Chapter 4
Intelligent Storage System

Business-critical applications require high levels of performance, availability, security, and scalability. A hard disk drive is a core element of storage that governs the performance of any storage system. Some of the older disk array technologies could not overcome performance constraints due to the limitations of a hard disk and its mechanical components. RAID technology made an important contribution to enhancing storage performance and reliability, but hard disk drives even with a RAID implementation could not meet performance requirements of today's applications.

KEY CONCEPTS

- Intelligent Storage System
- Front-End Command Queuing
- Cache Mirroring and Vaulting
- Logical Unit Number
- LUN Masking
- High-end Storage System
- Midrange Storage System

With advancements in technology, a new breed of storage solutions known as an *intelligent storage system* has evolved. The intelligent storage systems detailed in this chapter are the feature-rich RAID arrays that provide highly optimized I/O processing capabilities. These arrays have an operating environment that controls the management, allocation, and utilization of storage resources. These storage systems are configured with large amounts of memory called *cache* and use sophisticated algorithms to meet the I/O requirements of performance-sensitive applications.

4.1 Components of an Intelligent Storage System

An intelligent storage system consists of four key components: *front end, cache, back end,* and *physical disks*. Figure 4-1 illustrates these components and their interconnections. An I/O request received from the host at the front-end port is processed through cache and the back end, to enable storage and retrieval of data from the physical disk. A read request can be serviced directly from cache if the requested data is found in cache.

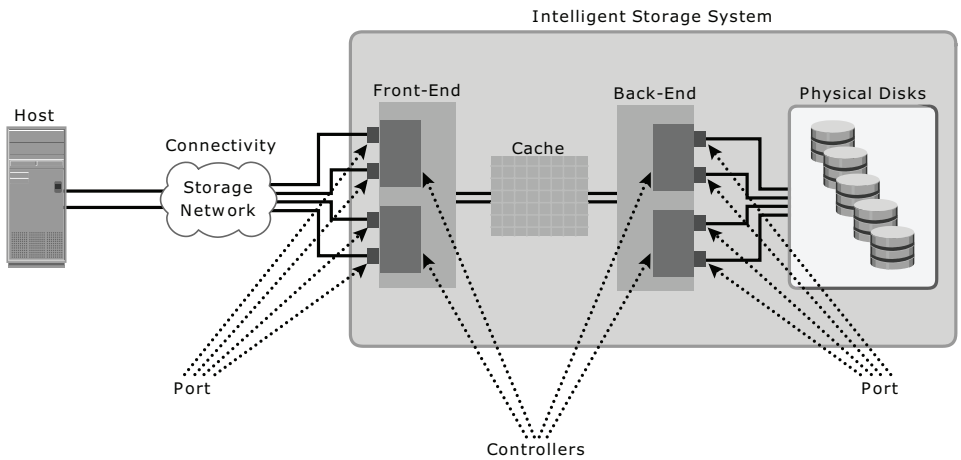

Figure 4-1: Components of an intelligent storage system

4.1.1 Front End

The front end provides the interface between the storage system and the host. It consists of two components: front-end ports and front-end controllers. The *front-end ports* enable hosts to connect to the intelligent storage system. Each front-end port has processing logic that executes the appropriate transport protocol, such as SCSI, Fibre Channel, or iSCSI, for storage connections. Redundant ports are provided on the front end for high availability.

Front-end controllers route data to and from cache via the internal data bus. When cache receives write data, the controller sends an acknowledgment message back to the host. Controllers optimize I/O processing by using command queuing algorithms.

Front-End Command Queuing

Command queuing is a technique implemented on front-end controllers. It determines the execution order of received commands and can reduce unnecessary drive head movements and improve disk performance. When a command is received for execution, the command queuing algorithms assigns

a tag that defines a sequence in which commands should be executed. With command queuing, multiple commands can be executed concurrently based on the organization of data on the disk, regardless of the order in which the commands were received.

The most commonly used command queuing algorithms are as follows:

- **First In First Out** (FIFO)**:** This is the default algorithm where commands are executed in the order in which they are received (Figure 4-2 [a]). There is no reordering of requests for optimization; therefore, it is inefficient in terms of performance.
- **Seek Time Optimization:** Commands are executed based on optimizing read/write head movements, which may result in reordering of commands. Without seek time optimization, the commands are executed in the order they are received. For example, as shown in Figure 4-2(a), the commands are executed in the order A, B, C and D. The radial movement required by the head to execute C immediately after A is less than what would be required to execute B. With seek time optimization, the command execution sequence would be A, C, B and D, as shown in Figure 4-2(b).

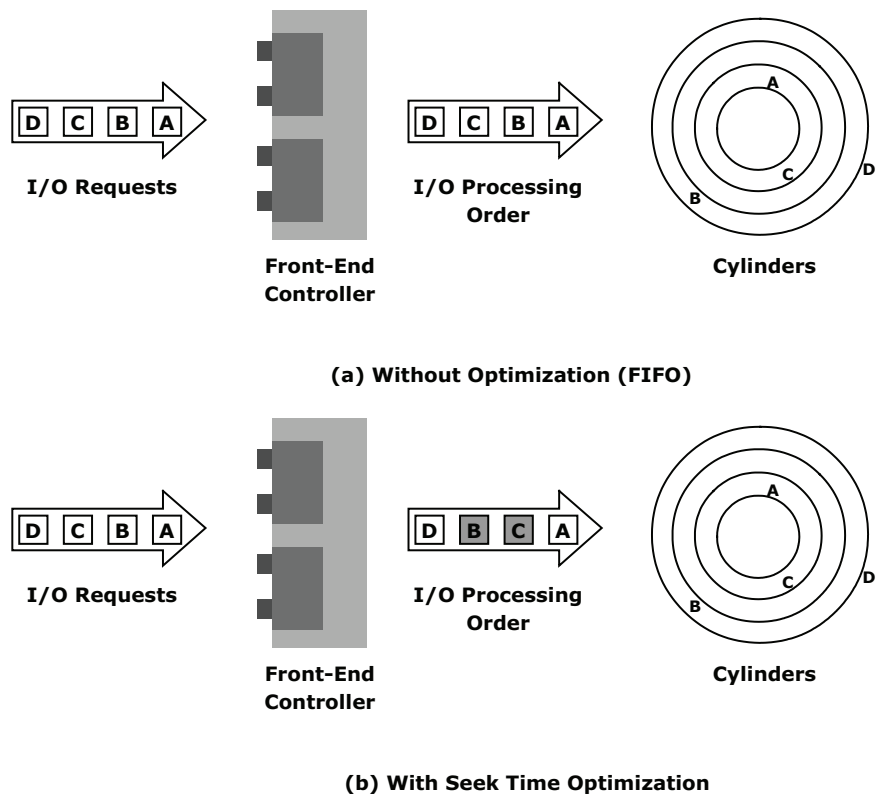

Figure 4-2: Front-end command queuing

- **Access Time Optimization:** Commands are executed based on the combination of seek time optimization and an analysis of rotational latency for optimal performance.

Command queuing can also be implemented on disk controllers and this may further supplement the command queuing implemented on the front-end controllers. Some models of SCSI and Fibre Channel drives have command queuing implemented on their controllers.

