UNIT-3

Data Protection: RAID

In 1980s data was stored on a single large, expensive disk drive called Single Large Expensive Drive (SLED)

Use of single disks could not meet the required performance levels.

HDDs are susceptible to failures due to mechanical wear and tear and other environmental factors. An HDD failure may result in data loss.

RAID is an enabling technology that leverages multiple disks as part of a set, which provides data protection against HDD failures. In general, RAID implementations also improve the I/O performance of storage systems by storing data across multiple HDDs.
In 1987, Patterson, Gibson, and Katz at the University of California, Berkeley, published a paper titled "A Case for Redundant Arrays of Inexpensive Disks

(RAID)." This paper described the use of small-capacity, inexpensive disk drives as an alternative to large-capacity drives common on mainframe computers. The term RAID has been redefined to refer to independent disks, to reflect advances in the storage technology. RAID storage has now grown from an academic concept to an industry standard.

Implementation of RAID

There are two types of RAID implementation, hardware and software. Both have their merits and demerits and are discussed in this section.

1.Software RAID

Software RAID uses host-based software to provide RAID functions. It is imple mented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.
Software RAID implementations offer cost and simplicity benefits when com

pared with hardware RAID. However, they have the following limitations:

Performance:Software RAID affects overall system performance. This is due to the additional CPU cycles required to perform RAID calculations.

Supported features:Software RAID does not support all RAID levels.

Operating system compatibility:Software RAID is tied to the host operating system hence upgrades to software RAID or to the operating system should be validated for compatibility. This leads to inflexibility in the data processing environment.

2.Hardware RAID

In hardware RAID implementations, a specialized hardware controller is implemented either on the host or on the array. These implementations vary in the way the storage array interacts with the host.
Controller card R AID is host-based hardware RAID implementation in which a specialized RAID controller is installed in the host and HDDs are connected to it. The RAID Controller interacts with the hard disks using a PCI bus.

The external RAID controller is an array-based hardware RAID. It acts as an interface between the host and disks. It presents storage volumes to the host, which manage the drives using the supported protocol. Key functions of RAID controllers are:
->Management and control of disk aggregations
->Translation of I/O requests between logical disks and physical disks
->Data regeneration in the event of disk failures

RAID Array Components

A RAID arrayis an enclosure that contains a number of HDDs and the support ing hardware and software to implement RAID.
A subset of disks within a RAID array can be grouped to form logical associations called logical arrays, also known as a RAID set or a RAID group.

Logical arrays are comprised of logical volumes (LV). The operating system recognizes the LVs as if they are physical HDDs managed by the RAID control ler. The number of HDDs in a logical array depends on the RAID level used. Configurations could have a logical array with multiple physical arrays or a physical array with multiple logical arrays.

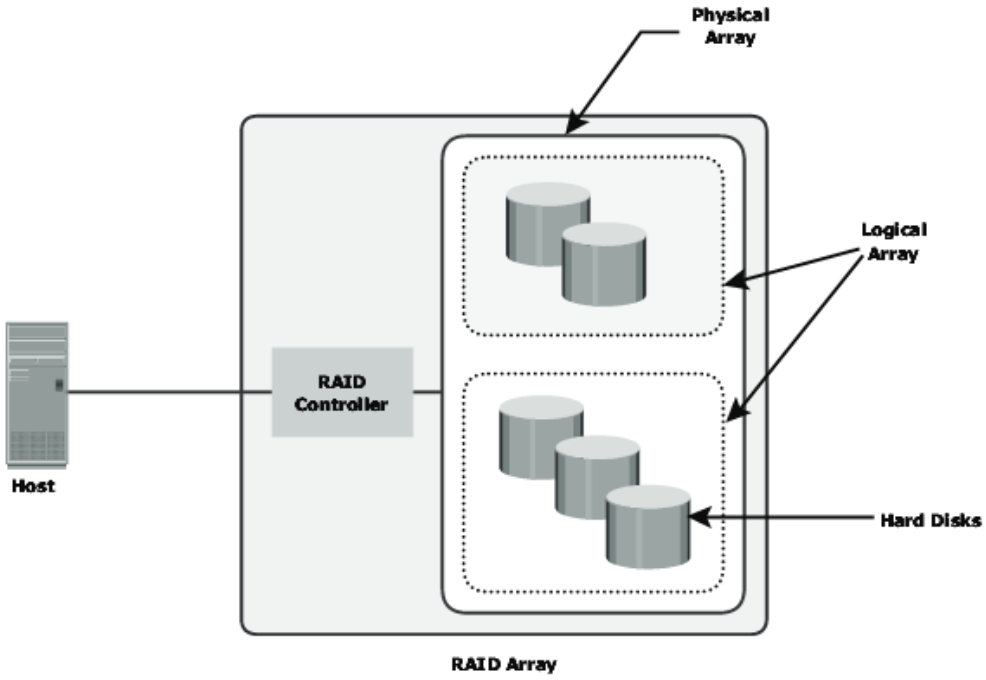


Figure 3-1: Components of RAID array

Raid Techniques

Striping, mirroring and parity form the basis for defining various RAID levels.

