Seventh Semester B.E. Degree Examination, June/July 2019

Storage Area Networks

Time: 3 hrs. Max. Marks: 80

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. What is a data center? List the core components of data center. Explain the characteristics of data center. **(08 Marks)**

b. Discuss volume manager and compute virtualization in detail. **(08 Marks)**

OR

2 a. Differentiate between software and hardware RAID. Illustrate how parity method is used for RAID levels. **(08 Marks)**

b. With a neat diagram explain ISS. Explain in detail the cache component of ISS. **(08 Marks)**

1.3 Data Center Infrastructure

- Organizations maintain data centers to provide centralized data processing capabilities across the enterprise.
- The data center infrastructure includes computers, storage systems, network devices, dedicated power backups, and environmental controls (such as air conditioning and fire suppression).

1.3.1 Key Data Center Elements

Five core elements are essential for the basic functionality of a data center:

1) **Application:** An application is a computer program that provides the logic for computing operations. Eg: order processing system.
2) **Database:** More commonly, a database management system (DBMS) provides a structured way to store data in logically organized tables that are interrelated. A DBMS optimizes the storage and retrieval of data.
3) **Host or compute:** A computing platform (hardware, firmware, and software) that runs applications and databases.
4) **Network:** A data path that facilitates communication among various networked devices.
5) **Storage array:** A device that stores data persistently for subsequent use.

1.4 Key characteristics for Data Center Elements

Key characteristics of data center elements are:

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. 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 (eg: servers, new applications, and additional databases) or storage on demand, without interrupting business operations. 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.

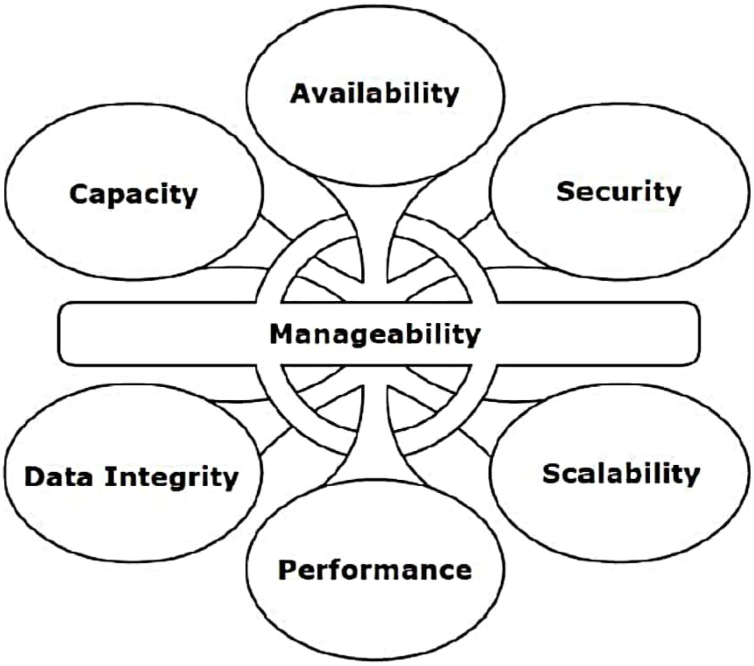


Fig 1.6: Key characteristics of data center elements

1b)

1.7.3.3 Volume Manager

- In the early days, disk drives appeared to the operating system as a number of continuous disk blocks. The entire disk drive would be allocated to the file system or other data entity used by the operating system or application.

  Disadvantages:

  - ✓ lack of flexibility.
  - ✓ When a disk drive ran out of space, there was no easy way to extend the file system's size.
  - ✓ as the storage capacity of the disk drive increased, allocating the entire disk drive for the file system often resulted in underutilization of storage capacity

  Solution: evolution of Logical Volume Managers (LVMs)

- LVM enabled dynamic extension of file system capacity and efficient storage management.

- The LVM is software that runs on the compute system and manages logical and physical storage.
- LVM is an intermediate layer between the file system and the physical disk.
- LVM can partition a larger-capacity disk into virtual, smaller-capacity volumes(called Partitioning) or aggregate several smaller disks to form a larger virtual volume. The process is called concatenation.
- Disk partitioning was introduced to improve the flexibility and utilization of disk drives.
- In partitioning, a disk drive is divided into logical containers called logical volumes (LVs) (see Fig 1.7)

