Module 1

STORAGE SYSTEM

Introduction to Information storage

Why Information management?

- Information is increasingly important in our daily lives. We have become information Dependents.
- We live in on-command, on-demand world that means we need information when and where it is required.
- We access the Internet every day to perform searches, participate in social networking, send and receive e-mails, share pictures and videos, and scores of other applications. Equipped with a growing number of content-generating devices, more information is being created by individuals than by businesses.
- The importance, dependency, and volume of information for the business world also continue to grow at astounding rates.
- Businesses depend on fast and reliable access to information critical to their success. Some of the business applications that process information include airline reservations, telephone billing systems, e-commerce, ATMs, product designs, inventory management, e-mail archives, Web portals, patient records, credit cards, life sciences, and global capital markets.
- The increasing criticality of information to the businesses has amplified the challenges in protecting and managing the data.
- Organizations maintain one or more data centers to store and manage information. A data center is a facility that contains information storage and other physical information technology (IT) resources for computing, networking, and storing information.

1.1 Information Storage

Organizations process data to derive the information required for their day-today operations. Storage is a repository that enables users to store and retrieve this digital data.

1.1.1 Data

- Data is a collection of raw facts from which conclusions may be drawn.

- Eg: a printed book, a family photograph, a movie on videotape, e-mail message, an e-book, a bitmapped image, or a digital movie are all examples of data.

- The data can be generated using a computer and stored in strings of 0s and 1s(as shown in Fig 1.1), is called digital data and is accessible by the user only after it is processed by a computer.

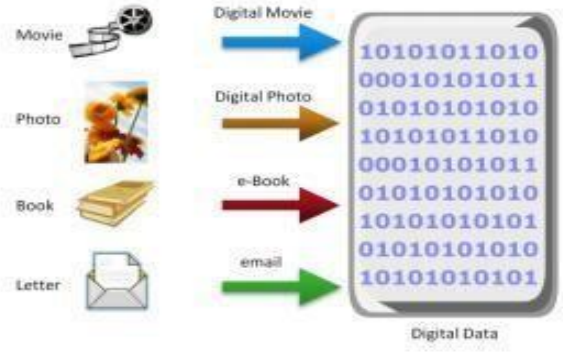


Fig 1.1: Digital data

The following is a list of some of the factors that have contributed to the growth of digital data:

1. **Increase in data processing capabilities**: Modern-day computers provide a significant increase in processing and storage capabilities. This enables the conversion of various types of content and media from conventional forms to digital formats.

2. **Lower cost of digital storage:** Technological advances and decrease in the cost of storage devices have provided low-cost solutions and encouraged the development of less expensive data storage devices. This cost benefit has increased the rate at which data is being generated and stored.

3. **Affordable and faster communication technology:** The rate of sharing digital data is now much faster than traditional approaches. A handwritten letter may take a week to reach its destination, whereas it only takes a few seconds for an e-mail message to reach its recipient.

4. **Proliferation of applications and smart devices:** Smartphones, tablets, and newer digital devices, along with smart applications, have significantly contributed to the generation of digital content.

1.1.2 Types of Data

Data can be classified as structured or unstructured (see Fig 1.2) based on how it is stored and managed.

- **Structured data:**
  - Structured data is organized in rows and columns in a rigidly defined format so that applications can retrieve and process it efficiently.
  - Structured data is typically stored using a database management system (DBMS).
- **Unstructured data:**
  - Data is unstructured if its elements cannot be stored in rows and columns, and is therefore difficult to query and retrieve by business applications.
  - Example: e-mail messages, business cards, or even digital format files such as .doc, .txt, and .pdf.

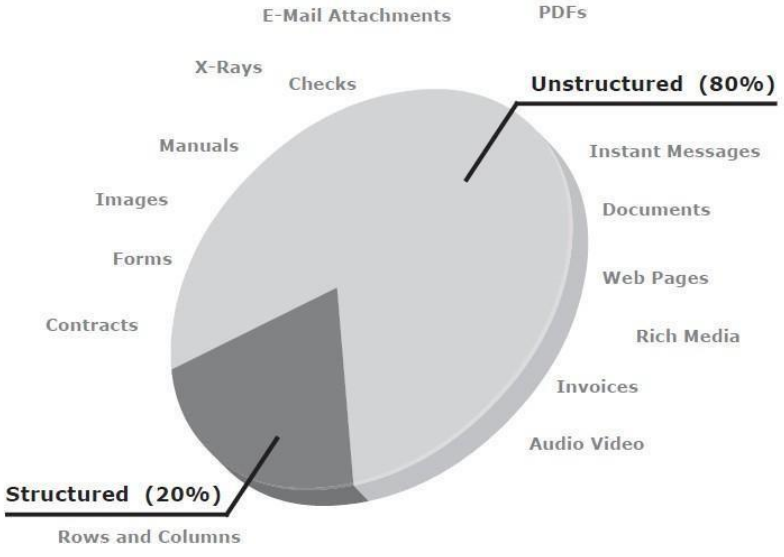


Fig 1.2: Types of data

1.1.3 Big Data

- Big data refers to data sets whose sizes are beyond the capability of commonly used software tools to capture, store, manage, and process within acceptable time limits.

- It includes both structured and unstructured data generated by a variety of sources, including business application transactions, web pages, videos, images, e-mails, social media, and so on.
- These data sets typically require real-time capture or updates for analysis, predictive modeling, and decision making.
- The big data ecosystem (see Fig 1.3) consists of the following:
  1. Devices that collect data from multiple locations and also generate new data about this data (metadata).
  2. Data collectors who gather data from devices and users.
  3. Data aggregators that compile the collected data to extract meaningful information.
  4. Data users and buyers who benefit from the information collected and aggregated by others in the data value chain.

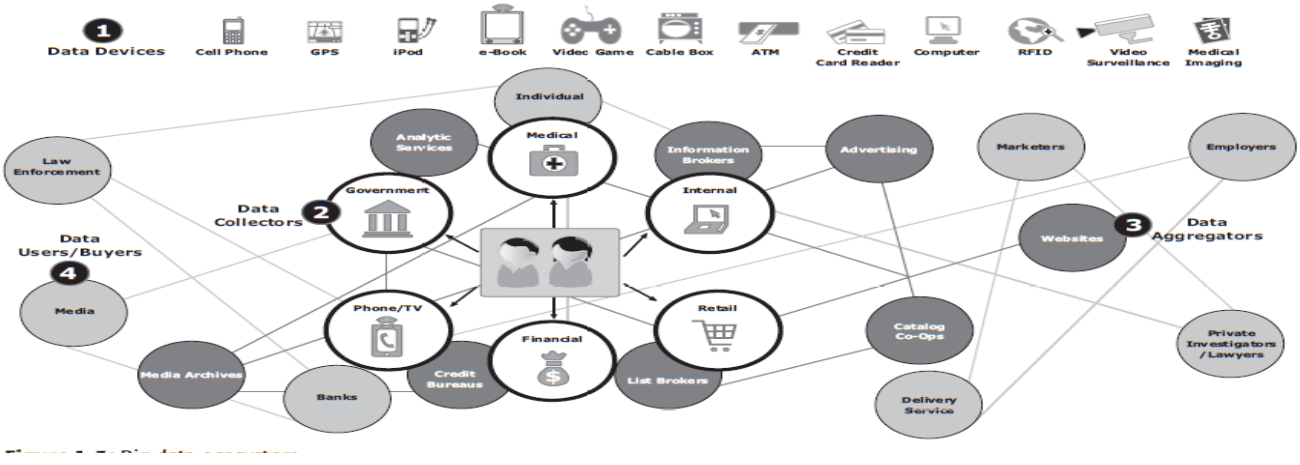


Fig 1.3: Big data Ecosystem

- Big data Analysis in real time requires new techniques, architectures, and tools that provide:
  1. high performance,
  2. massively parallel processing (MPP) data platforms,
  3. advanced analytics on the data sets.
- Big data Analytics provide an opportunity to translate large volumes of data into right decisions.

1.1.4 Information

- Data, whether structured or unstructured, does not fulfil any purpose for individuals or businesses unless it is presented in a meaningful form.

- Information is the intelligence and knowledge derived from data.
- Businesses analyze raw data in order to identify meaningful trends. On the basis of these trends, a company can plan or modify its strategy.
- For example, a retailer identifies customers' preferred products and brand names by analyzing their purchase patterns and maintaining an inventory of those products.
- Effective data analysis not only extends its benefits to existing businesses, but also creates the potential for new business opportunities by using the information in creative ways.