4.1.2 Cache

Cache is an important component that enhances the I/O performance in an intelligent storage system. Cache is semiconductor memory where data is placed temporarily to reduce the time required to service I/O requests from the host.

Cache improves storage system performance by isolating hosts from the mechanical delays associated with physical disks, which are the slowest components of an intelligent storage system. Accessing data from a physical disk usually takes a few milliseconds because of seek times and rotational latency. If a disk has to be accessed by the host for every I/O operation, requests are queued, which results in a delayed response. Accessing data from cache takes less than a millisecond. Write data is placed in cache and then written to disk. After the data is securely placed in cache, the host is acknowledged immediately.

Structure of Cache

Cache is organized into pages or slots, which is the smallest unit of cache allocation. The size of a cache page is configured according to the application I/O size. Cache consists of the *data store* and *tag RAM.* The data store holds the data while tag RAM tracks the location of the data in the data store (see Figure 4-3) and in disk.

Entries in tag RAM indicate where data is found in cache and where the data belongs on the disk. Tag RAM includes a *dirty bit* flag, which indicates whether the data in cache has been committed to the disk or not. It also contains time-based information, such as the time of last access, which is used to identify cached information that has not been accessed for a long period and may be freed up.

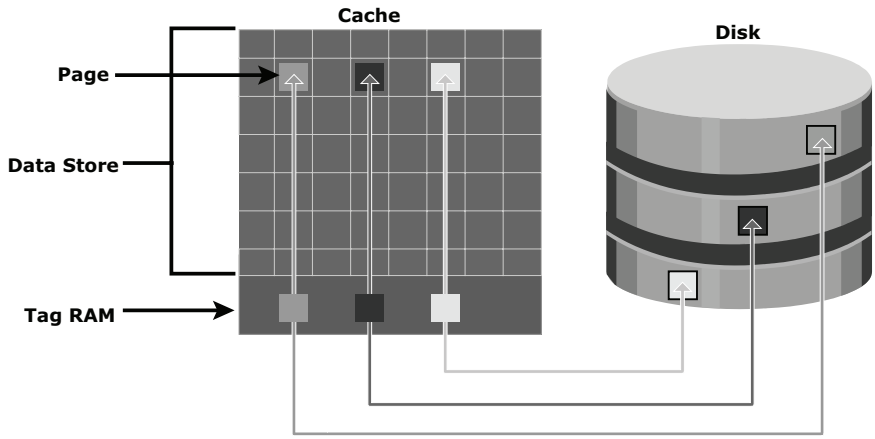

Figure 4-3: Structure of cache

Read Operation with Cache

When a host issues a read request, the front-end controller accesses the tag RAM to determine whether the required data is available in cache. If the requested data is found in the cache, it is called a *read cache hit* or *read hit* and data is sent directly to the host, without any disk operation (see Figure 4-4[a]). This provides a fast response time to the host (about a millisecond). If the requested data is not found in cache, it is called a *cache miss* and the data must be read from the disk (see Figure 4-4[b]). The back-end controller accesses the appropriate disk and retrieves the requested data. Data is then placed in cache and is finally sent to the host through the front-end controller. Cache misses increase I/O response time.

A *pre-fetch*, or *read-ahead*, algorithm is used when read requests are sequential. In a sequential read request, a contiguous set of associated blocks is retrieved. Several other blocks that have not yet been requested by the host can be read from the disk and placed into cache in advance. When the host subsequently requests these blocks, the read operations will be read hits. This process significantly improves the response time experienced by the host. The intelligent storage system offers fixed and variable pre-fetch sizes. In *fixed pre-fetch*, the intelligent storage system pre-fetches a fixed amount of data. It is most suitable when I/O sizes are uniform. In *variable pre-fetch*, the storage system pre-fetches an amount of data in multiples of the size of the host request. Maximum pre-fetch limits the number of data blocks that can be pre-fetched to prevent the disks from being rendered busy with pre-fetch at the expense of other I/O.

Read performance is measured in terms of the *read hit ratio,* or the *hit rate,* usually expressed as a percentage. This ratio is the number of read hits with respect to the total number of read requests. A higher read hit ratio improves the read performance.

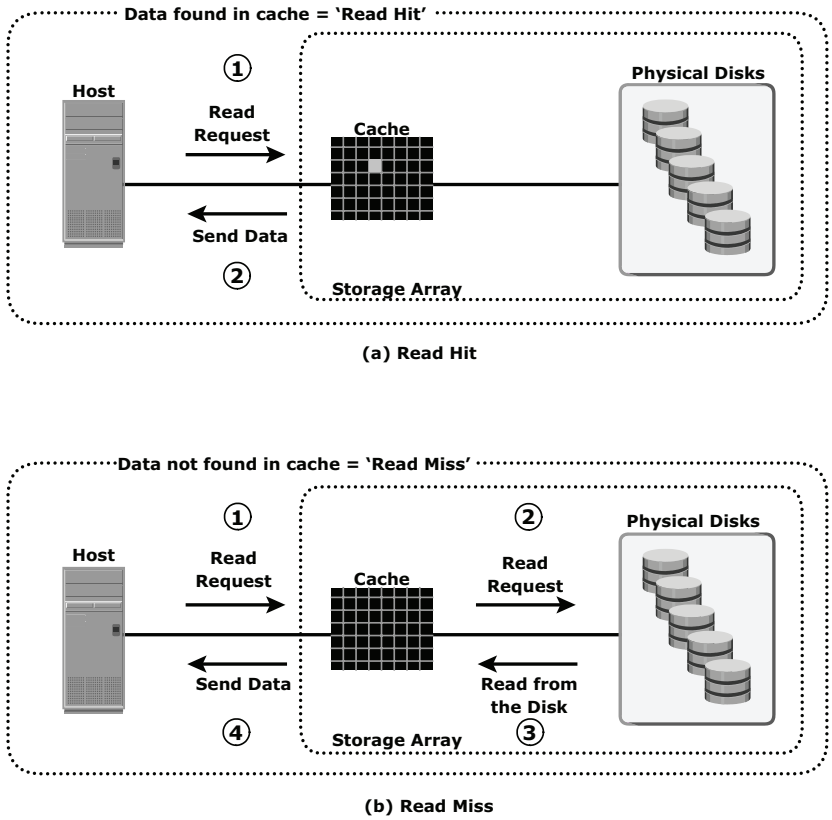

Figure 4-4: Read hit and read miss

Write Operation with Cache

Write operations with cache provide performance advantages over writing directly to disks. When an I/O is written to cache and acknowledged, it is completed in far less time (from the host's perspective) than it would take to write directly to disk. Sequential writes also offer opportunities for optimization because many smaller writes can be coalesced for larger transfers to disk drives with the use of cache.

A write operation with cache is implemented in the following ways:

- **Write-back cache:** Data is placed in cache and an acknowledgment is sent to the host immediately. Later, data from several writes are committed

(de-staged) to the disk. Write response times are much faster, as the write operations are isolated from the mechanical delays of the disk. However, uncommitted data is at risk of loss in the event of cache failures.

- **Write-through cache:** Data is placed in the cache and immediately written to the disk, and an acknowledgment is sent to the host. Because data is committed to disk as it arrives, the risks of data loss are low but write response time is longer because of the disk operations.