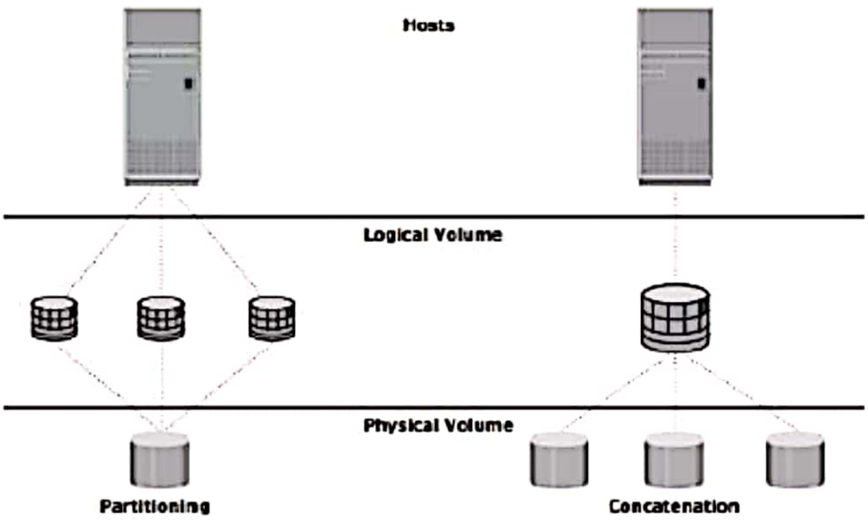


Fig 1.7: Disk Partitioning and concatenation

1.7.3.5 Compute Virtualization

- Compute virtualization is a technique for *masking* or *abstracting* the physical hardware from the operating system. It enables multiple operating systems to run concurrently on single or clustered physical machines.
- This technique enables creating portable virtual compute systems called *virtual machines* (VMs) running its own operating system and application instance in an isolated manner.
- Compute virtualization is achieved by a virtualization layer that resides between the hardware and virtual machines called the *hypervisor*. The hypervisor provides hardware resources, such as CPU, memory, and network to all the virtual machines.
- A virtual machine is a logical entity but appears like a physical host to the operating system, with its own CPU, memory, network controller, and disks. However, all VMs share the same underlying physical hardware in an isolated manner.
- Before Compute virtualization:
  - ✓ A physical server often faces resource-conflict issues when two or more applications running on the same server have conflicting requirements. As a result, only one application can be run on a server at a time, as shown in Fig 1.9 (a).
  - ✓ Due to this, organizations will need to purchase new physical machines for every application they deploy, resulting in expensive and inflexible infrastructure.
  - ✓ Many applications do not fully utilize complete hardware capabilities available to them. Resources such as processors, memory and storage remain underutilized.
  - ✓ Compute virtualization enables users to overcome these challenges (see Fig 1.9 (b)).

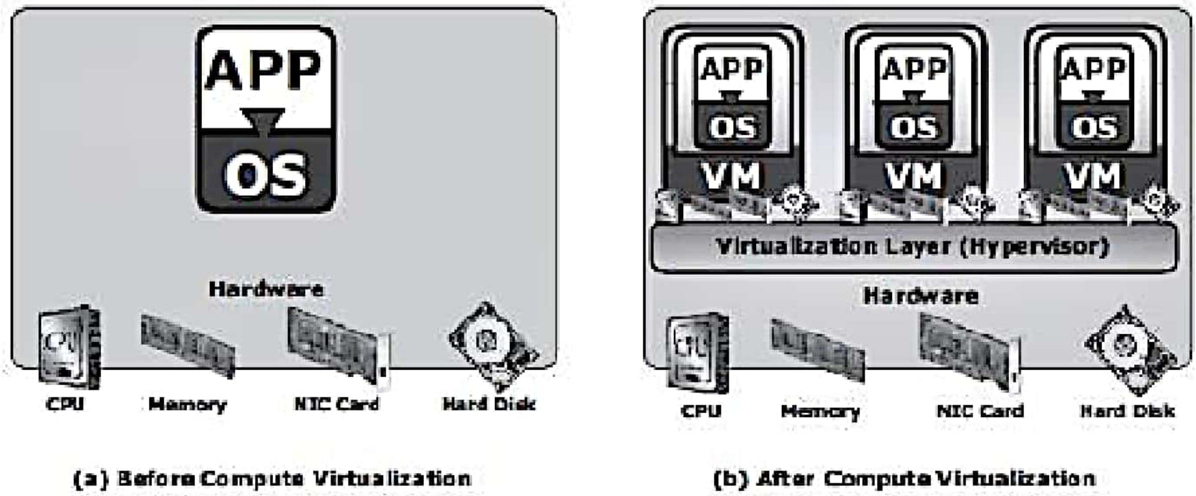


Fig 1.9: Server Virtualization

2a)

1.10 RAID Implementation Methods

- The two methods of RAID implementation are:
    1. Hardware RAID.
    2. Software RAID.