1.1.5 Storage

- Data created by individuals or businesses must be stored so that it is easily accessible for further processing.
- In a computing environment, devices designed for storing data are termed storage devices or simply storage.
- The type of storage used varies based on the type of data and the rate at which it is created and used.
  - Devices such as memory in a cell phone or digital camera, DVDs, CD-ROMs, and hard disks in personal computers are examples of storage devices.
- Businesses have several options available for storing data including internal hard disks, external disk arrays and tapes.

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

unprotected, unmanaged, fragmented islands of information and increased capital and operating expenses.

- To overcome these challenges, storage evolved from **server-centric to information-centric architecture** (see Fig 1.4 [b]).

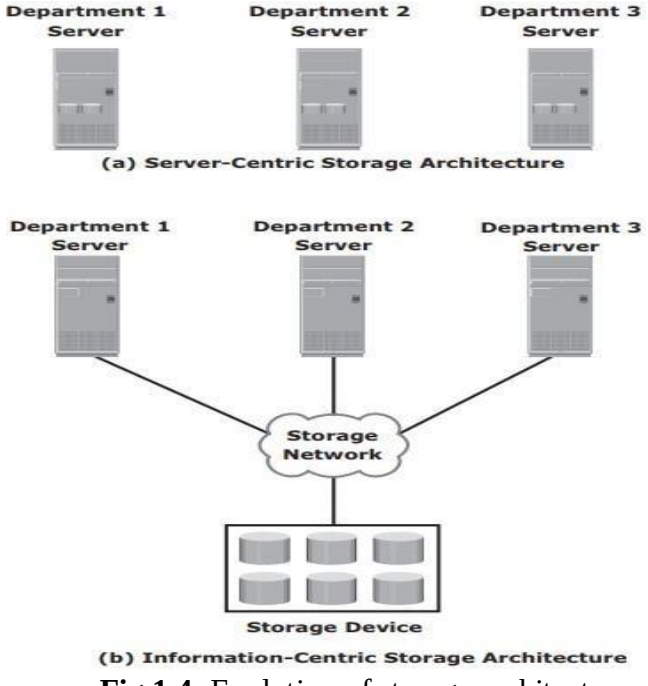


Fig 1.4: Evolution of storage architecture

- 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.
- In this architecture, information management is easier and cost-effective.
- Storage technology and architecture continues to evolve, which enables organizations to

consolidate, protect, optimize, and leverage their data to achieve the highest return on information assets.

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 Core Elements of Data Center

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.

- These core elements are typically viewed and managed as separate entities, but all the elements must work together to address data processing requirements.
- Fig 1.5 shows an example of an order processing system that involves the five core elements of a data center and illustrates their functionality in a business process.
    1) A customer places an order through a client machine connected over a LAN/ WAN to a host running an order-processing application.
    2) The client accesses the DBMS on the host through the application to provide order-related information, such as the customer name, address, payment method, products ordered, and quantity ordered.

3) The DBMS uses the host operating system to write this data to the database located on physical disks in the storage array.
4) The Storage Network provides the communication link between the host and the storage array and transports the request to read or write commands between them.
5) The storage array, after receiving the read or write request from the host, performs the necessary operations to store the data on physical disks.

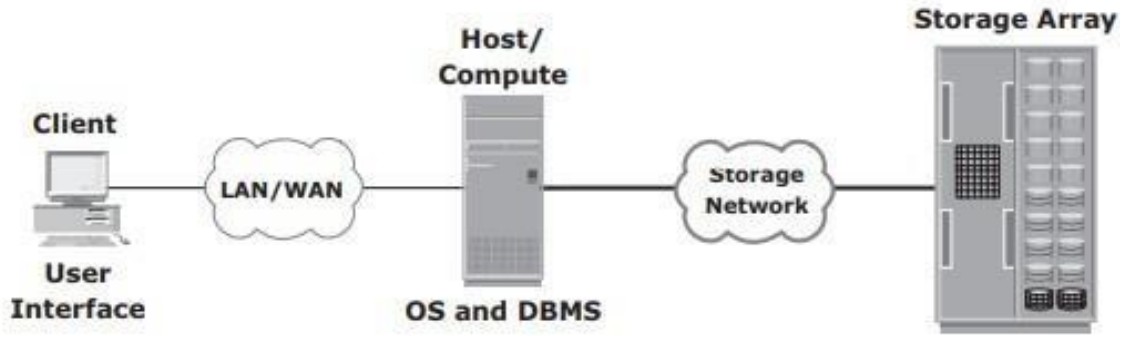


Fig 1.5: Example of online order transaction system

Fig 1.5: Example of an online order transaction system

1.3.2 Key characteristics of data center

Key characteristics of data center elements are:

1) **Availability:** A data center should ensure the availability of information when required. Unavailability of information could cost millions of dollars per hour to businesses, such as financial services, telecommunications, and e-commerce.
2) **Security:** Data centers must establish policies, procedures, and core element integration to prevent unauthorized access to information.
3) **Scalability:** Business growth often requires deploying more servers, new applications, and additional databases. Data center resources should scale based on requirements, without interrupting business operations
4) **Performance**: All the elements of the data center should provide optimal performance based on the required service levels.

5) **Data integrity:** Data integrity refers to mechanisms, such as error correction codes or parity bits, which ensure that data is stored and retrieved exactly as it was received.
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 provide additional capacity without interrupting availability or with minimal disruption. Capacity may be managed by reallocating the existing resources or by adding new resources.
7) **Manageability:** A data center should provide easy and integrated management of all its elements. Manageability can be achieved through automation and reduction of human (manual) intervention in common tasks.

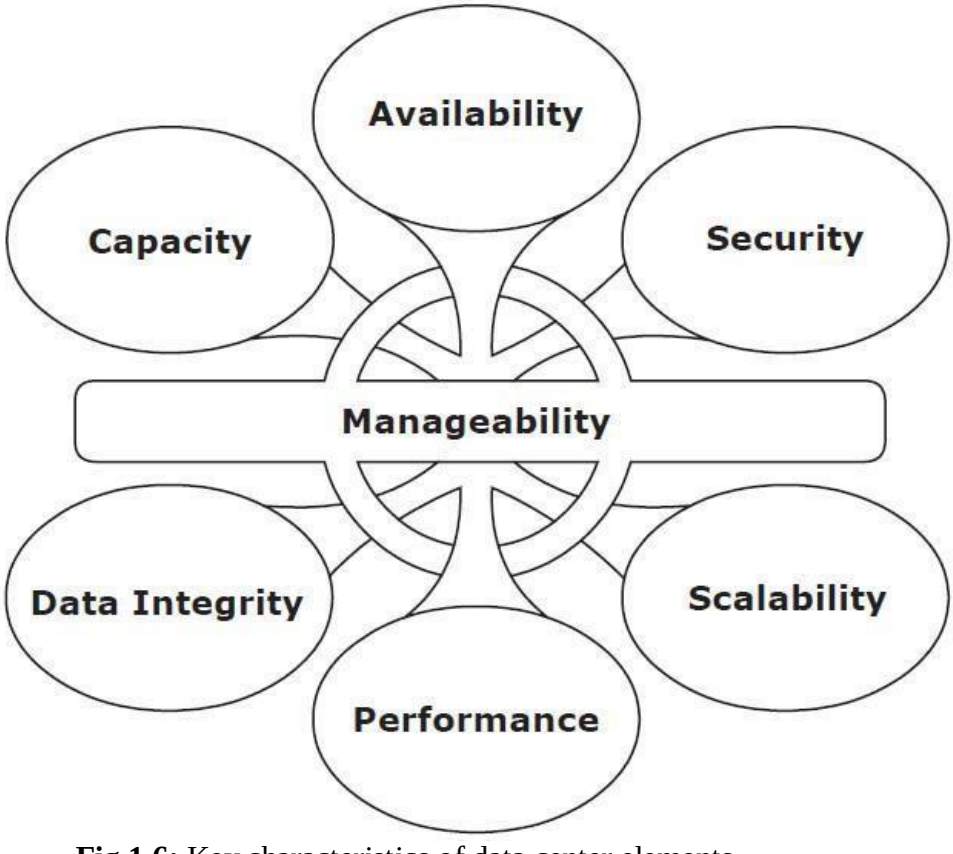


Fig 1.6: Key characteristics of data center elements

1.3.3 Managing a Data Center

Managing a data center involves many tasks. The key management activities include the following:

- **Monitoring:** It is a continuous process of gathering information on various elements and services running in a data center. The aspects of a data center that are monitored include security, performance, availability, and capacity

- **Reporting:** It is done periodically on resource performance, capacity, and utilization. Reporting tasks help to establish business justifications and chargeback of costs associated with data center operations.