Cache can be bypassed under certain conditions, such as very large size write I/O. In this implementation, if the size of an I/O request exceeds the predefined size, called *write aside size*, writes are sent to the disk directly to reduce the impact of large writes consuming a large cache area. This is particularly useful in an environment where cache resources are constrained and must be made available for small random I/Os.

Cache Implementation

Cache can be implemented as either dedicated cache or global cache. With dedicated cache, separate sets of memory locations are reserved for reads and writes. In global cache, both reads and writes can use any of the available memory addresses. Cache management is more efficient in a global cache implementation, as only one global set of addresses has to be managed.

Global cache may allow users to specify the percentages of cache available for reads and writes in cache management. Typically, the read cache is small, but it should be increased if the application being used is read intensive. In other global cache implementations, the ratio of cache available for reads versus writes is dynamically adjusted based on the workloads.

Cache Management

Cache is a finite and expensive resource that needs proper management. Even though intelligent storage systems can be configured with large amounts of cache, when all cache pages are filled, some pages have to be freed up to accommodate new data and avoid performance degradation. Various cache management algorithms are implemented in intelligent storage systems to proactively maintain a set of free pages and a list of pages that can be potentially freed up whenever required:

- **Least Recently Used (LRU):** An algorithm that continuously monitors data access in cache and identifies the cache pages that have not been accessed for a long time. LRU either frees up these pages or marks them for reuse. This algorithm is based on the assumption that data which hasn't been accessed for a while will not be requested by the host. However, if a page contains write data that has not yet been committed to disk, data will first be written to disk before the page is reused.

- **Most Recently Used (MRU):** An algorithm that is the converse of LRU. In MRU, the pages that have been accessed most recently are freed up or marked for reuse. This algorithm is based on the assumption that recently accessed data may not be required for a while.

As cache fills, the storage system must take action to flush dirty pages (data written into the cahce but not yet written to the disk) in order to manage its availability. Flushing is the process of committing data from cache to the disk. On the basis of the I/O access rate and pattern, high and low levels called *watermarks* are set in cache to manage the flushing process. *High watermark (HWM)* is the cache utilization level at which the storage system starts high-speed flushing of cache data. *Low watermark (LWM)* is the point at which the storage system stops the high-speed or forced flushing and returns to idle flush behavior. The cache utilization level, as shown in Figure 4-5, drives the mode of flushing to be used:

- **Idle flushing:** Occurs continuously, at a modest rate, when the cache utilization level is between the high and low watermark.
- **High watermark flushing:** Activated when cache utilization hits the high watermark. The storage system dedicates some additional resources to flushing. This type of flushing has minimal impact on host I/O processing.
- **Forced flushing:** Occurs in the event of a large I/O burst when cache reaches 100 percent of its capacity, which significantly affects the I/O response time. In forced flushing, dirty pages are forcibly flushed to disk.

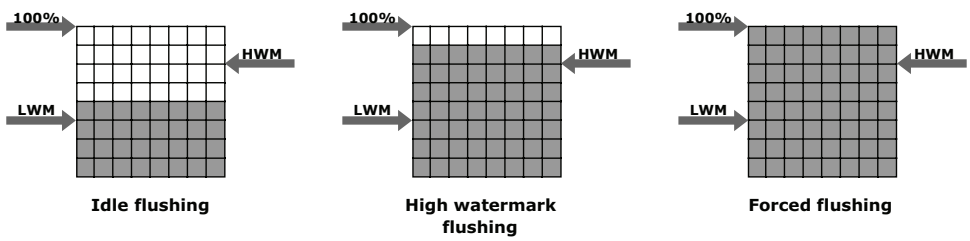

Figure 4-5: Types of flushing

Cache Data Protection

Cache is volatile memory, so a power failure or any kind of cache failure will cause the loss of data not yet committed to the disk. This risk of losing uncommitted data held in cache can be mitigated using *cache mirroring* and *cache vaulting*:

- **Cache mirroring:** Each write to cache is held in two different memory locations on two independent memory cards. In the event of a cache failure, the write data will still be safe in the mirrored location and can be committed to the disk. Reads are staged from the disk to the cache;

therefore, in the event of a cache failure, the data can still be accessed from the disk. As only writes are mirrored, this method results in better utilization of the available cache.

In cache mirroring approaches, the problem of maintaining *cache coherency* is introduced. Cache coherency means that data in two different cache locations must be identical at all times. It is the responsibility of the array operating environment to ensure coherency.

- **Cache vaulting:** Cache is exposed to the risk of uncommitted data loss due to power failure. This problem can be addressed in various ways: powering the memory with a battery until AC power is restored or using battery power to write the cache content to the disk. In the event of extended power failure, using batteries is not a viable option because in intelligent storage systems, large amounts of data may need to be committed to numerous disks and batteries may not provide power for sufficient time to write each piece of data to its intended disk. Therefore, storage vendors use a set of physical disks to dump the contents of cache during power failure. This is called cache vaulting and the disks are called vault drives. When power is restored, data from these disks is written back to write cache and then written to the intended disks.

4.1.3 Back End

The *back end* provides an interface between cache and the physical disks. It consists of two components: back-end ports and back-end controllers. The back end controls data transfers between cache and the physical disks. From cache, data is sent to the back end and then routed to the destination disk. Physical disks are connected to ports on the back end. The back end controller communicates with the disks when performing reads and writes and also provides additional, but limited, temporary data storage. The algorithms implemented on back-end controllers provide error detection and correction, along with RAID functionality.

For high data protection and availability, storage systems are configured with dual controllers with multiple ports. Such configurations provide an alternate path to physical disks in the event of a controller or port failure. This reliability is further enhanced if the disks are also dual-ported. In that case, each disk port can connect to a separate controller. Multiple controllers also facilitate load balancing.

4.1.4 Physical Disk

A physical disk stores data persistently. Disks are connected to the back-end with either SCSI or a Fibre Channel interface (discussed in subsequent chapters). An intelligent storage system enables the use of a mixture of SCSI or Fibre Channel drives and IDE/ATA drives.

SOLID-STATE DRIVES

Flash-based solid-state drives (SSDs) are a recent innovation for delivering ultra-high performance for mission-critical applications. Solid-state Flash drives utilize Flash memory to store and retrieve data. Unlike FC or SATA drives, Flash drives have no moving parts, and leverage semiconductor-based block storage devices, resulting in minimized response time and less power requirements to run. Flash drives are constructed with nonvolatile semiconductor memory to support persistent storage and they use either single-level cell (SLC) or multi-level cell (MLC) to store bits on each memory cell. SLC stores one bit per cell and is used in high-performance memory cards. MLC memory cards store more bits per cell and provide slower transfer speeds. The advantage of MLC over SLC memory cards is the lower manufacturing cost.

Flash drives that use SLC technology combined with sophisticated controllers can behave like virtual HDDs through a traditional storage interface (such as Fibre Channel) to achieve ultra-fast read/write performance, high reliability, and data integrity. Flash drives have been tested and qualified to withstand the intense workloads of high-end enterprise storage applications.

Flash storage technology is ideally suited to support applications that need to process massive amounts of information very quickly, such as currency exchange and electronic trading systems, real time data feed processing, mainframe transaction processing, and many others. Storage systems with enterprise-class Flash drives can deliver single-millisecond application response times, up to 10 times faster than those with traditional 15K RPM Fibre Channel disk drives.

