabled in this particular failure. It remains disabled until the failed system drive is replaced. The disabled read/write cache adversely affects overall storage system performance.

For example, if drive 0 on Bus 0 Enclosure 0 fails and there are no user LUNs, no hot spares are used in a swap. If there is a user LUN that is bound across that drive, only

the user LUN data is rebuilt to the hot spare. When the faulted drive is replaced, all data is copied/rebuilt to the drive 0 (OE DB from mirror, vault from parity, PSM from mirror, and user LUN from hot spare).

In addition, see "Standby power" on page 39 for the power failure protection applied to the DAE O/S.

Back-end ports

As explained in "Back end" on page 149, the storage system's back-end ports redundantly connect both SPs to the storage system's drives housed in the DPE or DAEs. The SAS back end provides two paths to the data on the drives.

There are two LCCs per DAE. Each SP attaches to a DAE port through an LCC by its SAS port(s). Should a DAE's LCC, fail an SP loses communication with all the drives in the DAE. In addition, communication is lost with DAEs further down the chain of DAEs.

The recovery from an LCC failure may result in the use of either a Background Verify (BV) or rebuild and possibly a host trespass. Note that recovery from an LCC failure can adversely affect host I/O performance. Details on LCC availability can be found in the *Best Practices* papers.

Global hot sparing

The operating environment constantly monitors the state of its drives. *Hot spares* are drives used to replace failing drives. Proactive sparing is the automatic activation of a hot spare when a drive indicates its pending failure through its SMART reporting feature.

Hot spares replace a failed drive in any redundant RAID group with bound LUNs. Hot spares do not replace unbound drives or drives in RAID groups of RAID Level 0 or individual disks.

The maximum number of hot spares configurable per storage system is model dependent. Entry-level storage systems may have fewer hot spares allocated than higher models.

Hot sparing occurs when a drive has failed. A spare is selected algorithmically from a pool of previously-designated hot spares. There are some restrictions on which types of drives may be used as hot spares.

In proactive sparing, a suspected failing drive is copied to a spare before it fails. The suspect drive continues servicing I/Os while this process goes forward. When the copying completes, the suspect drive is failed (powered down) and the hot spare takes its place in the RAID group. If the drive fails before the process completes, the remaining missing data is rebuilt to the hot spare by the process described in this section.

Copying data to the spare proactively is much less resource intensive and usually faster than rebuilding data from parity and the surviving drives. The disk-to-disk copy puts less load on the back end and the storage processor than a parity rebuild. This advantage extends to mirror-type RAID levels, where the copy is performed while the RAID group is not in a degraded mode. This is an availability advantage.

For example, if an uncorrectable disk error (UER) is encountered during the copy, the missing data can be reconstructed. In a rebuild-type data recovery, where the failed drive is no longer available, a UER results in loss of data. (The exception is RAID 6 rebuilds, which are immune to double drive failures.)

When a failed drive is manually replaced, a process called *equalization* restores the data from the hot spare to the RAID group's replacement drive. Equalization is a disk-to-disk copy operation and affects the backend port and affected drive the same as a RAID 1 rebuild or proactive copy. After equalization, the hot spare is returned to the global pool for reuse as a replacement.

Note that a RAID group's operation with a hot spare is not considered a normal operation. Normal operation is restored when the failed drive is replaced, its replacement is equalized, and it begins to handle I/O. No rebuild occurs if there are no hot spares configured of the appropriate type and size when a drive fails. The RAID group remains in a degraded state until the failed drive is replaced, and then the failed drive's RAID group rebuilds from parity or its mirror drive, depending on the RAID level.

Proactive hot sparing reduces the risk of a second disk failure during rebuild. A proactively spared RAID group always has its full number of storage devices. Note that it is still considered degraded. In addition, equalization, and not a rebuild, is needed to restore the RAID group's status. Equalization does not use the resources that a rebuild does.

Global hot sparing is not a requirement for normal operation enforced by the operating environment. However, it is a prudent choice if you wish to increase overall availability.

Rebuild logging

When a drive appears to be timing out, rebuild logging allows it additional time to recover from whatever it is doing, without delaying host I/O execution for an extended period of time. This also avoids performing a full rebuild operation to the drive. Rebuild logging is also called *drive probation*.

The situations in which rebuild logging get invoked are specific. If a drive actually faults — for example it no longer accepts commands, reports a fatal condition (such as "Hardware Error"), or is bypassed from the bus — it is immediately retired and a hot spare (if available) is employed. Rebuild logging gets invoked when commands already issued to the drive fail to complete in a reasonable amount of time — that is, they appear to be timing out, but the drive is still physically present, and has not faulted in any of the conditions mentioned.