- **Provisioning:** It is a process of providing the hardware, software, and other resources required to run a data center. Provisioning activities primarily include resources management to meet capacity, availability, performance, and security requirements.

1.4 Virtualization and Cloud Computing

Virtualization

- Virtualization is a technique of abstracting physical resources, such as compute, storage, and network, and making them appear as logical resources.
- Virtualization has existed in the IT industry for several years and in different forms.
- Common examples of virtualization are virtual memory used on compute systems and partitioning of raw disks.
- Virtualization enables pooling of physical resources and providing an aggregated view of the physical resource capabilities. For example, storage virtualization enables multiple pooled storage devices to appear as a single large storage entity.
- Similarly, by using compute virtualization, the CPU capacity of the pooled physical servers can be viewed as the aggregation of the power of all CPUs (in megahertz).
- Virtualization also enables centralized management of pooled resources.
- Virtual resources can be created and provisioned from the pooled physical resources. For example, a virtual disk of a given capacity can be created from a storage pool or a virtual server with specific CPU power and memory can be configured from a compute pool.
- These virtual resources share pooled physical resources, which improves the utilization of physical IT resources.
- Based on business requirements, capacity can be added to or removed from the virtual

resources without any disruption to applications or users.

- With improved utilization of IT assets, organizations save the costs associated management of new physical resources. Moreover, fewer physical resources means less space and energy, which leads to better economics and green computing.

Cloud Computing

- Cloud computing enables individuals or businesses to use IT resources as a service over the network.
- It provides highly scalable and flexible computing that enables provisioning of resources on demand.
- Users can scale up or scale down the demand of computing resources, including storage capacity, with minimal management effort or service provider interaction.
- Cloud computing empowers self-service requesting through a fully automated request-fulfillment process.
- Cloud computing enables consumption-based metering; therefore, consumers pay only for the resources they use, such as CPU hours used, amount of data transferred, and gigabytes of data stored.
- Cloud infrastructure is usually built upon virtualized data centers, which provide resource pooling and rapid provisioning of resources.

CHAPTER 2: Data Center Environment

2.1 Application

- An application is a computer program that provides the logic for computing operations.
- The application sends requests to the underlying operating system to perform read/write (R/W) operations on the storage devices.
- Applications deployed in a data center environment are commonly categorized as business applications, infrastructure management applications, data protection applications, and security applications.
- Some examples of these applications are e-mail, enterprise resource planning (ERP), decision support system (DSS), resource management, backup, authentication and antivirus applications, and so on

2.2 DBMS

- A database is a structured way to store data in logically organized tables that are interrelated.
- A DBMS controls the creation, maintenance, and use of a database.
- The DBMS processes an application's request for data and instructs the operating system to transfer the appropriate data from the storage.

2.3 Host(or) Compute

- The computers on which applications run are referred to as hosts. Hosts can range from simple laptops to complex clusters of servers.
- Hosts can be physical or virtual machines.
- A compute virtualization software enables creating virtual machines on top of a physical

compute infrastructure.

- A host consists of
  - ✓ CPU: The CPU consists of four components-Arithmetic Logic Unit (ALU), control unit, registers, and L1 cache
  - ✓ Memory: There are two types of memory on a host, Random Access Memory (RAM) and Read-Only Memory (ROM)
  - ✓ I/O devices : keyboard, mouse, monitor
  - ✓ a collection of software to perform computing operations- This software includes the operating system, file system, logical volume manager, device drivers, and so on.

The following section details various software components that are essential parts of a host system.

2.3.1 Operating System

- In a traditional computing environment, an operating system controls all aspects of computing.
- It works between the application and the physical components of a compute system.
- In a virtualized compute environment, the virtualization layer works between the operating system and the hardware resources.

Functions of OS

- data access
- monitors and responds to user actions and the environment
- organizes and controls hardware components
- manages the allocation of hardware resources
- It provides basic security for the access and usage of all managed resources
- performs basic storage management tasks
- manages the file system, volume manager, and device drivers.

Memory Virtualization

- Memory has been, and continues to be, an expensive component of a host.
- It determines both the size and number of applications that can run on a host.
- Memory virtualization is an operating system feature that virtualizes the physical memory

(RAM) of a host.

- It creates virtual memory with an address space larger than the physical memory space present in the compute system.
- The operating system utility that manages the virtual memory is known as the virtual memory manager (VMM).
- The space used by the VMM on the disk is known as a swap space.
- A swap space (also known as page file or swap file) is a portion of the disk drive that appears to be physical memory to the operating system.
- In a virtual memory implementation, the memory of a system is divided into contiguous blocks of fixed-size pages.
- A process known as paging moves inactive physical memory pages onto the swap file and brings them back to the physical memory when required.

2.3.2 Device Drivers

- A device driver is special software that permits the operating system to interact with a specific device, such as a printer, a mouse, or a disk drive.
- A device driver enables the operating system to recognize the device and to access and control devices.
- Device drivers are hardware-dependent and operating-system-specific.

2.3.3 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 2.1)

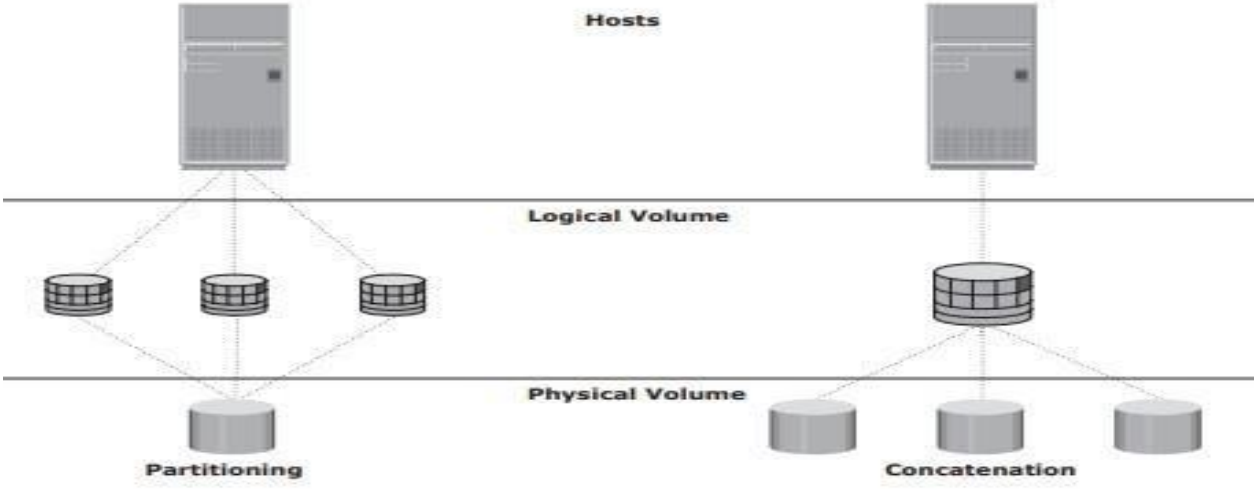


Fig 2.1: Disk Partitioning and concatenation

- Concatenation is the process of grouping several physical drives and presenting them to the host as one big logical volume.
- The basic LVM components are **physical volumes, volume groups, and logical volumes**.
- Each physical disk connected to the host system is a **physical volume (PV).**
- 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. 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 disk partition, whereas the volume group itself can be thought of as a disk.

2.3.4 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.
- An inode is associated with every file and directory and contains information such as the file length, ownership, access privileges, time of last access/modification, number of links, and the address of the data.
- A file system block is the smallest “unit” allocated for storing data.

- The following list shows the process of mapping user files to the disk storage subsystem with an LVM (see Fig 2.2)

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 theoperating 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.

