Module-3

IP SAN and FCoE

Technology of transporting block I/Os over an IP is referred to as IP SAN. With IP SAN, organizations can extend the geographical reach of their storage infrastructure

iSCSI

- iSCSI is an IP based protocol that establishes and manages connections between host and storage over IP, as shown in Fig below.
- iSCSI encapsulates SCSI commands and data into an IP packet and transports them using TCP/IP.
- iSCSI is widely adopted for connecting servers to storage because it is relatively inexpensive and easy to implement, especially in environments in which an FC SAN does not exist

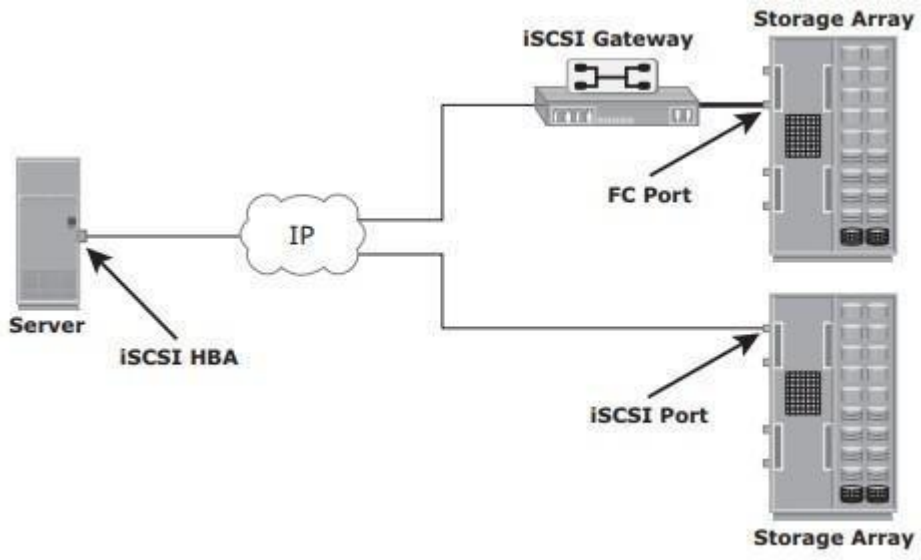


Fig : iSCSI implementation

Components of iSCSI

- An initiator (host), target (storage or iSCSI gateway), and an IP-based network are the key iSCSI components.
- If an iSCSI-capable storage array is deployed, then a host with the iSCSI initiator can directly communicate with the storage array over an IP network.
- However, in an implementation that uses an existing FC array for iSCSI communication, an iSCSI gateway is used.

- These devices perform the translation of IP packets to FC frames and vice versa, thereby bridging the connectivity between the IP and FC environments.

iSCSI Host Connectivity

The three iSCSI host connectivity options are:

- A standard NIC with software iSCSI initiator,
- a TCP offload engine (TOE) NIC with software iSCSI initiator,
- an iSCSI HBA

- The function of the iSCSI initiator is to route the SCSI commands over an IP network.
- A **standard NIC with a software iSCSI** initiator is the simplest and least expensive connectivity option. It is easy to implement because most servers come with at least one, and in many cases two, embedded NICs. It requires only a software initiator for iSCSI functionality. Because NICs provide standard IP function, encapsulation of SCSI into IP packets and decapsulation are carried out by the host CPU. This places additional overhead on the host CPU. If a standard NIC is used in heavy I/O load situations, the host CPU might become a bottleneck. TOE NIC helps reduce this burden.
- A **TOE NIC** offloads TCP management functions from the host and leaves only the iSCSI functionality to the host processor. The host passes the iSCSI information to the TOE card, and the TOE card sends the information to the destination using TCP/IP.
- Although this solution improves performance, the iSCSI functionality is still handled by a software initiatorthat requires host CPU cycles.
- An **iSCSI HBA** is capable of providing performance benefits because it offloads the entire iSCSI and TCP/IP processing from the host processor.
- The use of an iSCSI HBA is also the simplest way to boot hosts from a SAN environment via iSCSI.
- If there is no iSCSI HBA, modifications must be made to the basic operating system to boot a host from the storage devices because the NIC needs to obtain an IP address before the operating system loads
- The functionality of an iSCSI HBA is similar to the functionality of an FC HBA.

.

iSCSI Topologies

- Two topologies of iSCSI implementations are **native and bridged**.
- Native topology does not have FC components.
- The initiators may be either directly attached to targets or connected through the IP network.
- Bridged topology enables the coexistence of FC with IP by providing iSCSI-to-FC bridging functionality.
- For example, the initiators can exist in an IP environment while the storage remains in an FC environment.

Native iSCSI Connectivity

- FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed.
  - In Fig (a), the array has one or more iSCSI ports configured with an IP address and isconnected to a standard Ethernet switch.
  - After an initiator is logged on to the network, it can access the available LUNs on the storagearray.
  - A single array port can service multiple hosts or initiators as long as the array port can handlethe amount of storage traffic that the hosts generate.

Bridged iSCSI Connectivity

- A bridged iSCSI implementation includes FC components in its configuration.
- Fig (b), illustrates iSCSI host connectivity to an FC storage array. In this case, the array does not have any iSCSI ports. Therefore, an external device, called a gateway or a multiprotocol router, must be used to facilitate the communication between the iSCSI host and FC storage.
- The gateway converts IP packets to FC frames and vice versa.
- The bridge devices contain both FC and Ethernet ports to facilitate the communication between the FC and IP environments.
- In a bridged iSCSI implementation, the iSCSI initiator is configured with the gateway's IP address as its target destination.
- On the other side, the gateway is configured as an FC initiator to the storage array.
- **Combining FC and Native iSCSI Connectivity:** The most common topology is a combination of FC and native iSCSI. Typically, a storage array comes with both FC and iSCSI ports that enable iSCSI and FC connectivity in the same environment, as shown in Fig(c)

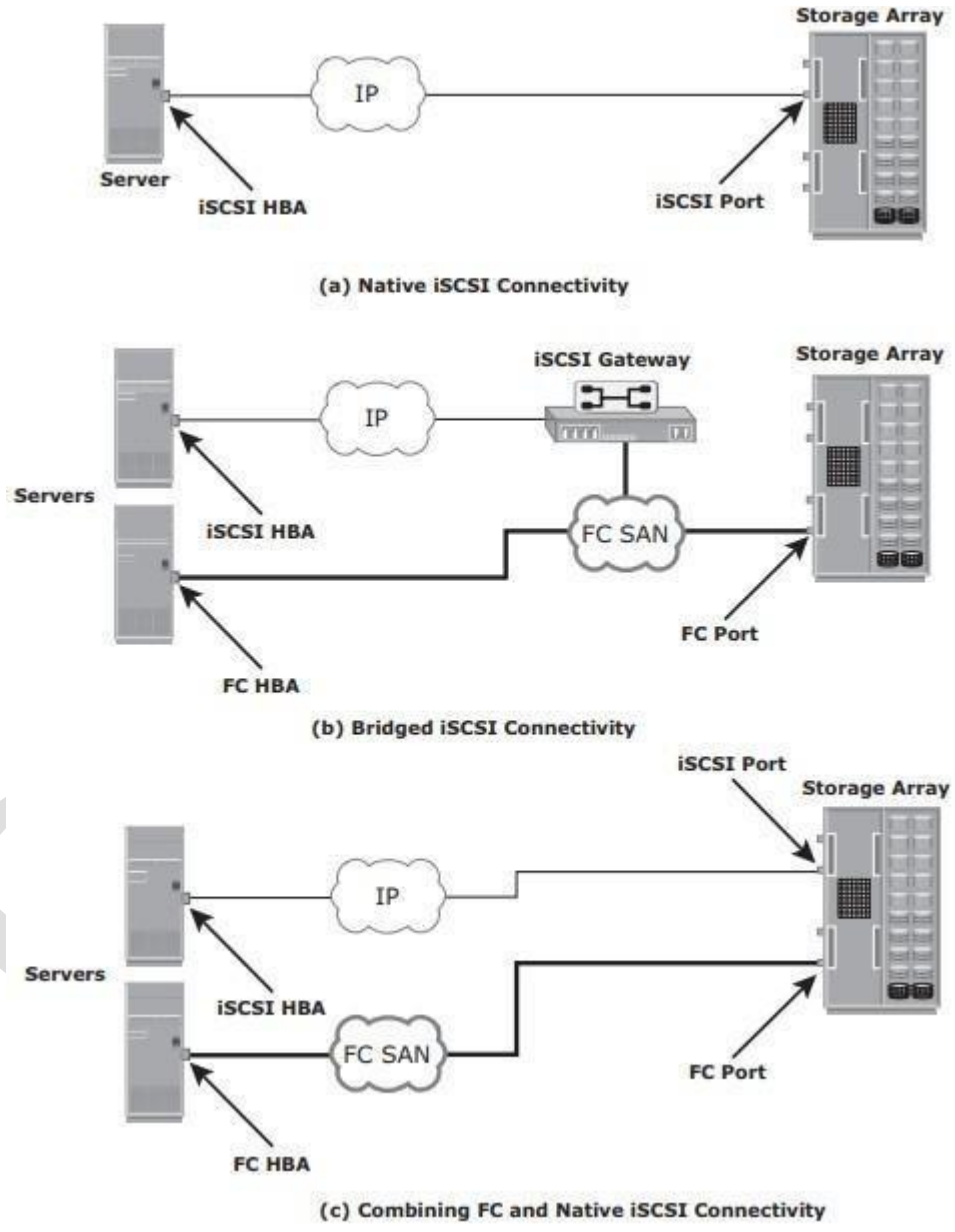


Fig c) : iSCSI Topologies

iSCSI Protocol Stack

- Fig 2.23 displays a model of the iSCSI protocol layers and depicts the encapsulation order of the SCSI commands for their delivery through a physical carrier.

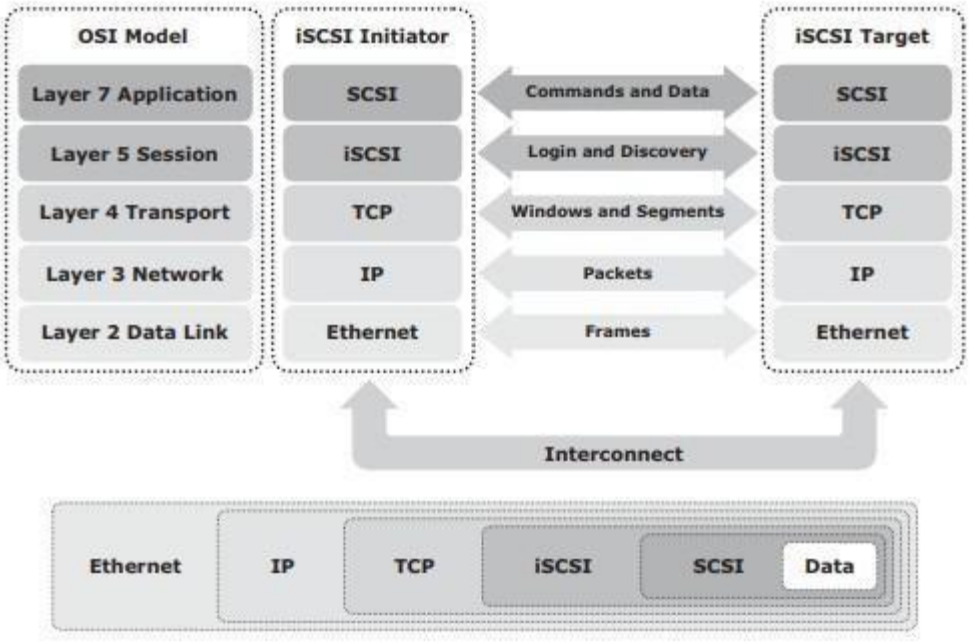


Fig 2.23: iSCSI protocol stack

- SCSI is the command protocol that works at the application layer of the Open System Interconnection (OSI) model.
- The initiators and targets use SCSI commands and responses to talk to each other.
- The SCSI command descriptor blocks, data, and status messages are encapsulated into TCP/IP and transmitted across the network between the initiators and targets.
- iSCSI is the session-layer protocol that initiates a reliable session between devices that recognize SCSI commands and TCP/IP.
- The iSCSI session-layer interface is responsible for handling login, authentication, target discovery, and session management.

- TCP is used with iSCSI at the transport layer to provide reliable transmission.
- TCP controls message flow, windowing, error recovery, and retransmission.
- It relies upon the network layer of the OSI model to provide global addressing and connectivity.
- The Layer 2 protocols at the data link layer of this model enable node-to-node communication through a physical network.

iSCSI PDU

- A *protocol data unit* (PDU) is the basic "information unit" in the iSCSI environment.
- The iSCSI initiators and targets communicate with each other using iSCSI PDUs. This communication includes establishing iSCSI connections and iSCSI sessions, performing iSCSI discovery, sending SCSI commands and data, and receiving SCSI status.
- All iSCSI PDUs contain one or more header segments followed by zero or more data segments.
- The PDU is then encapsulated into an IP packet to facilitate the transport.
- A PDU includes the components shown in Fig below.
- The IP header provides packet-routing information to move the packet across a network.
- The TCP header contains the information required to guarantee the packet delivery to the target.
- The iSCSI header (basic header segment) describes how to extract SCSI commands and data for the target. iSCSI adds an optional CRC, known as the *digest*, to ensure datagram integrity. This is in addition to TCP checksum and Ethernet CRC.
- The header and the data digests are optionally used in the PDU to validate integrity and data placement.

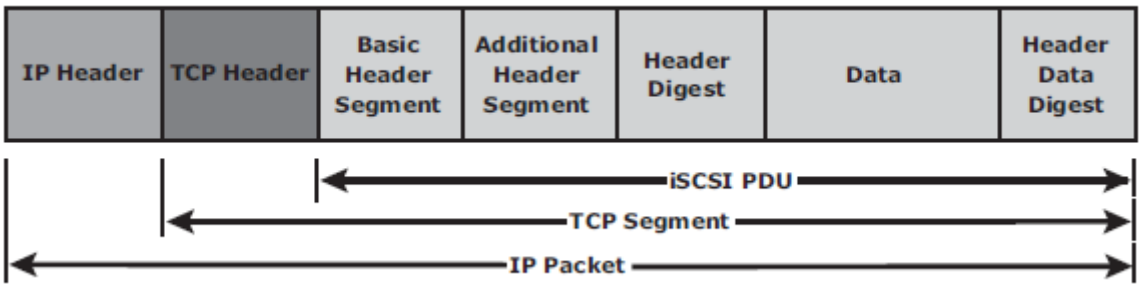


Fig : iSCSI PDU encapsulated in an IP packet

iSCSI Discovery

- An initiator must discover the location of its targets on the network and the names of the targets available to it before it can establish a session.
- This discovery can take place in two ways:
  - **SendTargets discovery**
  - **internet Storage Name Service (iSNS).**
- In *SendTargets discovery*, the initiator is manually configured with the target's network portal to establish a discovery session. The initiator issues the SendTargets command, and the target network portal responds with the names and addresses of the targets available to the host.
- iSNS (Fig below) enables automatic discovery of iSCSI devices on an IP network. The initiators and targets can be configured to automatically register themselves with the iSNS server. Whenever an initiator wants to now the targets that it can access, it can query the iSNS server for a list of available targets.
- The discovery can also take place by using service location protocol (SLP). However, this is less commonly used than SendTargets discovery and iSNS.

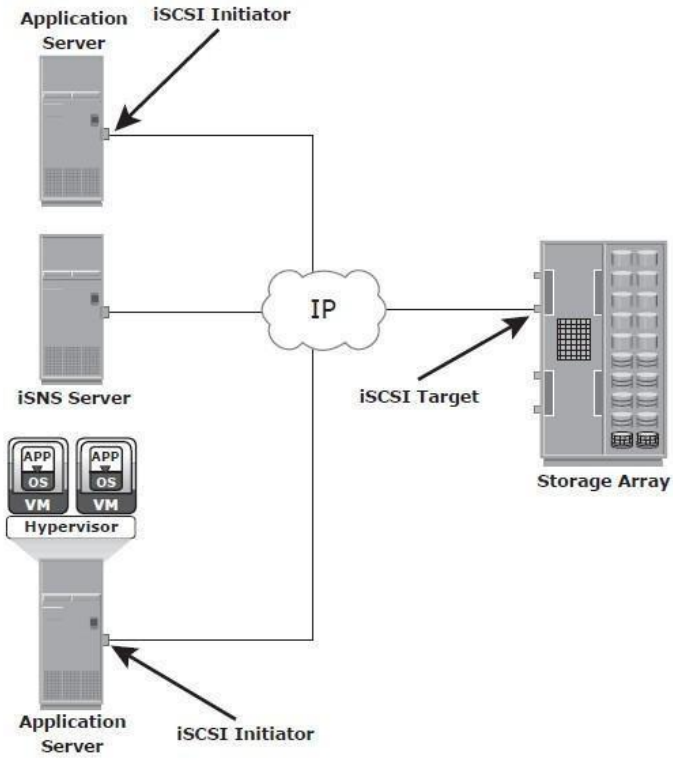


Fig : Discovery using iSNS

iSCSI Names

- A unique worldwide iSCSI identifier, known as an *iSCSI name*, is used to identify the initiators and targets within an iSCSI network to facilitate communication.
- The unique identifier can be a combination of the names of the department, application, or manufacturer, serial number, asset number, or any tag that can be used to recognize and manage the devices.
- Following are two types of iSCSI names commonly used:

  - **iSCSI Qualified Name (IQN):**
  - **Extended Unique Identifier (EUI)**

- **iSCSI Qualified Name (IQN):** An organization must own a registered domain name to generate iSCSI Qualifi ed Names. This domain name does not need to be active or resolve to an address. It just needs to be reserved to prevent other organizations from using the same domain name to generate iSCSI names. A date is included in the name to avoid potential conflicts caused by the transfer of domain names.

  An example of an IQN is iqn.2008-02.com.example:*optional_string.* The *optional_string* provides a serial number, an asset number, or any other device identifiers.

- **Extended Unique Identifi er (EUI):** An EUI is a globally unique identifier based on the IEEE EUI-64 naming standard. An EUI is composed of the eui prefix followed by a 16-character hexadecimal name, such aseui.0300732A32598D26.

- In either format, the allowed special characters are dots, dashes, and blank spaces.

iSCSI Session

- An iSCSI session is established between an initiator and a target, as shown in Fig.
- A session is identified by a session ID (SSID), which includes part of an initiator ID and a target ID.
- The session can be intended for one of the following:
  - The discovery of the available targets by the initiators and the location of a specific target on a network
  - The normal operation of iSCSI (transferring data between initiators and targets)
- There might be one or more TCP connections within each session. Each TCP connection within the session has a unique connection ID (CID).
- An iSCSI session is established via the iSCSI login process. The login process is started when the initiator establishes a TCP connection with the required target either via the well-known port 3260 or a specified target port.

- During the login phase, the initiator and the target authenticate each other and negotiate on various parameters.
- After the login phase is successfully completed, the iSCSI session enters the full-feature phase for normal SCSI transactions. In this phase, the initiator may send SCSI commands and data to the various LUNs on the target.
- The final phase of the iSCSI session is the connection termination phase, which is referred to as the logout procedure.
- The initiator is responsible for commencing the logout procedure; however, the target may also prompt termination by sending an iSCSI message, indicating the occurrence of an internal error condition.
- After the logout request is sent from the initiator and accepted by the target, no further request and response can be sent on that connection.

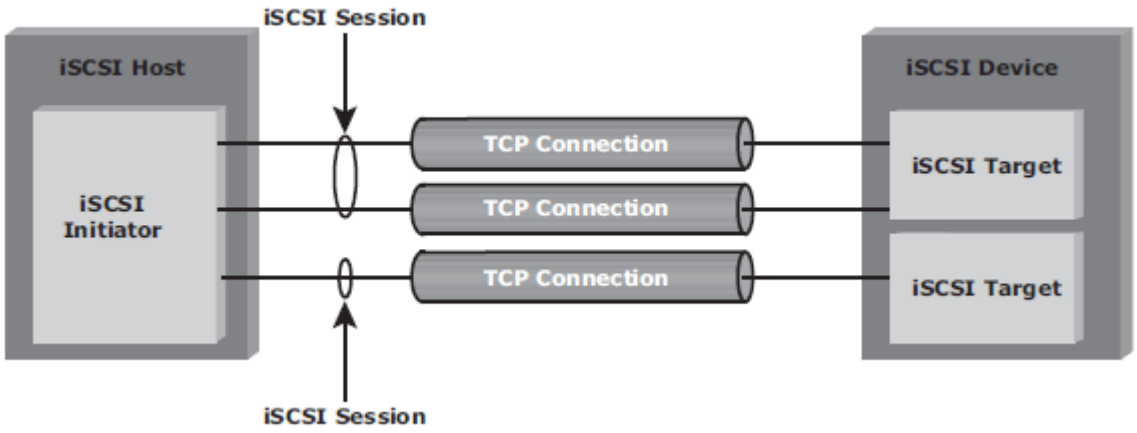


Fig : iSCSI session

iSCSI Command Sequencing

- The iSCSI communication between the initiators and targets is based on the request-response command sequences.