1.10.1 Hardware RAID

- In hardware RAID implementations, a specialized hardware controller is implemented either on the *host* or on the *array*.
- **Controller card RAID** is a *host-based hardware RAID* implementation in which a specialized RAID controller is installed in the host, and disk drives are connected to it.
- Manufacturers also integrate RAID controllers on motherboards.
- A host-based RAID controller is not an efficient solution in a data center environment with a large number of hosts.
- 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, and the host manages these volumes as physical drives.
- The key functions of the RAID controllers are as follows:
    - ✓ 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

1.10.2 Software RAID

- Software RAID uses host-based software to provide RAID functions.
- It is implemented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.
- Advantages when compared to Hardware RAID:
    - ✓ cost
    - ✓ simplicity benefits

- Limitations:
    - ✓ **Performance**: Software RAID affects overall system performance. This is due to additional CPU cycles required to perform RAID calculations.
    - ✓ **Supported features**: Software RAID does not support all RAID levels.

1.11.3 Parity

- Parity is a method to protect striped data from disk drive failure without the cost of mirroring.
- *An additional disk drive is added to hold parity*, a mathematical construct that allows re-creation of the missing data.
- Parity is a redundancy technique that ensures protection of data without maintaining a full set of duplicate data.
- Calculation of parity is a function of the RAID controller.
- Parity information can be stored on separate, dedicated disk drives or distributed across all the drives in a RAID set.
- Fig 1.13 shows a parity RAID set.

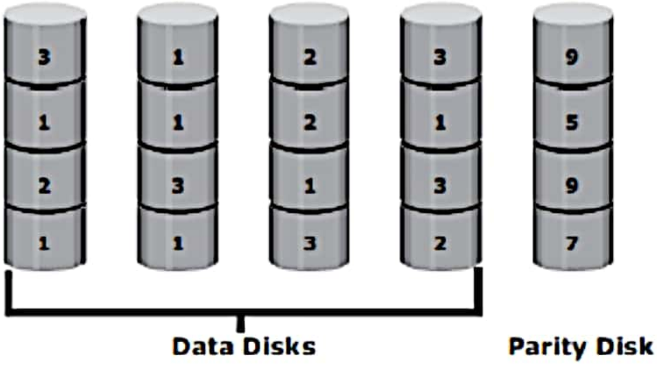


Fig 1.13: Parity RAID

- The first four disks, labeled "*Data Disks*," contain the data. The fifth disk, labeled "*Parity Disk*," stores the parity information, which, in this case, is the sum of the elements in each row.
- Now, if one of the data disks fails, the missing value can be calculated by subtracting the sum

2b)

1.14 Components of an Intelligent Storage System

- Intelligent Storage Systems are feature-rich RAID arrays that provide highly optimized I/O processing capabilities.
- These storage systems are configured with a large amount of memory (called *cache*) and multiple I/O paths and use sophisticated algorithms to meet the requirements of performance-sensitive applications.
- An intelligent storage system consists of four key components (Refer Fig 1.21):
  - ✓ Front End
  - ✓ Cache
  - ✓ Back end
  - ✓ Physical disks.
- 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.
- In modern intelligent storage systems, front end, cache, and back end are typically integrated on a single board (referred to as a storage processor or storage controller).