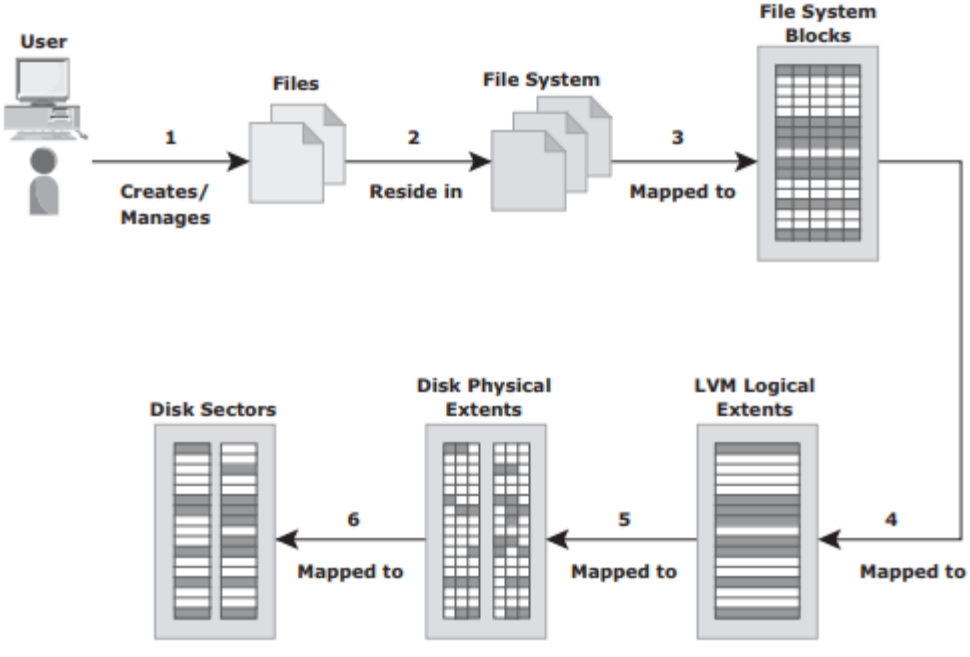


Fig 2.2: Process of mapping user files to disk storage

- The file system tree starts with the root directory. The root directory has a number of subdirectories.
- A file system can be either:
  - ✓ a journaling file system
  - ✓ a nonjournaling file system.

Nonjournaling file system: Nonjournaling file systems cause a potential loss of files because they use separate writes to update their data and metadata. If the system crashes during the write process, the metadata or data might be lost or corrupted. When the system reboots, the file system attempts to update the metadata structures by examining and repairing them. This operation takes a long time on large file systems. If there is insufficient information to re-create the wanted or original structure, the files might be misplaced or lost, resulting in corrupted file systems.

Journaling file system: Journaling File System uses a separate area called a *log* or *journal*. This journal might contain all the data to be written (physical journal) or just the metadata to be updated (logical journal). Before changes are made to the file system, they are written to this separate area. After the journal has been updated, the operation on the file system can be performed. If the system crashes during the operation, there is enough information in the log to "*replay*" the log record and complete the operation. Nearly all file system implementations today use journaling

Advantages:

- Journaling results in a quick file system check because it looks only at the active, most recently accessed parts of a large file system.
- Since information about the pending operation is saved, the risk of files being lost is reduced.

Disadvantage:

- they are slower than other file systems. This slowdown is the result of the extra operations that have to be performed on the journal each time the file system is changed.
- But the advantages of lesser time for file system checks and maintaining file system integrity far outweighs its disadvantage.

2.3.5 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 2.3 (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 2.3 (b)).

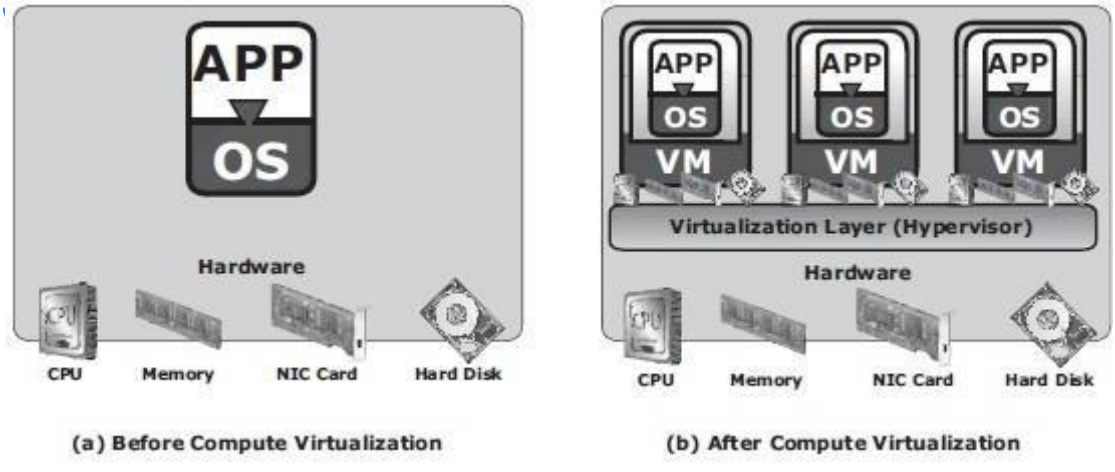


Fig 2.3: Server Virtualization

- After Compute virtualization:
  - ✓ This technique significantly improves server utilization and provides server consolidation.
  - ✓ *Server consolidation* enables organizations to run their data center with fewer physical servers.
  - ✓ This, in turn,

- reduces cost of new server acquisition,
- reduces operational cost,
- saves data center floor and rack space.

✓ Individual VMs can be restarted, upgraded, or even crashed, without affecting the other VMs.

✓ VMs can be copied or moved from one physical machine to another (non-disruptive migration) without causing application downtime. Nondisruptive migration of VMs is required for load balancing among physical machines, hardware maintenance, and availability purposes.

2.4 Connectivity

- Connectivity refers to the interconnection between hosts or between a host and peripheral devices, such as printers or storage devices.
- Connectivity and communication between host and storage are enabled using:
  - ✓ physical components
  - ✓ interface protocols.

2.4.1 Physical Components of Connectivity

- The physical components of connectivity are the hardware elements that connect the host to storage.
- Three physical components of connectivity between the host and storage are (refer Fig 2.4):
  - ✓ the host interface device
  - ✓ port
  - ✓ cable.

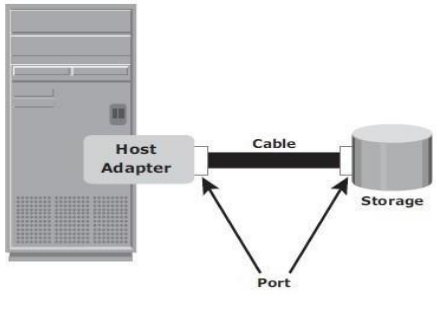


Fig 2.4: Physical components of connectivity

- A *host interface device* or *host adapter* connects a host to other hosts and storage devices.
  - ✓ Eg: host bus adapter (HBA) and network interface card (NIC).
  - ✓ HBA is an application-specific integrated circuit (ASIC) board that performs I/O interface functions between the host and storage, relieving the CPU from additional I/O processing workload.
  - ✓ A host typically contains multiple HBAs.
- A *port* is a specialized outlet that enables connectivity between the host and external devices. An HBA may contain one or more ports to connect the host.
- *Cables* connect hosts to internal or external devices using copper or fiber optic media.

2.4.2 Interface Protocols

- A protocol enables communication between the host and storage.
- Protocols are implemented using interface devices (or controllers) at both source and destination.
- The popular interface protocols used for host to storage communications are:
  i. Integrated Device Electronics/Advanced Technology Attachment (IDE/ATA)
  ii. Small Computer System Interface (SCSI),
  iii. Fibre Channel (FC)
  iv. Internet Protocol (IP)

IDE/ATA and Serial ATA:

- **IDE/ATA** is a popular interface protocol standard used for connecting storage devices, such as disk drives and CD-ROM drives.
- This protocol supports parallel transmission and therefore is also known as *Parallel ATA (PATA*) or simply ATA.
- IDE/ATA has a variety of standards and names.
- The Ultra DMA/133 version of ATA supports a throughput of **133 MB per second**.
- In a master-slave configuration, an ATA interface supports two storage devices per connector.
- If performance of the drive is important, sharing a port between two devices is not

recommended.