In a storage array, Flash drives can store a terabyte of data using 38 percent less energy than traditional mechanical disk drives. It would take 30 15K RPM Fibre Channel disk drives to deliver the same performance as a single Flash drive, which translates into a 98 percent reduction in power consumption in a transaction-per-second comparison.

Logical Unit Number

Physical drives or groups of RAID protected drives can be logically split into volumes known as logical volumes, commonly referred to as *Logical Unit Numbers* (LUNs). The use of LUNs improves disk utilization. For example, without the use of LUNs, a host requiring only 200 GB could be allocated an entire 1TB physical disk. Using LUNs, only the required 200 GB would be allocated to the host, allowing the remaining 800 GB to be allocated to other hosts.

In the case of RAID protected drives, these logical units are slices of RAID sets and are spread across all the physical disks belonging to that set. The logical

units can also be seen as a logical partition of a RAID set that is presented to a host as a physical disk. For example, Figure 4-6 shows a RAID set consisting of five disks that have been sliced, or partitioned, into several LUNs. LUNs 0 and 1 are shown in the figure.

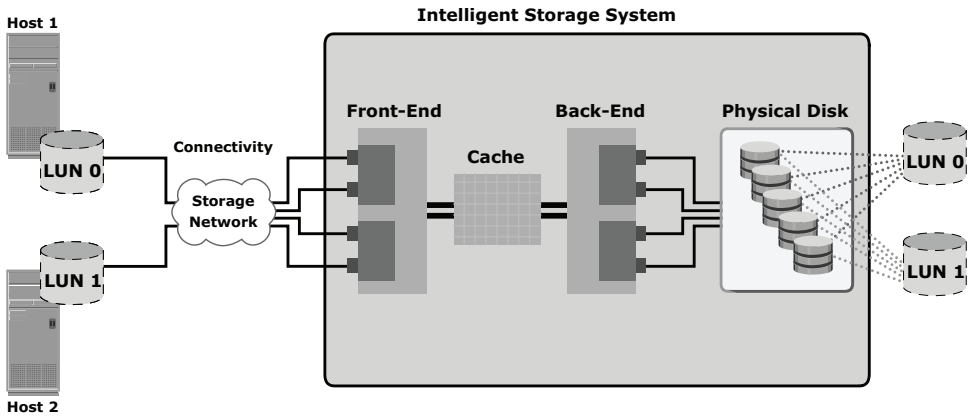

Figure 4-6: Logical unit number

Note how a portion of each LUN resides on each physical disk in the RAID set. LUNs 0 and 1 are presented to hosts 1 and 2, respectively, as physical volumes for storing and retrieving data. Usable capacity of the physical volumes is determined by the RAID type of the RAID set.

The capacity of a LUN can be expanded by aggregating other LUNs with it. The result of this aggregation is a larger capacity LUN, known as a *meta-LUN*. The mapping of LUNs to their physical location on the drives is managed by the operating environment of an intelligent storage system.

LUN Masking

LUN masking is a process that provides data access control by defining which LUNs a host can access. LUN masking function is typically implemented at the front end controller. This ensures that volume access by servers is controlled appropriately, preventing unauthorized or accidental use in a distributed environment.

For example, consider a storage array with two LUNs that store data of the sales and finance departments. Without LUN masking, both departments can easily see and modify each other's data, posing a high risk to data integrity and security. With LUN masking, LUNs are accessible only to the designated hosts.

4.2 Intelligent Storage Array

Intelligent storage systems generally fall into one of the following two categories:

- High-end storage systems
- Midrange storage systems

Traditionally, high-end storage systems have been implemented with *active-active arrays,* whereas midrange *storage systems* used typically in small- and medium-sized enterprises have been implemented with *active-passive arrays.* Active-passive arrays provide optimal storage solutions at lower costs. Enterprises make use of this cost advantage and implement active-passive arrays to meet specific application requirements such as performance, availability, and scalability. The distinctions between these two implementations are becoming increasingly insignificant.

4.2.1 High-end Storage Systems

High-end storage systems, referred to as *active-active arrays,* are generally aimed at large enterprises for centralizing corporate data. These arrays are designed with a large number of controllers and cache memory. An active-active array implies that the host can perform I/Os to its LUNs across any of the available paths (see Figure 4-7).

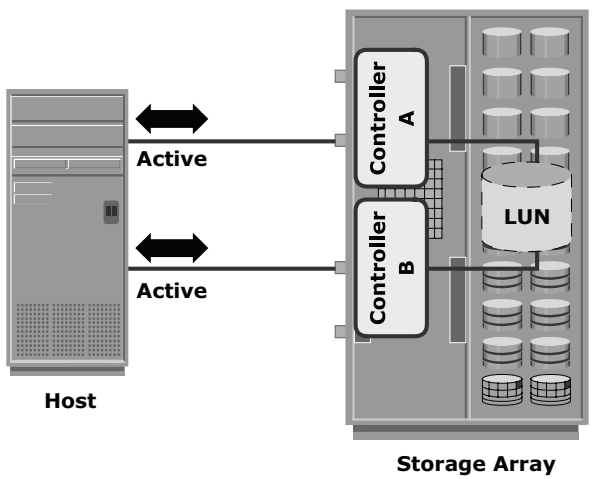

Figure 4-7: Active-active configuration

To address the enterprise storage needs, these arrays provide the following capabilities:

- Large storage capacity
- Large amounts of cache to service host I/Os optimally
- Fault tolerance architecture to improve data availability
- Connectivity to mainframe computers and open systems hosts
- Availability of multiple front-end ports and interface protocols to serve a large number of hosts
- Availability of multiple back-end Fibre Channel or SCSI RAID controllers to manage disk processing
- Scalability to support increased connectivity, performance, and storage capacity requirements
- Ability to handle large amounts of concurrent I/Os from a number of servers and applications
- Support for array-based local and remote replication

In addition to these features, high-end arrays possess some unique features and functionals that are required for mission-critical applications in large enterprises.

4.2.2 Midrange Storage System

Midrange storage systems are also referred to as *active-passive arrays* and they are best suited for small- and medium-sized enterprises. In an active-passive array, a host can perform I/Os to a LUN only through the paths to the owning controller of that LUN. These paths are called *active paths*. The other paths are passive with respect to this LUN. As shown in Figure 4-8, the host can perform reads or writes to the LUN only through the path to controller A, as controller A is the owner of that LUN. The path to controller B remains passive and no I/O activity is performed through this path.

Midrange storage systems are typically designed with two controllers, each of which contains host interfaces, cache, RAID controllers, and disk drive interfaces.