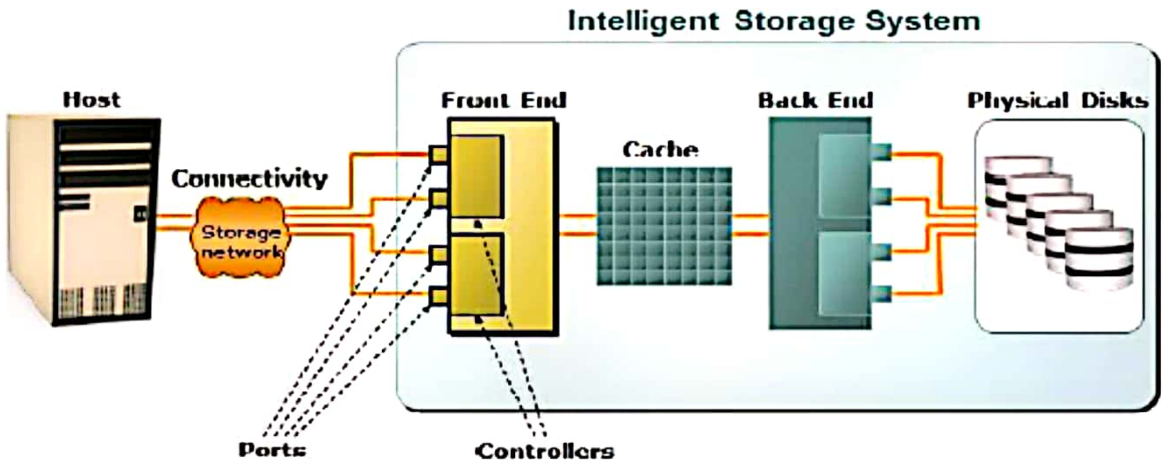


Fig 1.21 Components of an Intelligent Storage System

1.14.1 Front End

- The front end provides the interface between the storage system and the host.
- It consists of two components:
  i. Front-End Ports
  ii. Front-End Controllers.
- A front end has redundant controllers for high availability, and each controller contains multiple front-end ports that enable large numbers of hosts to connect to the intelligent storage system.
- Each front-end controller has processing logic that executes the appropriate transport protocol, such as Fibre Channel, iSCSI, FICON, or FCoE for storage connections.
- Front-end controllers route data to and from cache via the internal data bus.
- When the cache receives the write data, the controller sends an acknowledgment message back to the host.

1.14.2 Cache

- 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 rotating disks or hard disk drives (HDD).
- Rotating disks are the slowest component of an intelligent storage system. Data access on rotating disks usually takes several millisecond because of seek time and rotational latency.
- Accessing data from cache is fast and typically takes less than a millisecond.
- On intelligent arrays, write data is first placed in cache and then written to disk.

Seventh Semester B.E. Degree Examination, Dec.2018/Jan.2019
Storage Area Networks

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What is a data center? Explain key characteristics of data center elements with diagram. **(08 Marks)**

b. What is a file system? Explain the process of mapping user files to the disk storage. **(08 Marks)**

OR

2 a. What is RAID? Explain the RAID levels with reference to nested RAID, RAID3, RAID5 with neat diagram. **(08 Marks)**

b. With neat diagram, explain the structure of read and write operations with cache. **(08 Marks)**

1B)

1.7.3.4 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.
- A file system enables easy access to data files residing within a disk drive, a disk partition, or a logical volume.
- It provides users with the functionality to create, modify, delete, and access files.
- Access to files on the disks is controlled by the permissions assigned to the file by the owner, which are also maintained by the file system.
- 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.
- Examples of common file systems are:
  - ✓ 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
- The file system also includes a number of other related records, which are collectively called **the metadata.**
- For example, the metadata in a UNIX environment consists of the **superblock, the inodes, and the list of data blocks free and in use.**
- A superblock contains important information about the file system, such as the file system type, creation and modification dates, size, and layout.

- The following list shows the process of mapping user files to the disk storage subsystem with an LVM (see Fig 1.8)
    1. Files are created and managed by users and applications.
    2. These files reside in the file systems.
    3. The file systems are mapped to file system blocks.
    4. The file system blocks are mapped to logical extents of a logical volume.
    5. These logical extents in turn are mapped to the disk physical extents either by the operating system or by the LVM.
    6. These physical extents are mapped to the disk sectors in a storage subsystem.

If there is no LVM, then there are no logical extents. Without LVM, file system blocks are directly mapped to disk sectors.

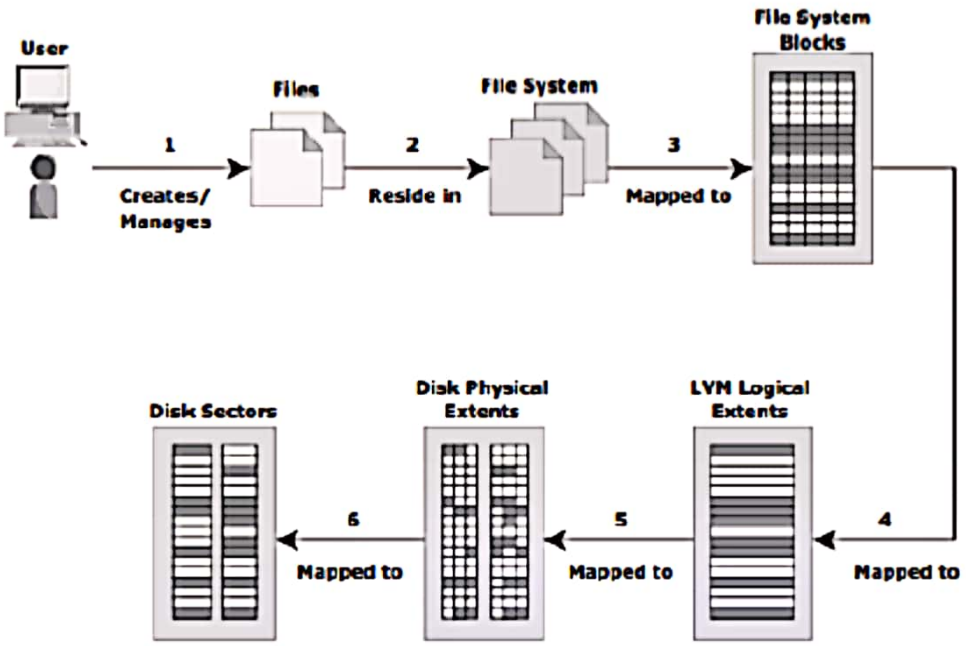


Fig 1.8: Process of mapping user files to disk storage

2A)