- The serial version of this protocol is known as Serial ATA (SATA) and supports single bit serial transmission.
- *High performance* and *low cost* SATA has replaced PATA in newer systems.
- SATA revision 3.0 provides a data transfer rate up to **6 Gb/s.**

SCSI and Serial SCSI:

- **SCSI** has emerged as a preferred connectivity protocol in high-end computers.
- This protocol supports parallel transmission and offers improved **performance**, **scalability,** and **compatibility** compared to ATA.
- The high cost associated with SCSI limits its popularity among home or personal desktop users.
- SCSI supports up to 16 devices on a single bus and provides data transfer rates up to **640 MB/s.**
- **Serial attached SCSI (SAS)** is a point-to-point serial protocol that provides an alternative to parallel SCSI.
- A newer version of serial SCSI (SAS 2.0) supports a data transfer rate up to **6 Gb/s.**

Fibre Channel (FC):

- **Fibre Channel** is a widely used protocol for high-speed communication to the storage device.
- Fibre Channel interface provides gigabit network speed.
- It provides a serial data transmission that operates over copper wire and optical fiber.
- The latest version of the FC interface (16FC) allows transmission of data up to **16 Gb/s.**

Internet Protocol (IP):

- IP is a network protocol that has been traditionally used for **host-to-host traffic**.
- With the emergence of new technologies, an IP network has become a viable option for host-to-storage communication.
- IP offers several advantages:
  - ✓ cost
  - ✓ maturity

- ✓ enables organizations to leverage their existing IP-based network.

- **iSCSI** and **FCIP** protocols are common examples that leverage IP for host-to-storage communication.

2.5 Storage

- Storage is a core component in a data center.
- A storage device uses magnetic, optic, or solid state media.
- Disks, tapes, and diskettes use magnetic media,
- CD/DVD uses optical media.
- Removable Flash memory or Flash drives uses solid state media.

Tapes

- In the past, **tapes** were the most popular storage option for backups because of their low cost.
- Tapes have various limitations in terms of performance and management, as listed below:
  - i. Data is stored on the tape linearly along the length of the tape. Search and retrieval of data are done sequentially, and it invariably takes several seconds to access the data. As a result, **random data access is slow and time-consuming**.
  - ii. In a shared computing environment, data stored on tape **cannot be accessed by multiple applications simultaneously**, restricting its use to one application at a time.
  - iii. On a tape drive, the read/write head touches the tape surface, so the tape degrades or wears out after repeated use.
  - iv. The storage and retrieval requirements of data from the tape and the overhead associated with managing the tape media are significant.
- Due to these limitations and availability of low-cost disk drives, tapes are no longer a preferred choice as a backup destination for enterprise-class data centers.

Optical Disc Storage:

- It is popular in small, single-user computing environments.
- It is frequently used by individuals to store photos or as a backup medium on personal or laptop computers.

- It is also used as a distribution medium for small applications, such as games, or as a means to transfer small amounts of data from one computer system to another.
- The capability to **write once and read many (WORM)** is one advantage of optical disc storage. Eg: CD-ROM
- Collections of optical discs in an array, called a **jukebox**, are still used as a fixed-content storage solution.
- Other forms of optical discs include CD-RW, Blu-ray disc, and other variations of DVD.

Disk Drives:

- **Disk drives** are the most popular storage medium used in modern computers for storing and accessing data for performance-intensive, online applications.
- Disks support rapid access to random data locations.
- Disks have large capacity.
- Disk storage arrays are configured with multiple disks to provide **increased capacity** and **enhanced performance**.
- Disk drives are accessed through predefined protocols, such as ATA, SATA, SAS, and FC.
- These protocols are implemented on the disk interface controllers.
- Disk interface controllers were earlier implemented as separate cards, which were connected to the motherboard.
- Modern disk interface controllers are integrated with the disk drives; therefore, disk drives are known by the protocol interface they support, for example SATA disk, FC disk, etc.

2.6 Disk Drive Components

- The key components of a hard disk drive are platter, spindle, read-write head, actuator arm assembly, and controller board (see Figure 2-5).
- I/O operations in a HDD are performed by rapidly moving the arm across the rotating flat platters coated with magnetic particles.
- Data is transferred between the disk controller and magnetic platters through the read-write (R/W) head which is attached to the arm. Data can be recorded and erased on magnetic platters any number of times.
- Following sections detail the different components of the disk drive, the mechanism for organizing and storing data on disks, and the factors that affect disk performance.

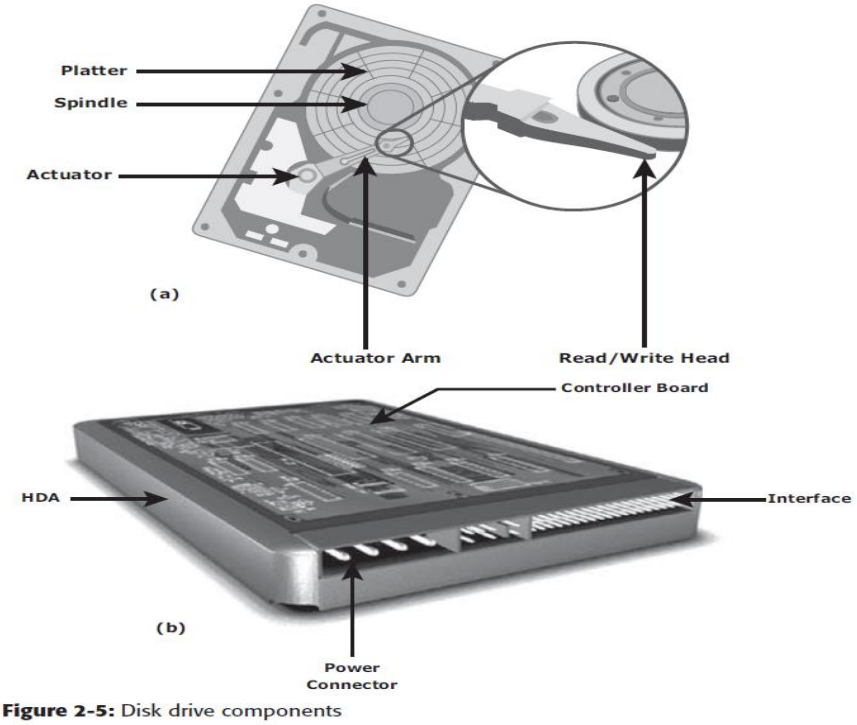


Figure 2-5: Disk drive components

2.6.1 Platter

- A typical HDD consists of one or more flat circular disks called platters (Figure 2-6). The data is recorded on these platters in binary codes (0s and 1s).
- The set of rotating platters is sealed in a case, called the Head Disk Assembly (HDA).
- A platter is a rigid, round disk coated with magnetic material on both surfaces (top and bottom). 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. The number of platters and the storage capacity of each platter determine the total capacity of the drive.

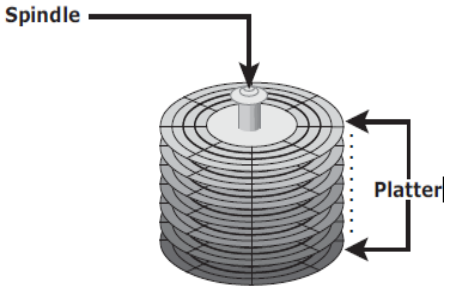


Figure 2-6: Spindle and platter

2.6.2 Spindle

- A spindle connects all the platters (refer to Figure 2-6) 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). Common spindle speeds are 5,400 rpm, 7,200 rpm, 10,000 rpm, and 15,000 rpm.
- The speed of the platter is increasing with improvements in technology, although the extent to which it can be improved is limited.

2.6.3 Read/Write Head

- Read/Write (R/W) heads, as shown in Figure 2-7, read and write data from or to platters. 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, the head detects the 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. When the spindle is rotating, there is a microscopic air gap maintained between the R/W heads and the platters, known as the head ***flying height.***
- This air gap is removed when the spindle stops rotating and the R/W head rests on a special area on the platter near the spindle. This area is called the ***landing zone.*** The landing zone is coated with a lubricant to reduce friction between the head and the platter.
- 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.

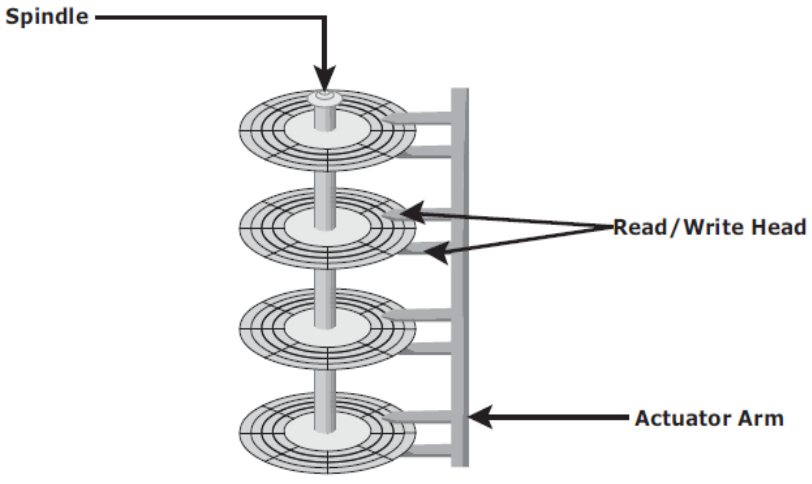


Figure 2-7: Actuator arm assembly

2.6.4 Actuator Arm Assembly

- R/W heads are mounted on the actuator arm assembly, which positions the R/W head at the location on the platter where the data needs to be written or read (refer to Figure 2-7).
- The R/W heads for all platters on a drive are attached to one actuator arm assembly and move across the platters simultaneously.