1.Striping

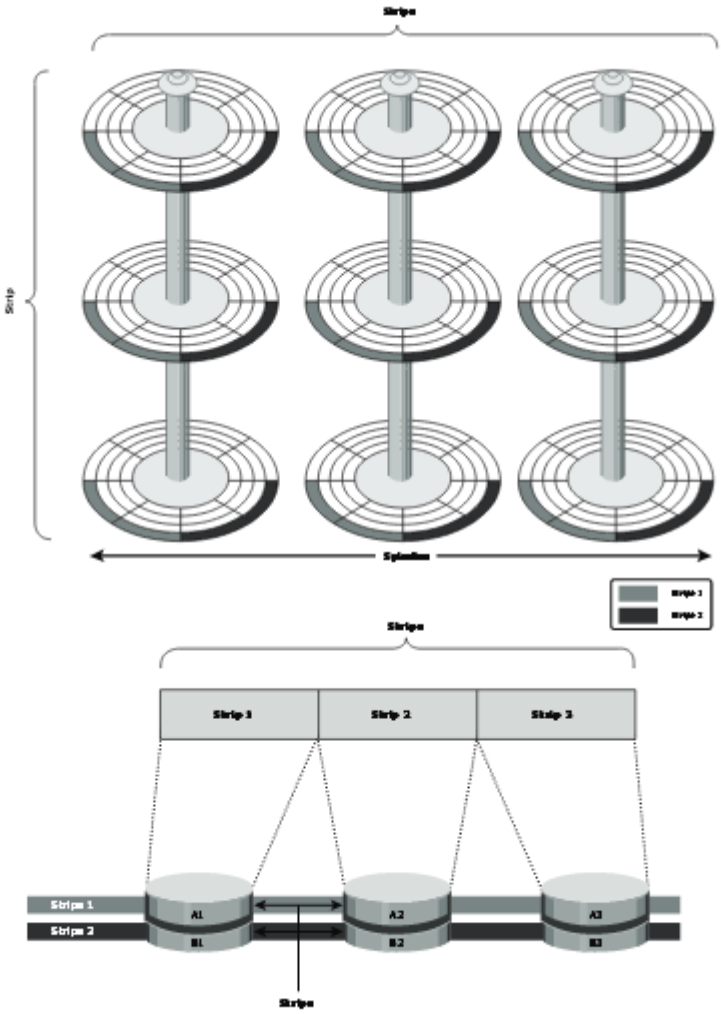


Figure 3-2: Striped RAID set

A RAID set is a group of disks. Within each disk, a predefined number of contiguously addressable disk blocks are defined as
strips.The set of aligned strips that spans across all the disks within the RAID set is called a stripe.
Strip size(also called stripe depth) describes the number of blocks in a strip and is the maximum amount of data that can be written to or read from a single HDD in the set.
Stripe width refers to the number of data strips in a stripe.
Striped RAID does not protect data unless parity or mirroring is used.
However, striping may significantly improve I/O performance.

2.Mirroring

Mirroring is a technique whereby data is stored on two different HDDs, yielding two copies of data.

