# DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
# INTERNAL ASSESSMENT TEST – 1
# STORAGE AREA NETWORKS (10IS765)
# QUESTION PAPER & KEY

Internal Assessment Test 1 – Sept. 2017

| Sub: | STORAGE AREA NETWORKS (SAN) | | | | | Sub Code: | 10IS765 | Branch: | ISE | |
|---|---|---|---|---|---|---|---|---|---|---|
| Date: | 22.09.2017 | Duration: | 90 min's | Max Marks: | 50 | Sem / Sec: | VII | | OBE | |

Answer any FIVE FULL Questions

| | | MARKS | CO | RBT |
|---|---|---|---|---|
| 1 (a) | What is a Data Center? Explain the key characteristics of a data center with a neat diagram. | [10] | CO1 | L4 |
| 2 (a) | Explain disk drive components, with suitable diagram. | [10] | CO1 | L4 |
| 3 (a) | Explain the RAID techniques with diagram. What is Hot Spare? | [10] | CO2 | L4 |
| 4 (a) | Describe RAID levels with reference to nested RAID, RAID 3 and RAID 5 with neat diagram. | [10] | CO2 | L1 |
| 5 (a) | Explain ILM process in detail with neat diagram. | [10] | CO1 | L4 |
| 6 (a) | With neat diagram explain the structure, read and write operations in cache. | [10] | CO2 | L4 |
| 7 (a) | Write a note on i) LVM ii) File System. | [10] | CO1 | L1 |

**Q1(a) What is a Data Center? Explain the key characteristics of a data center with a neat diagram. (10 Marks)**

**Solution:**
Data centers store and manage large amounts of mission-critical data. The data center infrastructure includes computers, storage systems, network devices, dedicated power backups, and environmental controls.

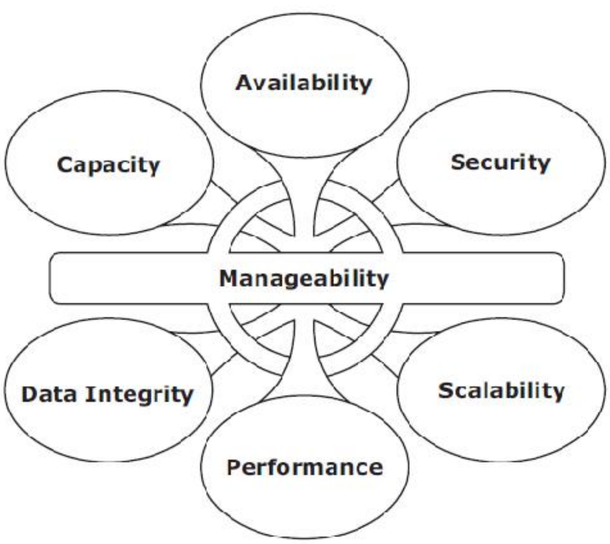



**Fig: Characteristics of data center**

The following are the characteristics of data center:

**1. Availability:** All data center elements should be designed to ensure accessibility. The inability of users to access data can have a significant negative impact on a business.

**2. Security:** Polices, procedures, and proper integration of the data center core elements that will prevent unauthorized access to information must be established. In addition to the security measures for client access, specific mechanisms must enable servers to access only their allocated resources on storage arrays.
**3. Scalability:** Data center operations should be able to allocate additional processing capabilities or storage on demand, without interrupting business operations. Business growth often requires deploying more servers, new applications, and additional databases. The storage solution should be able to grow with the business
**4. Performance:** All the core elements of the data center should be able to provide optimal performance and service all processing requests at high speed. The infrastructure should be able to support performance requirements.
**5. Data integrity:** Data integrity refers to mechanisms such as error correction codes or parity bits which ensure that data is written to disk exactly as it was received. Any variation in data during its retrieval implies corruption, which may affect the operations of the organization.
**6. Capacity:** Data center operations require adequate resources to store and process large amounts of data efficiently. When capacity requirements increase, the data center must be able to provide additional capacity without interrupting availability, or, at the very least, with minimal disruption. Capacity may be managed by reallocation of existing resources, rather than by adding new resources.
**7. Manageability:** A data center should perform all operations and activities in the most efficient manner. Manageability can be achieved through automation and the reduction of human (manual) intervention in common tasks.

**Q2(a) Explain disk drive components, with suitable diagram. (10 Marks)**

**Solution:**
A disk drive uses a rapidly moving arm to read and write data across a flat platter coated with magnetic particles. Data is transferred from the magnetic platter through the R/W head to the computer. Several platters are assembled together with the R/W head and controller, most commonly referred to as a hard disk drive (HDD). Data can be recorded and erased on a magnetic disk any number of times.
Key components of a disk drive are platter, spindle, read/write head, actuator arm assembly, and controller (Figure 2-2)

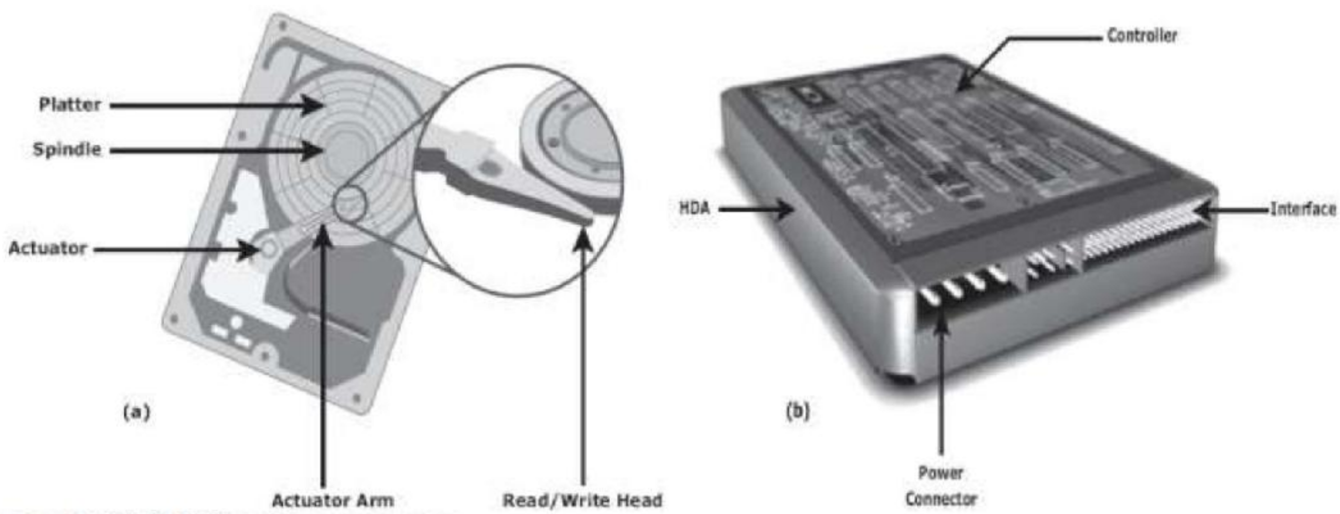



**Figure 2-2:** Disk Drive Components

**Platter:** A typical HDD consists of one or more flat circular disks called platters (Figure 2-3). The data is recorded on these platters in binary codes (0s and 1s). The set of rotating platters is sealed in a case, called a Head Disk Assembly (HDA). The data is encoded by polarizing the magnetic area, or domains, of the disk surface. Data can be written to or read from both surfaces of the platter.
**Spindle:** A spindle connects all the platters, as shown in Figure 2-3, and is connected to a motor. The motor of the spindle rotates with a constant speed. The disk platter spins at a speed of several