2.6.5 Drive Controller Board

- The controller (refer to Figure 2-5 [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 the power to the spindle motor and the speed of the motor. It also manages the communication between the drive and the host.
- In addition, it controls the R/W operations by moving the actuator arm and switching between different R/W heads, and performs the optimization of data access.

2.6.6 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 2-8.
- 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. The track and sector structure is written on the platter by the drive manufacturer using a low-level formatting operation.
- Typically, a sector holds 512 bytes of user data, although some disks can be formatted with larger sector sizes. In addition to user data, a sector also stores other information, such as the sector number, head number or platter number, and track number.
- A cylinder is a set of identical tracks on both surfaces of each drive platter. The location of R/W heads is referred to by the cylinder number, not by the track number.

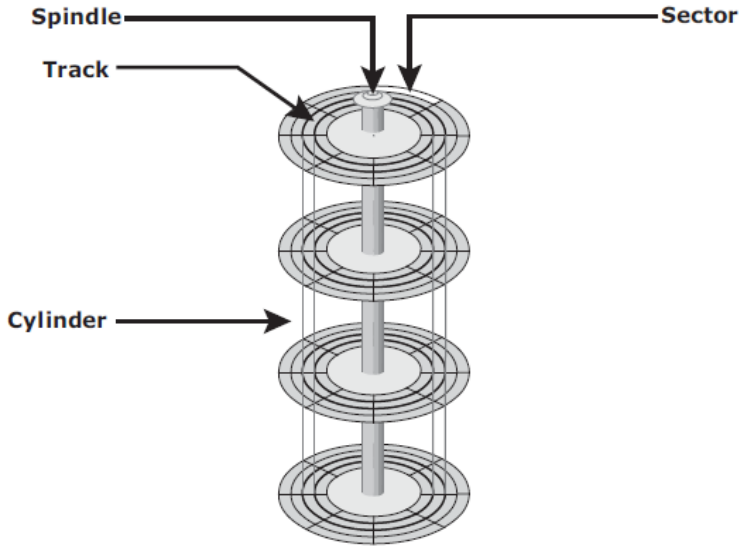


Figure 2-8: Disk structure: sectors, tracks, and cylinders

2.6.7 Zoned Bit Recording

- Platters are made of concentric tracks; the outer tracks can hold more data than the inner tracks because the outer tracks are physically longer than the inner tracks.
- On older disk drives, the outer tracks had the same number of sectors as the inner tracks, so data density was low on the outer tracks. This was an inefficient use of the available space, as shown in Figure 2-9 (a).
- Zoned bit recording uses the disk effi ciently. As shown in Figure 2-9 (b), this mechanism groups tracks into zones based on their distance from the center of the disk.
- The zones are numbered, with the outermost zone being zone 0. An appropriate number of sectors per track are assigned to each zone, so a zone near the center of the platter has fewer

sectors per track than a zone on the outer edge. However, tracks within a particular zone have the same number of sectors.

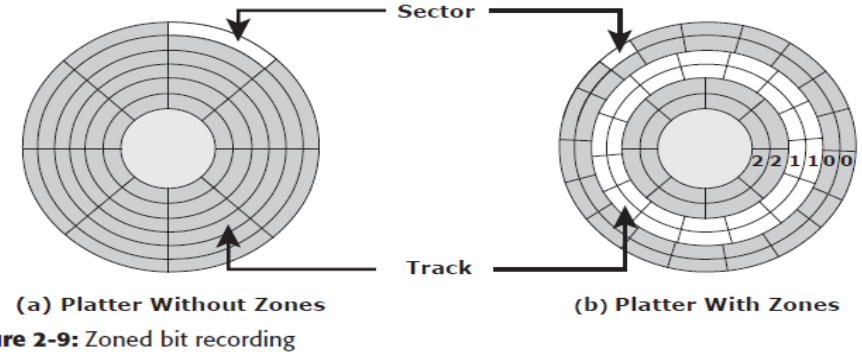


Figure 2-9: Zoned bit recording

2.6.8 Logical Block Addressing

- Earlier drives used physical addresses consisting of the cylinder, head, and sector (CHS) number to refer to specifi c locations on the disk, as shown in Figure 2-10 (a), and the host operating system had to be aware of the geometry of each disk used.
- Logical block addressing (LBA), as shown in Figure 2-10 (b), simplifies addressing by using a linear address to access physical blocks of data.
- The disk controller translates LBA to a CHS address, and the host needs to know only the size of the disk drive in terms of the number of blocks. The logical blocks are mapped to physical sectors on a 1:1 basis.
- In Figure 2-10 (b), the drive shows eight sectors per track, eight heads, and four cylinders. This means a total of 8 × 8 × 4 = 256 blocks, so the block number ranges from 0 to 255. Each block has its own unique address.
- Assuming that the sector holds 512 bytes, a 500 GB drive with a formatted capacity of 465.7 GB has in excess of 976,000,000 blocks.

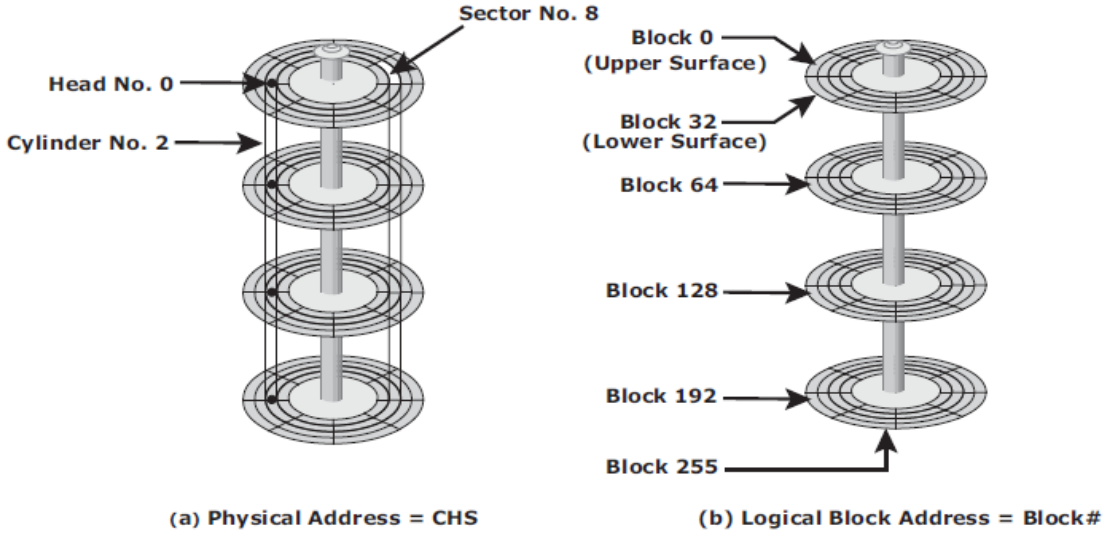


Figure 2-10: Physical address and logical block address

2.7 Disk Drive Performance

A disk drive is an electromechanical device that governs the overall performance of the storage system environment. The various factors that affect the performance of disk drives are discussed in this section.

2.7.1 Disk Service Time

Disk service time is the time taken by a disk to complete an I/O request. Components that contribute to the service time on a disk drive are ***seek time, rotational latency, and data transfer rate.***

- **Seek Time**

- The seek time (also called access time) describes the time taken to position the R/W heads across the platter with a radial movement (moving along the radius of the platter).
- In other words, it is the time taken to position and settle the arm and the head over the correct track. Therefore, the lower the seek time, the faster the I/O operation. Disk vendors publish the following seek time specifications:
  - **Full Stroke:** The time taken by the R/W head to move across the entire width of the disk, from the innermost track to the outermost track.
  - **Average**: The average time taken by the R/W head to move from one random track to another, normally listed as the time for one-third of a full stroke.
  - **Track-to-Track:** The time taken by the R/W head to move between adjacent tracks

❖ **Rotational Latency**

- To access data, the actuator arm moves the R/W head over the platter to a particular track while the platter spins to position the requested sector under the R/W head.
- The time taken by the platter to rotate and position the data under the R/W head is called **rotational latency.**
- This latency depends on the rotation speed of the spindle and is measured in milliseconds. The average rotational latency is one-half of the time taken for a full rotation.
- Average rotational latency is approximately 5.5 ms for a 5,400-rpm drive, and around 2.0 ms for a 15,000-rpm (or 250-rps revolution per second) drive as shown here:

Average rotational latency for a 15,000 rpm (or 250 rps)
drive = 0.5/250 = 2 milliseconds.