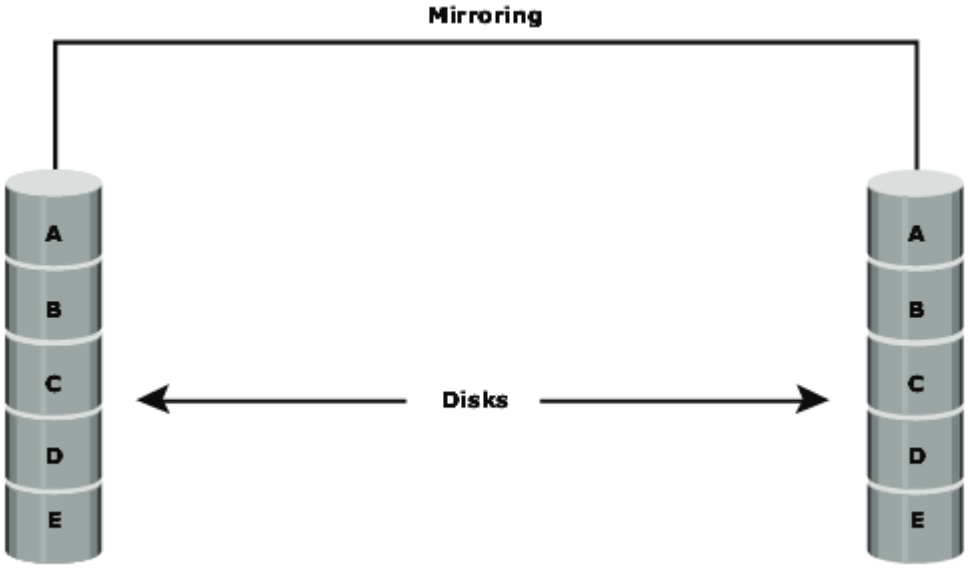


Figure 3-3: Mirrored disks in an array

When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair. This activity is transparent to the host.

n addition to providing complete data redundancy, mirroring enables faster recovery from disk failure.

Mirroring involves duplication of data the amount of storage capacity needed is twice the amount of data being stored. Therefore, mirroring is considered expensive and is preferred for mission-critical applications that cannot afford data loss. Mirroring improves read performance because read requests can be services by both disks.

3. Parity

Parity is a method of protecting striped data from HDD failure without the cost of mirroring. An additional HDD is added to the stripe width to hold parity, a mathematical construct that allows re-creation of the missing data. Parity is a redundancy check that ensures full protection of data without maintaining a full set of duplicate data.

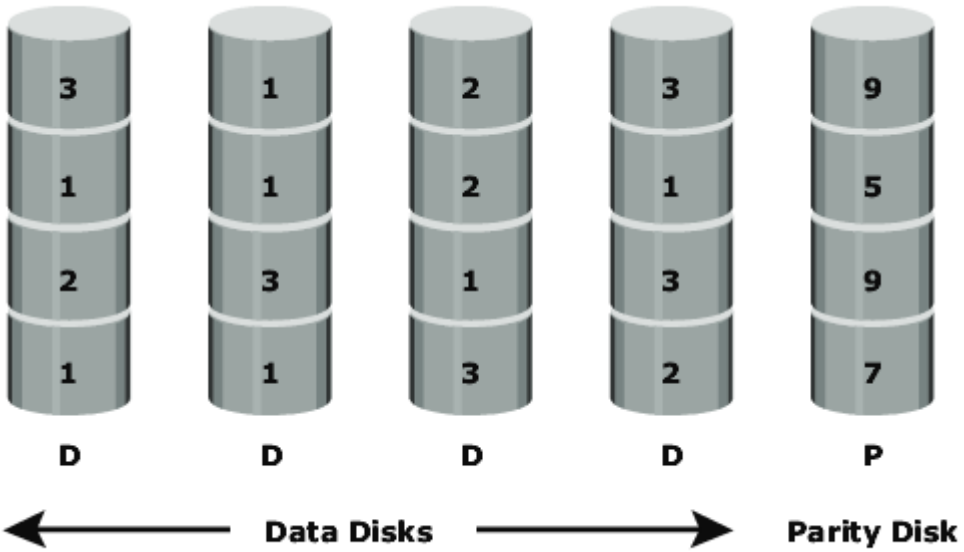


Figure 3-4: Parity RAID

RAID Levels

Table 3-1: Raid Levels

LEVELS	BRIEF DESCRIPTION
RAID 0	Striped array with no fault tolerance
RAID 1	Disk mirroring
RAID 3	Parallel access array with dedicated parity disk
RAID 4	Striped array with independent disks and a dedicated parity disk
RAID 5	Striped array with independent disks and distributed parity
RAID 6	Striped array with independent disks and dual distributed parity
Nested	Combinations of RAID levels. Example: RAID 1 + RAID 0

RAID 0

In a RAID 0 configuration, data is striped across the HDDs in a RAID set. It utilizes the full storage capacity by distributing strips of data over multiple HDDs in a RAID set. To read data, all the strips are put back together by the controller.

RAID 0 is used in applications that need high I/O throughput. However, if these applications require high availability, RAID 0 does not provide data protection and availability in the event of drive failures.

RAID 1

In a RAID 1 configuration, data is mirrored to improve fault tolerance.
A RAID 1 group consists of at least two HDDs. every write is written to both disks, which is transparent to the host in a

hardware RAID implementation. In the event of disk failure, the impact on data recovery is the least among all RAID implementations. This is because the RAID controller uses the mirror drive for data recovery and continuous operation. RAID 1 is suitable for applications that require high availability.