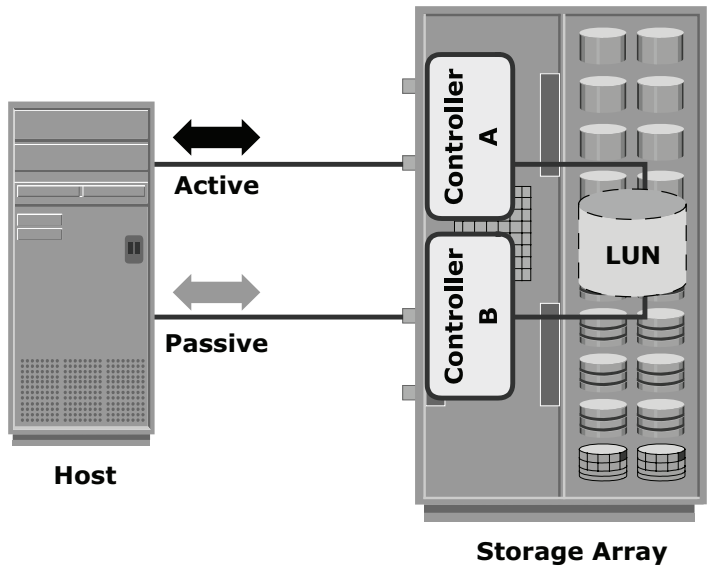

Figure 4-8: Active-passive configuration

Midrange arrays are designed to meet the requirements of small and medium enterprises; therefore, they host less storage capacity and global cache than active-active arrays. There are also fewer front-end ports for connection to servers. However, they ensure high redundancy and high performance for applications with predictable workloads. They also support array-based local and remote replication.

4.3 Concepts in Practice: EMC CLARiiON and Symmetrix

To illustrate the concepts just discussed, this section covers the EMC implementation of intelligent storage arrays.

The EMC CLARiiON storage array is an active-passive array implementation. It is the EMC midrange networked storage offering that delivers enterprise-quality features and functionality. It is ideally suited for applications with predictable workloads that need moderate to high response time and throughput.

The EMC Symmetrix networked storage array is an active-active array implementation. Symmetrix is a solution for customers who require an uncompromising level of service, performance, as well as the most advanced business continuity solution to support large and unpredictable application workloads. Symmetrix also provides built-in, advanced-level security features and offers the most efficient use of power and cooling to support enterprise-level data storage requirements.

For the latest information on CLARiiON and Symmetrix, please refer to `http://education.EMC.com/ismbook`.

4.3.1 CLARiiON Storage Array

The CX4 series is the fourth generation CLARiiON CX storage platform. Each generation has added enhancements to performance, availability, and scalability over the previous generation while the high-level architecture remains the same. Figure 4-9 shows an EMC CLARiiON storage array.

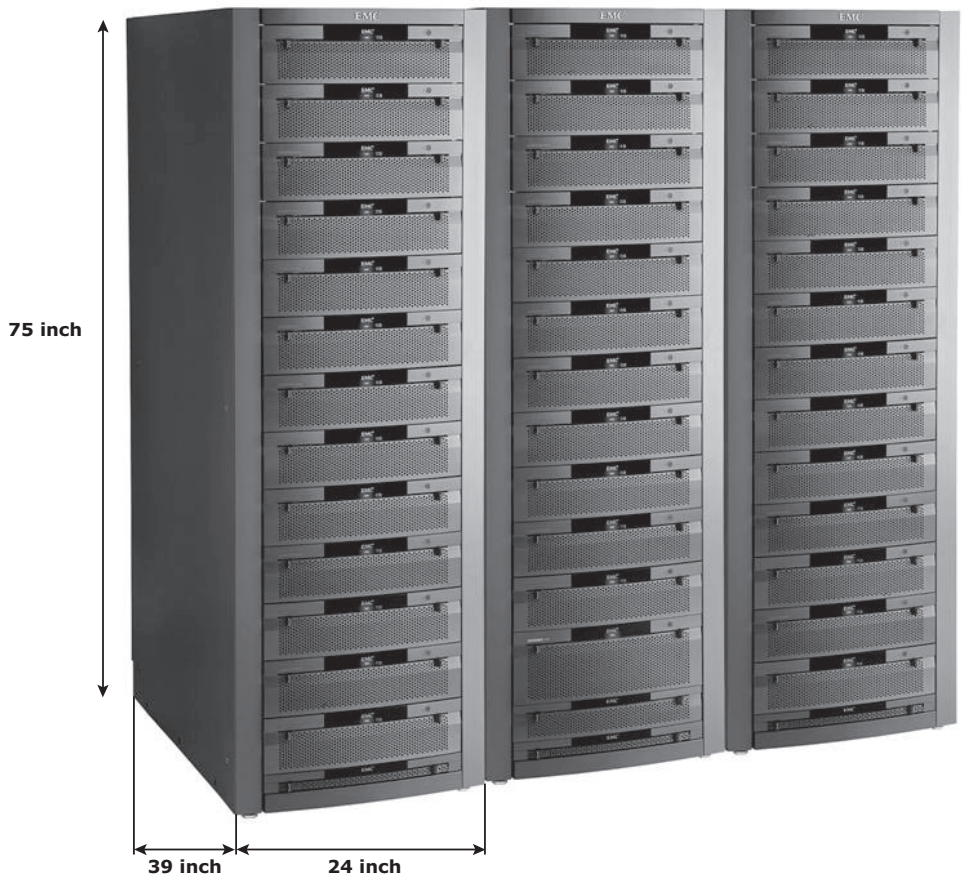

Figure 4-9: EMC CLARiiON

CLARiiON is built with modular building blocks and no single point of failure. CLARiiON CX4 is first midrange storage array that supports flash drives with 30 times more IOPS capability. The other features of CLARiiON are as follows:

- *UltraFlex* technology for dual protocol connectivity, online expansion via IO modules, and readiness for future technologies—such as 8 Gb/s Fibre Channel and 10 Gb/s iSCSI.
- Scalable up to 960 disks
- Supports different types and sizes of drives, and RAID types (0, 1, 1+0, 3, 5, 6)
- Supports up to 16 GB of available cache memory per controller (Storage Processor)
- Enhances availability with nondisruptive upgrade and failover
- Ensures data protection through mirrored write cache and cache vaulting
- Provides data integrity through disk scrubbing. The background verification process runs continually and reads all sectors of all the disks. If a block is unreadable, the back-end error handling recovers the bad sectors from parity or mirror data.
- Supports storage-based local and remote data replication for backup and disaster recovery through SnapView and MirrorView software.

4.3.2 CLARiiON CX4 Architecture

The *Storage Processor Enclosure (SPE)* and the *Disk Array Enclosure (DAE)* are the key modular building blocks of a CLARiiON. A DAE enclosure contains up to 15 disk drives, two link control cards (LCCs), two power supplies, and two fan modules. An SPE contains two storage processors, each consisting of one CPU module and slots for I/O modules. Figure 4-10 shows the CLARiiON CX4 architecture.