- A command sequence may generate multiple PDUs.
- A ***command sequence number*** **(CmdSN**) within an iSCSI session is used for numbering all initiator-to-target command PDUs belonging to the session.
- This number ensures that every command is delivered in the same order in which it is transmitted, regardless of the TCP connection that carries the command in the session.
- Command sequencing begins with the first login command, and the CmdSN is incremented by one for each subsequent command.
- The iSCSI target layer is responsible for delivering the commands to the SCSI layer in the order of their CmdSN.
- Similar to command numbering, a ***status sequence number*** **(StatSN)** is used to sequentially number status responses, as shown in Fig.
- These unique numbers are established at the level of the TCP connection.
- A target sends ***request-to-transfer*** **(R2T)** PDUs to the initiator when it is ready to accept data.
- A *data sequence number* (DataSN) is used to ensure in-order delivery of data within the same command.
- The DataSN and R2TSN are used to sequence data PDUs and R2Ts, respectively.

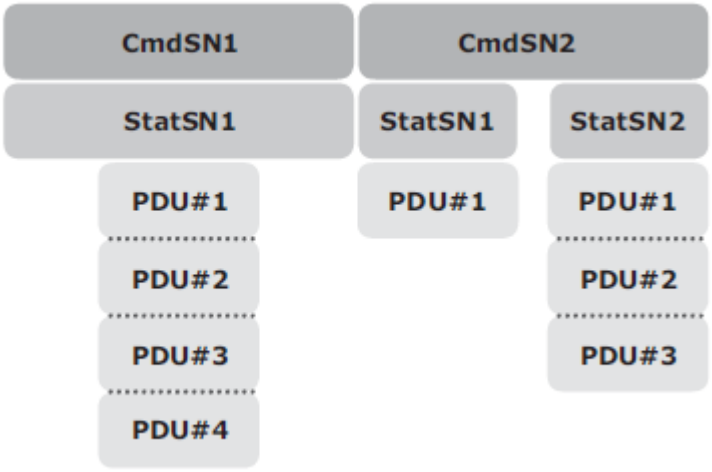


Fig : Command and status sequence number

FCIP (Fibre channel over IP)

- FCIP is a IP-based protocol that is used to connect distributed FC-SAN islands.
- Creates virtual FC links over existing IP network that is used to transport FC data between different FC SANS.
- It encapsulates FC frames into IP packet.
- It provides disaster recovery solution.

FCIP Protocol Stack

- The FCIP protocol stack is shown in Fig below. Applications generate SCSI commands and data, which are processed by various layers of the protocol stack.
- The upper layer protocol SCSI includes the SCSI driver program that executes the read-and-write commands.
- Below the SCSI layer is the Fibre Channel Protocol (FCP) layer, which is simply a Fibre Channel frame whose payload is SCSI.
- The FCP layer rides on top of the Fibre Channel transport layer. This enables the FC frames to run natively within a SAN fabric environment. In addition, the FC frames can be encapsulated into the IP packet and sent to a remote SAN over the IP.

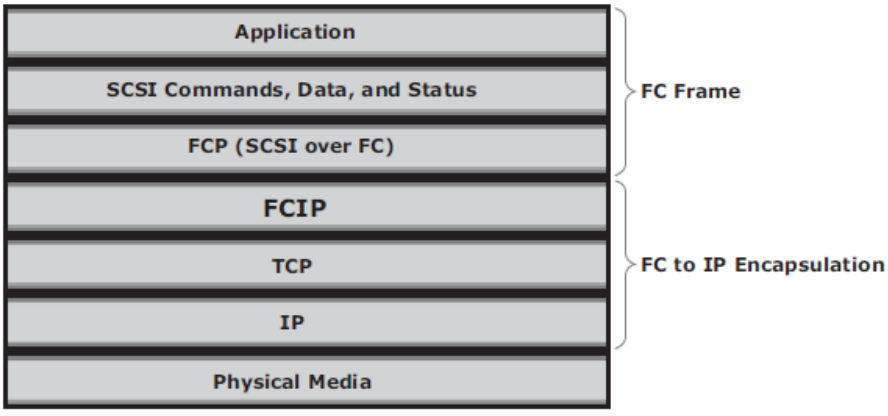


Fig : FCIP protocol stack

- The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Fig). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IP traffic.
- Encapsulation of FC frame into an IP packet could cause the IP packet to be fragmented when the data link cannot support the maximum transmission unit (MTU) size of an IP packet.
- When an IP packet is fragmented, the required parts of the header must be copied by all fragments.
- When a TCP packet is segmented, normal TCP operations are responsible for receiving and re-sequencing the data prior to passing it on to the FC processing portion of the device.

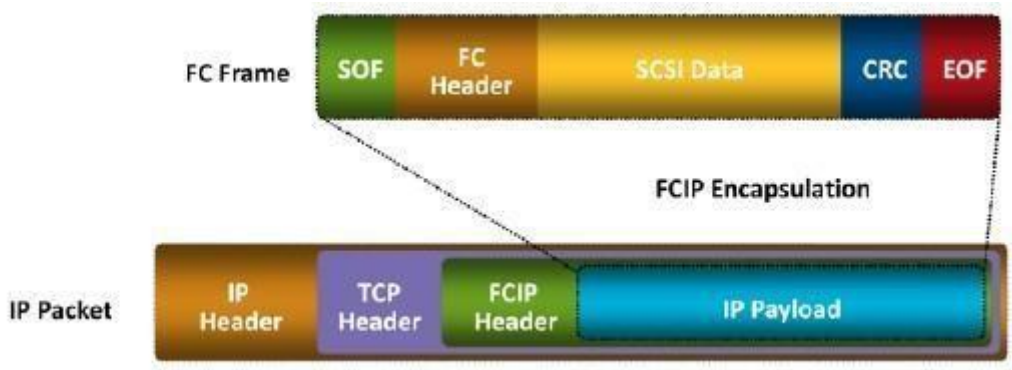


Fig FCIP encapsulation

FCIP Topology

- In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (Fig).
- The FCIP gateway at one end of the IP network encapsulates the FC frames into IP packets.
- The gateway at the other end removes the IP wrapper and sends the FC data to the layer 2 fabric.
- The fabric treats these gateways as layer 2 fabric switches.
- An IP address is assigned to the port on the gateway, which is connected to an IP network. After the IP connectivity is established, the nodes in the two independent fabrics can communicate with each other.

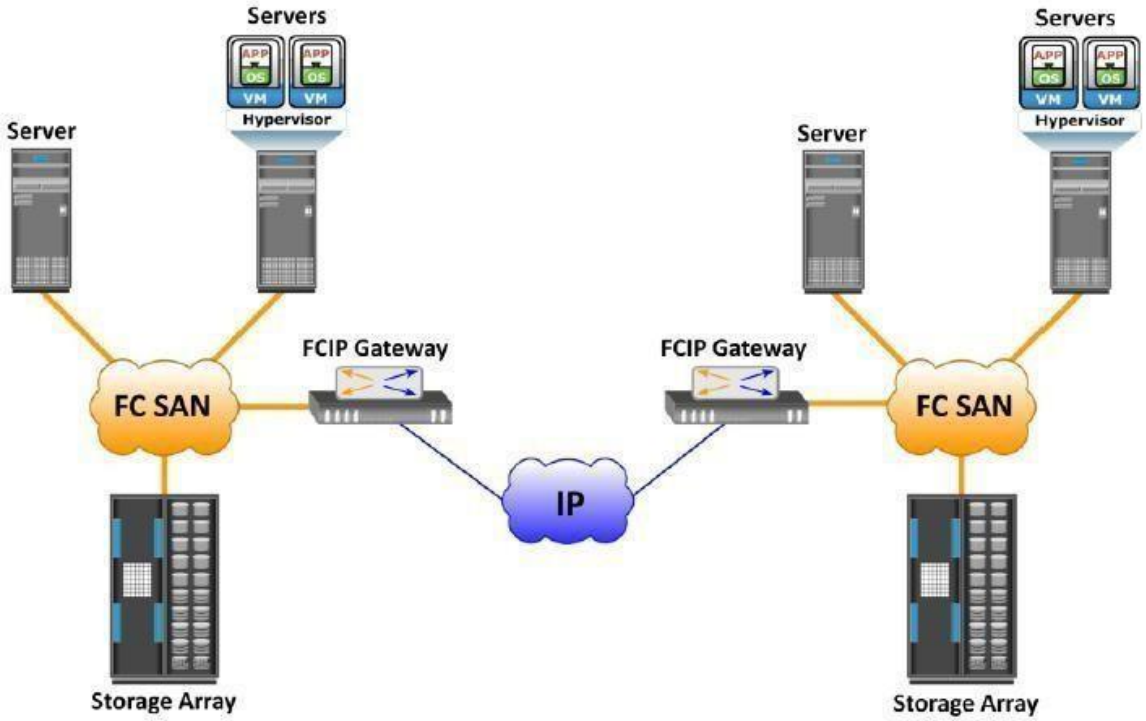


Fig : FCIP topology

FCoE (Fibre Channel over Ethernet)

- Data centers typically have multiple networks to handle various types of I/O traffic — for

example, an Ethernet network for TCP/IP communication and an FC network for FC communication.

- TCP/IP is typically used for client-server communication, data backup, infrastructure management communication, and so on.
- FC is typically used for moving block-level data between storage and servers.

To support multiple networks, servers in a data center are equipped with multiple redundant physical network interfaces — for example, multiple Ethernet and FC cards/adapters. In addition, to enable the communication, different types of networking switches and physical cabling infrastructure are implemented in data centers.

- The need for two different kinds of physical network infrastructure increases the overall cost and complexity of data center operation.
- Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffic over a single physical interface infrastructure.
- FCoE helps organizations address the challenges of having multiple discrete network infrastructures.
- FCoE uses the Converged Enhanced Ethernet (CEE) link (1*0* Gigabit Ethernet) to send FC frames over Ethernet.

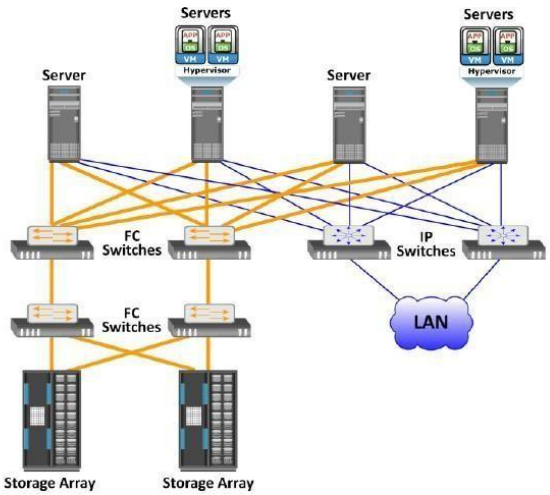


Fig: Infrastructure before using FCOE

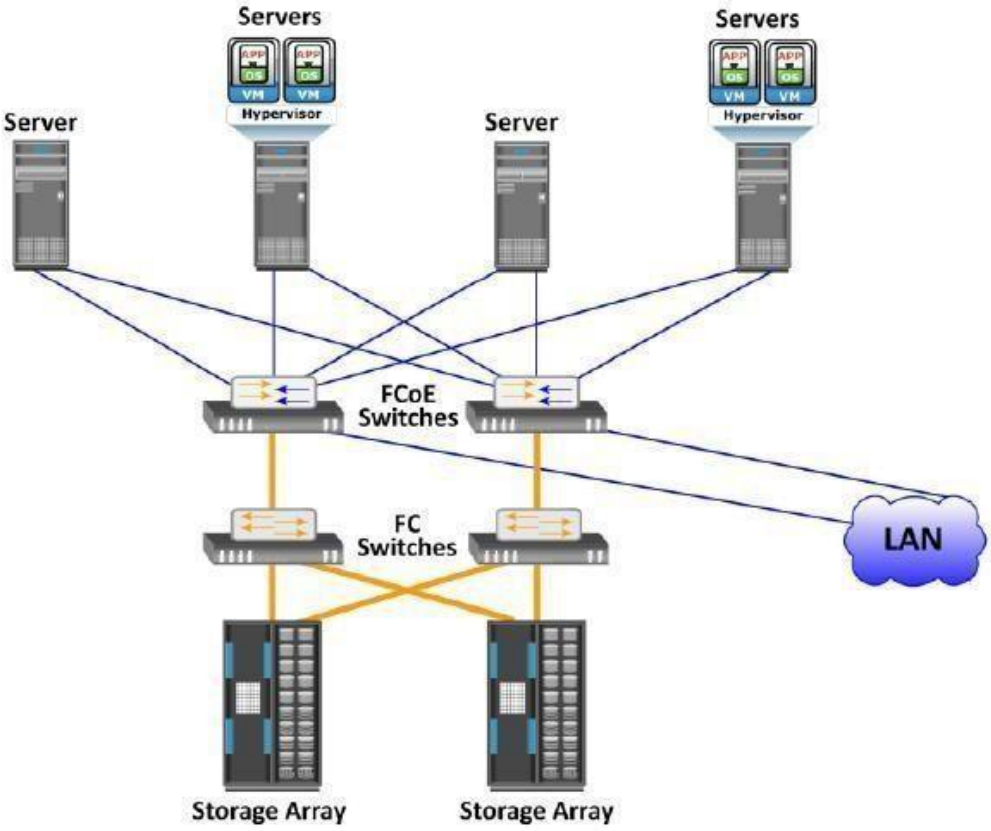


Fig Infrastructure after using FCOE

Components of FCOE Network

The key components of FCOE are :

- Converged Network Adaptors(CNA)
- Cables
- FCOE Switches

Converged Network Adaptors(CNA)

- A CNA provides the functionality of both a standard NIC and an FC HBA in a single adapter and consolidates both types of traffic. CNA eliminates the need to deploy separate adapters and cables for FC and Ethernet communications, thereby reducing the required number of server slots and switch ports.

- As shown in Fig below, a CNA contains separate modules for 10 Gigabit Ethernet, Fibre Channel, and FCoE Application Specific Integrated Circuits (ASICs). The FCoE ASIC encapsulates FC frames into Ethernet frames. One end of this ASIC is connected to 10GbE and FC ASICs for server connectivity, while the other end provides a 10GbE interface to connect to an FCoE switch.

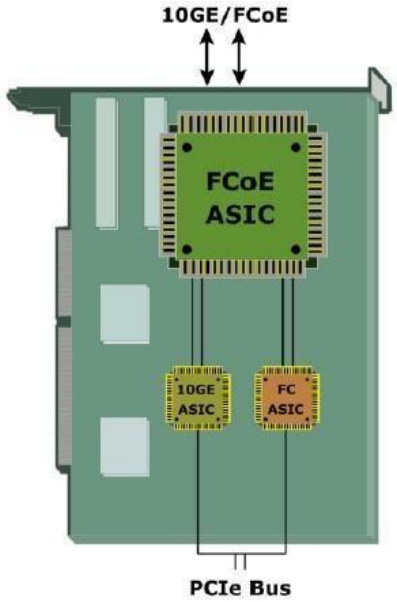


Fig : Converged Network Adapter

Cables

- There are two options available for FCoE cabling:
    1. Copper based Twinax
    2. Standard fiber optical cables.
- A Twinax cable is composed of two pairs of copper cables covered with a shielded casing. The Twinax cable can transmit data at the speed of 10 Gbps over shorter distances up to 10 meters. Twinax cables require less power and are less expensive than fiber optic cables.
- The Small Form Factor Pluggable Plus (SFP+) connector is the primary connector used for FCoE links and can be used with both optical and copper cables.

FCoE Switches

- An FCoE switch has both **Ethernet switch** and **Fibre Channel switch** functionalities.
- As shown in Fig below, FCoE switch consists of:
    1. *Fibre Channel Forwarder (FCF),*
    2. *Ethernet Bridge,*
    3. set of Ethernet ports
    4. optional FC ports
- The function of the FCF is to encapsulate the FC frames, received from the FC port, into the FCoE frames and also to de-encapsulate the FCoE frames, received from the Ethernet Bridge, to the FC frames.
- Upon receiving the incoming traffic, the FCoE switch inspects the **Ethertype** (used to indicate which protocol is encapsulated in the payload of an Ethernet frame) of the incoming frames and uses that to determine the destination.
    - If the Ethertype of the frame is FCoE, the switch recognizes that the frame contains an FC payload and forwards it to the FCF. From there, the FC is extracted from the FCoE frame and transmitted to FC SAN over the FC ports.
    - If the Ethertype is not FCoE, the switch handles the traffic as usual Ethernet traffic and forwards it over the Ethernet ports.

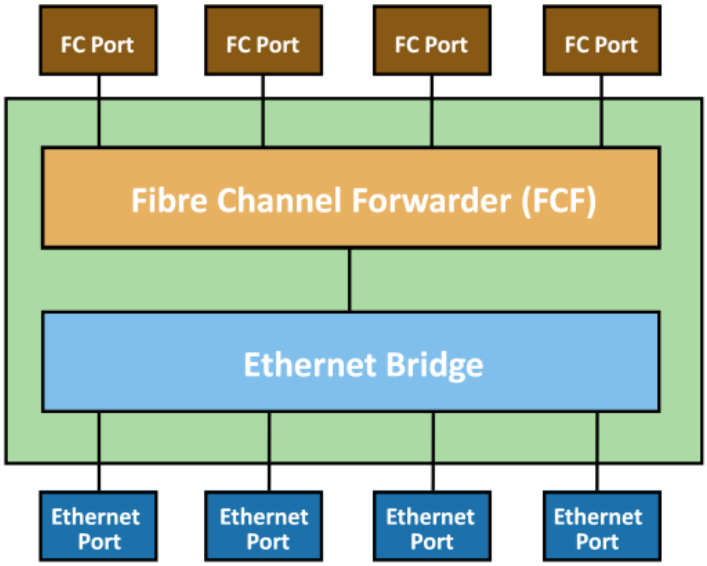


Fig FCoE switch generic architecture

NETWORK ATTACHED STORAGE (NAS)

- NAS is an IP based dedicated, high-performance file sharing and storage device.
- Enables NAS clients to share files over an IP network.
- Uses network and file-sharing protocols to provide access to the file data.
- Ex: Common Internet File System (CIFS) and Network File System (NFS).
- Enables both UNIX and Microsoft Windows users to share the same data seamlessly.
- NAS device uses its own operating system and integrated hardware and software components to meet specific file-service needs.
- Its operating system is optimized for file I/O which performs better than a general-purpose server.
- A NAS device can serve more clients than general-purpose servers and provide the benefit of server consolidation.

General-Purpose Servers versus NAS Devices

- A NAS device is optimized for file-serving functions such as storing, retrieving, and accessing files for applications and clients.
- As shown in Figure 1, a general-purpose server can be used to host any application because it runs a general-purpose operating system.
- Unlike a general-purpose server, a NAS device is dedicated to file-serving. It has specialized operating system dedicated to file serving by using industry-standard protocols.
- Some NAS vendors support features, such as native clustering for high availability.

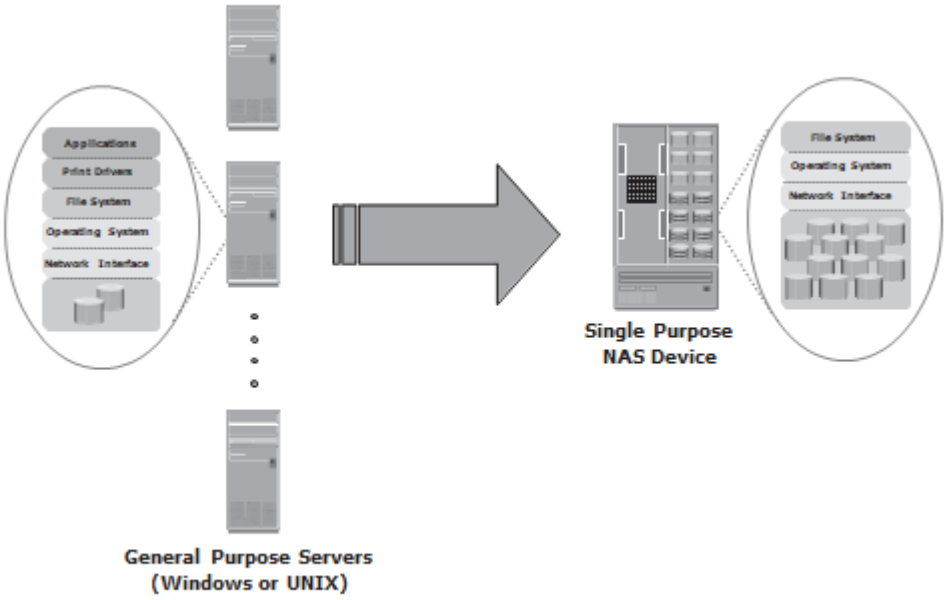


Figure 1: General purpose server versus NAS device

Benefits of NAS

NAS offers the following benefits:

- **Comprehensive access to information:** Enables efficient file sharing and supports many-to-one and one-to-many configurations. The many-to-one configuration enables a NAS device to

serve many clients simultaneously. The one-to-many configuration enables one client to connect with many NAS devices simultaneously.