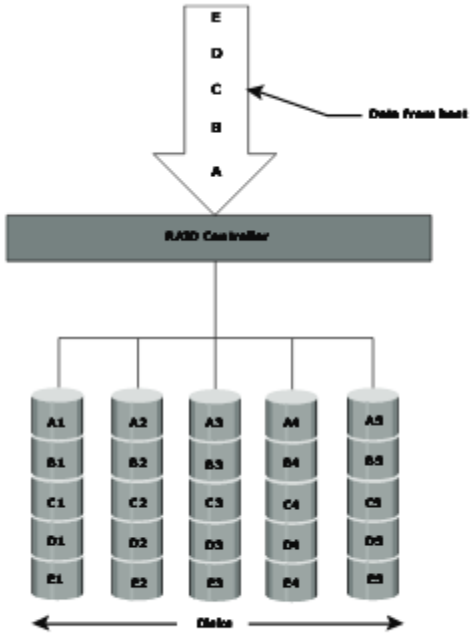


Figure 3-5: RAID 0

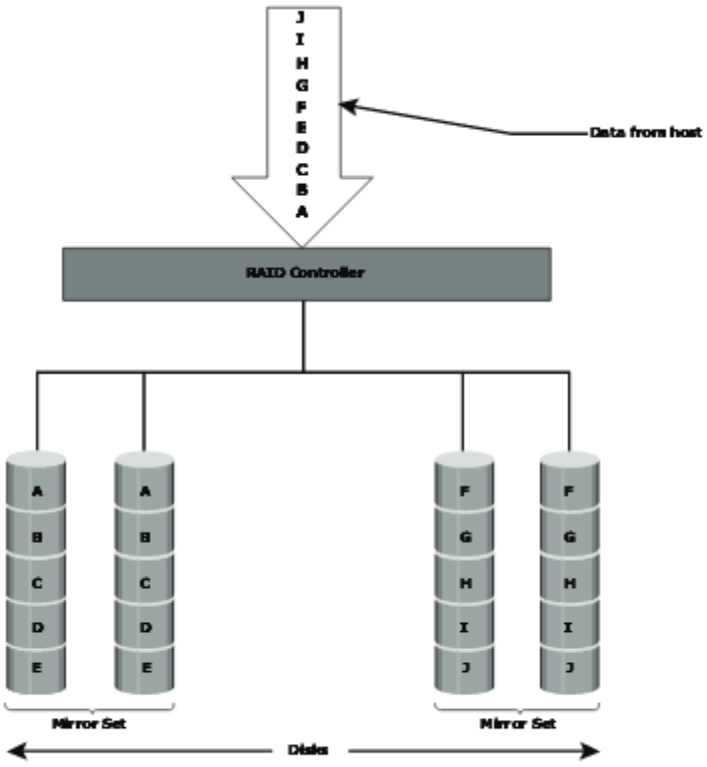


Figure 3-6: RAID 1

Nested RAID

Most data centers require data redundancy and performance from their RAID arrays. RAID 0+1 and RAID 1+0 combine the performance benefits of RAID 0 with the redundancy benefits of RAID 1.

They use striping and mirroring techniques and combine their benefits.
RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0. Similarly, RAID 0+1 is also known as RAID 01 or RAID 0/1. RAID 1+0 performs well for workloads that use small, random, write-intensive I/O. Some applications that benefit from RAID 1+0 include the following:
->High transaction rate Online Transaction Processing (OLTP)
->Large messaging installations
->Database applications that require high I/O rate, random access, and high availability

RAID 1+0 is also called striped mirror.The basic element of RAID 1+0 is a mirrored pair, which means that data is first mirrored and then both copies of data are striped across multiple HDDs in a RAID set. When replacing a failed drive, only the mirror is rebuilt.

RAID 0+1 is also called mirrored stripe.The basic element of RAID 0+1 is a stripe. This means that the process of striping data across HDDs is performed initially and then the entire stripe is mirrored. If one drive fails, then the entire stripe is faulted. A rebuild operation copies the entire stripe, copying data from each disk in the healthy stripe to an equivalent disk in the failed stripe.