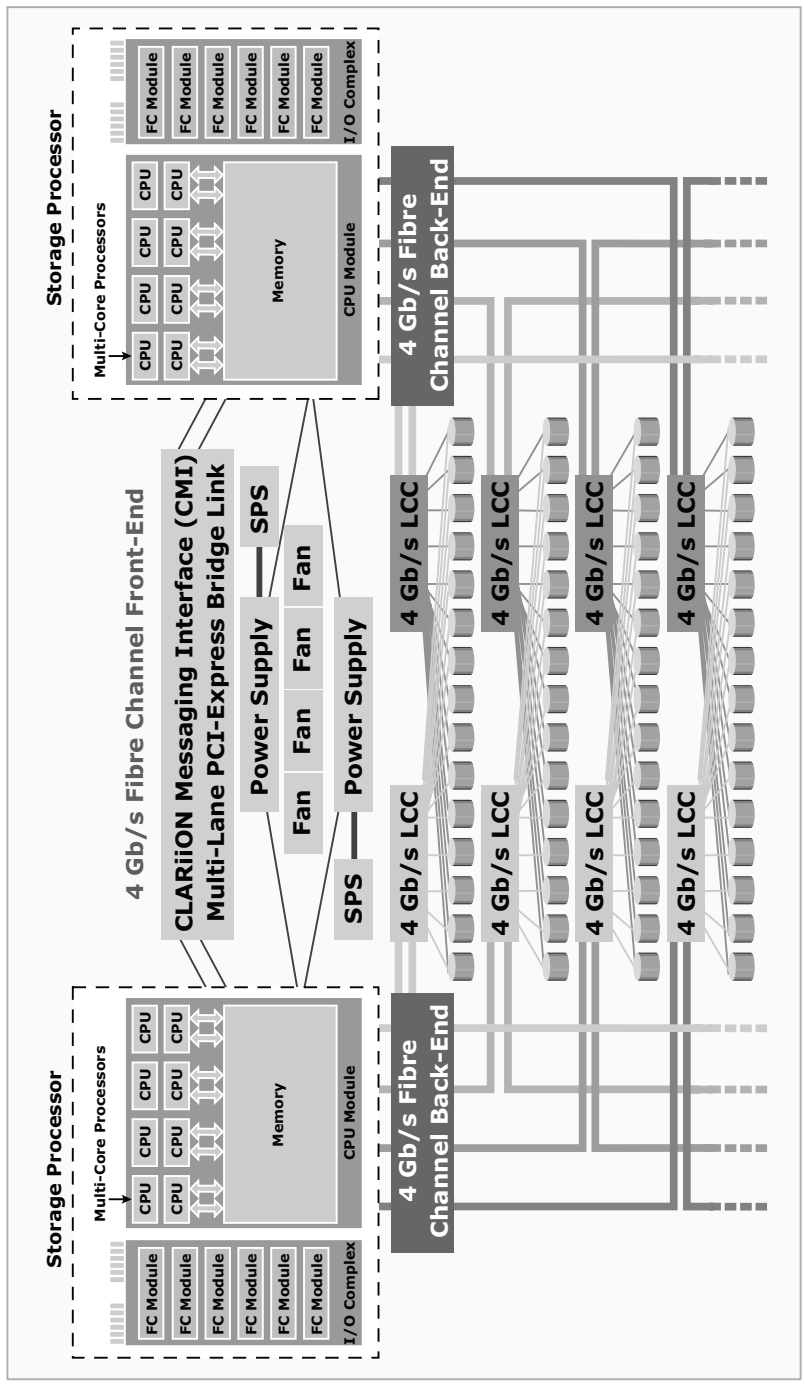

Figure 4-10: CLARiiON architecture

The CLARiiON architecture supports fully redundant, hot swappable components. This means that the system can survive with a failed component, which can be replaced without powering down the system. The important components of the CLARiiON storage system include the following:

- **Intelligent storage processor (SP):** Intelligent SP is the main component of the CLARiiON architecture. SP are configured in pairs for maximum availability. SP provide both front-end and back-end connectivity to the host and the physical disk, respectively. SP also include memory, most of which is used for cache. Depending on the model, each SP includes one or two CPUs.
- **CLARiiON Messaging Interface (CMI):** The SPs communicate to each other over the CLARiiON Messaging Interface, which transports commands, status information, and data for write cache mirroring between the SPs. CLARiiON uses PCI-Express as the high-speed CMI. The PCI Express architecture delivers high bandwidth per pin, has superior routing characteristics, and provides improved reliability.
- **Standby Power Supply (SPS):** In the event of a power failure, the SPS maintains a power supply to the cache for long enough to allow the content to be copied to the vault.
- **Link Control Card (LCC):** The LCC provides services to the drive enclosure, which includes the capability to control enclosure functionalities and monitor environmental status. Each drive enclosure has two LCCs. The other functions performed by LCCs are loop configuration control, failover control, marker LED control, individual disk port control, drive presence detection, and voltage status information.
- **FLARE Storage Operating Environment:** FLARE is a special software designed for EMC CLARiiON. Each storage system ships with a complete copy of the FLARE operating system installed on its first four disks. When CLARiiON is powered up, each SP boots and runs the FLARE operating system. FLARE performs resource allocation and other management tasks in the array.

4.3.3 Managing the CLARiiON

CLARiiON supports both command-line interface (CLI) and graphical user interface (GUI) for management. *Naviseccli* is a CLI-based management tool. Commands can be entered from the connected host system or from a remote server through Telnet/SSH to perform all management functions.

Navisphere management software is a GUI-based suite of tools that enables centralized management of CLARiiON storage systems. These tools are used to monitor, configure, and manage CLARiiON storage arrays. The Navisphere management suite includes the following:

- **Navisphere Manager:** A GUI-based tool for centralized storage system management that is used to configure and manage CLARiiON. It is a web-based user interface that helps to securely manage CLARiiON storage systems locally or remotely over the IP connection, using a common browser. Navisphere Manager provides the flexibility to manage single or multiple systems.
- **Navisphere Analyzer:** A performance analysis tool for CLARiiON hardware components.
- **Navisphere Agent:** A host-residing tool that provides a management communication path to the system and enables CLI access.

4.3.4 Symmetrix Storage Array

The EMC Symmetrix establishes the highest standards for performance and capacity for an enterprise information storage solution and is recognized as the industry's most trusted storage platform. Figure 4-11 shows the EMC Symmetrix DMX-4 storage array.

EMC Symmetrix uses the Direct Matrix Architecture and incorporates a fault-tolerant design. Other features of the Symmetrix are as follows:

- Incrementally scalable up to 2,400 disks
- Supports Flash-based solid-state drives
- Dynamic global cache memory (16 GB–512 GB)
- Advanced processing power (up to 130 PowerPC)
- Large number of concurrent data paths available (32–128 data paths) for I/O processing
- High data processing bandwidth (up to 128 GB/s)
- Data protection with RAID 1, 1+0 (also known as 10 for mainframe), 5, and 6
- Storage-based local and remote replication for business continuity through TimeFinder and SRDF software

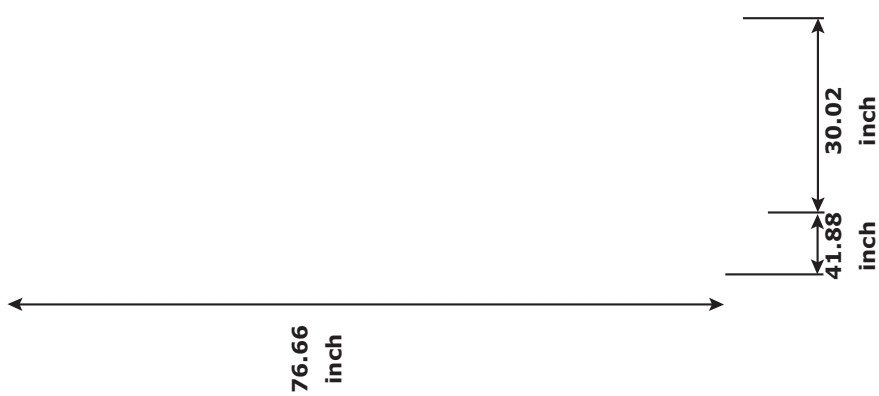

Figure 4-11: EMC Symmetrix

4.3.5 Symmetrix Component Overview

Figure 4-12 shows the basic block diagram of Symmetrix components.