1.12 RAID Levels

- RAID Level selection is determined by below factors:
    - ✓ Application performance
    - ✓ data availability requirements
    - ✓ cost
- RAID Levels are defined on the basis of:
    - ✓ Striping
    - ✓ Mirroring
    - ✓ Parity techniques
- Some RAID levels use a single technique whereas others use a combination of techniques.
- Table 1.2 shows the commonly used RAID levels

Table 1.2: RAID Levels

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

1.12.3 Nested RAID

- Most data centers require data redundancy and performance from their RAID arrays.
- RAID 1+0 and RAID 0+1 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 Fig 1.16).

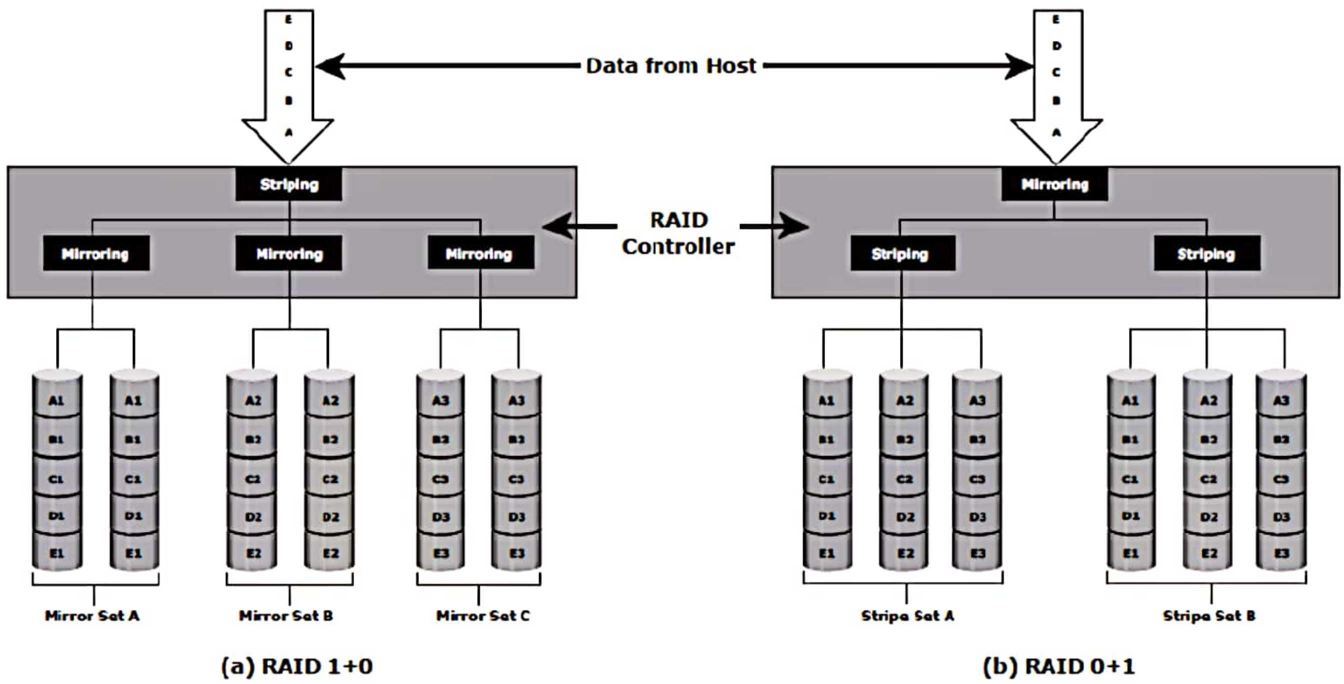


Fig 1.16: Nested RAID

1.12.4 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.
- 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 in a stripe.
- 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.
- Fig 1.17 shows the RAID 3 implementation