❖ **Data Transfer Rate**

- The data transfer rate (also called transfer rate) refers to the average amount of data per unit time that the drive can deliver to the HBA (Host Bus Adapter).
- In a read operation, the data first moves from disk platters to R/W heads; then it moves to the drive's internal buffer. Finally, data moves from the buffer through the interface to the host HBA.
- In a write operation, the data moves from the HBA to the internal buffer of the disk drive through the drive's interface. The data then moves from the buffer to the R/W heads. Finally, it moves from the R/W heads to the platters
- The data transfer rates during the R/W operations are measured in terms of internal and external transfer rates, as shown in Figure 2-11.
- **Internal transfer rate** is the speed at which data moves from a platter's surface to the internal buffer (cache) of the disk. The internal transfer rate takes into account factors such as the seek time and rotational latency.
- **External transfer rate** is the rate at which data can move through the interface to the HBA. The external transfer rate is generally the advertised speed of the interface, such as 133 MB/s for ATA. The sustained external transfer rate is lower than the interface speed.

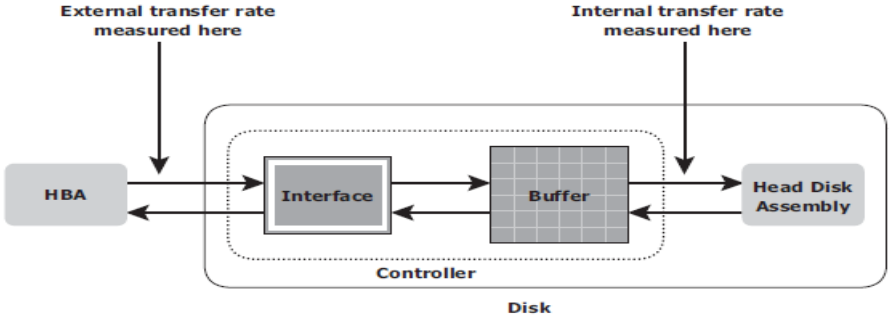


Figure 2-11: Data transfer rate

2.7.2 Disk I/O Controller Utilization

Utilization of a disk I/O controller has a signifi cant impact on the I/O response time. To understand this impact, consider that a disk can be viewed as a black box consisting of two elements:

- **Queue:** The location where an I/O request waits before it is processed by the I/O controller
- **Disk I/O Controller:** Processes I/Os waiting in the queue one by one

- The I/O requests arrive at the controller at the rate generated by the application. This rate is also called the **arrival rate.** These requests are held in the I/O queue, and the I/O controller processes them one by one, as shown in Figure 2-12.
- The I/O arrival rate, the queue length, and the time taken by the I/O controller to process each request determines the I/O response time. If the controller is busy or heavily utilized, the queue size will be large and the response time will be high.

Figure 2-12: I/O processing

- Based on the fundamental laws of disk drive performance, the relationship between controller utilization and average response time is given as

$$\text{Average response time } (T_r) = \text{Service time } (T_s) / (1 - \text{Utilization})$$

where T_s is the time taken by the controller to serve an I/O.

- As the utilization reaches 100 percent — that is, as the I/O controller saturates — the response time is closer to infinity.
- In essence, the saturated component, or the bottleneck, forces the serialization of I/O requests, meaning that each I/O request must wait for the completion of the I/O requests that preceded it. Figure 2-13 shows a graph plotted between utilization and response time.

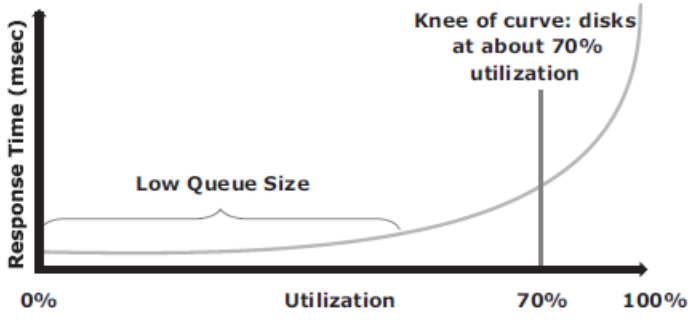


Figure 2-13: Utilization versus response time

- The graph indicates that the response time changes are nonlinear as the utilization increases. When the average queue sizes are low, the response time remains low.
- The response time increases slowly with added load on the queue and increases exponentially when the utilization exceeds 70 percent. Therefore, for performance-sensitive applications, it is common to utilize disks below their 70 percent of I/O serving capability.

2.8 Host Access to Data

- Data is accessed and stored by applications using the underlying infrastructure. The key components of this infrastructure are the operating system (or file system), connectivity, and storage.
- In either case, the host controller card accesses the storage devices using predefined protocols, such as IDE/ATA, SCSI, or Fibre Channel (FC). IDE/ATA and SCSI are popularly used in small and personal computing environments for accessing internal storage.
- FC and iSCSI protocols are used for accessing data from an external storage device (or subsystems). External storage devices can be connected to the host directly or through the storage network. When the storage is connected directly to the host, it is referred as **direct-attached storage (DAS).**
- Understanding access to data over a network is important because it lays the foundation for storage networking technologies. Data can be accessed over a network in one of the following ways: **block level, file level, or object level.**
- In general, the application requests data from the fi le system (or operating system) by specifying the filename and location. The file system maps the fi le attributes to the logical block address of the data and sends the request to the storage device. The storage device

converts the logical block address (LBA) to a cylinder-head-sector (CHS) address and fetches the data.

- In a **block-level** access, the fi le system is created on a host, and data is accessed on a network at the block level, as shown in Figure 2-14 (a). In this case, raw disks or logical volumes are assigned to the host for creating the fi le system.
- In a **file-level** access, the fi le system is created on a separate fi le server or at the storage side, and the file-level request is sent over a network, as shown in Figure 2-14 (b). Because data is accessed at the file level, this method has higher overhead, as compared to the data accessed at the block level.
- **Object-level** access is an intelligent evolution, whereby data is accessed over a network in terms of self-contained objects with a unique object identifier.