- **Improved efficiency:** NAS delivers better performance compared to a general-purpose file server because NAS uses an operating system specialized for file serving.
- **Improved flexibility:** Compatible with clients on both UNIX and Windows platforms using industry-standard protocols. NAS is flexible and can serve requests from different types of clients from the same source.
- **Centralized storage:** Centralizes data storage to minimize data duplication on client workstations, and ensure greater data protection
- **Simplified management:** Provides a centralized console that makes it possible to manage file systems efficiently
- **Scalability:** Scales well with different utilization profiles and types of business applications because of the high-performance and low-latency design
- **High availability:** Offers efficient replication and recovery options, enabling high data availability. NAS uses redundant components that provide maximum connectivity options. A NAS device supports clustering tech- nology for failover.
- **Security:** Ensures security, user authentication, and file locking with industry-standard security schemas
- **Low cost:** NAS uses commonly available and inexpensive Ethernet components.
- **Ease of deployment:** Configuration at the client is minimal, because the clients have required NAS connection software built in.

File Systems and Network File Sharing

- A *file system* is a structured way to store and organize data files. Many file systems maintain a file access table to simplify the process of searching and accessing files.

1.Accessing a File System

- A file system must be mounted before it can be used. In most cases, the operating system mounts a local file system during the boot process.
- The mount process creates a link between the file system on the NAS and the operating system on the client.
- When mounting a file system, the operating system organizes files and directories in a tree-like structure and grants the privilege to the user to access this structure.
- The tree is rooted at a mount point. The mount point is named using operating system conventions. Users and applications can traverse the entire tree from the root to the leaf nodes as file system permissions allow.
- Files are located at leaf nodes, and directories and subdirectories are located at intermediate roots. The access to the file system terminates when the file system is unmounted. Figure 7 shows an example of a UNIX directory structure.

2.Network File Sharing

- Network file sharing refers to storing and accessing files over a network.
- In a file-sharing environment, the user who creates a file (the creator or owner of a file) determines the type of access (such as read, write, execute, append, and delete) to be given to other users and controls changes to the file.
- When multiple users try to access a shared file at the same time, a locking scheme is required to maintain data integrity and, at the same time, make this sharing possible.

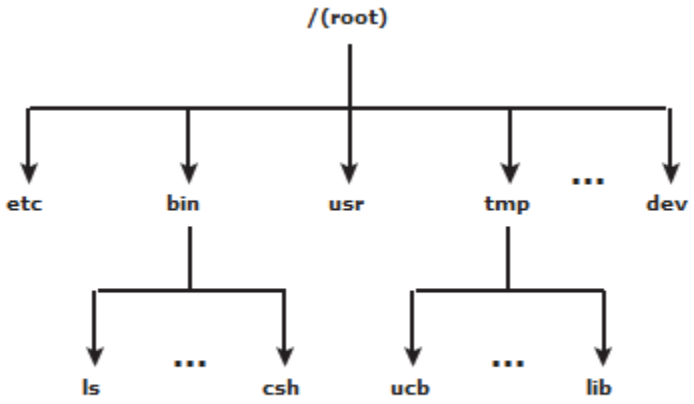


Figure 7 UNIX directory structure

- Some examples of file-sharing methods are file transfer protocol (FTP), Distributed File System (DFS), client-server models that use file-sharing protocols such as NFS and CIFS, and the peer-to-peer (P2P) model
- *FTP* is a client-server protocol that enables data transfer over a network. An FTP server and an FTP client communicate with each other using TCP as the transport protocol. FTP, as defined by the standard, is not a secure method of data transfer because it uses unencrypted data transfer over a network. FTP over Secure Shell (SSH) adds security to the original FTP specification. When FTP is used over SSH, it is referred to as Secure FTP (SFTP).
- A *distributed file system* (DFS) is a file system that is distributed across several hosts. A DFS can provide hosts with direct access to the entire file system, while ensuring efficient management and data security.
- Standard client-server file sharing protocols, such as NFS and CIFS, enable the owner of a file to set the required type of access, such as read-only or read-write, for a particular user or group of users. Using this protocol, the clients mount remote file systems that are available on dedicated file servers.
  - *A name service*, such as Domain Name System (DNS), and directory services such as Microsoft Active Directory, and Network Information Services (NIS), helps users identify

and access a unique resource over the network. A *name service protocol* such as the Lightweight Directory Access Protocol (LDAP) creates a namespace, which holds the unique name of every network resource and helps recognize resources on the network.

- A *peer-to-peer* (P2P) file sharing model uses a peer-to-peer network. P2P enables client machines to directly share files with each other over a network. Clients use a file sharing software that searches for other peer clients. This differs from the client-server model that uses file servers to store files for sharing.

Components of NAS

- NAS device has *two* key components (as shown in Fig 2.33): **NAS head** and **storage**.
- In some NAS implementations, the storage could be external to the NAS device and shared with other hosts.
- NAS head includes the following components:
  - CPU and memory
  - One or more network interface cards (NICs), which provide connectivity to the client network.
  - An optimized operating system for managing the NAS functionality. It translates file-level requests into block-storage requests and further converts the data supplied at the block level to file data
  - NFS, CIFS, and other protocols for file sharing
  - Industry-standard storage protocols and ports to connect and manage physical disk resources
- The NAS environment includes clients accessing a NAS device over an IP network using file-sharing protocols.

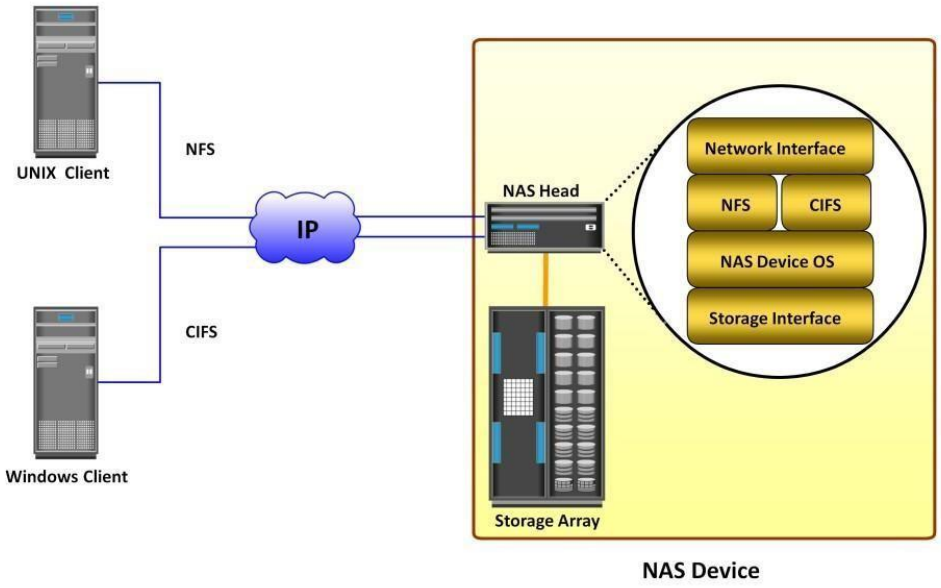


Fig 2.33 Components of NAS

NAS I/O Operation

- NAS provides *file-level data access* to its clients. File I/O is a high-level request that specifies the file to be accessed.
- Eg: a client may request a file by specifying its name, location, or other attributes. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrieve data.
- The process of handling I/Os in a NAS environment is as follows:

    1. The requestor (client) packages an I/O request into TCP/IP and forwards it through the network stack. The NAS device receives this request from the network.

2. The NAS device converts the I/O request into an appropriate physical storage request, which is a block-level I/O, and then performs the operation on the physical storage.
3. When the NAS device receives data from the storage, it processes and repackages the data into an appropriate file protocol response.
4. The NAS device packages this response into TCP/IP again and forwards it to the client through the network.

- Fig 2.34 illustrates the NAS I/O operation

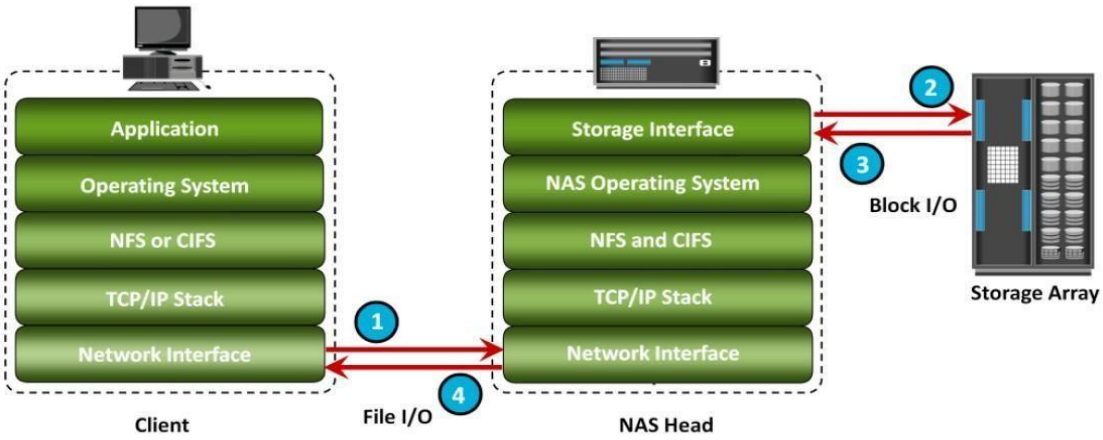


Fig 2.34 NAS I/O Operation

NAS Implementations

- Three common NAS implementations are unified, gateway, and scale-out.
  - The *unified* NAS consolidates NAS-based and SAN-based data access within a unified storage platform and provides a unified management interface for managing both the environments.
  - In a *gateway* implementation, the NAS device uses external storage to store and retrieve data, and unlike unified storage, there are separate administrative tasks for the NAS device and storage.
  - The *scale-out* NAS implementation pools multiple nodes together in a cluster. A node may consist of either the NAS head or storage or both. The cluster performs the NAS operation as a single entity.

1)Unified NAS

- Unified NAS performs file serving and storing of file data, along with providing access to block-level data.
- It supports both CIFS and NFS protocols for file access and iSCSI and FC protocols for block level access.
- Due to consolidation of NAS-based and SAN-based access on a single storage platform, unified NAS reduces an organization's infrastructure and management costs.
- A unified NAS contains one or more NAS heads and storage in a single system.
- NAS heads are connected to the storage controllers (SCs), which provide access to the storage. These storage controllers also provide connectivity to iSCSI and FC hosts. The storage may consist of different drive types, such as SAS, ATA, FC, and flash drives, to meet different workload requirements.

2)Unified NAS Connectivity

- Each NAS head in a unified NAS has front-end Ethernet ports, which connect to the IP network. The front-end ports provide connectivity to the clients and service the file I/O requests.
- Each NAS head has back-end ports, to provide connectivity to the storage controllers.
- iSCSI and FC ports on a storage controller enable hosts to access the storage directly or through a storage network at the block level.
- Figure 7-5 illustrates an example of unified NAS connectivity.

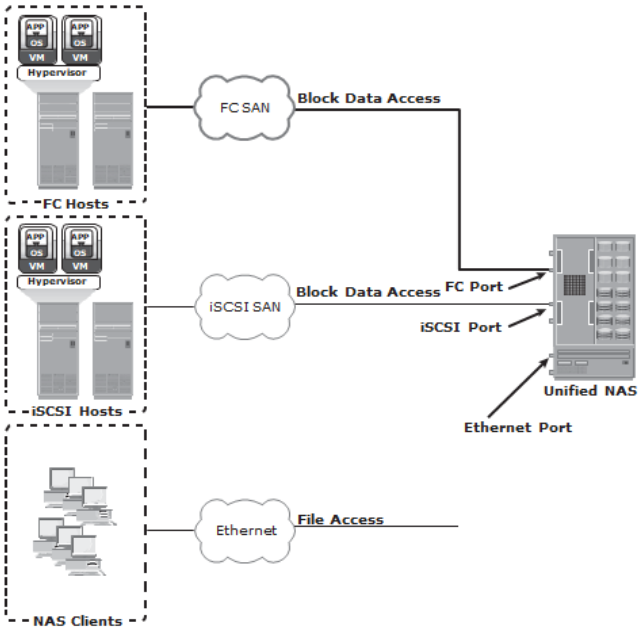


Figure 7-5: Unified NAS connectivity

3.Gateway NAS

- A gateway NAS device consists of one or more NAS heads and uses external and independently managed storage.
- Similar to unified NAS, the storage is shared with other applications that use block-level I/O. Management functions in this type of solution are more complex than those in a unified NAS environment because there are separate administrative tasks for the NAS head and the storage.
- A gateway solution can use the FC infrastructure, such as switches and directors for accessing SAN-attached storage arrays or direct- attached storage arrays.
- The gateway NAS is more scalable compared to unified NAS because NAS heads and storage arrays can be independently scaled up when required.
- Similar to a unified NAS, a gateway NAS also enables high utilization of storage capacity by sharing it with the SAN environment

..

4) Gateway NAS Connectivity

- In a gateway solution, the front-end connectivity is similar to that in a unified storage solution. Communication between the NAS gateway and the storage system in a gateway solution is achieved through a traditional FC SAN.
- To deploy a gateway NAS solution, factors, such as multiple paths for data, redundant fabrics, and load distribution, must be considered. Figure 7-6 illustrates an example of gateway NAS connectivity.

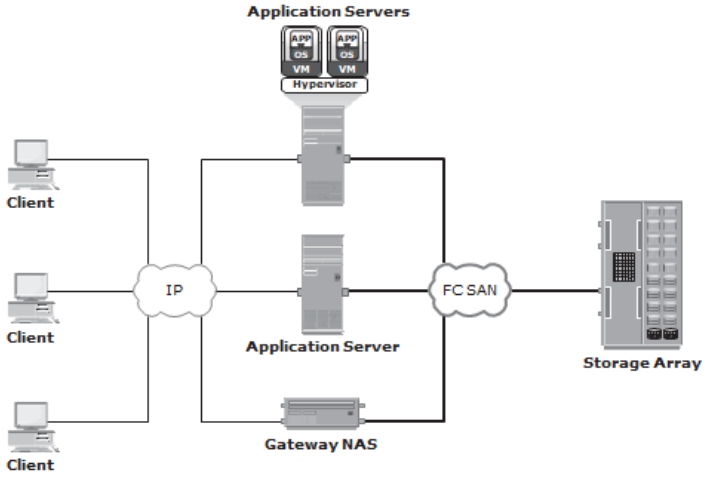


Figure 7-6: Gateway NAS connectivity

- Implementation of both unified and gateway solutions requires analysis of the SAN environment.
- This analysis is required to determine the feasibility of combining the NAS workload with the SAN workload.

.

5.)Scale-Out NAS

- Both unified and gateway NAS implementations provide the capability to scale- up their resources based on data growth and rise in performance requirements.
- Scaling up these NAS devices involves adding CPUs, memory, and storage to the NAS device.
- Scalability is limited by the capacity of the NAS device to house and use additional NAS heads and storage
- Scale-out NAS enables grouping multiple nodes together to construct a clustered NAS system.
- A scale-out NAS provides the capability to scale its resources by simply adding nodes to a clustered NAS architecture. The cluster works as a single NAS device and is managed centrally.
- Nodes can be added to the cluster, when more performance or more capacity is needed, without causing any downtime.
- Scale-out NAS provides the flexibility to use many nodes of moderate performance and availability characteristics to produce a total system that has better aggregate performance and availability.
- It also provides ease of use, low cost, and theoretically unlimited scalability.
- Scale-out NAS creates a single file system that runs on all nodes in the cluster. All information is shared among nodes, so the entire file system is accessible by clients connecting to any node in the cluster.
- Scale-out NAS stripes data across all nodes in a cluster along with mirror or parity protection. As data is sent from clients to the cluster, the data is divided and allocated to different nodes in parallel.
- When a client sends a request to read a file, the scale-out NAS retrieves the appropriate blocks from multiple nodes, recombines the blocks into a file, and presents the file to the client. As nodes are added, the file system grows dynamically and data is evenly distributed to every node.
- Each node added to the cluster increases the aggregate storage, memory, CPU, and network capacity. Hence, cluster performance also increases.

- Scale-out NAS is suitable to solve the "Big Data" challenges that enterprises and customers face today.
- It provides the capability to manage and store large, high-growth data in a single place with the flexibility to meet a broad range of performance requirements.

6 Scale-Out NAS Connectivity

- Scale-out NAS clusters use separate internal and external networks for back-end and front-end connectivity, respectively.
- An internal network provides connections for intra cluster communication, and an external network connection enables clients to access and share file data.
- Each node in the cluster connects to the internal network. The internal network offers high throughput and low latency and uses high-speed networking technology
- To enable clients to access a node, the node must be connected to the external Ethernet network.
- Redundant internal or external networks may be used for high availability.
- Figure 7-7 illustrates an example of scale-out NAS connectivity.

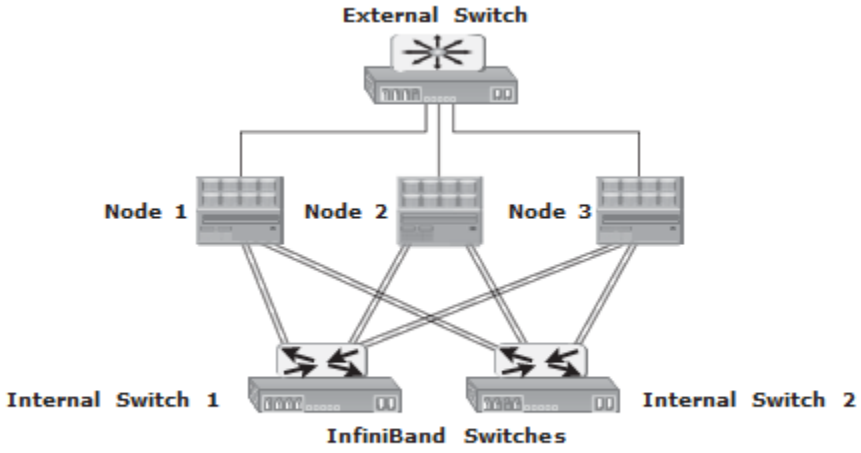


Figure 7-7: Scale-out NAS with dual internal and single external networks

NAS File Sharing Protocols

- NAS devices support multiple file-service protocols to handle file I/O requests
- Two common NAS file sharing protocols are:

- Common Internet File System (CIFS)
- Network File System (NFS)

- NAS devices enable users to share file data across different operating environments
- It provides a means for users to migrate transparently from one operating system to another

Network File System (NFS)

- NFS is a **client-server protocol** for file sharing that is commonly used on **UNIX systems**.
- NFS was originally based on the connectionless *User Datagram Protocol (UDP).*
- It uses *Remote Procedure Call (RPC)* as a method of inter-process communication between two computers.
- The NFS protocol provides a set of RPCs to access a remote file system for the following operations:
  - Searching files and directories
  - Opening, reading, writing to, and closing a file
  - Changing file attributes
  - Modifying file links and directories
- NFS creates a connection between the client and the remote system to transfer data.
- NFSv3 and earlier is a stateless protocol
- It does not maintain any kind of table to store information about open files and associated pointers. Each call provides a full set of arguments - a file handle, a particular position to read or write, and the versions of NFS - to access files on the server .
- Currently, three versions of NFS are in use:
  1. **NFS version 2 (NFSv2):** Uses *UDP* to provide a *stateless* network connection between a client and a server. Features, such as locking, are handled outside the protocol.
  2. **NFS version 3 (NFSv3):** Uses *UDP or TCP*, and is based on the *stateless protocol* design. It includes some new features, such as a 64-bit file size, asynchronous writes, and

additional file attributes to reduce refetching.