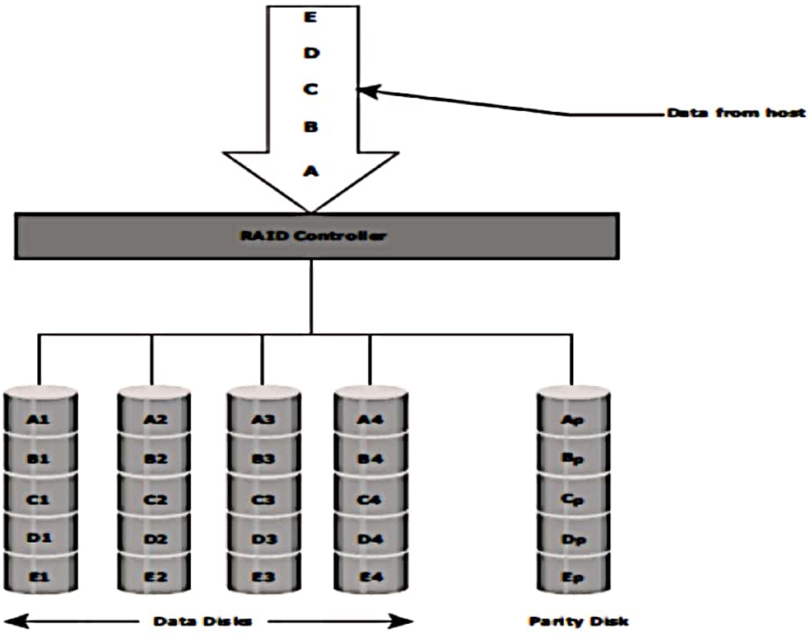


Fig 1.17: RAID 3

1.12.6 RAID 5

- RAID 5 is a versatile RAID implementation.
- It is similar to RAID 4 because it uses striping. The drives (strips) are also 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. Below Figure illustrates the RAID 5 implementation.
- Fig 1.18 illustrates the RAID 5 implementation.
- RAID 5 is good for random, read-intensive I/O applications and preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations, in which database administrators (DBAs) optimize data access.

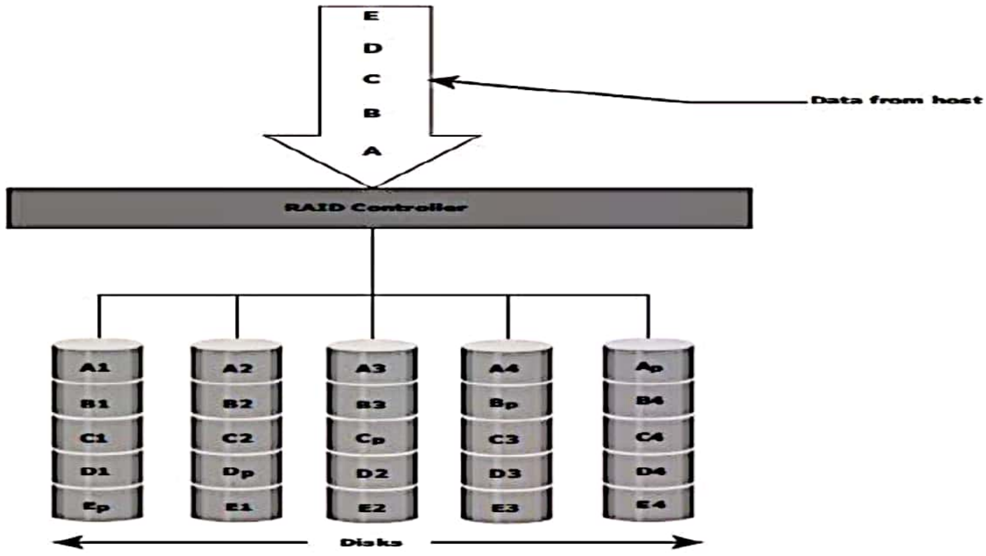
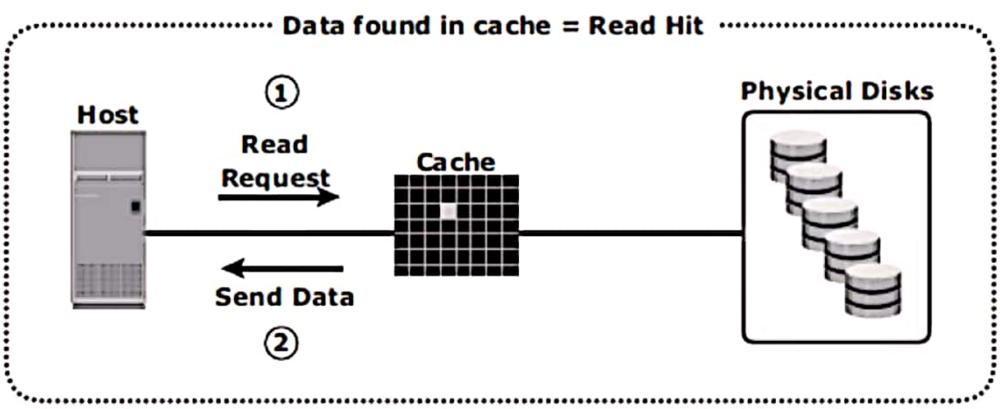