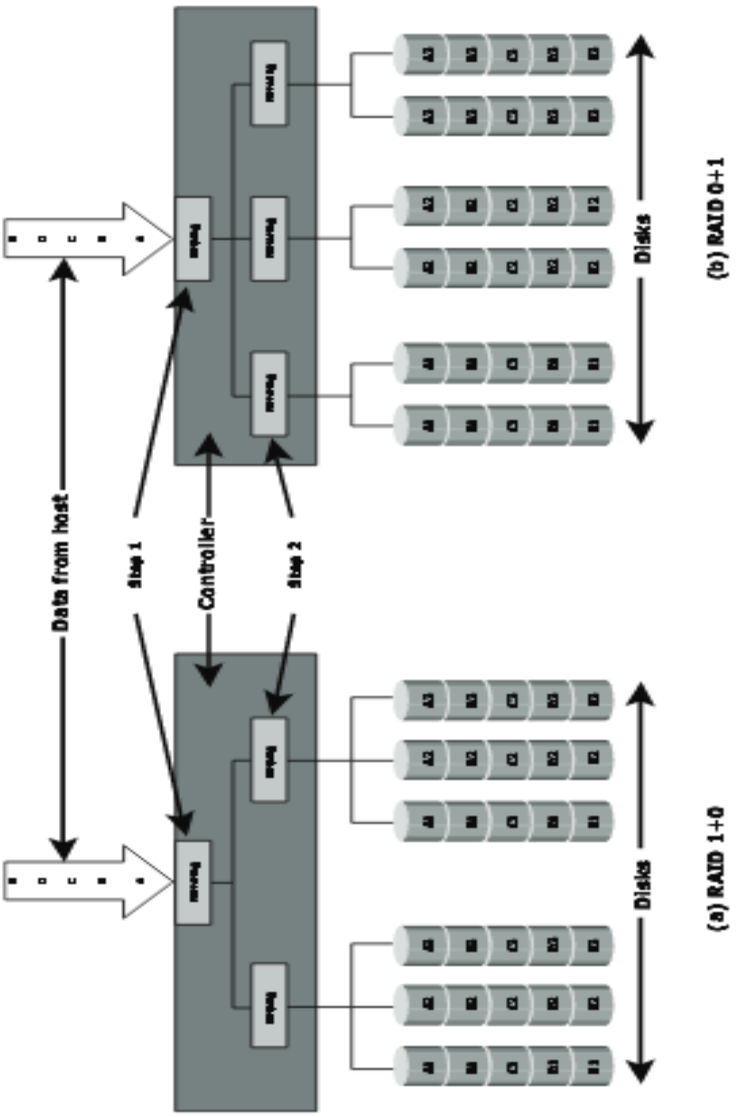


Figure 3-7: Nested RAID

RAID 3

RAID 3 stripes data for high performance and uses parity for improved fault tolerance. Parity information is stored on a dedicated drive so that data can be reconstructed if a drive fails. For example, of five disks, four are used for data and on

issused for parity. Therefore, the total disk space required is 1.25 times the size of the data disks.

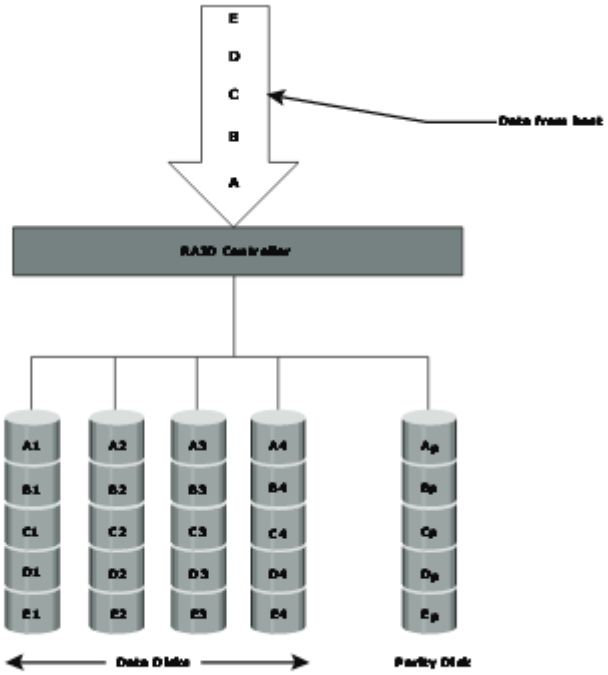


Figure 3-8: RAID 3

RAID 3 provides good bandwidth for the transfer of large volumes of data. RAID 3 is used in applications that involve large sequential data access, such as video streaming.

RAID 4

Similar to RAID 3, RAID 4 stripes data for high performance and uses parity for improved fault tolerance .Data is striped across all disks except the parity disk in the array. Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails. Striping is done at the block level.

RAID 5

RAID 5 is a very versatile RAID implementation. It is similar to RAID 4 because it uses striping and the drives (strips) are independently accessible. The difference between RAID 4 and RAID 5 is the parity location. In RAID 4, parity is written to a dedicated drive, creating a write bottleneck for the parity disk. In RAID 5, parity is distributed across all disks. The distribution of parity in RAID 5 overcomes the write bottleneck.