3. **NFS version 4 (NFSv4):** Uses TCP and is based on a *stateful protocol* design. It offers enhanced security. The latest NFS version 4.1 is the enhancement of NFSv4 and includes some new features, such as session model, parallel NFS (pNFS), and data retention.

Common Internet File System (CIFS)

- CIFS is a *client-server application* protocol
- It enables clients to access files and services on remote computers over **TCP/IP**.
- It is a public, or open, variation of **Server Message Block (SMB)** protocol.
- It provides following features to ensure data integrity:
    - It uses file and record locking to prevent users from overwriting the work of another user on a file or a record.
    - It supports fault tolerance and can automatically restore connections and reopen files that were open prior to an interruption. This feature depends on whether an application is written to take advantage of this.
    - CIFS is a stateful protocol because the CIFS server maintains connection information regarding every connected client. If a network failure or CIFS server failure occurs, the client receives a disconnection notification. User disruption is minimized if the application has the embedded intelligence to restore the connection. However, if the embedded intelligence is missing, the user must take steps to reestablish the CIFS connection.

Factors Affecting NAS Performance

- NAS uses IP network; therefore, bandwidth and latency issues associated with IP affect NAS performance.

- Network congestion is one of the most significant sources of latency (Figure 7-8) in a NAS environment.

Other factors that affect NAS performance at different levels follow:

1.Number of hops:

- A large number of hops can increase latency because IP processing is required at each hop, adding to the delay caused at the router.

2.Authentication with a directory service such as Active Directory or NIS:

- The authentication service must be available on the network with enough resources to accommodate the authentication load.
- Otherwise, a large number of authentication requests can increase latency.

3.Retransmission:

- Link errors and buffer overflows can result in retransmission. This causes packets that have not reached the specified destination to be re-sent.
- Care must be taken to match both speed and duplex settings on the network devices and the NAS heads. Improper configuration might result in errors and retransmission, adding to latency.

4.Overutilized routers and switches:

- The amount of time that an over utilized device in a network takes to respond is always more than the response time of an optimally utilized or underutilized device.
- Network administrators can view utilization statistics to determine the optimum utilization of switches and routers in a network.
- Additional devices should be added if the current devices are over utilized.

5.File system lookup and metadata requests:

- NAS clients access files on NAS devices.
- The processing required to reach the appropriate file or directory can cause delays.
- Sometimes a delay is caused by deep directory structures and can be resolved by flattening the directory structure.
- Poor file system layout and an over utilized disk system can also degrade performance.

6.Over utilized NAS devices:

- Clients accessing multiple files can cause high utilization levels on a NAS device, which can be determined by viewing utilization statistics. High memory,
- CPU, or disk subsystem utilization levels can be caused by a poor file system structure or insufficient resources in a storage subsystem.

7.Over utilized clients:

- The client accessing CIFS or NFS data might also be over utilized.
- An over utilized client requires a longer time to process the requests and responses. Specific performance-monitoring tools are available for various operating systems to help determine the utilization of client resources.

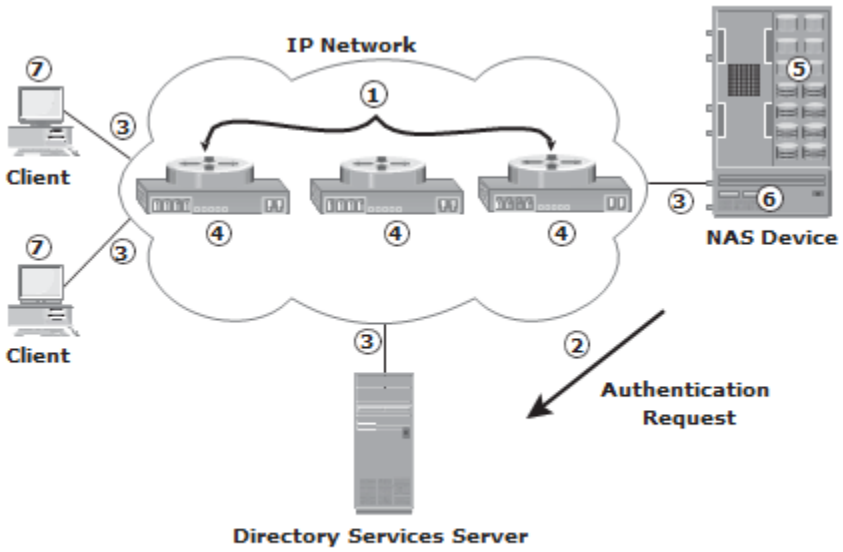


Figure 7-8: Causes of latency

MODULE-2

- In 1987, Patterson, Gibson, and Katz at the University of California, Berkeley, published a paper titled "A Case for **Redundant Arrays of Inexpensive Disks (RAID)**."
- RAID is the use of small-capacity, inexpensive disk drives as an alternative to large-capacity drives common on mainframe computers.
- Later RAID has been redefined to refer to ***independent*** disks to reflect advances in the storage technology.

RAID Implementation Methods

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

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

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

RAID Array Components

- A *RAID array* is an enclosure that contains a number of disk drives and sup- porting hardware to implement RAID. A subset of disks within a RAID arraycan be grouped to form logical associations called logical arrays, also known as a *RAID set* or a *RAID group* (see Figure 3-1).

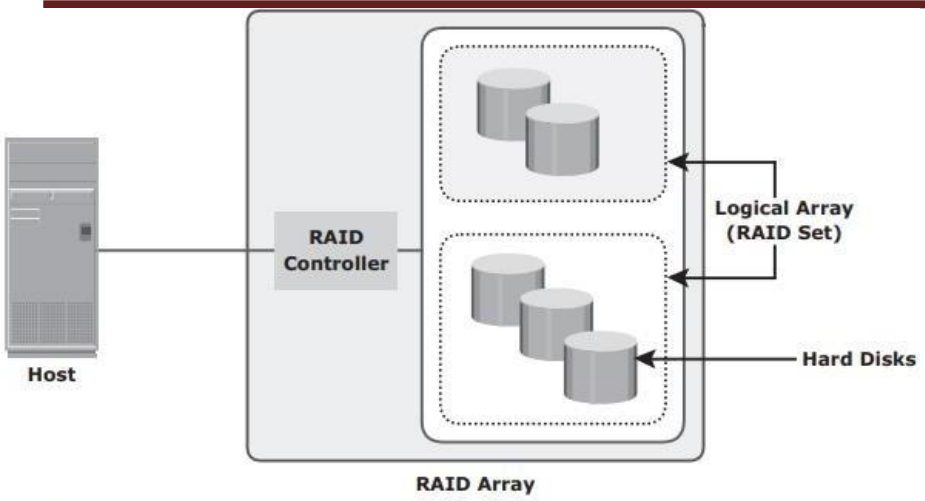


Figure 3-1: Components of a RAID array

RAID Techniques

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

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**.
- The below figure shows physical and logical representations of a striped RAID set.
- **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 disk in the set.
- All strips in a stripe have the same number of blocks.
    - ✓ Having a smaller strip size means that data is broken into smaller pieces while spread across the disks.

- **Stripe size** is a multiple of strip size by the number of **data** disks in the RAID set.
  - ✓ Eg: In a 5 disk striped RAID set with a strip size of 64 KB, the stripe size is 320KB (64KB x 5).
- **Stripe width** refers to the number of *data* strips in a stripe.
- Striped RAID does not provide any data protection unless parity or mirroring is used.

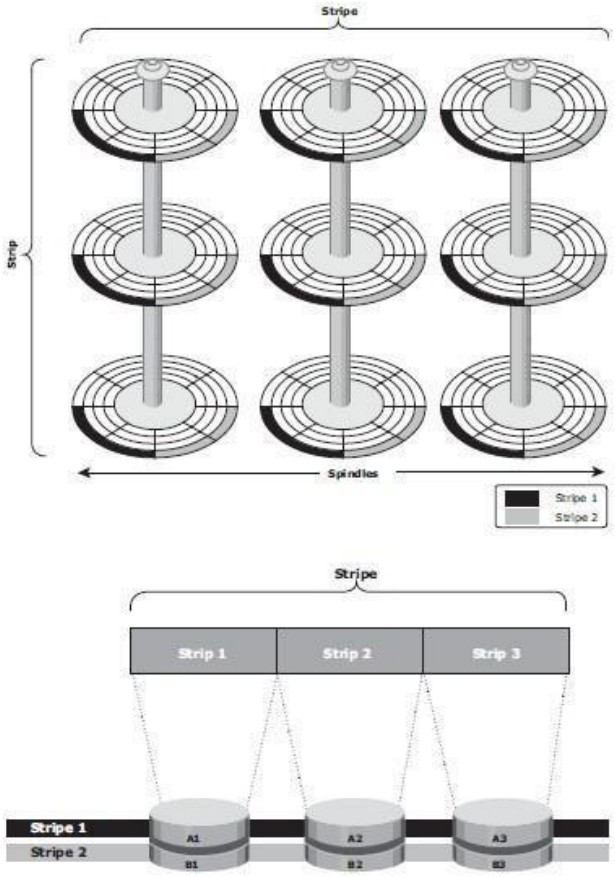


Fig : Striped RAID set

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 below) 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.

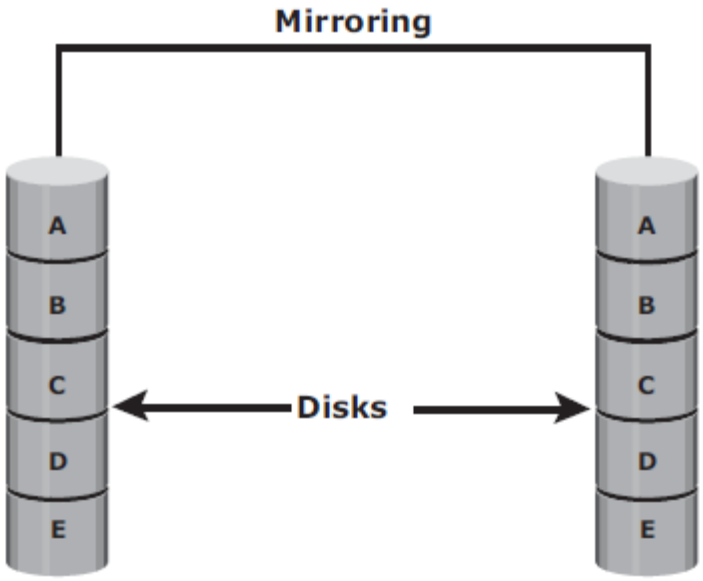


Fig: Mirrored disks in an array

- Advantages:
  - ✓ complete data redundancy
  - ✓ mirroring enables fast recovery from disk failure.
  - ✓ data protection
- Mirroring is not a substitute for data backup. Mirroring constantly captures changes in thedata, whereas a backup captures point-in-time images of the data.

Disadvantages:

- ✓ Mirroring involves duplication of data — the amount of storage capacity needed is twice the amount of data being stored.
- ✓ Expensive

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 shows a parity RAID set.

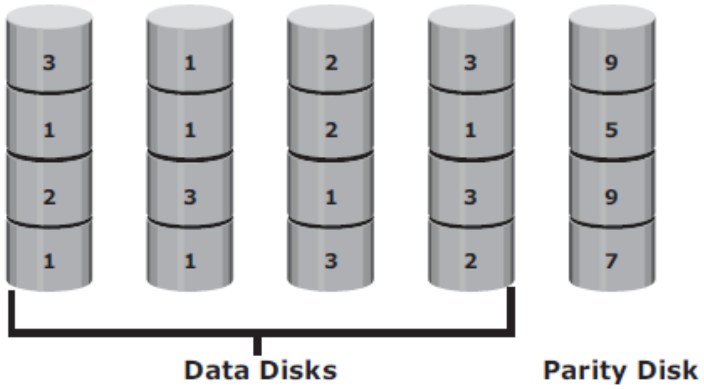


Fig : Parity RAID

- 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 of the rest of the elements from the parity value.
- Here, computation of parity is represented as an arithmetic sum of the data. However, parity calculation is a bitwise XOR operation.

XOR Operation:

- A bit-by-bit Exclusive -OR (XOR) operation takes two bit patterns of equal length and performs the logical XOR operation on each pair of corresponding bits.
- The result in each position is 1 if the two bits are different, and 0 if they are the same.

- The truth table of the XOR operation is shown below (A and B denote inputs and C, the output the XOR operation).

Table 1.1: Truth table for XOR Operation

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

- If any of the data from A, B, or C is lost, it can be reproduced by performing an XOR operation on the remaining available data.
- Eg: if a disk containing all the data from A fails, the data can be regenerated by performing an XOR between B and C.
- Advantages:
  - ✓ Compared to mirroring, parity implementation considerably reduces the **cost** associated with data protection.
- Disadvantages:
  - ✓ 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 array.
- For parity RAID, the stripe size calculation does not include the parity strip.
- Eg: in a five (4 + 1) disk parity RAID set with a strip size of 64 KB, the stripe size will be 256 KB (64 KB x 4).

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 shows the commonly used RAID levels

Table : 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

RAID 0

- **RAID 0** configuration uses *data striping techniques*, where data is striped across all the disks within a RAID set. Therefore it utilizes the full storage capacity of a RAID set.
- To read data, all the strips are put back together by the controller.
- Fig shows RAID 0 in an array in which data is striped across five disks.

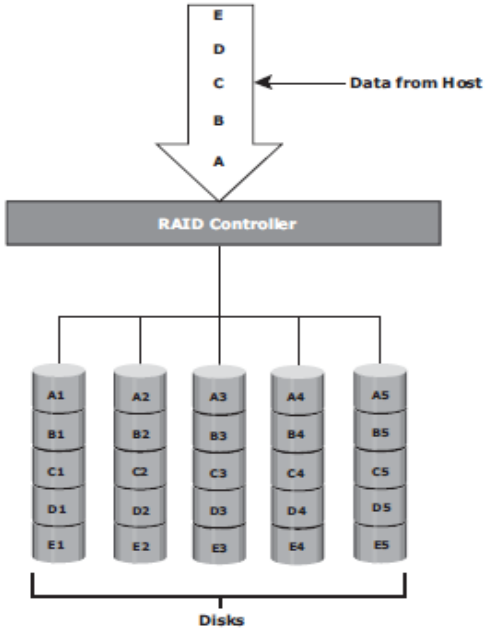


Fig : RAID 0

- When the number of drives in the RAID set increases, performance improves because more data can be read or written simultaneously.
- RAID 0 is a good option for applications that need high I/O throughput.
- However, if these applications require high availability during drive failures, RAID 0 does not provide data protection and availability.

RAID 1

- **RAID 1** is based on the *mirroring* technique.
- In this RAID configuration, data is mirrored to provide *fault tolerance* (see Fig). A
- RAID 1 set consists of two disk drives and every write is written to both disks.
- The mirroring is transparent to the host.
- During disk failure, the impact on data recovery in RAID 1 is the least among all RAID implementations. This is because the RAID controller uses the mirror drive for data recovery.
- RAID 1 is suitable for applications that require high availability and cost is no constraint.

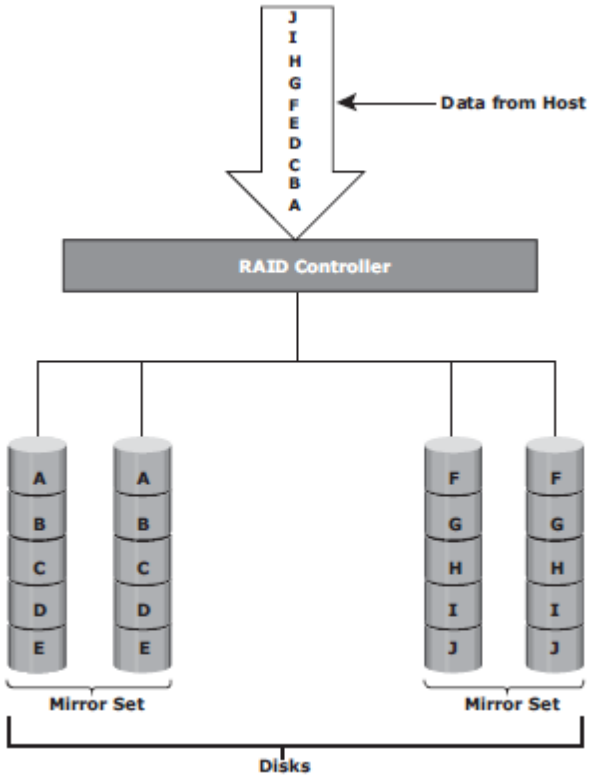


Fig : RAID 1

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).

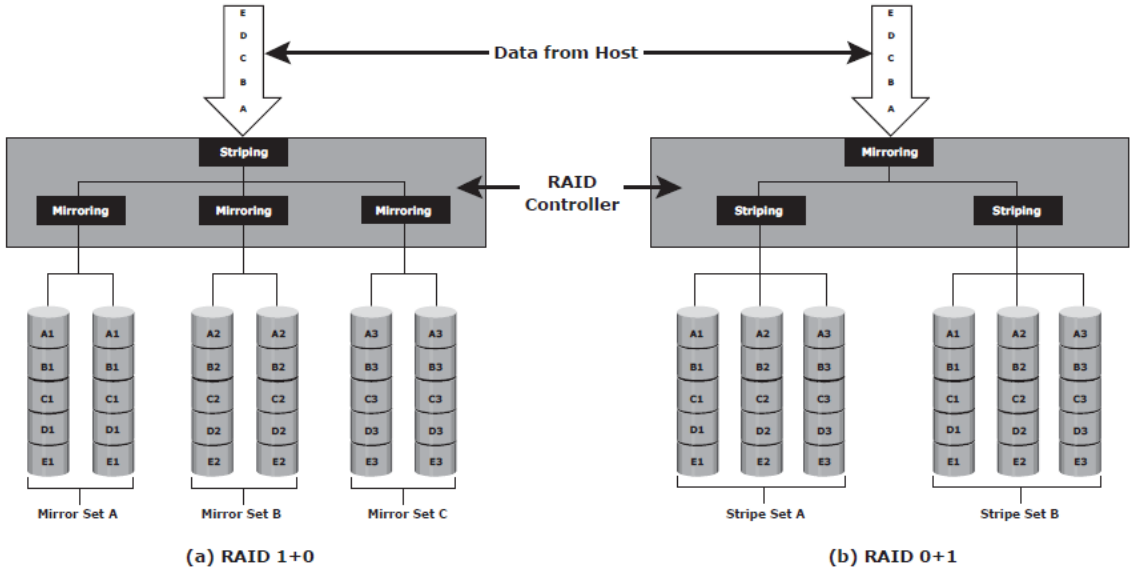


Fig: Nested RAID

RAID 1+0:

- RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0.
- RAID 1+0 performs well for workloads with small, random, write-intensive I/Os.
- Some applications that benefit from RAID 1+0 include the following:
  - ✓ High transaction rate Online Transaction Processing (OLTP)
  - ✓ Large messaging installations
  - ✓ Database applications with write intensive random access workloads
- **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 the data are striped across multiple disk drive pairs in a RAID set.
- When replacing a failed drive, only the mirror is rebuilt. The disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation.

Working of RAID 1+0:

- Eg: consider an example of six disks forming a RAID 1+0 (RAID 1 first and then RAID 0) set.
- These six disks are paired into three sets of two disks, where each set acts as a RAID 1 set (mirrored pair of disks). Data is then striped across all the three mirrored sets to form RAID 0.

- Following are the steps performed in RAID 1+0 (see Fig 1.16 [a]):
  - ✓ Drives 1+2 = RAID 1 (Mirror Set A)
  - ✓ Drives 3+4 = RAID 1 (Mirror Set B)
  - ✓ Drives 5+6 = RAID 1 (Mirror Set C)
- Now, RAID 0 striping is performed across sets A through C.
- In this configuration, if drive 5 fails, then the mirror set C alone is affected. It still has drive 6 and continues to function and the entire RAID 1+0 array also keeps functioning.
- Now, suppose drive 3 fails while drive 5 was being replaced. In this case the array still continues to function because drive 3 is in a different mirror set.
- So, in this configuration, up to three drives can fail without affecting the array, as long as they are all in different mirror sets.
- **RAID 0+1** is also called a mirrored stripe.
- The basic element of RAID 0+1 is a stripe. This means that the process of striping data across disk drives is performed initially, and then the entire stripe is mirrored.
- In this configuration if one drive fails, then the entire stripe is faulted.