thousands of revolutions per minute (rpm). Disk drives have spindle speeds of 7,200 rpm, 10,000 rpm, or 15,000 rpm. Disks used on current storage systems have a platter diameter of 3.5” (90 mm).

**Read/Write Head:** Read/Write (R/W) heads, shown in Figure 2-4, read and write data from or to a platter. Drives have two R/W heads per platter, one for each surface of the platter. The R/W head changes the magnetic polarization on the surface of the platter when writing data. While reading data, this head detects magnetic polarization on the surface of the platter. During reads and writes, the R/W head senses the magnetic polarization and never touches the surface of the platter.

The logic on the disk drive ensures that heads are moved to the landing zone before they touch the surface. If the drive malfunctions and the R/W head accidentally touches the surface of the platter outside the landing zone, a head crash occurs. In a head crash, the magnetic coating on the platter is scratched and may cause damage to the R/W head. A head crash generally results in data loss.

**Actuator Arm Assembly:** The R/W heads are mounted on the actuator arm assembly (refer to Figure 2-2 [a]), which positions the R/W head at the location on the platter where the data needs to be written or read. The R/W heads for all platters on a drive are attached to one actuator arm assembly and move across the platters simultaneously.

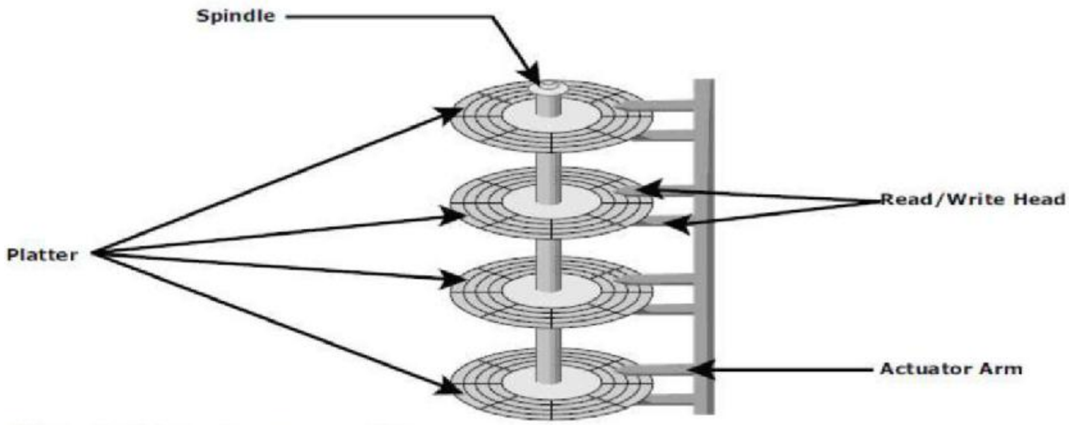



**Figure 2-4:** Actuator arm assembly

**Controller:** The controller (see Figure 2-2 [b]) is a printed circuit board, mounted at the bottom of a disk drive. It consists of a microprocessor, internal memory, circuitry, and firmware. The firmware controls power to the spindle motor and the speed of the motor. It also manages communication between the drive and the host.

**Physical Disk Structure:** Data on the disk is recorded on tracks, which are concentric rings on the platter around the spindle, as shown in Figure 2-5. The tracks are numbered, starting from zero, from the outer edge of the platter. The number of tracks per inch (TPI) on the platter (or the track density) measures how tightly the tracks are packed on a platter. Each track is divided into smaller units called sectors. A sector is the smallest, individually addressable unit of storage.