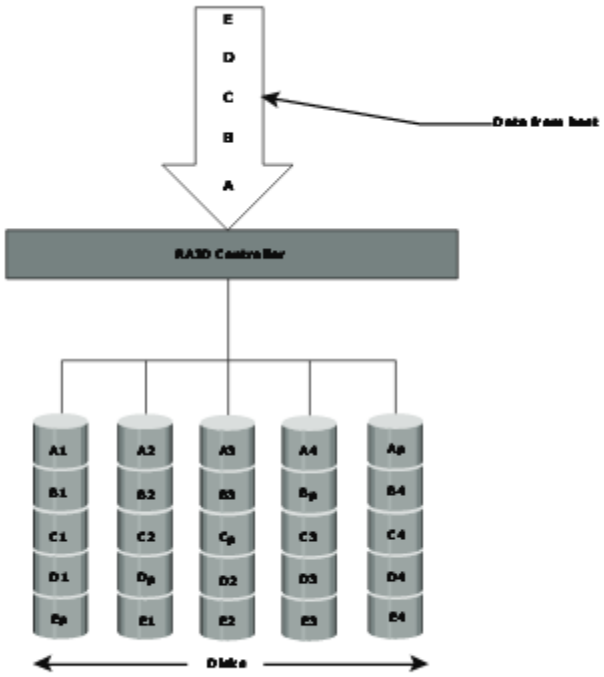


Figure 3-9: RAID 5

RAID 5 is preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations in which database administrators (DBAs) optimize data access.

RAID 6

RAID 6 works the same way as RAID 5 except that RAID 6 includes a second parity element to enable survival in the event of the failure of two disks in a RAID group. Therefore, a RAID 6 implementation requires at least four disks. RAID 6 distributes the parity across all the disks. The write penalty in RAID 6 is more than that in RAID 5; therefore, RAID 5 writes perform better than RAID 6. The rebuild operation in RAID 6 may take longer than that in RAID 5 due to the presence of two parity sets.

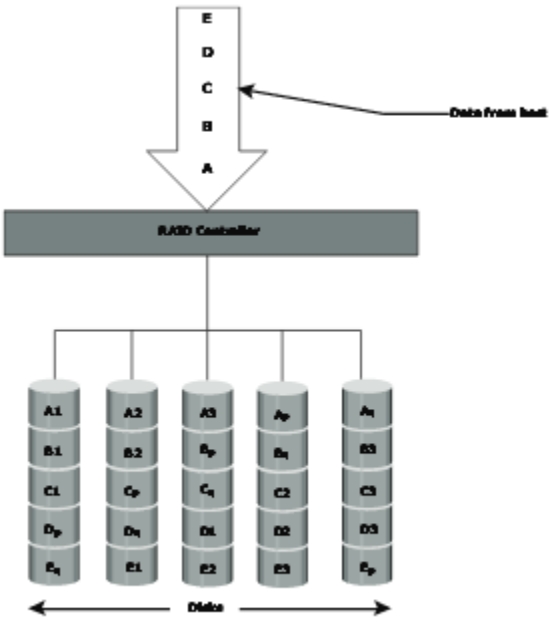


Figure 3-10: RAID 6

Intelligent Storage System(ch 4)

Disk drive →RAID →Intelligent Storage system

Intelligent Storage system is 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.

Components of an intelligent storage system

An intelligent storage system consists of four key components: front end, cache, backend,and physical disks.
An I/O request received from the host at the front-end port is processed through cache and theback 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.

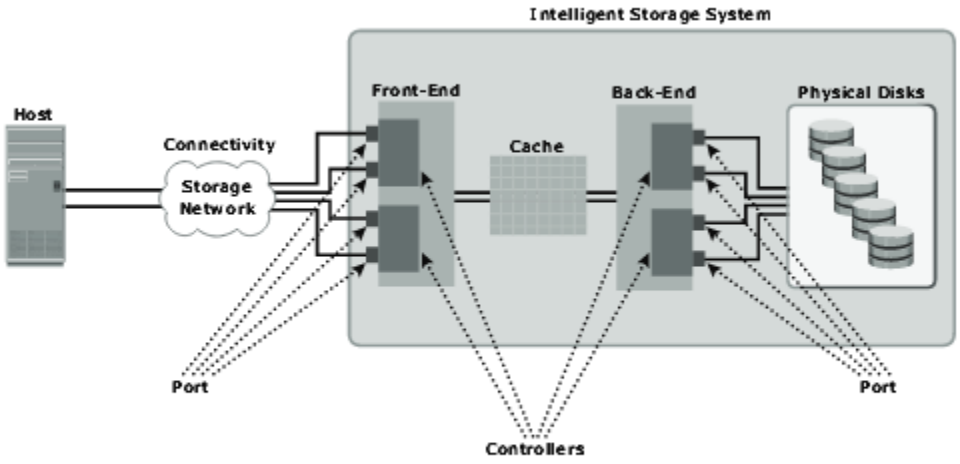


Figure 4-1: Components of an intelligent storage system

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.