Working of RAID 0+1:

- Eg: Consider the same example of six disks forming a RAID 0+1 (that is, RAID 0 first and then RAID 1).
- Here, six disks are paired into two sets of three disks each.
- Each of these sets, in turn, act as a RAID 0 set that contains three disks and then these two sets are mirrored to form RAID 1.
- Following are the steps performed in RAID 0+1 (see Fig 1.16 [b]):
  - ✓ Drives 1 + 2 + 3 = RAID 0 (Stripe Set A)
  - ✓ Drives 4 + 5 + 6 = RAID 0 (Stripe Set B)
- These two stripe sets are mirrored.
- If one of the drives, say drive 3, fails, the entire stripe set A fails.
- A rebuild operation copies the entire stripe, copying the data from each disk in the healthy stripe to an equivalent disk in the failed stripe.
- 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.
- 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 shows the RAID 3 implementation

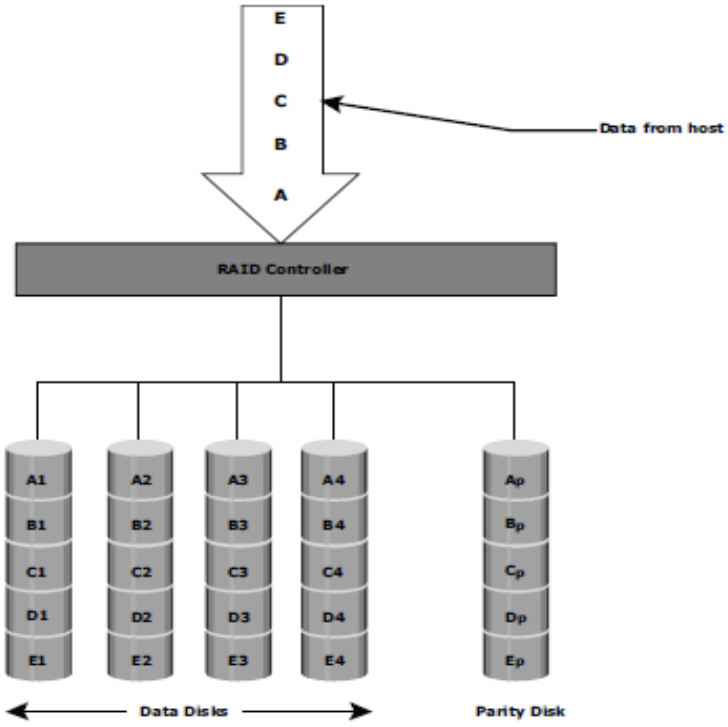


Fig: RAID 3

RAID 4

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

- Unlike RAID 3, data disks in RAID 4 can be accessed independently so that specific data elements can be read or written on single disk without read or write of an entire stripe. RAID 4 provides good read throughput and reasonable write throughput.

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 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.

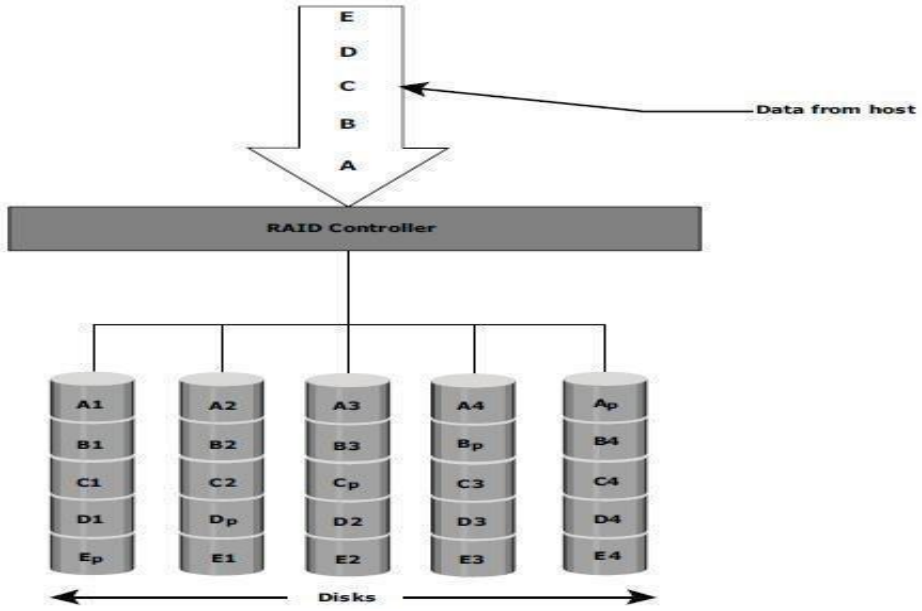


Fig : RAID 5

RAID 6

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

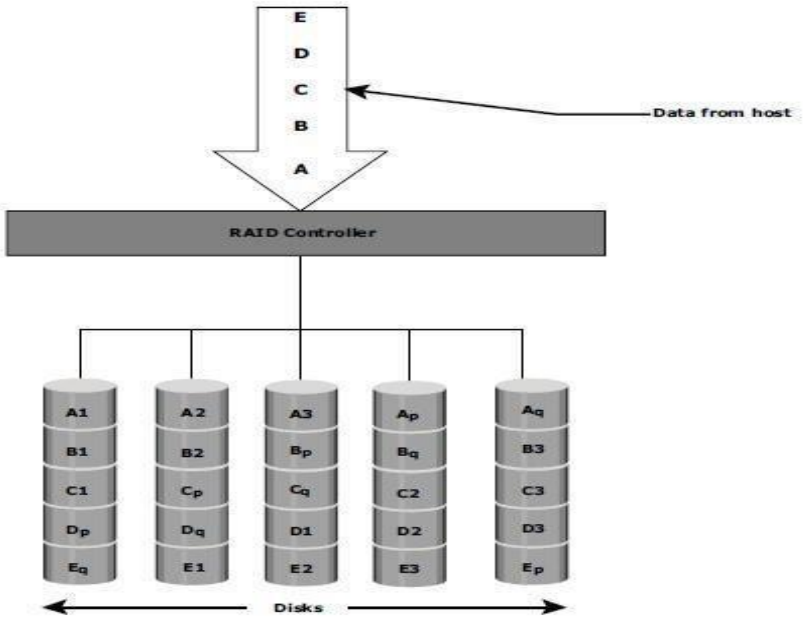


Fig: RAID 6

RAID Impact on Disk Performance

- When choosing a RAID type, it is imperative to consider its impact on disk performance and application IOPS.
- In both mirrored (RAID 1) and parity RAID (RAID 5) configurations, every write operation translates into more I/O overhead for the disks which is referred to as **write penalty**.
- In a RAID 1 implementation, every write operation must be performed on two disks configured as a mirrored pair. **The write penalty is 2.**
- In a RAID 5 implementation, a write operation may manifest as four I/O operations. When performing small I/Os to a disk configured with RAID 5, the controller has to read, calculate, and write a parity segment for every data write operation.
- Fig illustrates a single write operation on RAID 5 that contains a group of five disks.

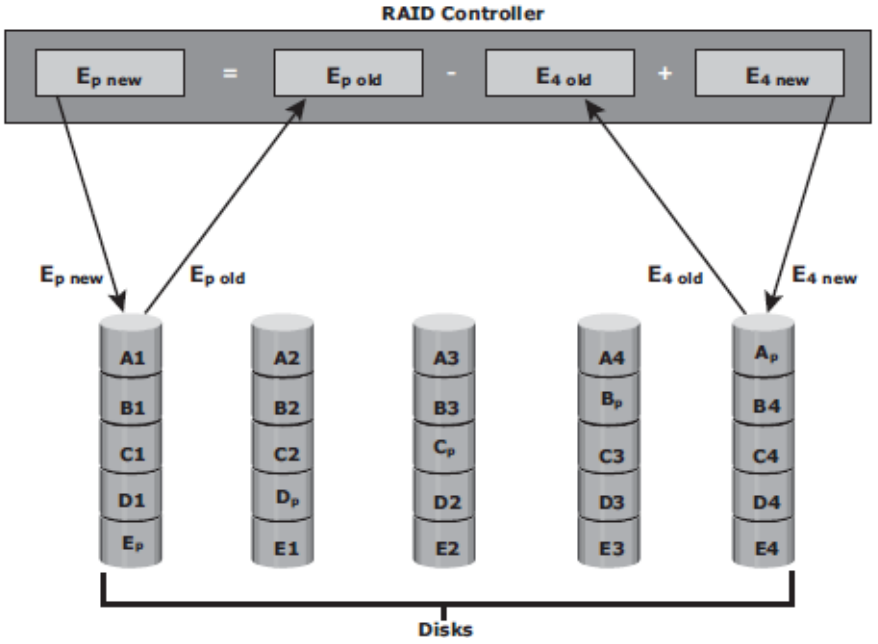


Fig : Write Penalty in RAID 5

- Four of these disks are used for data and one is used for parity.
  - ✓ The **parity (E_p)** at the controller is calculated as follows:

$$E_p = E_1 + E_2 + E_3 + E_4 \text{ (XOR operations)}$$

- Whenever the controller performs a write I/O, parity must be computed by reading the old parity (E_p old) and the old data (E_4 old) from the disk, which means two read I/Os.
- The new parity (E_p new) is computed as follows:

$$E_p \text{ new} = E_p \text{ old} - E_4 \text{ old} + E_4 \text{ new (XOR operations)}$$

- After computing the new parity, the controller completes the write I/O by doing two write I/Os for the new data and the new parity onto the disks..
- Therefore, the controller performs two disk reads and two disk writes for every write operation, and **the write penalty is 4**.
- In RAID 6, which maintains dual parity, a disk write requires **three read operations**: two parity and one data.
- After calculating both new parities, the controller performs **three write operations**: two parity and an I/O.
- Therefore, in a RAID 6 implementation, the controller performs six I/O operations for each write I/O, and the **write penalty is 6**.

RAID Comparison

RAID	MIN. DISKS	STORAGE EFFICIENCY %	COST	READ PERFORMANCE	WRITE PERFORMANCE	WRITE PENALTY	PROTECTION
0	2	100	Low	Good for both random and sequential reads	Good	No	No protection
1	2	50	High	Better than single disk	Slower than single disk because every write must be committed to all disks	Moderate	Mirror protection
3	3	[(n-1)/n] ¥ 100 where n= number of disks	Moderate	Fair for random reads and good for sequential reads	Poor to fair for small random writes and fair for large, sequential writes	High	Parity protection for single disk failure
4	3	[(n-1)/n] ¥ 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
5	3	[(n-1)/n] ¥ 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
6	4	[(n-2)/n] ¥ 100 where n= number of disks	Moderate but more than RAID 5.	Good for random and sequential reads	Poor to fair for random writes and fair for sequential writes	Very High	Parity protection for two disk failures
1+0 and 0+1	4	50	High	Good	Good	Moderate	Mirror protection

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):
  - ✓ 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).

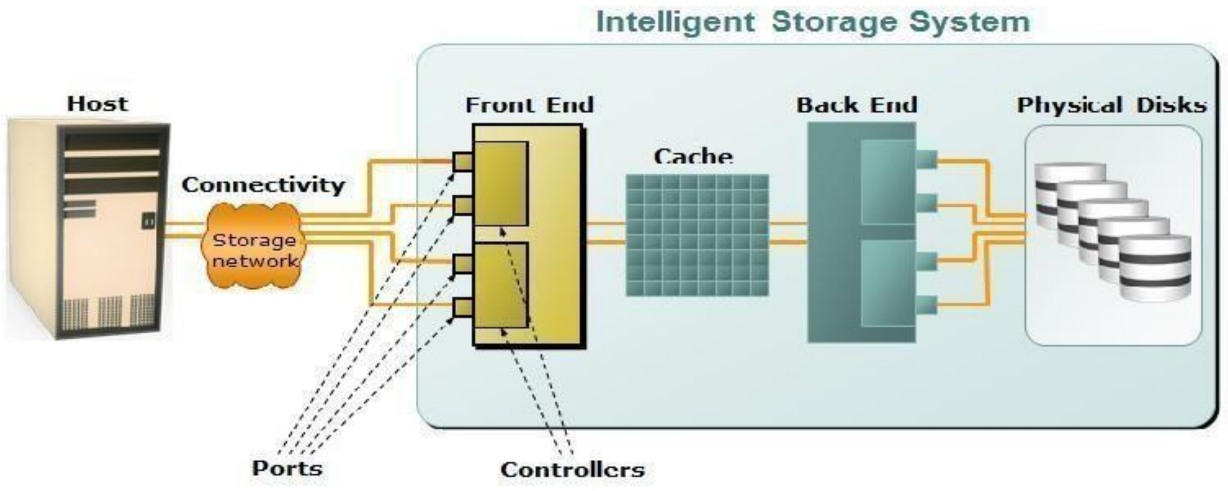


Fig : Components of an Intelligent Storage System

Front End

- The front end provides the interface between the storage system and the host.
- It consists of two components:
  - i. Front-End Ports 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.

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.

Structure Of Cache

- Cache is organized into pages, 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 whereas the tag RAM tracks the location of the data in the data store (see Fig) and in the disk.
- Entries in tag RAM indicate where data is found in cache and where the data belongs on the disk.
- Tag RAM includes a dirty bit flag, which indicates whether the data in cache has been committed to the disk.
- 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.

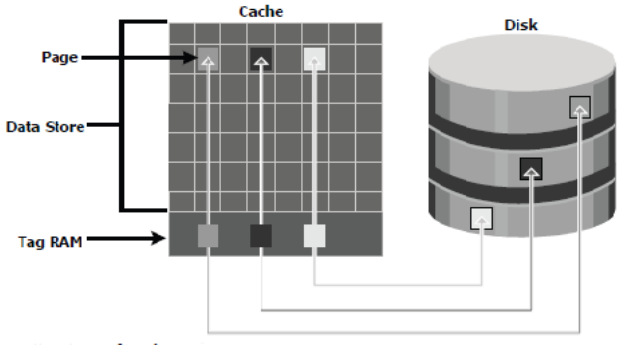


Fig : Structure of cache

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 [a]).This provides a fast response time to the host (about a millisecond).
- If the requested data is not found in cache, it is called a **cache miss** and the data must be read from the disk(see Fig [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 prefetch 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.

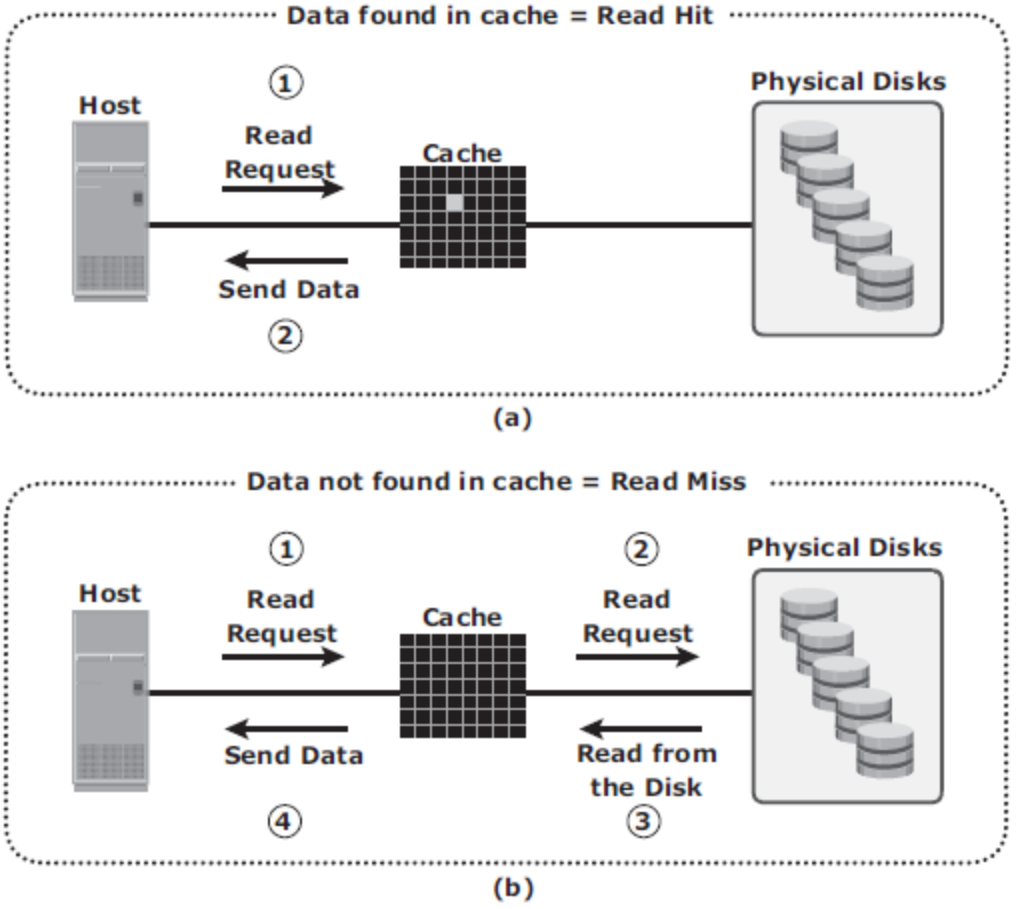


Fig: Read hit and read miss

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,

the risks of data loss are low but write response time is longer because of the disk operations.

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

Cache Implementation

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

Cache Management

- Cache is a finite and expensive resource that needs proper management.
- Even though modern intelligent storage systems come with a large amount of cache, when all cache pages are filled, some pages have to be freed up to accommodate new data and avoid performance degradation.
- Various cache management algorithms are implemented in intelligent storage systems to proactively maintain a set of free pages and a list of pages that can be potentially freed up whenever required.
- The most commonly used algorithms are listed below:
  - ✓ **Least Recently Used (LRU):** An algorithm that continuously monitors data access in cache and identifies the cache pages that have not been accessed for a long time. LRU either frees up these pages or marks them for reuse. This algorithm is based on the assumption that data which hasn't been accessed for a while will not be requested by the host.

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

- As cache fills, the storage system must take action to **flush dirty pages** (data written into the cache but not yet written to the disk) to manage space availability.
- **Flushing** is the process that commits data from cache to the disk.
- On the basis of the I/O access rate and pattern, high and low levels called **watermarks** are set in cache to manage the flushing process.
- **High watermark (HWM)** is the cache utilization level at which the storage system starts high-speed flushing of cache data.
- **Low watermark (LWM)** is the point at which the storage system stops flushing data to the disks.
- The *cache utilization level*, as shown in Fig, drives the mode of flushing to be used:
  - ✓ **Idle flushing:** Occurs continuously, at a modest rate, when the cache utilization level is between the high and low watermark.
  - ✓ **High watermark flushing:** Activated when cache utilization hits the high watermark. The storage system dedicates some additional resources for flushing. This type of flushing has some impact on I/O processing.
  - ✓ **Forced flushing:** Occurs in the event of a large I/O burst when cache reaches 100 percent of its capacity, which significantly affects the I/O response time. In forced flushing, system flushes the cache on priority by allocating more resources.

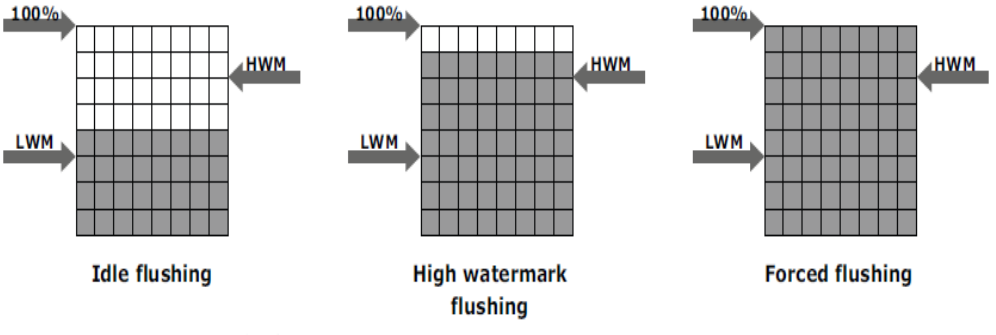


Fig : Types of flushing

Cache Data Protection

- Cache is volatile memory, so a power failure or any kind of cache failure will cause loss of the data that is not yet committed to the disk.

- This risk of losing uncommitted data held in cache can be mitigated using
  i. cache mirroring
  ii. cache vaulting

Cache mirroring

- ✓ Each write to cache is held in two different memory locations on two independent memory cards. In the event of a cache failure, the write data will still be safe in the mirrored location and can be committed to the disk.
- ✓ Reads are staged from the disk to the cache, therefore, in the event of a cache failure, the data can still be accessed from the disk.
- ✓ In cache mirroring approaches, the problem of maintaining *cache coherency* is introduced.
- ✓ Cache coherency means that data in two different cache locations must be identical at all times. It is the responsibility of the array operating environment to ensure coherency.