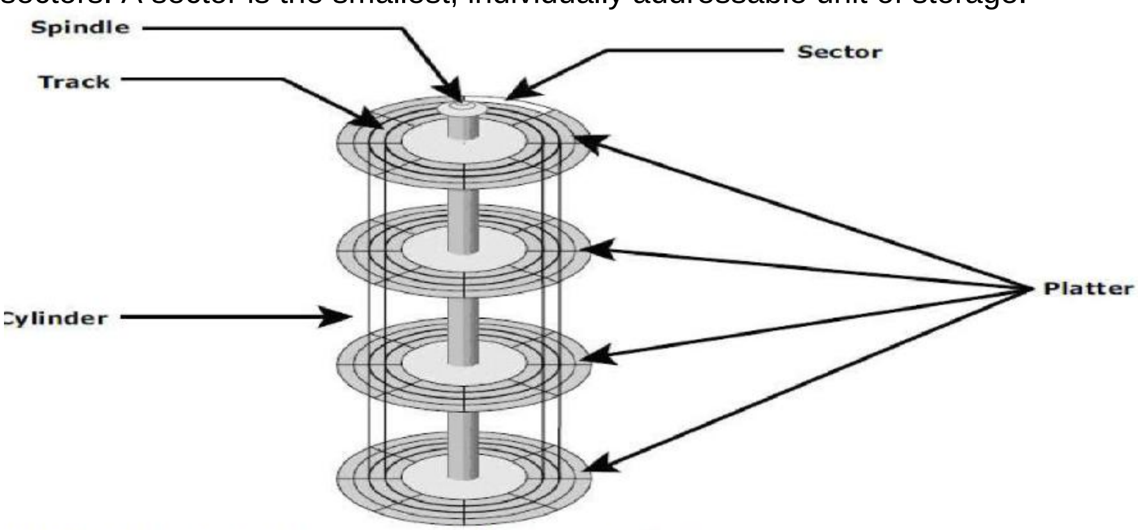
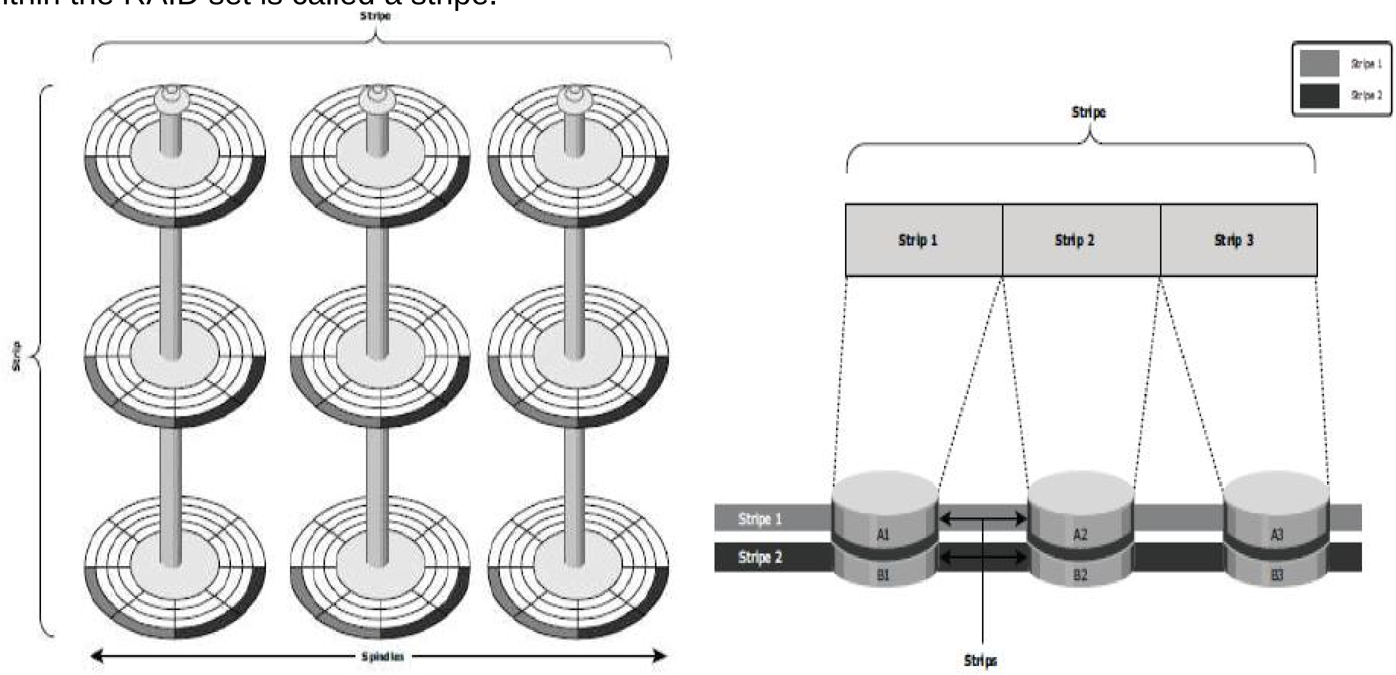



**Figure 2-5:** Disk structure: sectors, tracks, and cylinders

**Q3(a) Explain the RAID techniques with diagram. What is Hot Spare? (10 Marks)**

**Solution:**

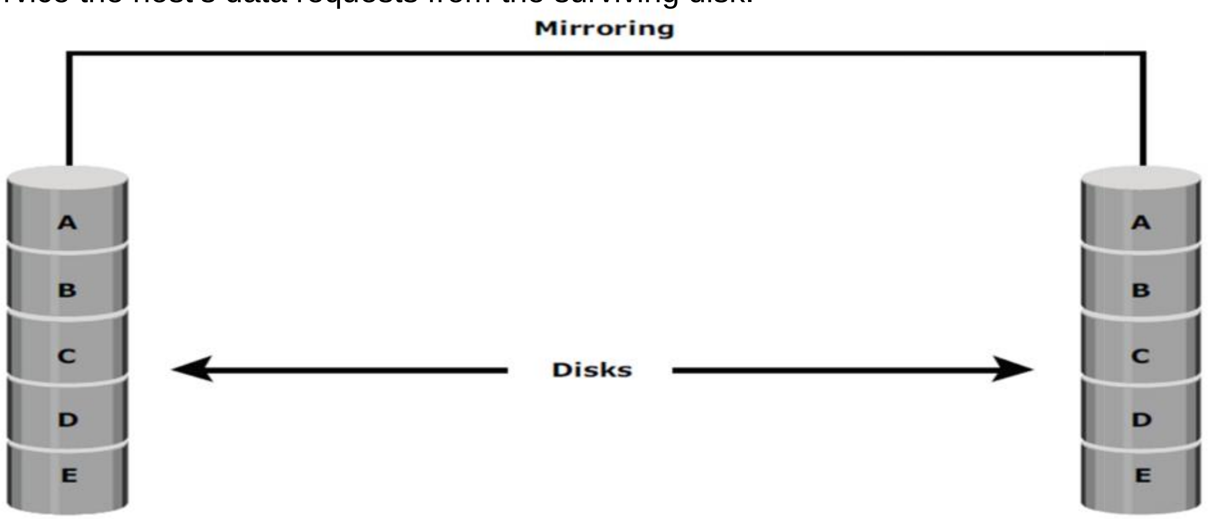
**Striping:** A RAID set is a group of disks. Within each disk, a predefined number of contiguously addressable disk blocks are defined as strips. The set of aligned strips that spans across all the disks within the RAID set is called a stripe.




Striping

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. All strips in a stripe have the same number of blocks. Stripe width refers to the number of data strips in a stripe. Striped RAID does not protect data unless parity or mirroring is used. Striping significantly improves I/O performance.

**Mirroring:** Is a technique where data is stored on two different HDDs, yielding two copies of data. In the event of one HDD failure, the data is intact on the surviving HDD. And the controller continues to service the host's data requests from the surviving disk.




When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair. Mirroring provide complete data redundancy, and also enables faster recovery from disk failure. Mirroring is not a substitute for data backup. Mirroring constantly captures changes in the data, whereas a backup captures point-in-time images of data. Mirroring involves duplication of data - 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 serviced by both disks. However, write performance decreases, as each write request must perform two writes on the HDDs.

**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. Parity is a redundancy check that ensures full protection of data without maintaining a full set of duplicate data. Parity information can be stored on separate, dedicated HDDs or distributed across all the drives in a RAID set. Figure shows a parity RAID. The first four disks, labeled D, contain the data. The fifth disk, labeled P, stores the parity information, which is the sum of the elements in each row. Now, if one of the Ds fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value.