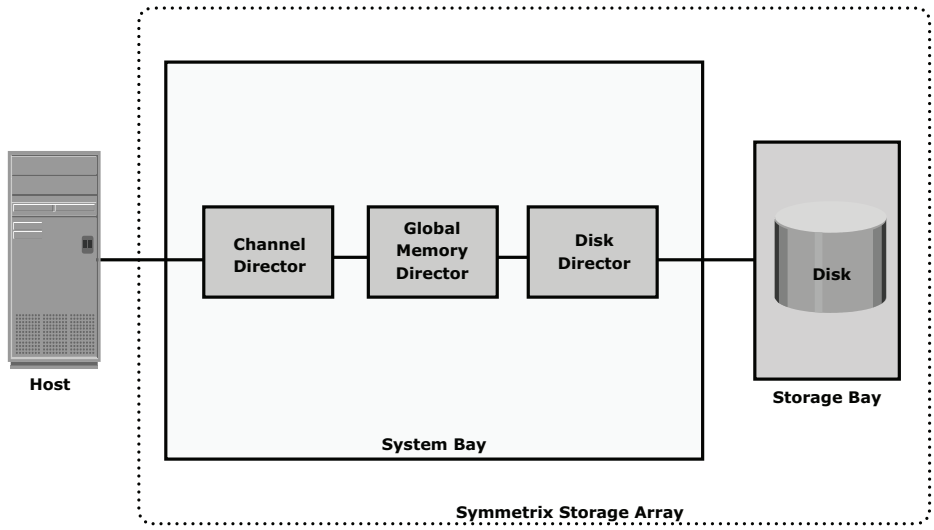

Figure 4-12: Basic building blocks of Symmetrix

The Symmetrix system bay consists of front-end controllers (called *Channel Directors*) for host connectivity, large amounts of global cache (called *Global Memory Director [GMD]*), and back-end controllers (called *Disk Directors*) for disk connectivity. The storage bay is a disk enclosure that can house 240 drives in a cabinet. Figure 4-13 shows the components of the Symmetrix system bay and storage bay.

The system bay consists of a card cage structure whereby controller cards are inserted in cage slots. The 24 slots at the front contain the GMD, disk director, and channel director cards. The rear slots contain channel adapters, disk adapters, and environmental control module cards.

The system bay contains up to 12 front-end channel directors and adapters. Channel directors provide connectivity options for Fibre Channel, ESCON, FICON, iSCSI, or GigE connectivity to the host or the network.

The Symmetrix DMX-4 system also comprises up to eight GMD cards in the card cage. Individual memory directors are available from 2 GB to 64 GB. Symmetrix uses Double Data Rate Synchronous Dynamic Random-Access Memory (DDR SDRAM), the latest generation of the SDRAM chip technology.

Up to eight disk directors/adapters, in pairs, provide back-end 2 GB/s connectivity to the Fibre Channel disk drives. Each disk director can support a maximum of 240 drives.

Two communication and environmental control modules, also called *cross communication modules* (XCMs), are provided for configuration and other communication purposes. The XCMs contain the Ethernet interface between the directors (channel, memory, and disk) and the service processor.

The service processor contains a keyboard, a video display, a mouse, and a server that connects to the Symmetrix subsystem through the private Ethernet interface. The service processor can be configured with an external modem to communicate with the EMC Customer Support Center. The storage bay is configured with up to 240 disk drives, with each cabinet containing a redundant power supply and cooling modules for the disk drives, two LCCs, and 4 to 15 Fibre Channel disk drives per enclosure.

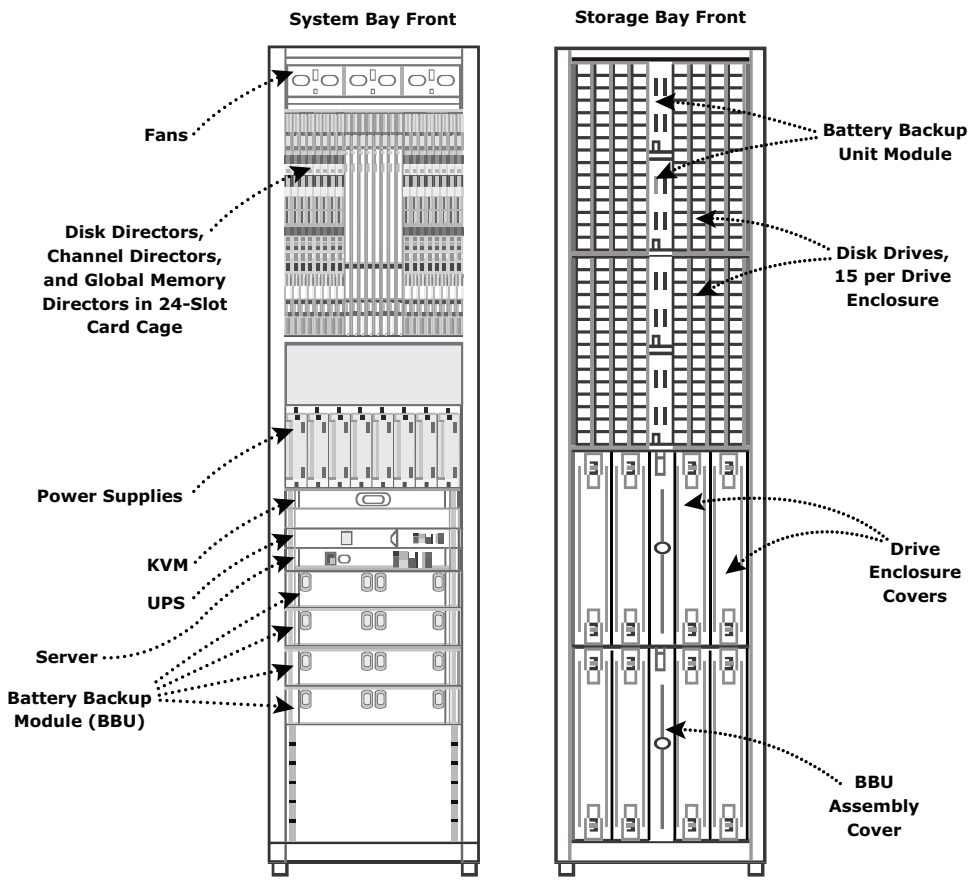

Figure 4-13: Symmetrix system bay and storage bay

4.3.6 Direct Matrix Architecture

Symmetrix uses the Direct Matrix Architecture consisting of dedicated paths for data transfer between the front end, global memory, and the back end. Key components of Symmetrix DMX are as follows:

- **Front end:** The host connects to Symmetrix via a front-end port on the channel director. Multiple directors are configured, each with multiple ports to provide host connectivity and redundancy. Each Symmetrix channel director supports eight internal links to global memory. Data transfers between host and global memory can execute concurrently across multiple ports on a director (refer to Figure 4-14).
- **Back end:** Back end disk directors manage the interface to the disk drives and are responsible for data movement between the disk drives and global memory. Each disk director on a Symmetrix system supports 8 internal links to global memory.
- **Global Memory:** The Symmetrix global memory is its most important component. All read and write operations are performed through global memory. Host I/Os are received at the front end and processed through global memory at much greater electronic speeds than transfers involving disks. The global memory directors work in pairs. The hardware writes to the one global memory director first and then writes are mirrored to the secondary global memory director, for data protection. Each global memory director has 16 ports with full-duplex serial connections between the global memory director and the channel or disk directors (a total of 16 directors) through the *direct matrix*. Each of the 8 director ports on the 16 directors connects to one of the 16 memory ports on each of the 8 global memory directors, as shown in Figure 4-14. These 128 individual point-to-point connections facilitate up to 128 concurrent cache operations in the system, providing ultra-high bandwidth for I/O processing.
- **XCM:** XCM is the communication agent between the service processor and all the processing nodes (channel, disk, and memory director) within the system. External connections to the service processor provide dial-home capability for remote monitoring and diagnostics. The XCM also has the capability to issue remote commands to the director boards, global memory directors, and itself. These commands can be issued from the service processor or remotely, by the EMC Customer Support Center, providing a rich set of intelligent serviceability functions.

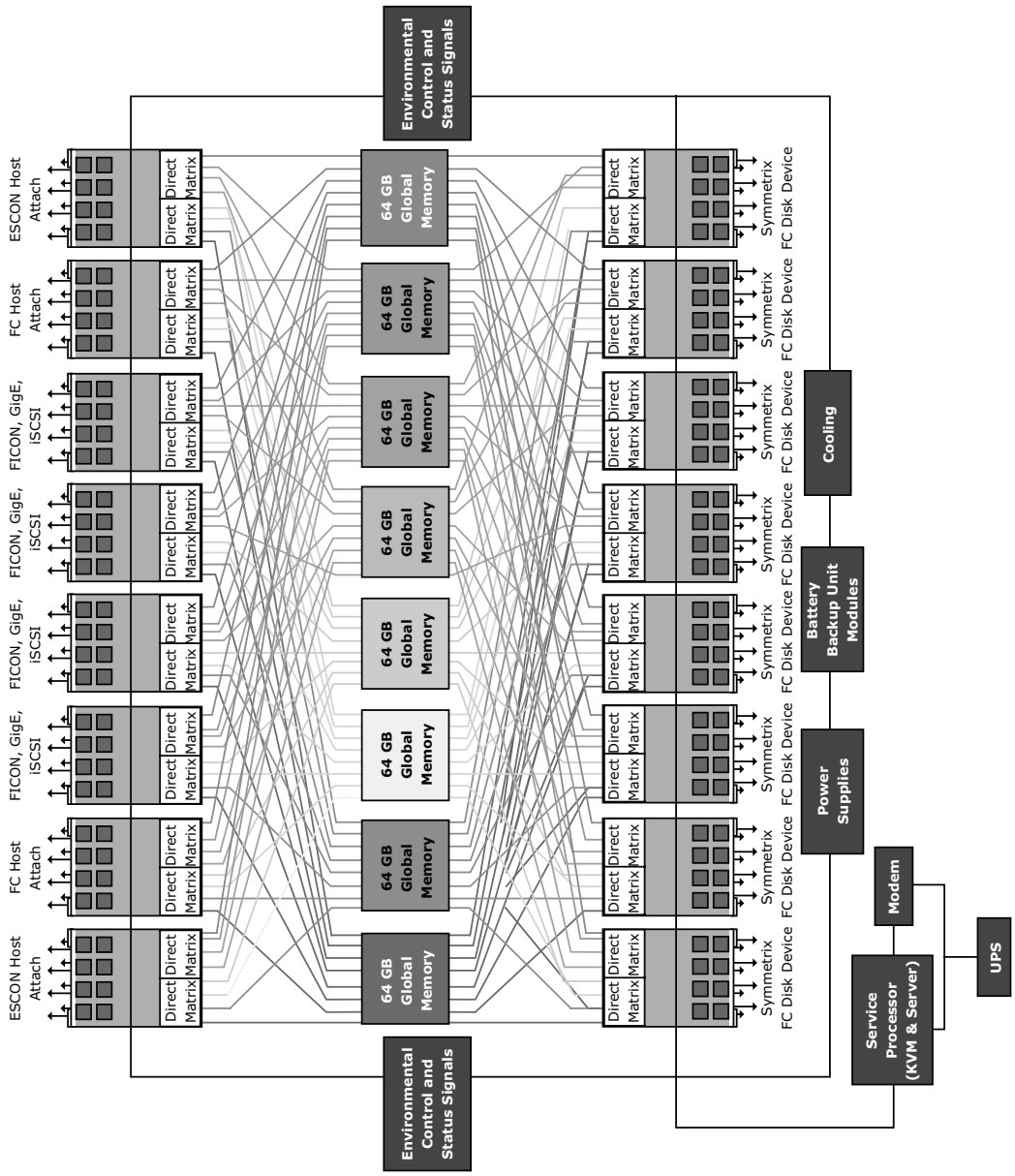

Figure 4-14: Direct matrix architecture

- **Symmetrix Enginuity:** This is the operating environment for EMC Symmetrix. Enginuity manages and ensures the optimal flow and integrity of information through the various hardware components of the Symmetrix system. It manages all Symmetrix operations and system resources to optimize performance intelligently. Enginuity ensures system availability through advanced fault monitoring, detection, and correction capabilities and provides concurrent maintenance and serviceability features. It also offers a foundation for specific software features for disaster recovery, business continuance, and storage management.

Summary

This chapter detailed the features and components of the intelligent storage system — front end, cache, back end, and physical disks. The active-active and active-passive implementations of intelligent storage systems were also described. An intelligent storage system provides the following benefits to an organization:

- Increased capacity
- Improved performance
- Easier storage management
- Improved data availability
- Improved scalability and flexibility
- Improved business continuity
- Improved security and access control

An intelligent storage system is now an integral part of every mission-critical data center. Although a high-end intelligent storage system addresses information storage requirements, it poses a challenge for administrators to share information easily and securely across the enterprise.

Storage networking is a flexible information-centric strategy that extends the reach of intelligent storage systems throughout an enterprise. It provides a common way to manage, share, and protect information. Storage networking is detailed in the next section.

EXERCISES

1. Consider a scenario in which an I/O request from track 1 is followed by an I/O request from track 2 on a sector that is 180 degrees away from the first request. A third request is from a sector on track 3, which is adjacent to the sector on which the first request is made. Discuss the advantages and disadvantages of using the command queuing algorithm in this scenario.
2. Which type of application benefits the most by bypassing write cache? Why?
3. An Oracle database uses a block size of 4 KB for its I/O operation. The application that uses this database primarily performs a sequential read operation. Suggest and explain the appropriate values for the following cache parameters: cache page size, cache allocation (read versus write), pre-fetch type, and write aside cache.
4. Download Navisphere Simulator and the lab guide from `http://education.EMC.com/ismbook` and perform the tasks listed.