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.

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 structure

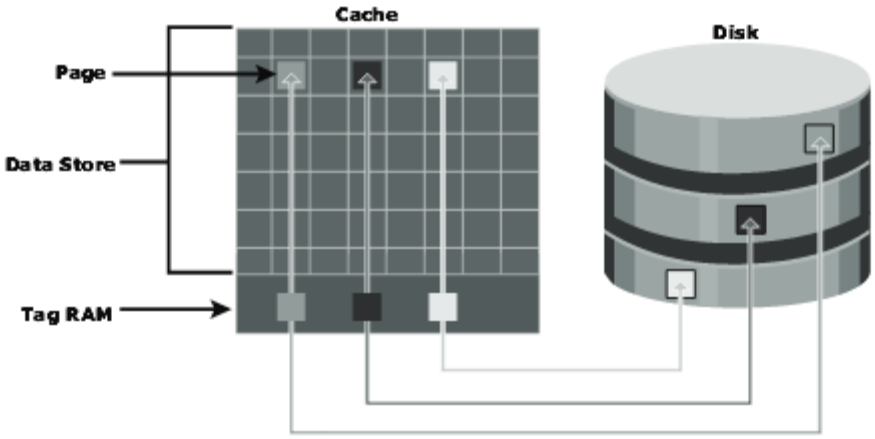


Figure 4-3: Structure of cache

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 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.

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 and data is sent directly to the host, without any disk operation .

If the requested data is not found in cache, it is called a cache miss and the data must be read from the disk . Cache misses increase I/O response time.
A pre-fetch, or read-ahead, algorithm is used when read requests are sequential. 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.
Read performance is measured in terms of the read hit ratio,or the hit rate usually expressed as a percentage.

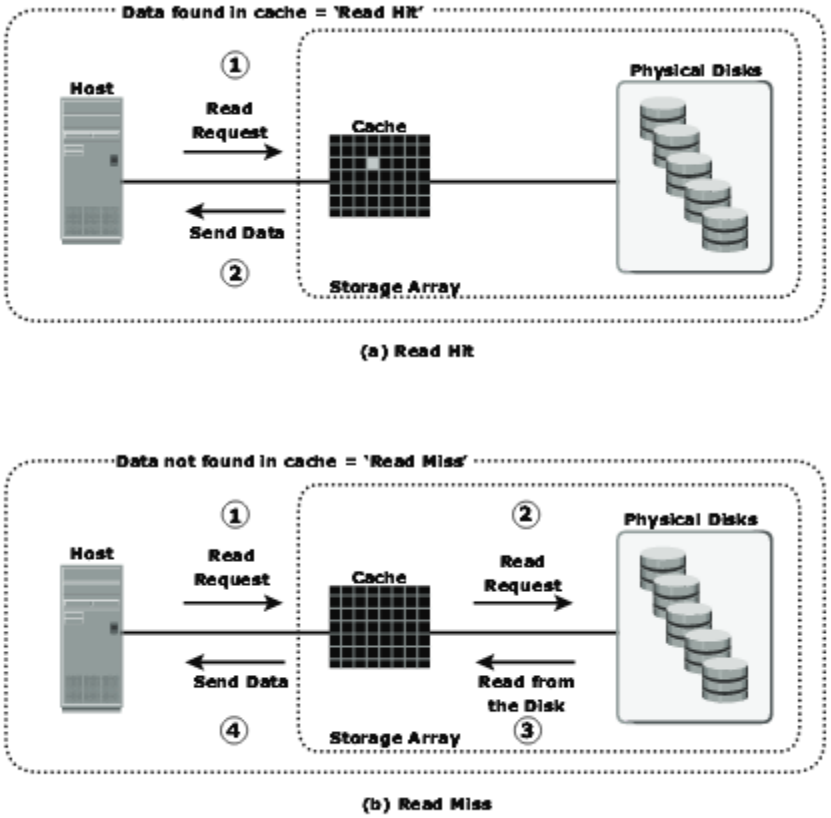


Figure 4-4: Read hit and read miss

Write operation with 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 writ-

ten 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

Cache can be implemented as either dedicated cache or global cache. With dedi
-
cated 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.

Cache Management

Cache is a finite and exexpensive resource that needs proper management.

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.

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.