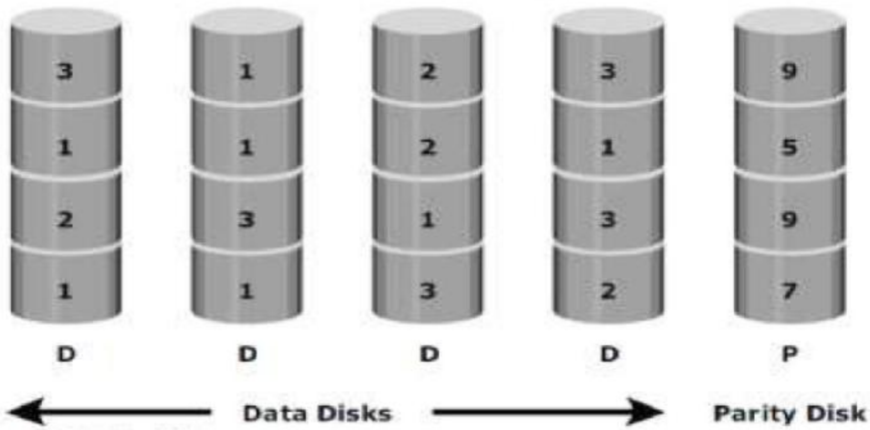



**Figure 3-4:** Parity RAID

The computation of parity is represented as a simple arithmetic operation on the data. Parity calculation is a bitwise XOR operation. Calculation of parity is a function of the RAID controller.

**Advantage:** Compared to mirroring, parity implementation considerably reduces the cost associated with data protection. Consider a RAID configuration with five disks. Four of these disks hold data, and the fifth holds parity information. Parity requires 25 percent extra disk space compared to mirroring, which requires 100 percent extra disk space.

**Disadvantage:** Parity information is generated from data on the data disk. Therefore, parity is recalculated every time there is a change in data. This recalculation is time-consuming and affects the performance of the RAID controller.

**Hot Spare:** A hot spare refers to a spare HDD in a RAID array that temporarily replaces a failed HDD of a RAID set. A hot spare takes the identity of the failed HDD in the array. One of the following methods of data recovery is performed depending on the RAID implementation:

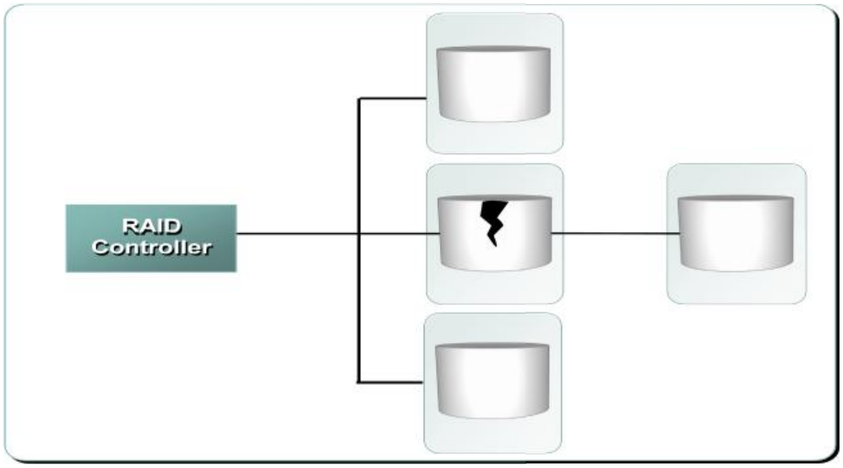
1. If parity RAID is used, then the data is rebuilt onto the hot spare from the parity and the data on the surviving HDDs in the RAID set.
2. If mirroring is used, then the data from the surviving mirror is used to copy the data.

When the failed HDD is replaced with a new HDD, one of the following takes place:

1. The hot spare replaces the new HDD permanently. This means that it is no longer a hot spare, and a new hot spare must be configured on the array.
2. When a new HDD is added to the system, data from the hot spare is copied to it. The hot spare returns to its idle state, ready to replace the next failed drive.

A hot spare should be large enough to accommodate data from a failed drive. System can implement multiple hot spares to improve data availability. A hot spare can be configured as automatic or user initiated, which specifies how it will be used in the event of disk failure.

In an automatic configuration, when the recoverable error rates for a disk exceed a predetermined threshold, the disk subsystem tries to copy data from the failing disk to the hot spare automatically. If this task is completed before the damaged disk fails, then the subsystem switches to the hot spare and marks the failing disk as unusable.




**Q4(a) Describe RAID levels with reference to nested RAID, RAID 3 and RAID 5 with neat diagram. (10 Marks)**

**Solution:**

**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. These types of RAID require an even number of disks, the minimum being four (see Figure below). 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. A common misconception is that RAID 1+0 and RAID 0+1 are the same.

Some applications that benefit from RAID 1+0 include the following:

1. High transaction rate Online Transaction Processing (OLTP)
2. Large messaging installations
3. Database applications that require high I/O rate, random access, and high availability.