Cache vaulting

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

Back End

- The **back end** provides an interface between cache and the physical disks.
- It consists of two components:
  i. Back-end ports
  ii. Back-end controllers.
- The back end controls data transfers between cache and the physical disks.
- From cache, data is sent to the back end and then routed to the destination disk.

- Physical disks are connected to *ports* on the back end.
- The *back end controller* communicates with the disks when performing reads and writes and also provides additional, but limited, temporary data storage.
- The algorithms implemented on back-end controllers provide error detection and correction, and also RAID functionality.
- For high data protection and high availability, storage systems are configured with dual controllers with multiple ports.

Physical Disk

- A physical disk stores data persistently.
- Physical disks are connected to the back-end storage controller and provide persistent data storage.
- Modern intelligent storage systems provide support to a variety of disk drives with different speeds and types, such as FC, SATA, SAS, and flash drives.
- They also support the use of a mix of flash, FC, or SATA within the same array.

Types of Intelligent Storage Systems

- An intelligent storage system is divided into following two categories:
    1. High-end storage systems
    2. Midrange storage systems
- High-end storage systems have been implemented with active-active configuration, whereas midrange storage systems have been implemented with active-passive configuration.
- The distinctions between these two implementations are becoming increasingly insignificant.

High-end Storage Systems

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

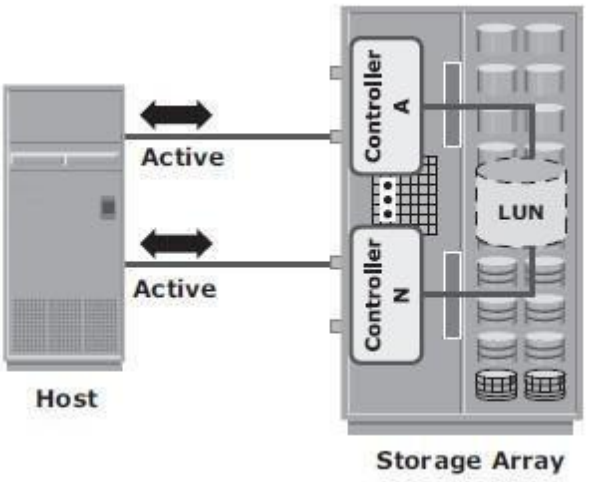


Fig : Active-active configuration

Advantages of High-end storage:

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

Midrange Storage System

- Midrange storage systems are also referred to as **Active-Passive Arrays** and they are best suited for small- and medium-sized enterprises.
- They also provide optimal storage solutions at a *lower cost*.
- In an *active-passive* array, a host can perform I/Os to a LUN only through the paths to the **owning controller** of that LUN. These paths are called *Active Paths*. The other paths are *passive* with respect to this LUN.

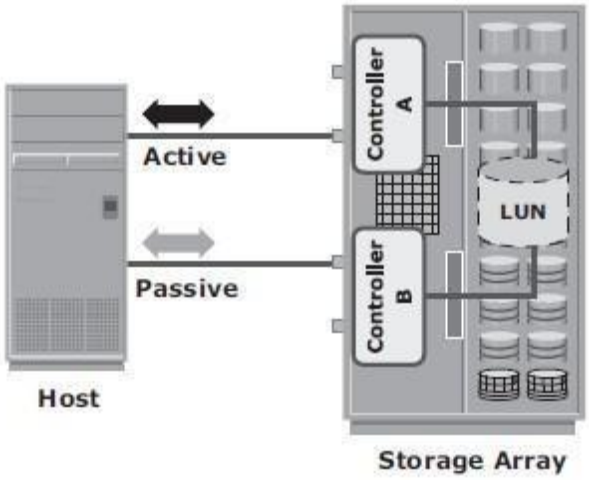


Fig : Active-passive configuration

- As shown in Fig, the host can perform reads or writes to the LUN only through the path to controller A, as controller A is the owner of that LUN.
- The path to controller B remains **Passive** and no I/O activity is performed through this path.
- Midrange storage systems are typically designed with two controllers, each of which contains host interfaces, cache, RAID controllers, and disk drive interfaces.
- Midrange arrays are designed to meet the requirements of small and medium enterprise applications; therefore, they host less storage capacity and cache than high-end storage arrays.
- There are also fewer front-end ports for connection to hosts.
- But they ensure high redundancy and high performance for applications with predictable workloads.
- They also support array-based local and remote replication.

FIBRE CHANNEL STORAGE AREA NETWORKS

SAN is a high-speed dedicated network of servers and shared storage. Commom SAN deployments are:

- ✓ FC SAN
- ✓ IP SAN

Fibre Channel: Overview

- The FC architecture forms the fundamental construct of the SAN infrastructure.
- **Fibre Channel** is a high-speed network technology that runs on high-speed optical fiber cables (preferred for front-end SAN connectivity) and serial copper cables (preferred for back-end disk connectivity).
- The FC technology was created to meet the demand for increased speeds of data transfer among computers, servers, and mass storage subsystems.
- High data transmission speed is an important feature of the FC networking technology.
- The initial implementation offered a throughput of 200 MB/s (equivalent to a raw bit rate of 1Gb/s), which was greater than the speeds of Ultra SCSI (20 MB/s), commonly used in DAS environments
- The FC architecture is highly scalable, and theoretically, a single FC network can accommodate approximately 15 million devices.

The SAN and Its Evolution

- A SAN carries data between servers (or *hosts*) and storage devices through Fibre Channel network (Figure).
- A SAN enables storage consolidation and enables storage to be shared across multiple servers.
- This improves the utilization of storage resources compared to direct-attached storage architecture and reduces the total amount of storage an organization needs to purchase and manage
- SAN also enables organizations to connect geographically dispersed servers and storage.

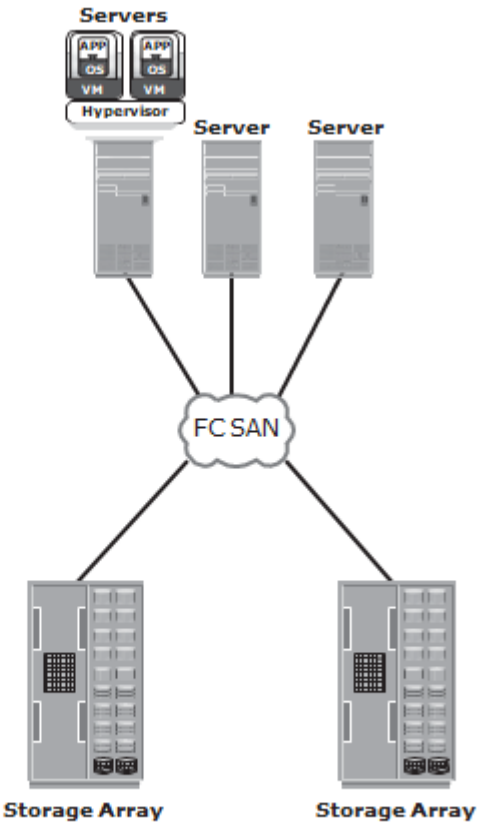


Figure : FC SAN implementation

- In its earliest implementation, the FC SAN was a simple grouping of hosts and storage devices connected to a network using an FC hub as a connectivity device.
- This configuration of an FC SAN is known as a *Fibre Channel Arbitrated Loop* (FC-AL). Use of hubs resulted in isolated FC-AL SAN islands because hubs provide limited connectivity and bandwidth.
- The inherent limitations associated with hubs gave way to high-performance FC *switches*.
- Use of switches in SAN improved connectivity and performance and enabled FC SANs to be highly scalable. This enhanced data accessibility to applications across the enterprise.
- Now, FC-AL has been almost abandoned for FC SANs due to its limitations but still survives as a back-end connectivity option to disk drives.
- Below Figure illustrates the FC SAN evolution from FC-AL to enterprise SANs.

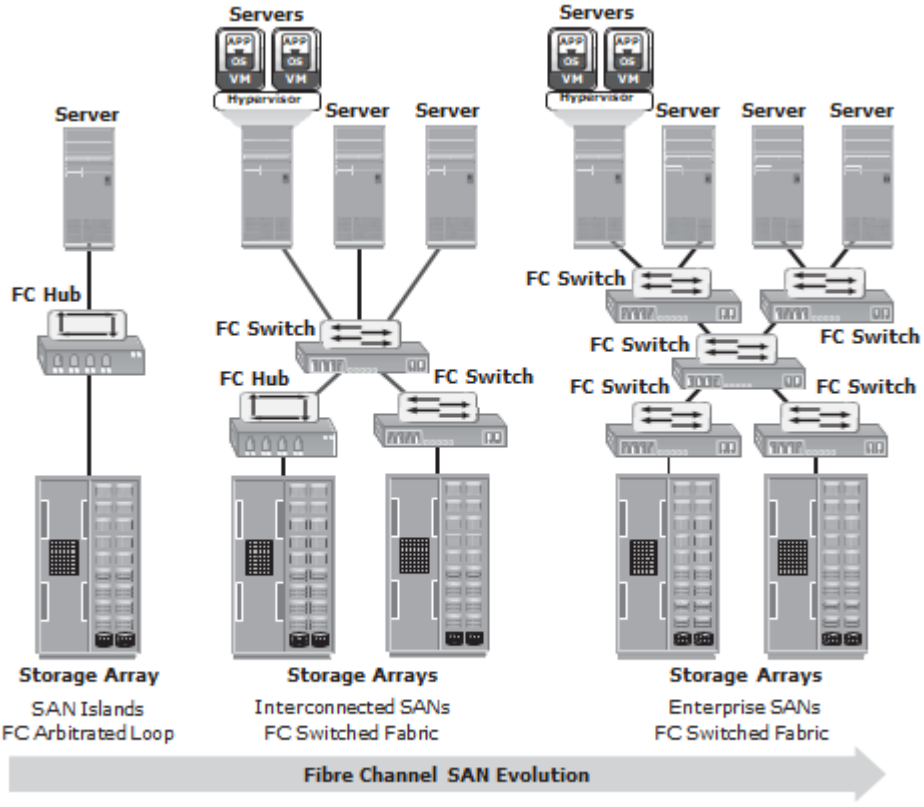


Figure FC SAN evolution

Components of FC SAN

- Components of FC SAN infrastructure are:

1) **Node Ports,**
2) **Cables**
3) **Connectors,**
4) **Interconnecting Devices (Such As Fc Switches Or Hubs),**
5) **San Management Software**.

Node Ports

- In fibre channel, devices such as hosts, storage and tape libraries are all referred to as **Nodes.**
- Each node is a **source or destination** of information for one or more nodes.

- Each node requires one or more ports to provide a physical interface for communicating with other nodes.

A port operates in full-duplex data transmission mode with a **transmit (Tx) link and receive (Rx) link** (see Fig 2.1).

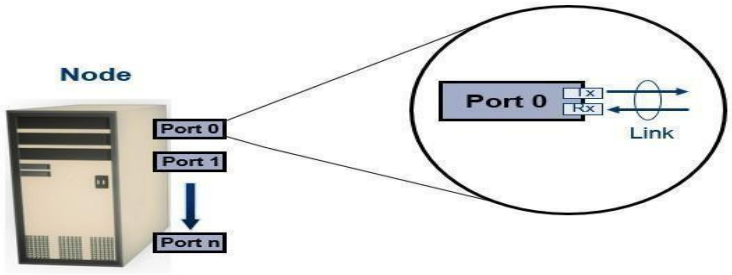


Fig 2.1: Nodes, Ports, links

Cables

- SAN implementations use optical fiber cabling.
- Copper can be used for shorter distances for back-end connectivity
- Optical fiber cables carry data in the form of light.
- There are two types of optical cables :**Multi-Mode And Single-Mode.**

1) **Multi-mode fiber (MMF)** cable carries multiple beams of light projected at different angles simultaneously onto the core of the cable (see Fig 2.2 (a)).
    - In an MMF transmission, multiple light beams traveling inside the cable tend to disperse and collide. This collision weakens the signal strength after it travels a certain distance — a process known as *modal dispersion*.
    - MMFs are generally used within data centers for shorter distance runs
2) **Single-mode fiber (SMF)** carries a single ray of light projected at the center of the core (see Fig 2.2 (b)).
    - In an SMF transmission, a single light beam travels in a straight line through the core of the fiber.
    - The small core and the single light wave limits modal dispersion. Among all types of fibre cables, single-mode provides minimum signal attenuation over maximum

distance (up to 10 km).

- A single-mode cable is used for long-distance cable runs, limited only by the power of the laser at the transmitter and sensitivity of the receiver.
- SMFs are used for longer distances.

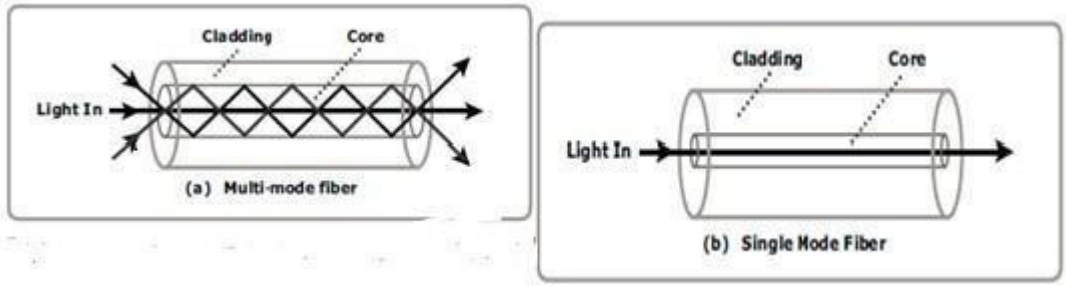


Fig 2.2: Multimode fiber and single-mode fiber

Connectors

- They are attached at the end of the cable to enable swift connection and disconnection of the cable to and from a port.
- A **Standard connector (SC)** (see Fig 2.3 (a)) and a **Lucent connector (LC)** (see Fig 2.3 (b)) are two commonly used connectors for fiber optic cables.
- An SC is used for data transmission speeds up to 1 Gb/s, whereas an LC is used for speeds up to 4 Gb/s.
- Figure 2.3 depicts a Lucent connector and a Standard connector.
- A Straight Tip (ST) is a fiber optic connector with a plug and a socket that is locked with a half-twisted bayonet lock (see Fig 2.3 (c)).

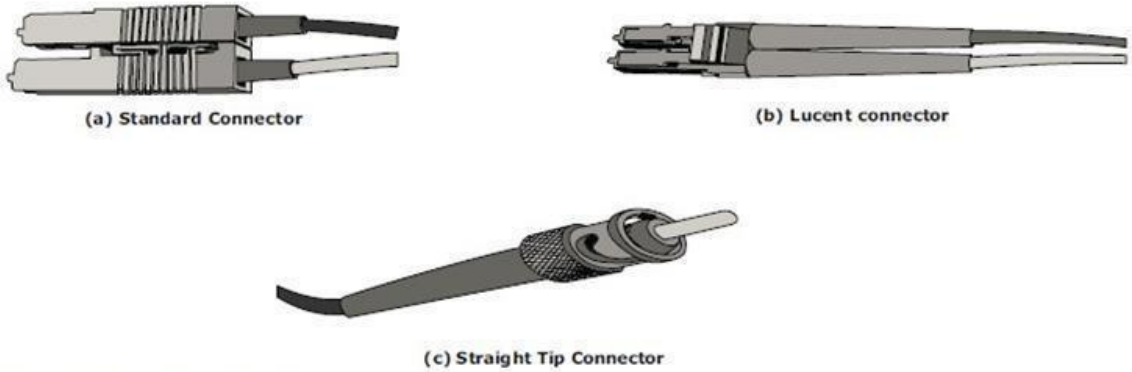


Fig 2.3: SC,LC, and ST connectors

Interconnect Devices

The commonly used interconnecting devices in SAN are

1) **Hubs,**
2) **Switches,**
3) **Directors**

- **Hubs** are used as communication devices in FC-AL implementations. Hubs physically connect nodes in a logical loop or a physical star topology.
- All the nodes must share the bandwidth because data travels through all the connection points. Because of availability of low cost and high performance switches, hubs are no longer used in SANs.
- **Switches** are more **intelligent** than hubs and directly **route data from one physical port to another**. Therefore, nodes do not share the bandwidth. Instead, each node has a dedicated communication path, resulting in bandwidth aggregation.
- Switches are available with:
  - ✓ Fixed port count
  - ✓ Modular design : port count is increased by installing additional port cards to open slots.
- **Directors are larger than switches** and are deployed for data center implementations.
- The function of directors is similar to that of FC switches, but directors have higher port count and fault tolerance capabilities.
- Port card or blade has multiple ports for connecting nodes and other FC switches

SAN Management Software

- SAN management software manages the interfaces between hosts, interconnect devices, and storage arrays.
- The software provides a view of the SAN environment and enables management of various resources from one central console.

- It provides key management functions, including mapping of storage devices, switches, and servers, monitoring and generating alerts for discovered devices, and logical partitioning of the SAN, called *zoning*

MODULE – 4

INTRODUCTION TO BUSINESS CONTINUITY, BACKUP AND ARCHIVE

❖ INTRODUCTION TO BUSINESS CONTINUITY

Business Continuity (BC):

Business continuity (BC) is an integrated and enterprise wide process that includes all activities (internal and external to IT) that a business must perform to mitigate the impact of planned and unplanned downtime.

BC entails preparing for, responding to, and recovering from a system outage that adversely affects business operations. It involves proactive measures, such as business impact analysis, risk assessments, deployment of BC technology solutions (backup and replication), and reactive measures, such as disaster recovery and restart, to be invoked in the event of a failure.

The goal of a BC solution is to ensure the **"information availability"** required to conduct vital business operations.

Information Availability:

Information availability (IA) refers to the ability of the infrastructure to function according to business expectations during its specified time of operation. Information availability ensures that people (employees, customers, suppliers, and partners) can access information whenever they need it. Information availability can be defined in terms of:

1. Reliability,
2. Accessibility
3. Timeliness.

1. **Reliability:** This reflects a component's ability to function without failure, under stated conditions, for a specified amount of time.
2. **Accessibility:** This is the state within which the required information is accessible at the right place, to the right user. The period of time during which the system is in an accessible state is termed **system uptime;** when it is not accessible it is termed **system**

downtime.

3. **Timeliness:** Defines the exact moment or the time window (a particular time of the day, week, month, and/or year as specified) during which information must be accessible. For example, if online access to an application is required between 8:00 am and 10:00 pm each day, any disruptions to data availability outside of this time slot are not considered to affect timeliness.

Causes of Information Unavailability

Various planned and unplanned incidents result in data unavailability.

- **Planned outages** include installation/integration/maintenance of new hardware, software upgrades or patches, taking backups, application and data restores, facility operations (renovation and construction), and refresh/migration of the testing to the production environment.
- **Unplanned outages** include failure caused by database corruption, component failure, and human errors.
- **Disasters (natural or man-made)** such as flood, fire, earthquake, and contamination are another type of incident that may cause data unavailability.

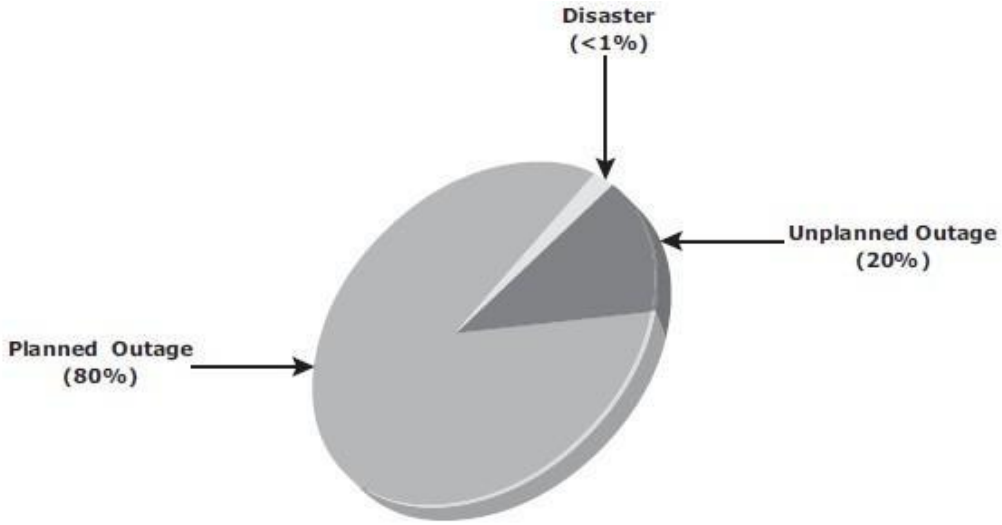


Fig 3.1: Disruptors of Information Availability

As illustrated in Fig 3.1 above, the majority of outages are planned. Planned outages are expected and scheduled, but still cause data to be unavailable.