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 watermarksare 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.
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

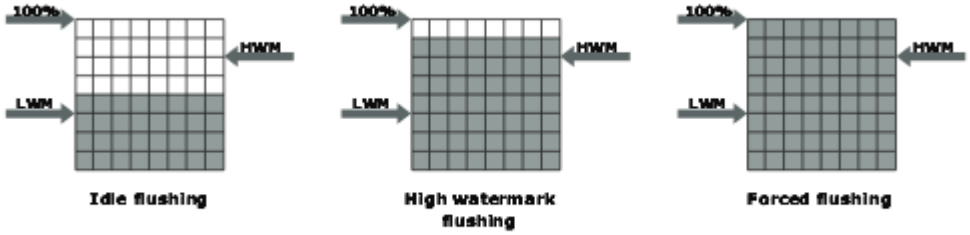


Figure 4-5: Types of flushing

<u>Cache Data Protection</u>
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.
→Cache vaulting
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.

Back End

The back end provides an interface between cache and the physical disks. It con sists 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.

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 chap ters). An intelligent storage system enables the use of a mixture of SCSI or Fibre Channel drives and IDE/ATA drives.

Intelligent Storage Array

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

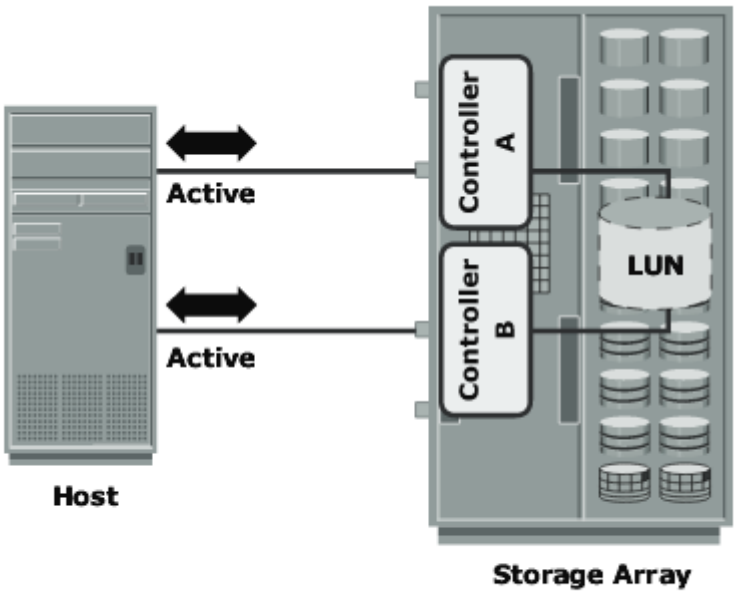
→High-end storage systems

→Midrange end storage systems

Traditionally, high-end storage systems have been implemented with active-active arrays,whereas midrangestorage systemsused typically in small- and medium-sized enterprises have been implemented with active-passive arrays.

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.



igure 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-critica.

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 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,disk drive interfaces.

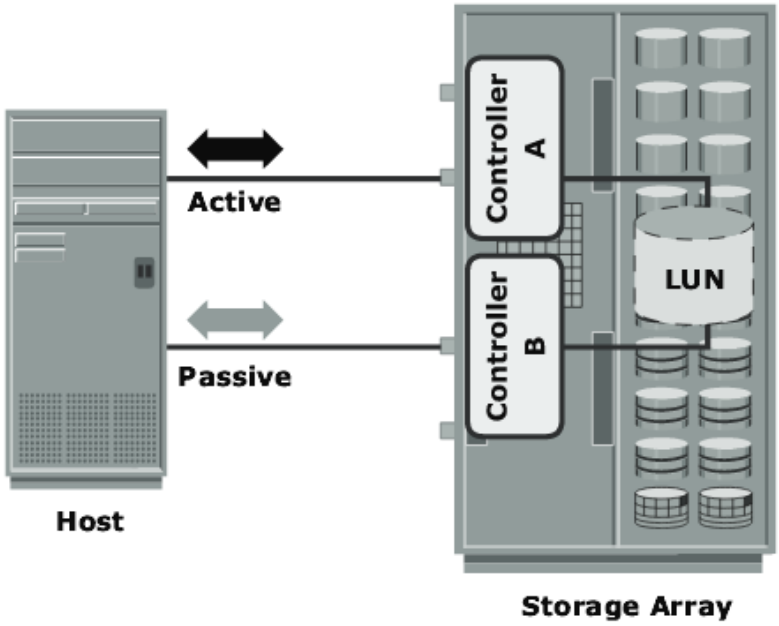


Figure 4-8: Active-passive configuration