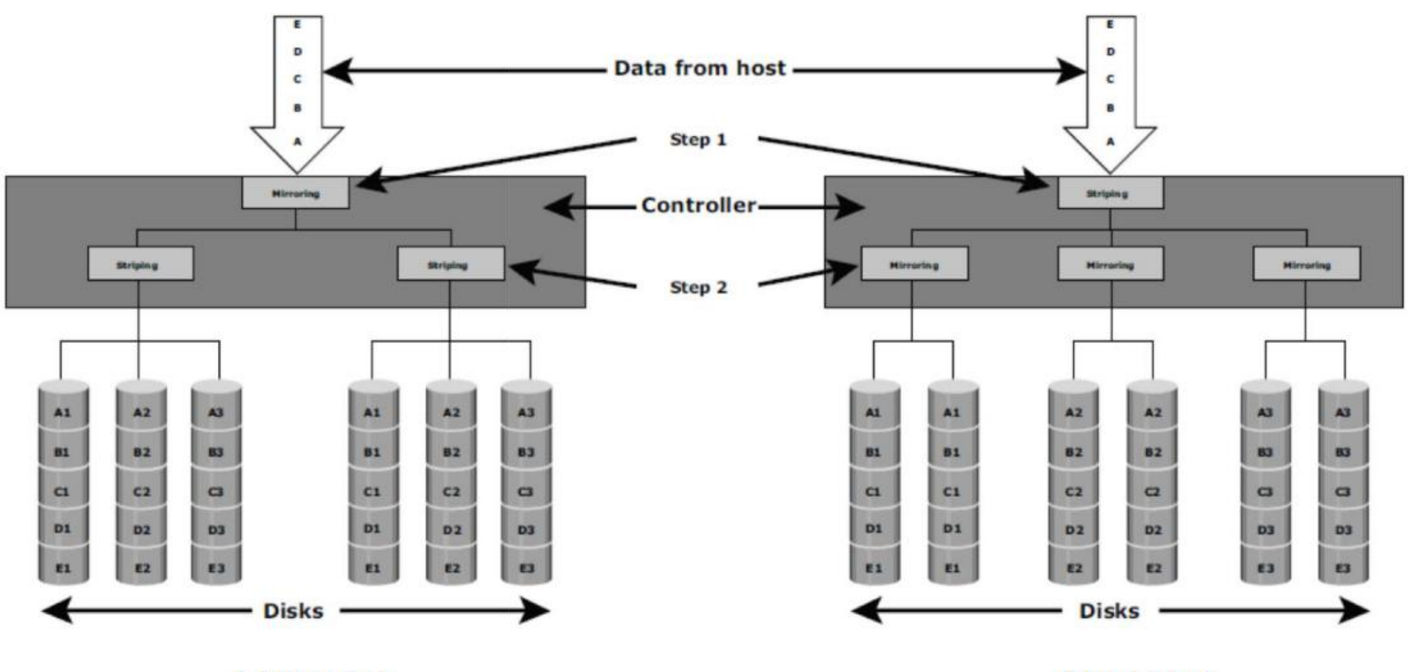



**Figure: Nested RAID**

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, i.e. the disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation. Data from the surviving disk is copied to the replacement disk.

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.
Disadv: This causes increased and unnecessary I/O load on the surviving disks and makes the RAID set more vulnerable to a second disk failure.

**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 one is used for parity. Therefore, the total disk space required is 1.25 times the size of the data disks. RAID 3 always reads and writes complete stripes of data across all disks, as the drives operate in parallel. There are no partial writes that update one out of many strips. Figure 3-8 illustrates the RAID 3 implementation.\ 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.

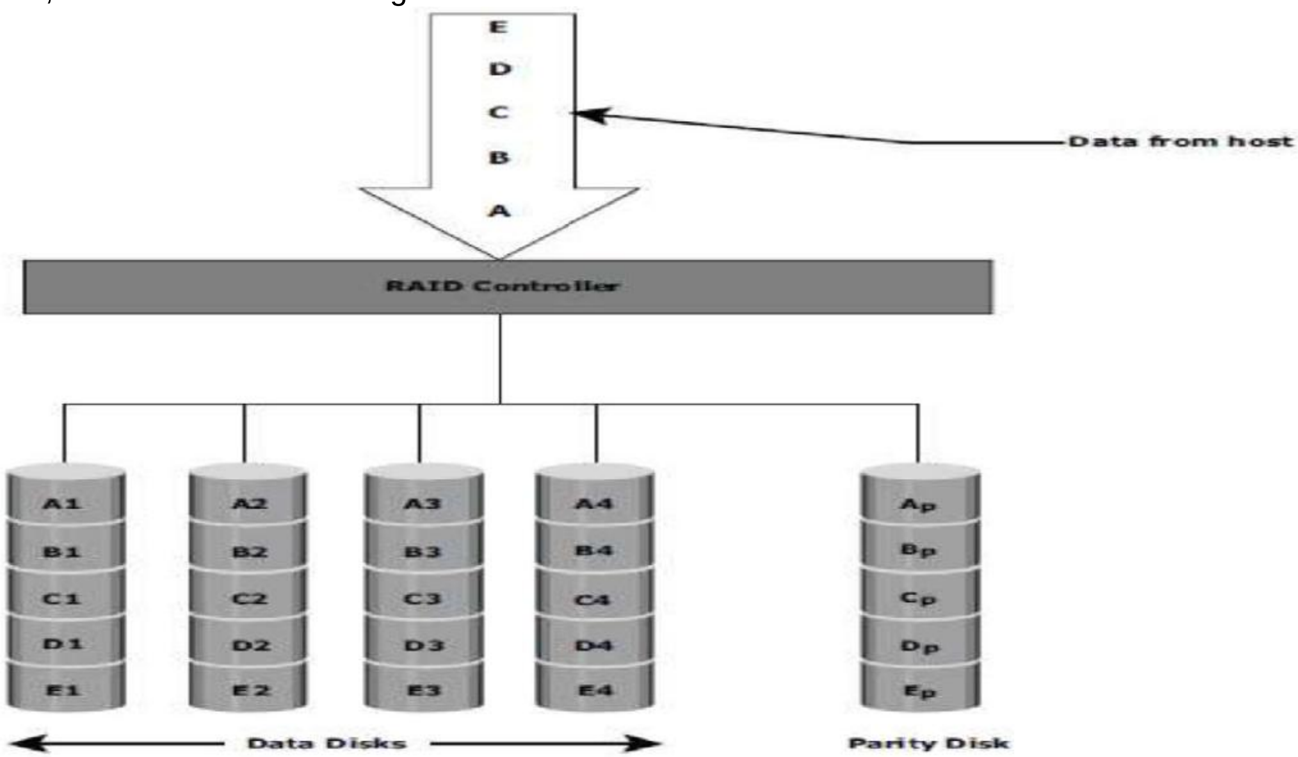



**Figure: RAID 3**

**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 illustrates the RAID 5 implementation. 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.