Fig 1.18: RAID 5

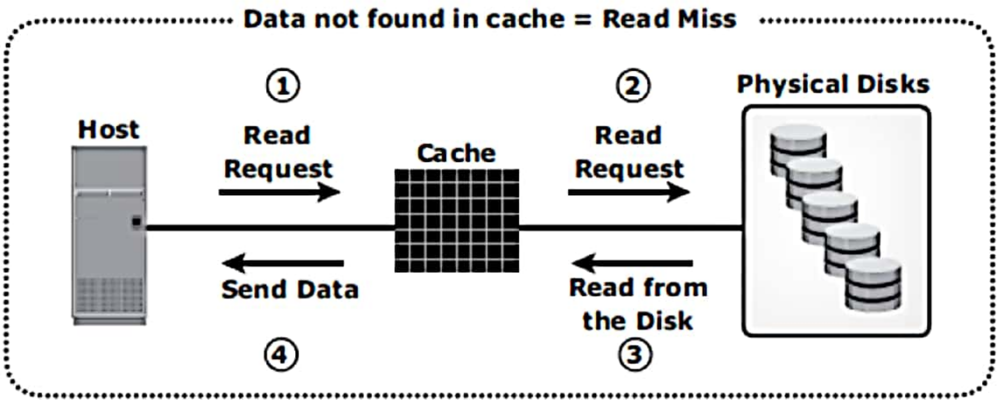
2B)

Read Operation with Cache

- When a host issues a read request, the storage controller reads 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 Fig 1.23[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. 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



(a)



(b)

Fig 1.23 : 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 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,

Seventh Semester B.E. Degree Examination, Dec.2019/Jan.2020

Storage Area Networks

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1	a.	Explain with neat diagram the Evolution of storage Architecture.	(06 Marks)
	b.	Discuss core Elements of Data center and key characteristics of Data center.	(10 Marks)

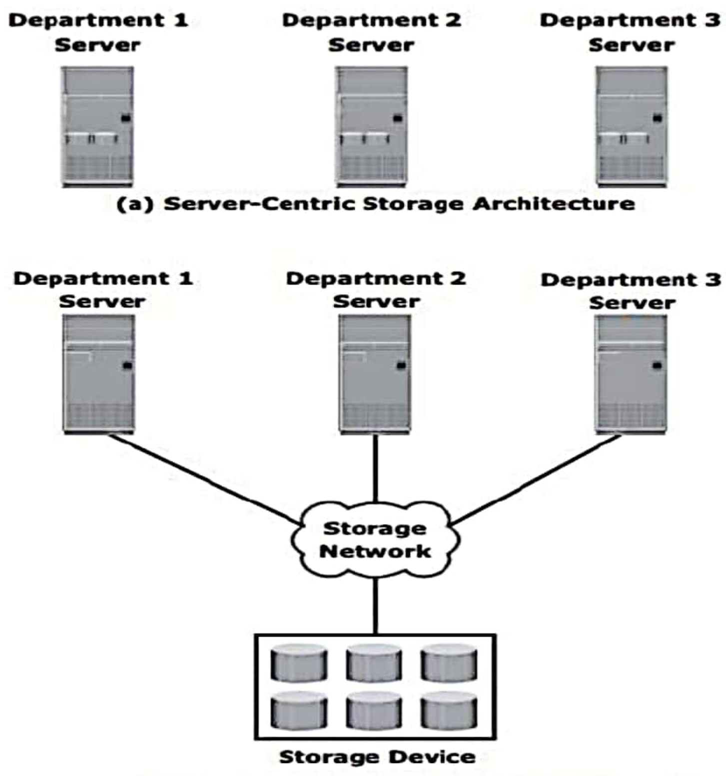
OR

2	a.	Describe with neat block diagram the components of Intelligent storage system.	(08 Marks)
	b.	With diagram explain different RAID Techniques.	(08 Marks)

1A)

1.2 Introduction to Evolution of Storage Architecture

- Historically, organizations had centralized computers (mainframe) and information storage devices (tape reels and disk packs) in their data center.
- The evolution of open systems and the affordability and ease of deployment that they offer made it possible for business units/departments to have their own servers and storage.
- In earlier implementations of open systems, the storage was typically internal to the server. This approach is referred to as **server-centric storage architecture** (see Fig 1.4 [a]).
- In this server-centric storage architecture, each server has a limited number of storage devices, and any administrative tasks, such as maintenance of the server or increasing storage capacity, might result in unavailability of information.
- The rapid increase in the number of departmental servers in an enterprise resulted in



(b) Information-Centric Storage Architecture

- In information-centric architecture, storage devices are managed centrally and independent of servers.
- These centrally-managed storage devices are shared with multiple servers.
- When a new server is deployed in the environment, storage is assigned from the same shared storage devices to that server.
- The capacity of shared storage can be increased dynamically by adding more storage devices without impacting information availability.