Consequences of Downtime

- Information unavailability or downtime results in loss of productivity, loss of revenue, poor financial performance, and damage to reputation.
- Loss of productivity includes reduced output per unit of labor, equipment, and capital.
- Loss of revenue includes direct loss, compensatory payments, future revenue loss, billing loss, and investment loss.
- Poor financial performance affects revenue recognition, cash flow, discounts, payment guarantees, credit rating, and stock price.
- Damages to reputations may result in a loss of confidence or credibility with customers, suppliers, financial markets, banks, and business partners.
- An important metric, *average cost of downtime per hour*, provides a key estimate in determining the appropriate BC solutions. It is calculated as follows:

  Average cost of downtime per hour = average productivity loss per hour + average revenue loss per hour

  Where:

  Productivity loss per hour = (total salaries and benefits of all employees per week) /(average number of working hours per week)

  Average revenue loss per hour = (total revenue of an organization per week) /(average number of hours per week that an organization is open for business)

Measuring Information Availability

- Information availability (IA) relies on the availability of physical and virtual components of a data center. Failure of these components might disrupt IA. A failure is the termination of a component's capability to perform a required function. The component's capability can be restored by performing an external corrective action, such as a manual reboot, a repair, or replacement of the failed component(s).
- Proactive risk analysis performed as part of the BC planning process considers the component failure rate and average repair time, which are measured by MTBF and MTTR:

→ **Mean Time Between Failure (MTBF):** It is the average time available for a system or component to perform its normal operations between failures.

→ **Mean Time To Repair (MTTR):** It is the average time required to repair a failed component. MTTR includes the total time required to do the following activities: Detect the fault, mobilize the maintenance team, diagnose the fault, obtain the spare parts, repair, test, and restore the data.

Fig 3.2 illustrates the various information availability metrics that represent system uptime and downtime.

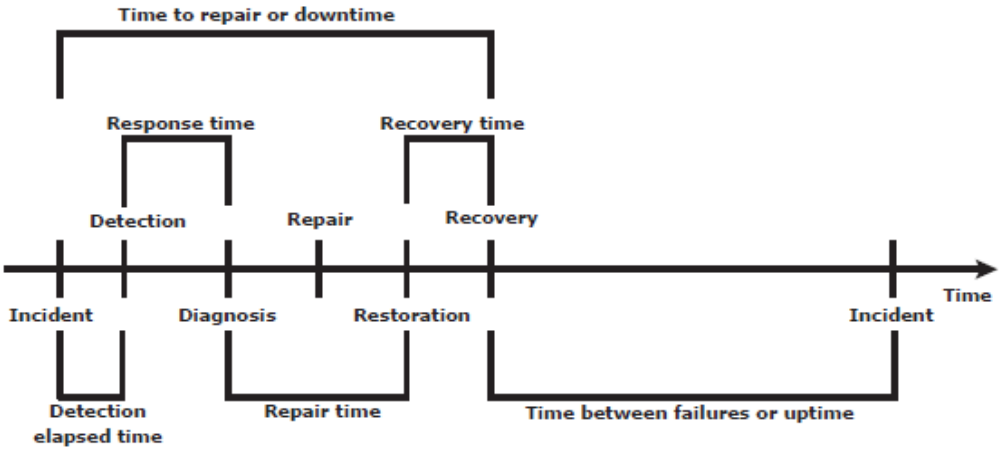


Fig 3-2: Information availability metrics

IA is the time period that a system is in a condition to perform its intended function upon demand. It can be expressed in terms of system uptime and downtime and measured as the amount or percentage of system uptime:

IA = system uptime / (system uptime + system downtime)

In terms of MTBF and MTTR, IA could also be expressed as

IA = MTBF / (MTBF + MTTR)

Uptime per year is based on the exact timeliness requirements of the service, this calculation leads to the number of "9s" representation for availability metrics.

Table 3-1 lists the approximate amount of downtime allowed for a service to achieve certain levels of 9s availability. For example, a service that is said to be "five 9s available" is available for 99.999 percent of the scheduled time in a year (24 × 365).

UPTIME (%)	DOWNTIME (%)	DOWNTIME PER YEAR	DOWNTIME PER WEEK
98	2	7.3 days	3 hr, 22 minutes
99	1	3.65 days	1 hr, 41 minutes
99.8	0.2	17 hr, 31 minutes	20 minutes, 10 secs
99.9	0.1	8 hr, 45 minutes	10 minutes, 5 secs
99.99	0.01	52.5 minutes	1 minute
99.999	0.001	5.25 minutes	6 secs
99.9999	0.0001	31.5 secs	0.6 secs

Table 3-1: Availability percentage and Allowable downtime

BC Terminology

This section defines common terms related to BC operations which are used in this module to explain advanced concepts:

- **Disaster recovery:** This is the coordinated process of restoring systems, data, and the infrastructure required to support key ongoing business operations in the event of a disaster. It is the process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known point of consistency. Once all recoveries are completed, the data is validated to ensure that it is correct.
- **Disaster restart:** This is the process of restarting business operations with mirrored consistent copies of data and applications.
- **Recovery-Point Objective (RPO):** This is the point in time to which systems and data must be recovered after an outage. It defines the amount of data loss that a business can endure. A large RPO signifies high tolerance to information loss in a business. Based on the RPO, organizations plan for the minimum frequency with which a backup or replica must be made. For example, if the RPO is six hours, backups or replicas must be made at least once in 6 hours. Fig 3.3 (a) shows various RPOs and their corresponding ideal recovery strategies. An organization can plan for an appropriate BC technology solution on the basis of the RPO it sets. For example:

  → **RPO of 24 hours:** This ensures that backups are created on an offsite tape drive every midnight. The corresponding recovery strategy is to restore data from the set of last

backup tapes.

→ **RPO of 1 hour:** Shipping database logs to the remote site every hour. The corresponding recovery strategy is to recover the database at the point of the last log shipment.

→ **RPO in the order of minutes:** Mirroring data asynchronously to a remote site

→ **Near zero RPO:** This mirrors mission-critical data synchronously to a remote site.

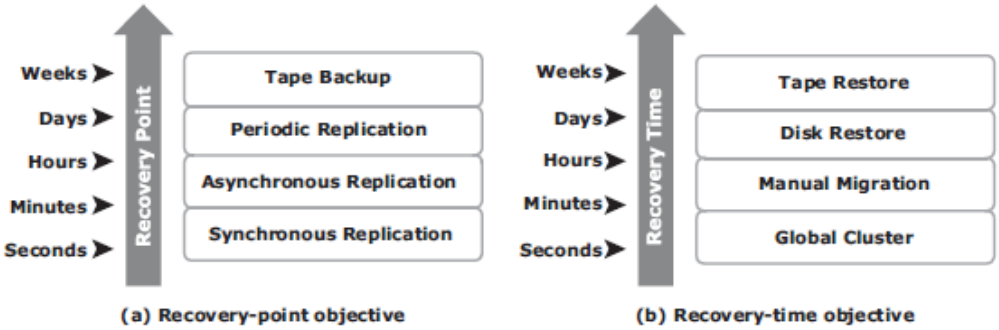


Fig 3.3: Strategies to meet RPO and RTO targets

- **Recovery-Time Objective (RTO):** The time within which systems and applications must be recovered after an outage. It defines the amount of downtime that a business can endure and survive. Businesses can optimize disaster recovery plans after defining the RTO for a given system. For example, if the RTO is two hours, then use a disk backup because it enables a faster restore than a tape backup. However, for an RTO of one week, tape backup will likely meet requirements. Some examples of RTOs and the recovery strategies to ensure data availability are listed below (refer to Fig 3.3 (b)):

  → **RTO of 72 hours:** Restore from backup tapes at a cold site.

  → **RTO of 12 hours:** Restore from tapes at a hot site.

  → **RTO of few hours:** Use a data vault to a hot site.

  → **RTO of a few seconds:** Cluster production servers with bidirectional mirroring, enabling the applications to run at both sites simultaneously.

- **Data vault:** A repository at a remote site where data can be periodically or continuously copied (either to tape drives or disks) so that there is always a copy at another site

- **Hot site:** A site where an enterprise's operations can be moved in the event of disaster. It is a site with the required hardware, operating system, application, and network support to perform business operations, where the equipment is available and running at all times.

- **Cold site:** A site where an enterprise's operations can be moved in the event of disaster, with minimum IT infrastructure and environmental facilities in place, but not activated

- **Server Clustering:** A group of servers and other necessary resources coupled to operate as a single system. Clusters can ensure high availability and load balancing. Typically, in failover clusters, one server runs an application and updates the data, and another server is kept as standby to take over completely, as required. Server clustering provides load balancing by distributing the application load evenly among multiple servers within the cluster.

BC Planning Life Cycle

BC planning must follow a disciplined approach like any other planning process. Organizations today dedicate specialized resources to develop and maintain BC plans. From the conceptualization to the realization of the BC plan, a life cycle of activities can be defined for the BC process.

The BC planning lifecycle includes five stages shown below (Fig 3.4):

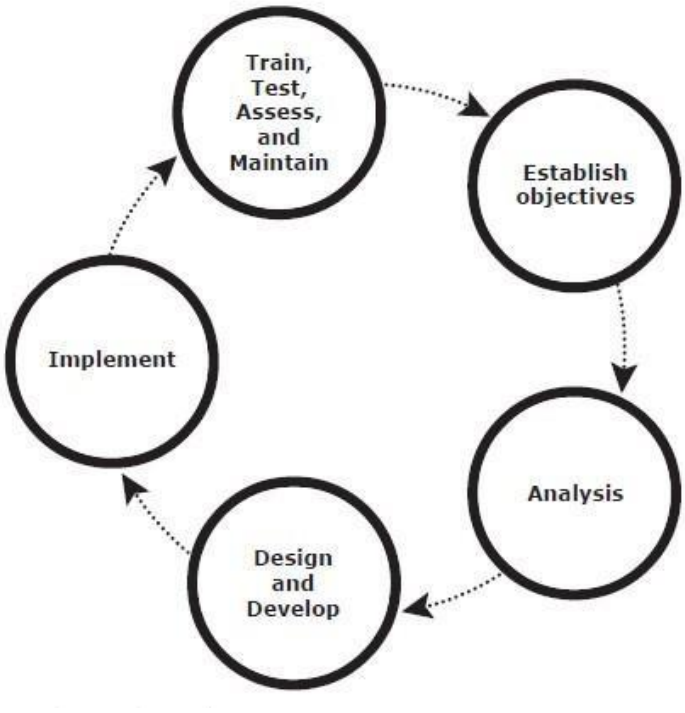


Fig 3.4: BC Planning Lifecycle

Several activities are performed at each stage of the BC planning lifecycle, including the following key activities:

1. **Establishing objectives**

→ Determine BC requirements.

→ Estimate the scope and budget to achieve requirements.

→ Select a BC team by considering subject matter experts from all areas of the business, whether internal or external.

→ Create BC policies.

2. Analyzing

→ Collect information on data profiles, business processes, infrastructure support, dependencies, and frequency of using business infrastructure.

→ Identify critical business needs and assign recovery priorities.

→ Create a risk analysis for critical areas and mitigation strategies.

→ Conduct a Business Impact Analysis (BIA).

→ Create a cost and benefit analysis based on the consequences of data unavailability.

3. Designing and developing

→ Define the team structure and assign individual roles and responsibilities. For example, different teams are formed for activities such as emergency response, damage assessment, and infrastructure and application recovery.

→ Design data protection strategies and develop infrastructure.

→ Develop contingency scenarios.

→ Develop emergency response procedures.

→ Detail recovery and restart procedures.

4. Implementing

→ Implement risk management and mitigation procedures that include backup, replication, and management of resources.

→ Prepare the disaster recovery sites that can be utilized if a disaster affects the primary data center.

→ Implement redundancy for every resource in a data center to avoid single points of failure.

5. Training, testing, assessing, and maintaining

→ Train the employees who are responsible for backup and replication of business-critical data on a regular basis or whenever there is a modification in the BC plan

→ Train employees on emergency response procedures when disasters are declared.

→ Train the recovery team on recovery procedures based on contingency scenarios.

→ Perform damage assessment processes and review recovery plans.

→ Test the BC plan regularly to evaluate its performance and identify its limitations.

→ Assess the performance reports and identify limitations.

→ Update the BC plans and recovery/restart procedures to reflect regular changes within the data center.

Failure Analysis

Single Point of Failure

- A **single point of failure** refers to the failure of a component that can terminate the availability of the entire system or IT service.
- Fig 3.5 depicts a system setup in which an application, running on a VM, provides an interface to the client and performs I/O operations.
- The client is connected to the server through an IP network, the server is connected to the storage array through a FC connection, an HBA installed at the server sends or receives data to and from a storage array, and an FC switch connects the HBA to the storage port
- In a setup where **each component must function as required to ensure data availability**, the failure of a single physical or virtual component causes the failure of the entire data center or an application, resulting in disruption of business operations.
- In this example, failure of a hypervisor can affect all the running VMs and the virtual network, which are hosted on it.
- The can be several similar single points of failure identified in this example. A VM, a hypervisor, an HBA/NIC on the server, the physical server, the IP network, the FC switch, the storage array ports, or even the storage array could be a potential single point of failure. To avoid single points of failure, it is essential to implement a fault-tolerant mechanism.

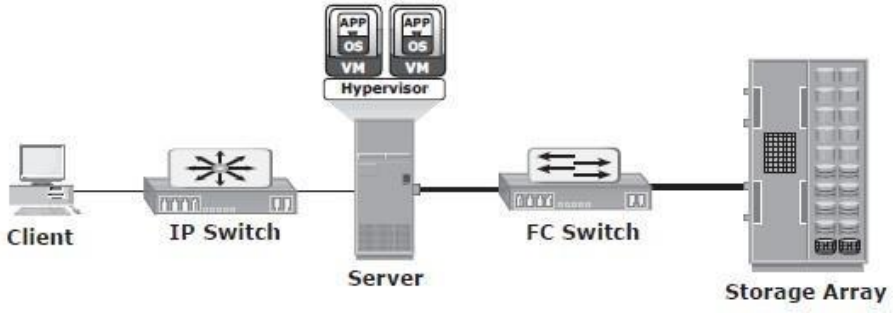


Fig 3.5: Single Point of Failure

Resolving Single Points of Failure

- To mitigate a single point of failure, systems are designed with redundancy, such that the system will fail only if all the components in the redundancy group fail. This ensures that the failure of a single component does not affect data availability.
- Data centers follow stringent guidelines to implement fault tolerance for uninterrupted information availability. Careful analysis is performed to eliminate every single point of failure.
- The example shown in Fig 3.6 represents all enhancements of the system shown in Fig 3.5 in the infrastructure to mitigate single points of failure:
  - Configuration of redundant HBAs at a server to mitigate single HBA failure
  - Configuration of NIC (network interface card) teaming at a server allows protection against single physical NIC failure. It allows grouping of two or more physical NICs and treating them as a single logical device. NIC teaming eliminates the single point of failure associated with a single physical NIC.
  - Configuration of redundant switches to account for a switch failure
  - Configuration of multiple storage array ports to mitigate a port failure
  - RAID and hot spare configuration to ensure continuous operation in the event of disk failure
  - Implementation of a redundant storage array at a remote site to mitigate local site failure
  - Implementing server (or compute) clustering, a fault-tolerance mechanism whereby two or more servers in a cluster access the same set of data volumes. Clustered servers exchange a heartbeat to inform each other about their health. If one of the servers or hypervisors fails, the other server or hypervisor can take up the workload.
  - Implementing a VM Fault Tolerance mechanism ensures BC in the event of a server failure. This technique creates duplicate copies of each VM on another server so that when a VM failure is detected, the duplicate VM can be used for failover. The two VMs are kept in synchronization with each other in order to perform successful failover.

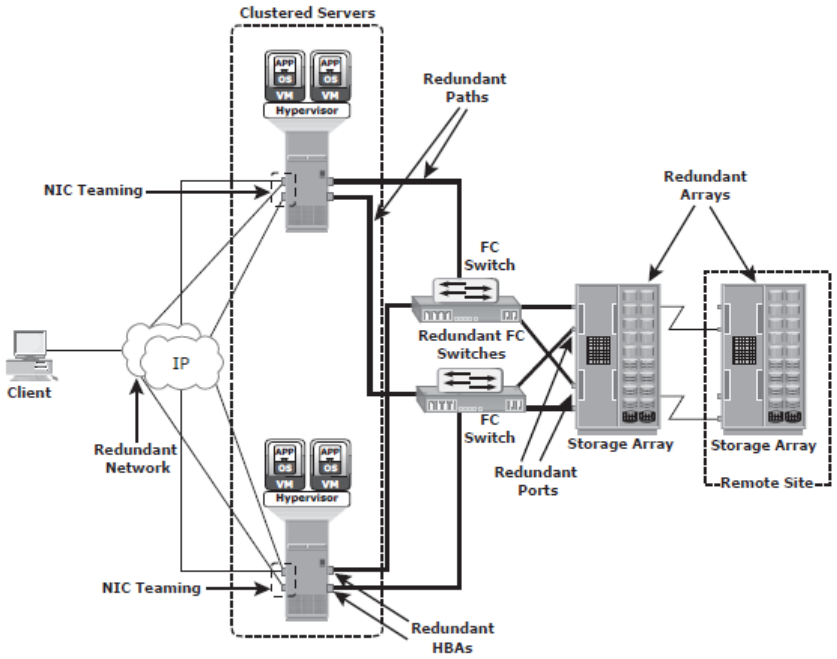


Fig 3.6: Resolving single points of failure

Multipathing Software

- Configuration of multiple paths increases the data availability through path failover. If servers are configured with one I/O path to the data there will be no access to the data if that path fails. Redundant paths eliminate the path to become single points of failure.
- Multiple paths to data also improve I/O performance through load sharing and maximize server, storage, and data path utilization.
- In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/O will not reroute unless the system recognizes that it has an alternate path.
- Multipathing software provides the functionality to recognize and utilize alternate I/O path to data. Multipathing software also manages the load balancing by distributing I/Os to all available, active paths.
- In a virtual environment, multipathing is enabled either by using the hypervisor's built-in capability or by running a third-party software module, added to the hypervisor.

Business Impact Analysis

- A *business impact analysis* (BIA) identifies which business units, operations, and processes are essential to the survival of the business.

- It evaluates the financial, operational, and service impacts of a disruption to essential business processes.

- The BIA process leads to a report detailing the incidents and their impact over business functions. The impact may be specified in terms of money or in terms of time.

- Based on the potential impacts associated with downtime, businesses can prioritize and implement countermeasures to mitigate the likelihood of such disruptions. These are detailed in the BC plan.

  A BIA includes the following set of tasks:
  - Determine the business areas.
  - For each business area, identify the key business processes critical to its operation.
  - Determine the attributes of the business process in terms of applications, databases, and hardware and software requirements.
  - Estimate the costs of failure for each business process.
  - Calculate the maximum tolerable outage and define RTO and RPO for each business process.
  - Establish the minimum resources required for the operation of business processes.
  - Determine recovery strategies and the cost for implementing them.
  - Optimize the backup and business recovery strategy based on business priorities.
  - Analyze the current state of BC readiness and optimize future BC planning.

BC Technology Solutions

After analyzing the business impact of an outage, designing appropriate solutions to recover from a failure is the next important activity. One or more copies of the original data are maintained using any of the following strategies, so that data can be recovered and business operations can be restarted using an alternate copy:

1. **Backup:** Data backup is a predominant method of ensuring data availability. The frequency of backup is determined based on RPO, RTO, and the frequency of data changes.
2. **Storage array-based replication (local):** Data can be replicated to a separate location within the same storage array. The replica is used independently for other business operations. Replicas can also be used for restoring operations if data corruption occurs.
3. **Storage array-based replication (remote):** Data in a storage array can be replicated to another storage array located at a remote site. If the storage array is lost due to a disaster, business operations can be started from the remote storage array.

BACKUP AND ARCHIVE

- **Data Backup** is a copy of production data, created and retained for the sole purpose of recovering lost or corrupted data.
- Evaluating the various backup methods along with their recovery considerations and retention requirements is an essential step to implement a successful backup and recovery solution.
- Organizations generate and maintain large volumes of data, and most of the data is fixed content. This fixed content is rarely accessed after a period of time. Still, this data needs to be retained for several years to meet regulatory compliance.
- **Data archiving** is the process of moving data that is no longer actively used, from primary storage to a low-cost secondary storage. This data is retained in the secondary storage for a long term to meet regulatory requirements. This reduces the amount of data to be backed up and the time required to back up the data.