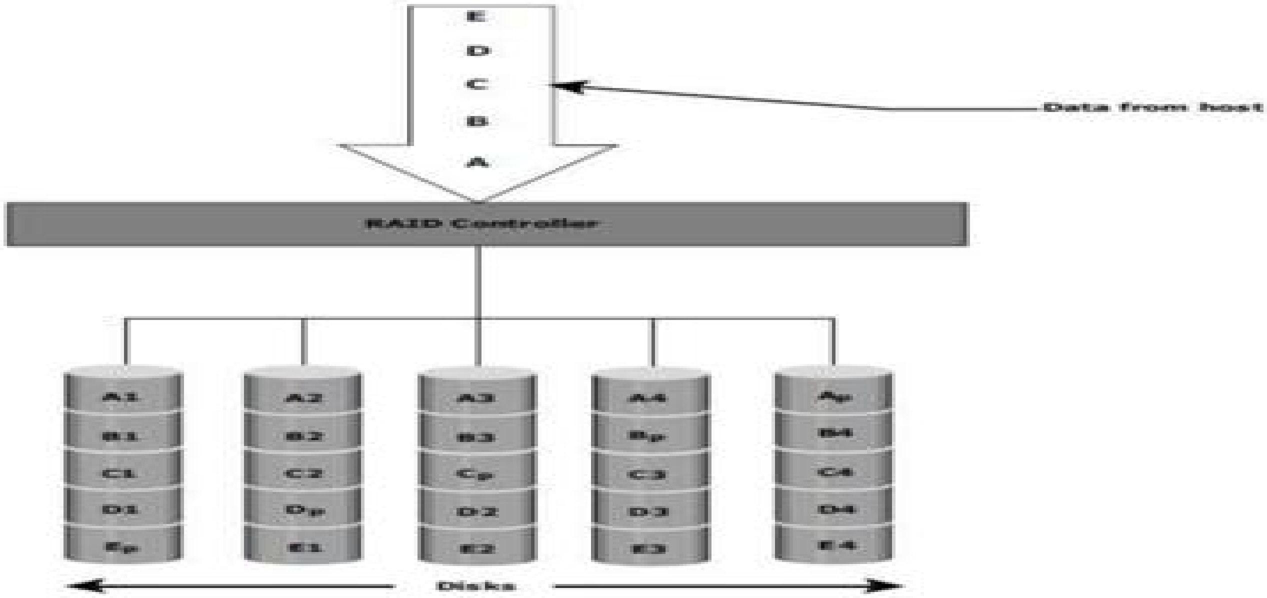



**Figure: RAID 5**

**Q5(a) Explain ILM process in detail with neat diagram. (10 Marks)**

**Solution:**

The information lifecycle is the "change in the value of information" over time. When data is first created, it often has the highest value and is used frequently. As data ages, it is accessed less frequently and is of less value to the organization.

For example, in a sales order application, the value of the information changes from the time the order is placed until the time that the warranty becomes void (see Figure 1-7). The value of the information is highest when a company receives a new sales order and processes it to deliver the product.

After order fulfillment, the customer or order data need not be available for real-time access. The company can transfer this data to less expensive secondary storage with lower accessibility and availability requirements unless or until a warranty claim or another event triggers its need.

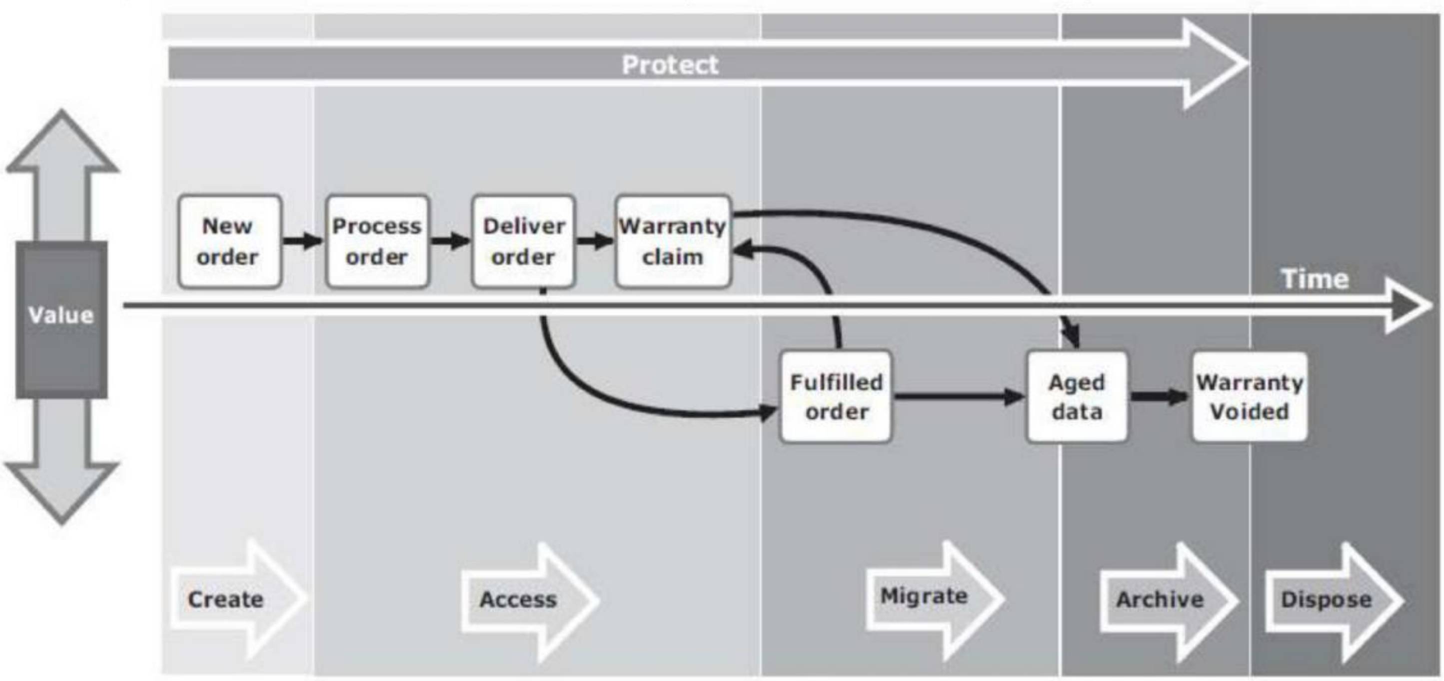



**Figure 1-7:** Changing value of sales order information

Today's business requires data to be protected and available 24 × 7. Data centers can accomplish this with the optimal and appropriate use of storage infrastructure.
Information lifecycle management (ILM) is a proactive strategy that enables an IT organization to effectively manage the data throughout its lifecycle, based on predefined business policies. This allows an IT organization to optimize the storage infrastructure for maximum return on investment.

An ILM strategy should include the following characteristics:
**1 Business-centric:** It should be integrated with key processes, applications, and initiatives of the business to meet both current and future growth in information.
**2 Centrally managed:** All the information assets of a business should be under the purview of the ILM strategy.
**3 Policy-based:** The implementation of ILM should not be restricted to a few departments. ILM should be implemented as a policy and encompass all business applications, processes, and resources.
**4 Heterogeneous:** An ILM strategy should take into account all types of storage platforms and operating systems.
**5 Optimized:** Because the value of information varies, an ILM strategy should consider the different storage requirements and allocate storage resources based on the information's value to the business.

**Q6(a) With neat diagram explain the structure, read and write operations in cache. (10 Marks)**

**Solution:**

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 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.