Software and firmware update

The operating environment can be updated to incorporate new features, and to correct discovered errors.

The disk array operating system can be upgraded online using the nondisruptive update (NDU) process. The DAE LCC firmware (FRUMON) is upgraded online using the NDU process (contained within operating environment upgrades). Disk firmware

is not usually recommended nor required for upgrade (unless a field change order or ETA is implemented) and requires scheduled downtime for it to be upgraded

Chapter 14 Conclusion

Having read this document, you should now be familiar with the fundamental concepts and features of the VNX. This paves the way to understanding the details of performance and availability tuning described in *EMC Unified Best Practices for Performance and Availability — Applied Best Practices*. (This paper is also referred to as the *VNX Best Practices* paper.)

A lot of information is presented in this paper. However, there are some very simple points that you must take away to get the most out of your VNX and to understand the tuning recommendations in *VNX Best Practices*. They are:

- A storage infrastructure is a balance of cost, capacity, performance, and availability.
- There are many ways to meet the capacity requirements of your workload.
- Performance enables behavior to be evaluated. To do so you need to:
  - Know how to describe your workload.
  - Have anticipated, or already have past and current performance metrics.
- VNX is a computer-based device with resources that need to be managed.
- Availability is about redundancy and data integrity.

Several physical and logical storage technologies exist for protecting your data. Cost, capacity, and ease-of-use typically drive performance and availability decisions.

To get the highest performance, you need to understand a storage system's workload. This includes knowledge of the host applications. You should understand how your applications are generating I/O. The workload may be simple, coming from a single application; it might also be complex, meaning it is generated by more than one application on one or more hosts with different I/O profiles. You need to know the throughput, bandwidth, and required response time of your I/O to determine the needed performance. If you can characterize your workload, tuning is a straightforward process. In addition to characterizing the workload, the information gathered is important for creating the historical baselines for future tuning, and investigations. Note that, if the workload's demands exceed the underlying storage system's performance capabilities, applying performance tuning has little effect.

A VNX is actually two computers with a lot of attached drives. It has CPU, memory (cache), and I/O (SAS back-end ports and storage) resources that need to be managed and provisioned for it to reach optimal performance and highest availability. Be aware of the usable bandwidth of its back-end and front-end ports, its usable CPU cycles, and the throughput, bandwidth, and storage capacity of its drives. A part of these resources is consumed by the workload. Another part is consumed by overall system maintenance, such as backup and storage system management tasks like LUN expansion. In addition, reserves need to be maintained to ensure acceptable performance in degraded modes of operation and unexpected increases in usage.

The VNX's built-in reliability and redundant hardware make it a highly available system. Redundancy extends to data protection. Careful provisioning is required to ensure the highest data protection *and* performance. Remember that configuring the storage system for high availability may mean sacrificing some degree of performance. (Performance and availability are both dependent on the same resources.) Evaluating the allocation of resources needed to meet the immediate demands of the workload, while maintaining reserves for maintenance and to ensure high-availability, is a continuous process.

This document and others mentioned here are found on Powerlink®, EMC's password-protected extranet for customers and partners. We encourage you to read the best practices information for your workload's applications and its host hardware and software. The best practices from the vendors of your network elements should also be of interest. An additional networking information resource is the *EMC Networked Storage Topology Guide* found on Powerlink.

Finally, note that the storage system is an important, but dependent, part of a much larger system. Host and networking-related fundamentals are beyond the scope of this document. However, the storage system's performance and availability are greatly affected by the performance and availability of the networks attaching it to hosts, host applications, and the host's hardware and operating system software. In many cases, improving the performance or availability of the network or the host results in better performance and higher availability than tuning the storage system.

Appendix A Glossary

"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 — When data written from or requested by the host is found in the cache, avoiding a wait for a drive request.

Cache miss — When data requested by the host is not found in the cache, so that 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 — Official name of the Celerra product, referred to both the File 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 to 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 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 (X-Blade) running VNX Operating Environment for 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 a 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 a 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.

Failback — Restoring the original data path after correcting a failure.

Failover — 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 — Of 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.

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.

LACP — Link Aggregation Control Protocol.

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.

MIB — Management Information Base.

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 of 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 of 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 drive’s 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.

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 results 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.

SNMP — Simple Network Management Protocol

Solid state disk — A drive using non-volatile semiconductor memory for data storage.

SP — Storage processor.

SPE — Storage processor enclosure.

Specification — A document that details, 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 axle that 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 a 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 — Distributing sequential chunks of storage across many drives in a RAID group.

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.

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 communication 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 that 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.