2B)

1.11 RAID Techniques

- There are three RAID techniques
    1. striping
    2. mirroring
    3. parity

1.11.1 Striping

- **Striping** is a technique to spread data across multiple drives (more than one) to use the drives in parallel.
- All the read-write heads work simultaneously, allowing more data to be processed in a shorter time and increasing performance, compared to reading and writing from a single disk.
- Within each disk in a RAID set, a **predefined number of contiguously addressable** disk blocks are defined as a **strip**.
- The set of aligned strips that spans across all the disks within the RAID set is called a **stripe.**

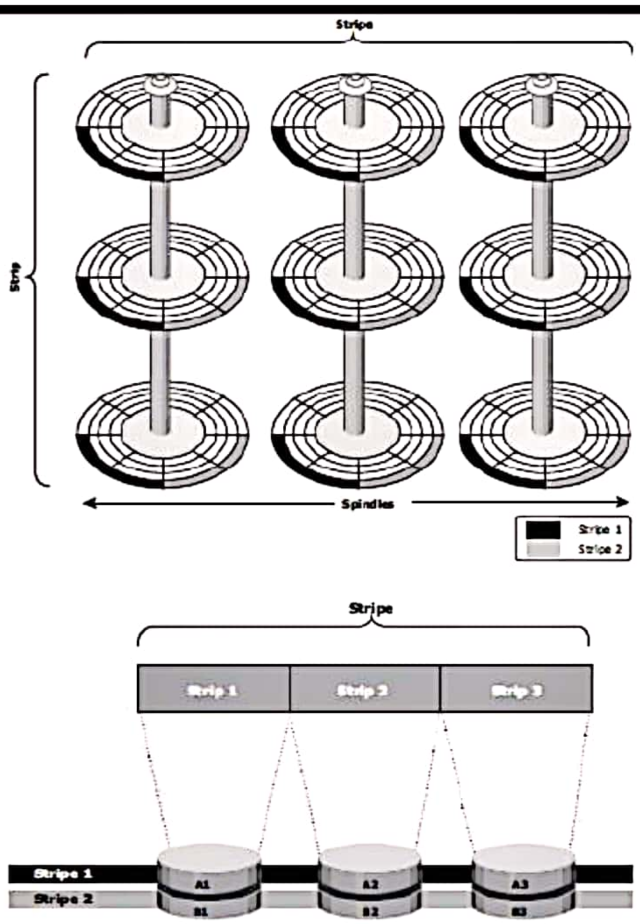


Fig 1.11: Striped RAID set

1.11.2 Mirroring

- **Mirroring** is a technique whereby the same data is stored on two different disk drives, yielding two copies of the data.
- If one disk drive failure occurs, the data is intact on the surviving disk drive (see Fig 1.12) and the controller continues to service the host's data requests from the surviving disk of a mirrored pair.
- 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.

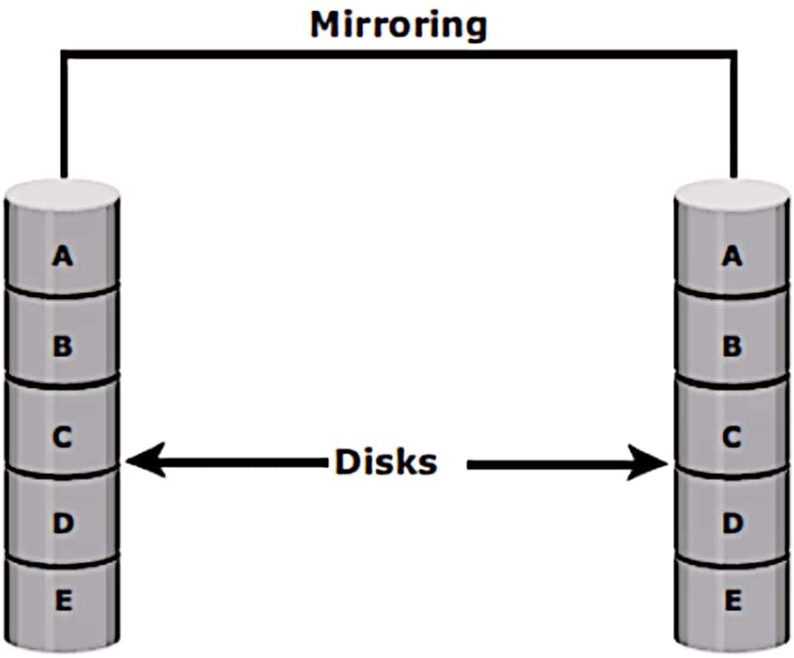


Fig 1.12: Mirrored disks in an array

3.Parity method REPEATEd-page no 6