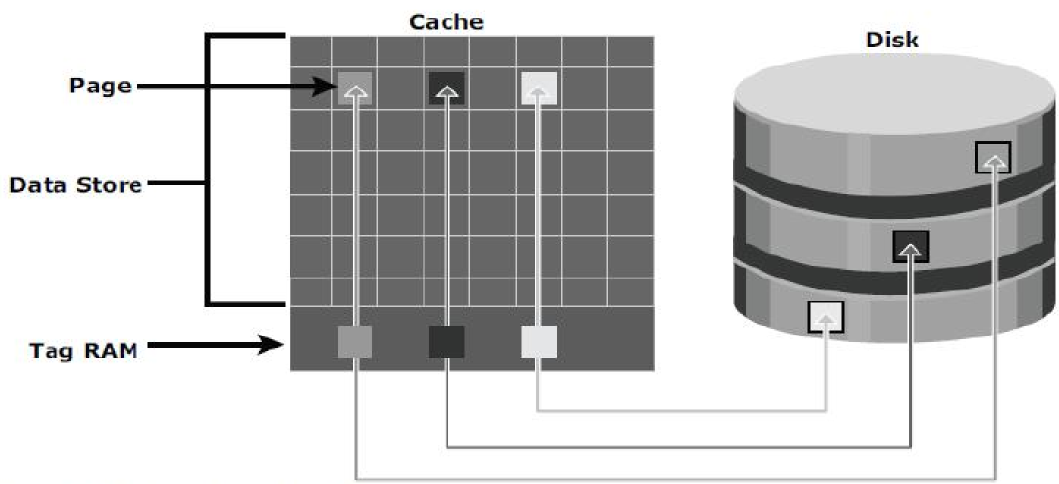
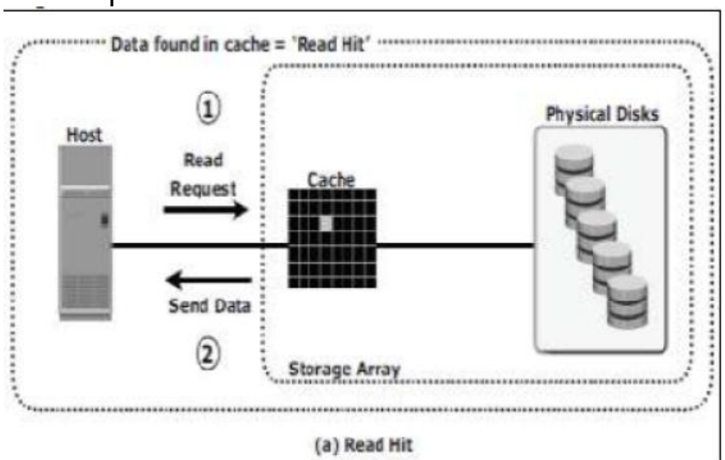



**Figure 4-3:** Structure of cache

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.

**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 read cache miss or read 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.



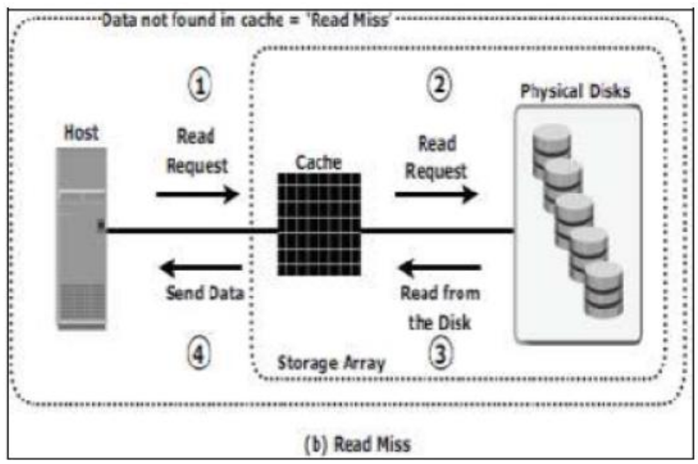
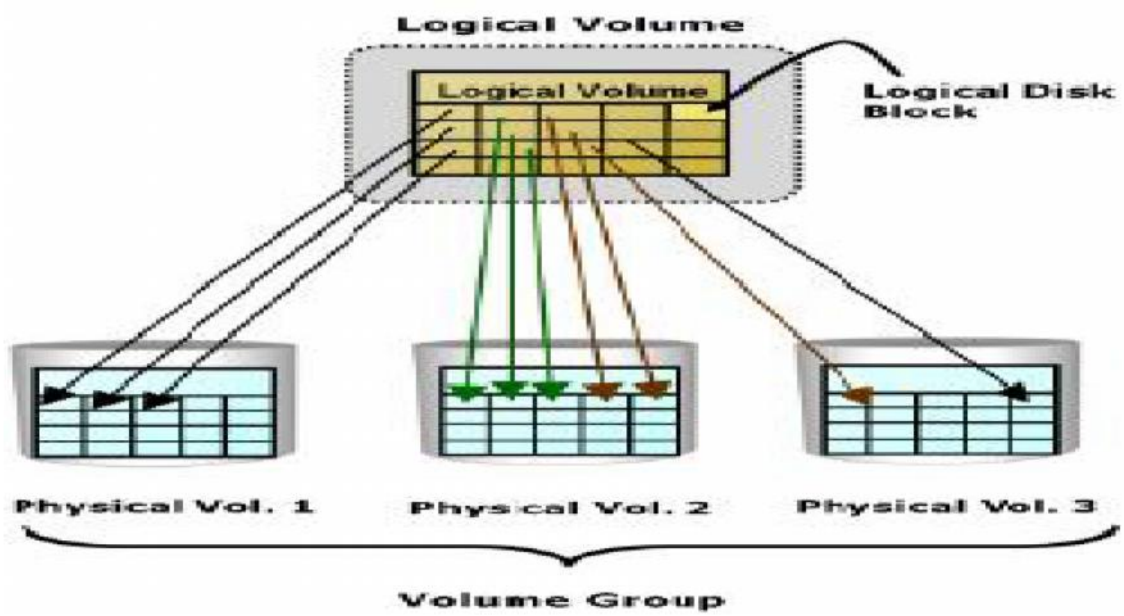




Figure 4-4: Read hit and read miss

**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 less time (from the host's perspective) than it would take to write directly to disk.
A write operation with cache is implemented in the following ways:
1. **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.
2. **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.

**Q7(a) Write a note on i) LVM ii) File System. (10 Marks)**

**Solution:**
**i)LVM**
Physical Volume is the actual space of storage in the disk. Volume group is the combination of one or more volume groups and logical volume is the part of physical volume that is given to the user.
LVM is an optional, intermediate layer between the file system and the physical disk. It can aggregate several smaller disks to form a larger virtual disk or to partition a larger-capacity disk into virtual, smaller capacity disks, which are then presented to applications.




**Functions:**

- The LVM provides optimized storage access and simplifies storage resource management.
- LVM hides details about the physical disk and the location of data on the disk.
- LVM enables administrators to change the storage allocation without changing the hardware, even when the application is running.
- LVM converts the physical storage provided by the physical volumes to a logical view of storage, which is then used by the operating system and applications. A volume group is created by grouping together one or more physical volumes.
- A unique physical volume identifier (PVID) is assigned to each physical volume when it is initialized for use by the LVM. Physical volumes can be added or removed from a volume group dynamically.