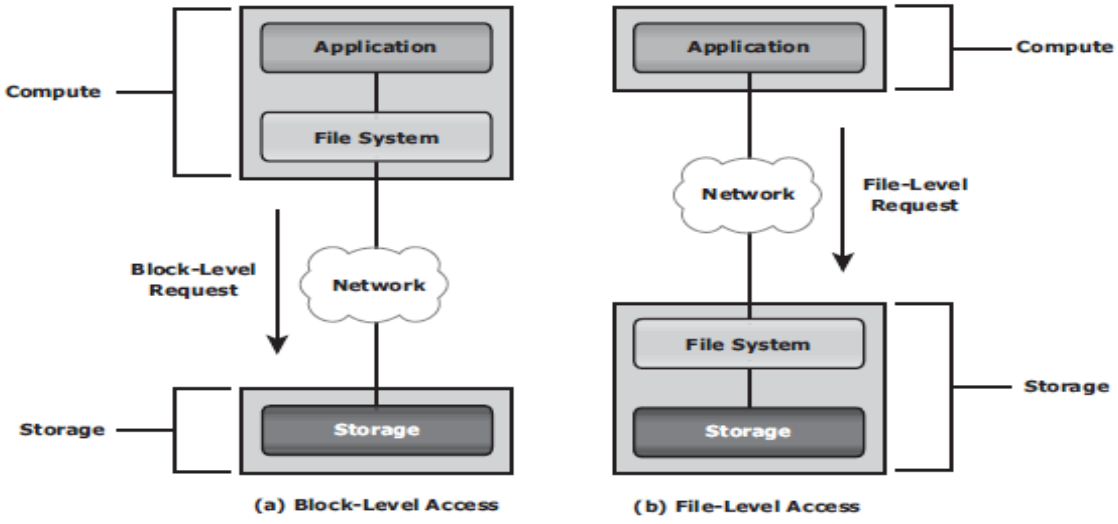


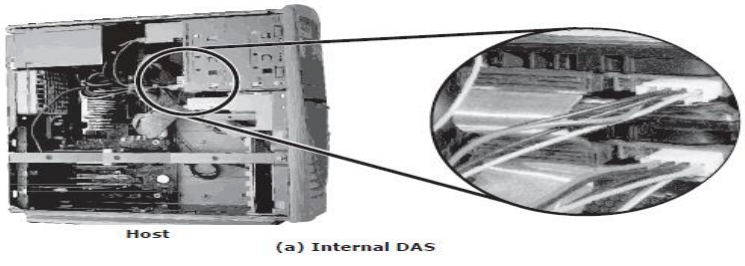
Figure 2-14: Host access to storage

2.9 Direct-Attached Storage

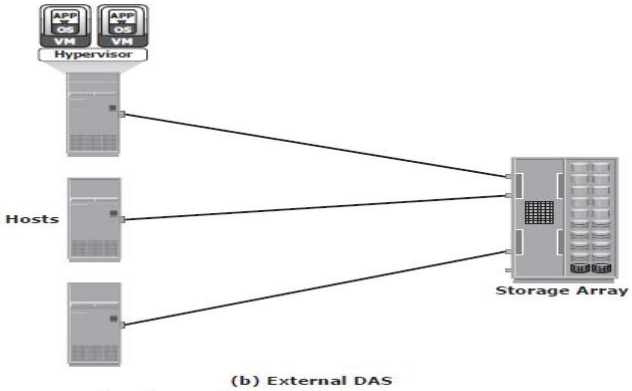
- DAS is an architecture in which storage is connected directly to the hosts. The internal disk drive of a host and the directly-connected external storage array are some examples of DAS.
- DAS is classified as internal or external, based on the location of the storage device with respect to the host.
- In **internal DAS** architectures, the storage device is internally connected to the host by a serial or parallel bus (see Figure 2-15 [a]).
    - The physical bus has distance limitations and can be sustained only over a shorter

distance for highspeed connectivity. In addition, most internal buses can support only a limited number of devices, and they occupy a large amount of space inside the host, making maintenance of other components difficult.

- On the other hand, in external DAS architectures, the host connects directly to the external storage device, and data is accessed at the block level (see Figure 2-15 [b]).
  - In most cases, communication between the host and the storage device takes place over a SCSI or FC protocol. Compared to internal DAS, an external DAS overcomes the distance and device count limitations and provides centralized management of storage devices.



(a) Internal DAS



(b) External DAS

Figure 2-15: Internal and external DAS architecture

2.9.1 DAS Benefits and Limitations

Benefits:

- DAS requires a relatively lower initial investment than storage networking architectures.
- The DAS configuration is simple and can be deployed easily and rapidly.
- The setup is managed using host-based tools, such as the host OS, which makes storage management tasks easy for small environments. Because DAS has a simple architecture, it requires fewer management tasks and less hardware and software elements to set up and operate.

Limitations:

- However, DAS does not scale well. A storage array has a limited number of ports, which restricts the number of hosts that can directly connect to the storage.
- When capacities are reached, the service availability may be compromised. DAS does not make optimal use of resources due to its limited capability to share front-end ports.
- In DAS environments, unused resources cannot be easily reallocated, resulting in islands of over-utilized and under-utilized storage pools.

2.10 Storage Design Based on Application Requirements and Disk Performance

- Determining storage requirements for an application begins with determining the required storage capacity. This is easily estimated by the size and number of file systems and database components used by applications.
- The I/O size, I/O characteristics, and the number of I/Os generated by the application at peak workload are other factors that affect disk performance, I/O response time, and design of storage systems.
- The I/O block size depends on the file system and the database on which the application is built. Block size in a database environment is controlled by the underlying database engine and the environment variables.
- The disk service time (T_S) for an I/O is a key measure of disk performance; T_S, along with disk utilization rate (U), determines the I/O response time for an application. As discussed earlier in this chapter, the total disk service time (T_S) is the sum of the seek time (T), rotational latency (L), and internal transfer time (X):

$$T_s = T + L + X$$

Consider an example with the following specifications provided for a disk:

- The average seek time is 5 ms in a random I/O environment; therefore, T = 5 ms.
- Disk rotation speed of 15,000 revolutions per minute or 250 revolutions per second — from which rotational latency (L) can be determined, which is one-half of the time taken for a full rotation or L = (0.5/250 rps expressed in ms).
- 40 MB/s internal data transfer rate, from which the internal transfer time (X) is derived based on the block size of the I/O — for example, an I/O with a block size of 32 KB; therefore X = 32 KB/40 MB.

Consequently, the time taken by the I/O controller to serve an I/O of block size 32 KB is $(T_S) = 5 \text{ ms} + (0.5/250) + 32 \text{ KB}/40 \text{ MB} = 7.8 \text{ ms}$.

Therefore, the maximum number of I/Os serviced per second or IOPS is $(1/T_S) = 1/(7.8 \times 10^{-3}) = 128$ IOPS.

- Table 2-1 lists the maximum IOPS that can be serviced for different block sizes using the previous disk specifications.

Table 2-1: IOPS Performed by Disk Drive

BLOCK SIZE	$T_s = T + L + X$	IOPS = $1/T_s$
4 KB	5 ms + (0.5/250 rps) + 4 K/40 MB = 5 + 2 + 0.1 = 7.1	140
8 KB	5 ms + (0.5/250 rps) + 8 K/40 MB = 5 + 2 + 0.2 = 7.2	139
16 KB	5 ms + (0.5/250 rps) + 16 K/40 MB = 5 + 2 + 0.4 = 7.4	135
32 KB	5 ms + (0.5/250 rps) + 32 K/40 MB = 5 + 2 + 0.8 = 7.8	128
64 KB	5 ms + (0.5/250 rps) + 64 K/40 MB = 5 + 2 + 1.6 = 8.6	116

- The IOPS ranging from 116 to 140 for different block sizes represents the IOPS that can be achieved at potentially high levels of utilization (close to 100 percent). As discussed in Section 2.7.2, the application response time, R, increases with an increase in disk controller utilization.
  - For the same preceding example, the response time (R) for an I/O with a block size of 32 KB at 96 percent disk controller utilization is

$$R = T_s/(1 - U) = 7.8/(1 - 0.96) = 195 \text{ ms}$$

- If the application demands a faster response time, then the utilization for the disks should be maintained below 70 percent. For the same 32-KB block size, at 70-percent disk utilization, the response time reduces drastically to 26 ms. However, at lower disk utilization, the number of IOPS a disk can perform is also reduced.
  - In the case of a 32-KB block size, a disk can perform 128 IOPS at almost 100 percent utilization, whereas the number of IOPS it can perform at 70-percent utilization is 89 (128 x 0.7). This indicates that the number of I/Os a disk can perform is an important

factor that needs to be considered while designing the storage requirement for an application.

- Therefore, the storage requirement for an application is determined in terms of both the capacity and IOPS. If an application needs 200 GB of disk space, then this capacity can be provided simply with a single disk. However, if the application IOPS requirement is high, then it results in performance degradation because just a single disk might not provide the required response time for I/O operations.
- Based on this discussion, the total number of disks required (D_R) for an application is computed as follows:

$$D_R = \text{Max}\ (D_C, D_I)$$

Where D_C is the number of disks required to meet the capacity, and D_I is the number of disks required to meet the application IOPS requirement.

- Let's understand this with the help of an example. Consider an example in which the capacity requirement for an application is 1.46 TB. The number of IOPS generated by the application at peak workload is estimated at 9,000 IOPS. The vendor specifies that a 146-GB, 15,000-rpm drive is capable of doing a maximum 180 IOPS.
    - In this example, the number of disks required to meet the capacity requirements will be 1.46 TB/146 GB = 10 disks.
- To meet the application IOPS requirements, the number of disks required is 9,000/180 = 50. However, if the application is response-time sensitive, the number of IOPS a disk drive can perform should be calculated based on 70- percent disk utilization.
    - Considering this, the number of IOPS a disk can perform at 70 percent utilization is 180×0.7 = 126 IOPS. Therefore, the number of disks required to meet the application IOPS requirement will be 9,000/126 = 72.
- As a result, the number of disks required to meet the application requirements will be Max (10, 72) = 72 disks.
    - The preceding example indicates that from a capacity-perspective, 10 disks are sufficient; however, the number of disks required to meet application performance is 72.