They cannot be shared between volume groups; the entire physical volume becomes part of a volume group. Each physical volume is partitioned into equal-sized data blocks called physical extents when the volume group is created Logical volumes are created within a given volume group. A logical volume can be thought of as a virtual disk partition, while the volume group itself can be thought of as a disk. A volume group can have a number of logical volumes. The size of a logical volume is based on a multiple of the physical extents. The logical volume appears as a physical device to the operating system. A logical volume can be made up of noncontiguous physical partitions and can span multiple physical volumes.

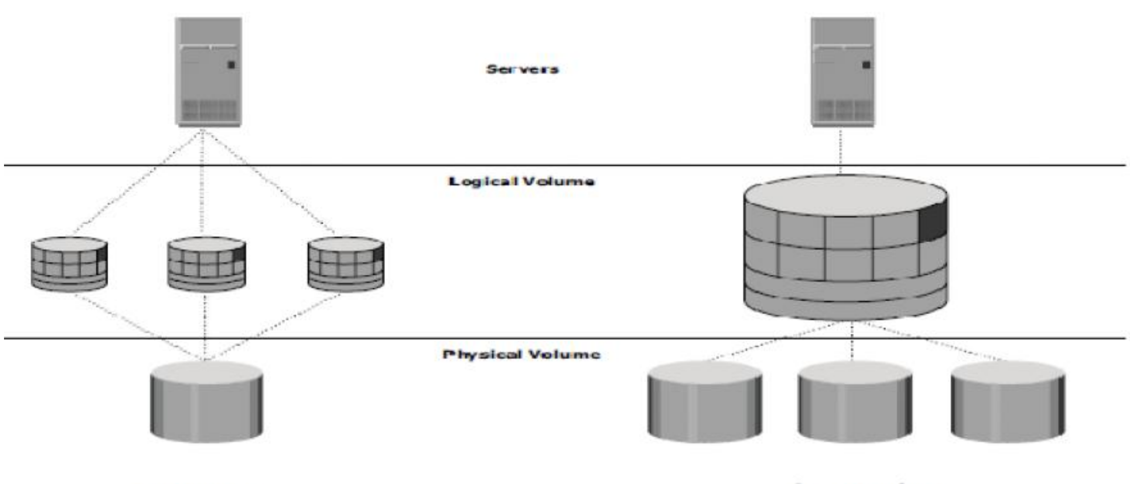



**Fig: Partitioning and concatenation in LVM**

**ii) File System**

A file is a collection of related records or data stored as a unit with a name. A file system is a hierarchical structure of files. File systems enable easy access to data files residing within a disk drive, a disk partition, or a logical volume. A file system needs host-based logical structures and software routines that control access to files. It provides users with the functionality to create, modify, delete, and access files. A file system organizes data in a structured hierarchical manner via the use of directories, which are containers for storing pointers to multiple files. All file systems maintain a pointer map to the directories, subdirectories, and files that are part of the file system. Some of the common file systems are as follows:

- FAT 32 (File Allocation Table) for Microsoft Windows
- NT File System (NTFS) for Microsoft Windows
- UNIX File System (UFS) for UNIX
- Extended File System (EXT2/3) for Linux

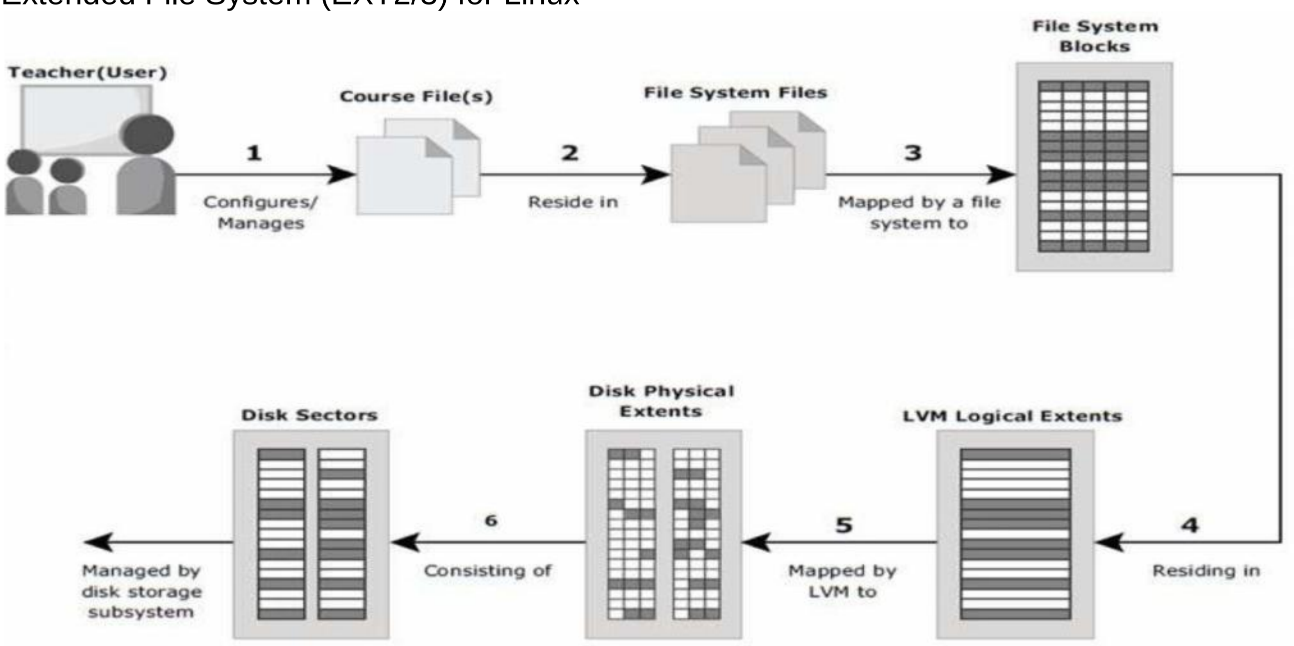



**Figure 2-12:** Process of mapping user files to disk storage

Figure 2-12 shows the following process of mapping user files to the disk storage subsystem with an LVM:

1. Files are created and managed by users and applications.
2. These files reside in the file systems.
3. The file systems are then mapped to units of data, or file system blocks.
4. The file system blocks are mapped to logical extents.
5. These in turn are mapped to disk physical extents either by the operating system or by the LVM.
6. These physical extents are mapped to the disk storage subsystem.