Backup Purpose

Backups are performed to serve three purposes: ***disaster recovery, operational recovery, and archival***. These are discussed in the following sections.

Disaster Recovery

- Backups are performed to address disaster recovery needs.
- The backup copies are used for restoring data at an alternate site when the primary site is incapacitated due to a disaster. Based on RPO and RTO requirements, organizations use different backup strategies for disaster recovery.
- When a tape-based backup method is used as a disaster recovery strategy, the backup tape media is shipped and stored at an offsite location. These tapes can be recalled for restoration at the disaster recovery site.
- Organizations with stringent RPO and RTO requirements use remote replication technology to replicate data to a disaster recovery site. Organizations can bring production systems online in a relatively short period of time if a disaster occurs.

Operational Recovery

- Data in the production environment changes with every business transaction and operation.
- Operational recovery is the use of backups to restore data if data loss or logical

corruption occurs during routine processing.

- For example, it is common for a user to accidentally delete an important email or for a file to become corrupted, which can be restored from operational backup.

Archival

- Backups are also performed to address archival requirements.
- Traditional backups are still used by small and medium enterprises for long-term preservation of transaction records, e- mail messages, and other business records required for regulatory compliance.

Apart from addressing disaster recovery, archival, and operational requirements, backups serve as a protection against data loss due to physical damage of a storage device, software failures, or virus attacks. Backups can also be used to protect against accidents such as a deletion or intentional data destruction.

Backup Considerations

- The amount of data loss and downtime that a business can endure in terms of RPO and RTO are the primary considerations in selecting and implementing a specific backup strategy.
- RPO refers to the point in time to which data must be recovered, and the point in time from which to restart business operations.
- This specifies the time interval between two backups. In other words, the RPO determines backup frequency.
- For example, if an application requires an RPO of 1 day, it would need the data to be backed up at least once every day.
- Another consideration is the retention period, which defines the duration for which a business needs to retain the backup copies.
- Some data is retained for years and some only for a few days. For example, data backed up for archival is retained for a longer period than data backed up for operational recovery.
- The backup media type or backup target is another consideration, that is driven by RTO and impacts the data recovery time.
- The time-consuming operation of starting and stopping in a tape-based system affects the backup performance, especially while backing up a large number of small files.
- The development of a backup strategy must include a decision about the most appropriate time for performing a backup to minimize any disruption to production

operations.

- The location, size, number of files, and data compression should also be considered because they might affect the backup process. Location is an important consideration for the data to be backed up.
- Many organizations have dozens of heterogeneous platforms locally and remotely supporting their business.
- The file size and number of files also influence the backup process. Backing up large-size files (for example, ten 1 MB files) takes less time, compared to backing up an equal amount of data composed of small-size files (for example, ten thousand 1 KB files).

Backup Granularity

- Backup granularity depends on business needs and the required RTO/RPO.
- Based on the granularity, backups can be categorized as full, incremental and cumulative (differential).
- Most organizations use a combination of these three backup types to meet their backup and recovery requirements.
- The below figure shows the different backup granularity levels.

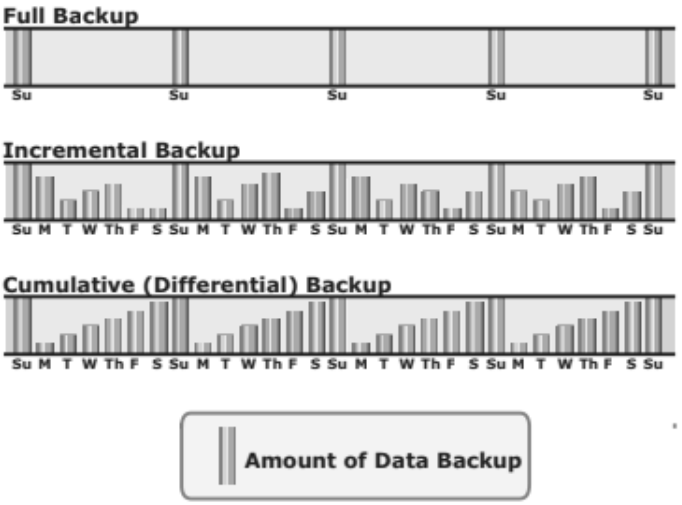


Figure: Different Granularity levels

- Full backup is a backup of the complete data on the production volumes.
- A full backup copy is created by copying the data in the production volumes to a backup

storage device.

- It provides a faster recovery but requires more storage space and also takes more time to back up.

Incremental Backup

- Incremental backup copies the data that has changed since the last full or incremental backup, whichever has occurred more recently.
- This is much faster than a full backup (because the volume of data backed up is restricted to the changed data only) but takes longer to restore.

Cumulative Backup

- Cumulative backup copies the data that has changed since the last full backup.
- This method takes longer than an incremental backup but is faster to restore.

- Restore operations vary with the granularity of the backup.
- A full backup provides a single repository from which the data can be easily restored.
- The process of restoration from an incremental backup requires the last full backup and all the incremental backups available until the point of restoration.
- A restore from a cumulative backup requires the last full backup and the most recent cumulative backup.
- The below figure shows an example of restoring data from incremental backup.

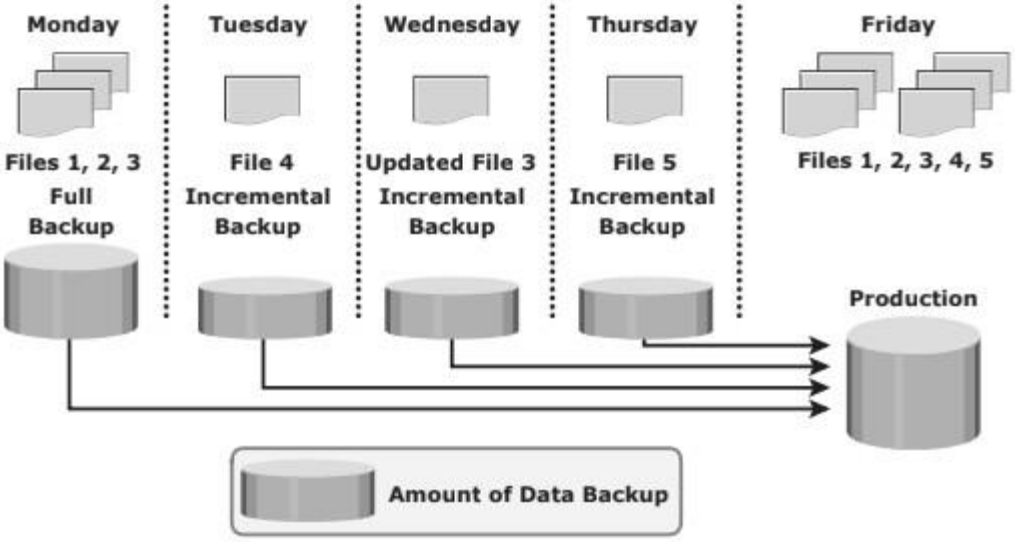


Figure: Restoring from Incremental Backup

- In this example, a full backup is performed on Monday evening. Each day after that, an incremental backup is performed.

- On Tuesday, a new file (File 4 in the figure) is added, and no other files have changed.
- Consequently, only File 4 is copied during the incremental backup performed on Tuesday evening.
- On Wednesday, no new files are added, but File 3 has been modified. Therefore, only the modified File 3 is copied during the incremental backup on Wednesday evening.
- Similarly, the incremental backup on Thursday copies only File 5.
- On Friday morning, there is data corruption, which requires data restoration from the backup.
- The first step toward data restoration is restoring all data from the full backup of Monday evening. The next step is applying the incremental backups of Tuesday, Wednesday, and Thursday.
- In this manner, data can be successfully recovered to its previous state, as it existed on Thursday evening.
- The below figure shows an example of restoring data from cumulative backup.

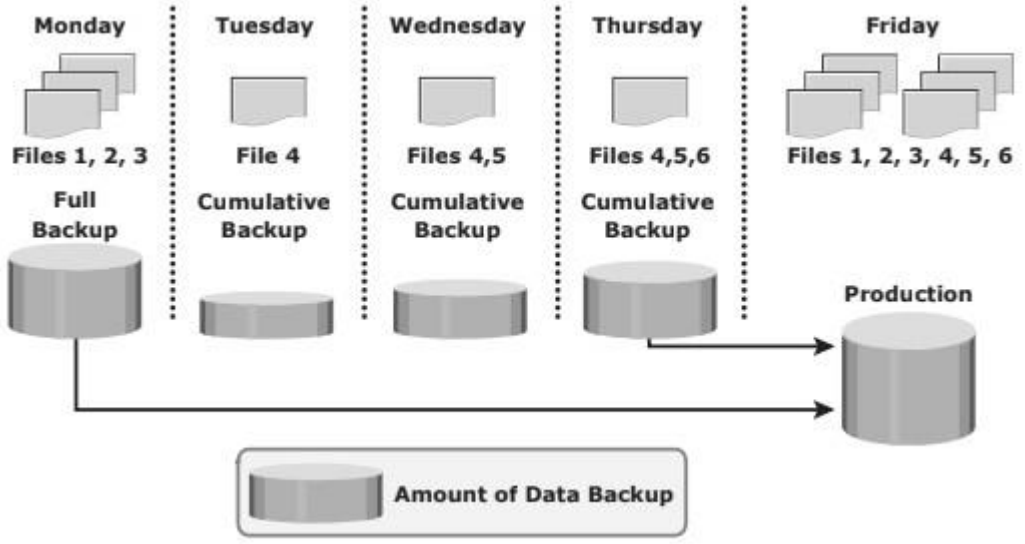


Figure: Restoring from Cumulative Backup

- In this example, a full backup of the business data is taken on Monday evening. Each day after that, a cumulative backup is taken.
- On Tuesday, File 4 is added and no other data is modified since the previous full backup of Monday evening. Consequently, the cumulative backup on Tuesday evening copies only File 4.
- On Wednesday, File 5 is added. The cumulative backup taking place on Wednesday evening copies both File 4 and File 5 because these files have been added or modified since the last

full backup.

- Similarly, on Thursday, File 6 is added. Therefore, the cumulative backup on Thursday evening copies all three files: File 4, File 5, and File 6.
- On Friday morning, data corruption occurs that requires data restoration using backup copies.
- The first step in restoring data is to restore all the data from the full backup of Monday evening. The next step is to apply only the latest cumulative backup, which is taken on Thursday evening.
- In this way, the production data can be recovered faster because its needs only two copies of data — the last full backup and the latest cumulative backup.

Recovery Considerations

- The retention period is a key consideration for recovery. The retention period for a backup is derived from an RPO.
- For example, users of an application might request to restore the application data from its backup copy, which was created a month ago. This determines the retention period for the backup.
- Therefore, the minimum retention period of this application data is one month. However, the organization might choose to retain the backup for a longer period of time because of internal policies or external factors, such as regulatory directives.
- If the recovery point is older than the retention period, it might not be possible to recover all the data required for the requested recovery point.
- Long retention periods can be defined for all backups, making it possible to meet any RPO within the defined retention periods. However, this requires a large storage space, which translates into higher cost. Therefore, while defining the retention period, analyze all the restore requests in the past and the allocated budget.
- RTO relates to the time taken by the recovery process.
- To meet the defined RTO, the business may choose the appropriate backup granularity to minimize recovery time.
- In a backup environment, RTO influences the type of backup media that should be used. For example, a restore from tapes takes longer to complete than a restore from disks.

Backup Methods

- **Hot backup and cold backup** are the two methods deployed for backup. They are based on the state of the application when the backup is performed.
- In a **hot backup**, the application is up and running, with users accessing their data during the backup process. This method of backup is also referred to as an *online backup***.**

- In a **cold backup**, the application is not active or shutdown during the backup process and is also called as *offline backup*.
- The hot backup of online production data becomes more challenging because data is actively used and changed.
- An open file is locked by the operating system and is not backed up during the backup process. In such situations, an *open file agent* is required to back up the open file.
- In database environments, the use of open file agents is not enough, because the agent should also support a consistent backup of all the database components.
- For example, a database is composed of many files of varying sizes occupying several file systems. To ensure a consistent database backup, all files need to be backed up in the same state. That does not necessarily mean that all files need to be backed up at the same time, but they all must be synchronized so that the database can be restored with consistency.
- The disadvantage associated with a hot backup is that the agents usually affect the overall application performance.

- Consistent backups of databases can also be done by using a cold backup. This requires the database to remain inactive during the backup. Of course, the disadvantage of a cold backup is that the database is inaccessible to users during the backup process.
- Hot backup is used in situations where it is not possible to shut down the database. This is facilitated by database backup agents that can perform a backup while the database is active. The disadvantage associated with a hot backup is that the agents usually affect overall application performance.
- A **point-in-time (PIT)** copy method is deployed in environments where the impact of downtime from a cold backup or the performance resulting from a hot backup is unacceptable. The PIT copy is created from the production volume and used as the source for the backup. This reduces the impact on the production volume.
- Certain attributes and properties attached to a file, such as permissions, owner, and other metadata, also need to be backed up. These attributes are as important as the data itself and must be backed up for consistency.
- Backup of boot sector and partition layout information is also critical for successful recovery.
- In a disaster recovery environment, **bare-metal recovery (BMR)** refers to a backup in which all metadata, system information, and application configurations are appropriately backed up for a full system recovery. BMR builds the base system, which includes partitioning, the file system layout, the operating system, the applications, and all the relevant configurations. BMR recovers the base system first, before starting the recovery of data files. Some BMR technologies can recover a server onto dissimilar hardware.

Backup Architecture

- A backup system commonly uses the client-server architecture with a backup server and multiple backup clients.
- The below figure illustrates the backup architecture.
- The backup server manages the backup operations and maintains the backup catalog, which contains information about the backup configuration and backup metadata.
- Backup configuration contains information about when to run backups, which client data to be backed up, and so on, and the backup metadata contains information about the backed up data.
- The role of a backup client is to gather the data that is to be backed up and send it to the storage node. It also sends the tracking information to the backup server.

- The storage node is responsible for writing the data to the backup device. (In a backup environment, a storage node is a host that controls backup devices.)
- The storage node also sends tracking information to the backup server. In many cases, the storage node is integrated with the backup server, and both are hosted on the same physical platform.
- A backup device is attached directly or through a network to the storage node's host platform. Some backup architecture refers to the storage node as the media server because it manages the storage device.
- Backup software provides reporting capabilities based on the backup catalog and the log files. These reports include information, such as the amount of data backed up, the number of completed and incomplete backups, and the types of errors that might have occurred.
- Reports can be customized depending on the specific backup software used.

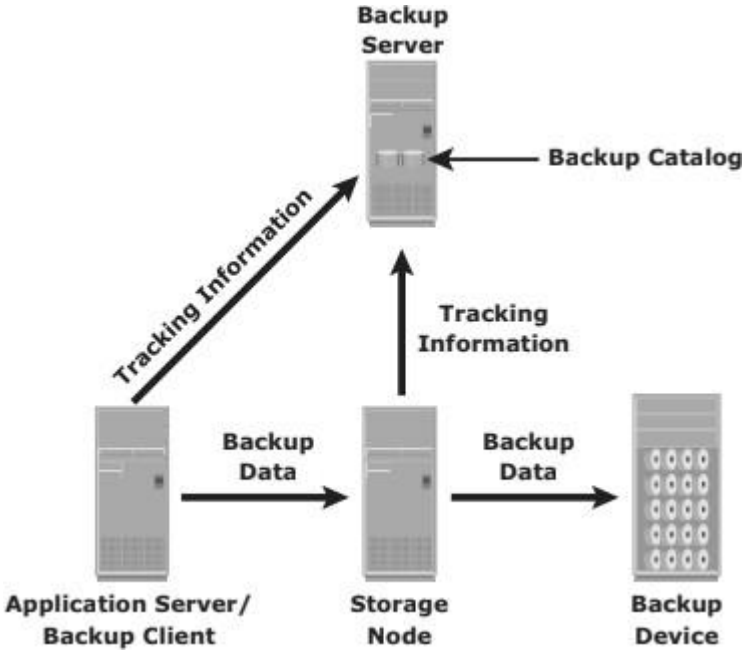


Figure: Backup Architecture

Backup and Restore Operations

- When a backup operation is initiated, significant network communication takes place between the different components of a backup infrastructure.
- The backup operation is typically initiated by a server, but it can also be initiated by a client.
- The backup server initiates the backup process for different clients based on the backup schedule

configured for them.

- For example, the backup for a group of clients may be scheduled to start at 11:00 p.m. every day.
- The backup server coordinates the backup process with all the components in a backup environment (see Figure below).
- The backup server maintains the information about backup clients to be backed up and storage nodes to be used in a backup operation.
- The backup server retrieves the backup-related information from the backup catalog and, based on this information, instructs the storage node to load the appropriate backup media into the backup devices.
- Simultaneously, it instructs the backup clients to gather the data to be backed up and send it over the network to the assigned storage node.
- After the backup data is sent to the storage node, the client sends some backup metadata (the number of files, name of the files, storage node details, and so on) to the backup server.
- The storage node receives the client data, organizes it, and sends it to the backup device.
- The storage node then sends additional backup metadata (location of the data on the backup device, time of backup, and so on) to the backup server.
- The backup server updates the backup catalog with this information.

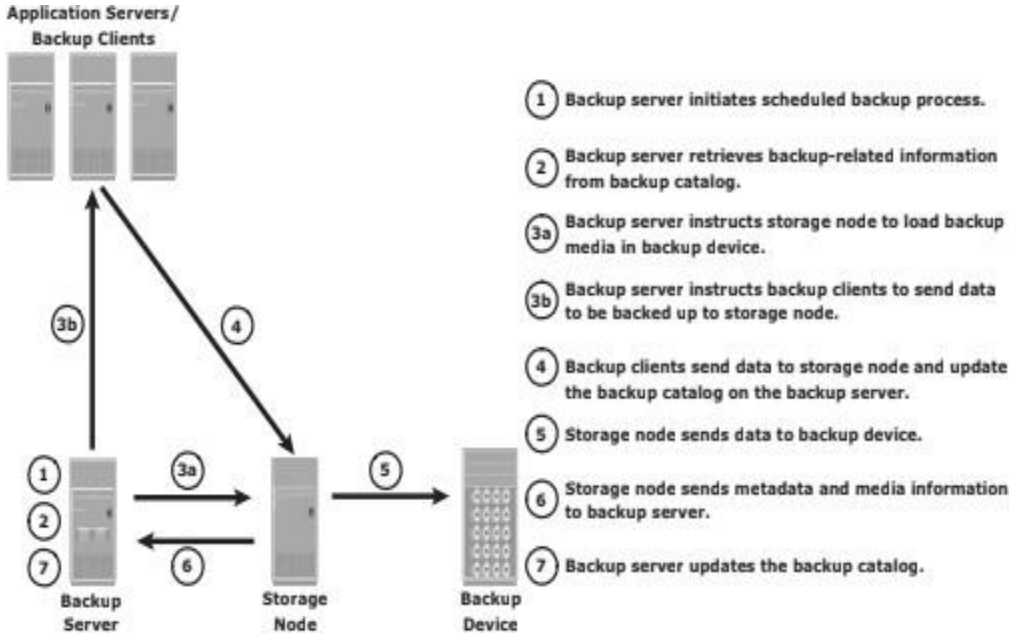


Figure: Backup Operation

- After the data is backed up, it can be restored when required.
- A restore process must be manually initiated from the client.

- Some backup software has a separate application for restore operations.
- These restore applications are usually accessible only to the administrators or backup operators.
- The below Figure shows a restore operation.
- Upon receiving a restore request, an administrator opens the restore application to view the list of clients that have been backed up.
- While selecting the client for which a restore request has been made, the administrator also needs to identify the client that will receive the restored data. Data can be restored on the same client for whom the restore request has been made or on any other client.
- The administrator then selects the data to be restored and the specified point in time to which the data has to be restored based on the RPO.
- Because all this information comes from the backup catalog, the restore application needs to communicate with the backup server.
- The backup server instructs the appropriate storage node to mount the specific backup media onto the backup device.
- Data is then read and sent to the client that has been identified to receive the restored data.
- Some restorations are successfully accomplished by recovering only the requested production data. For example, the recovery process of a spreadsheet is completed when the specific file is restored.
- In database restorations, additional data, such as log files, must be restored along with the production data. This ensures consistency for the restored data.
- In these cases, the RTO is extended due to the additional steps in the restore operation.
-

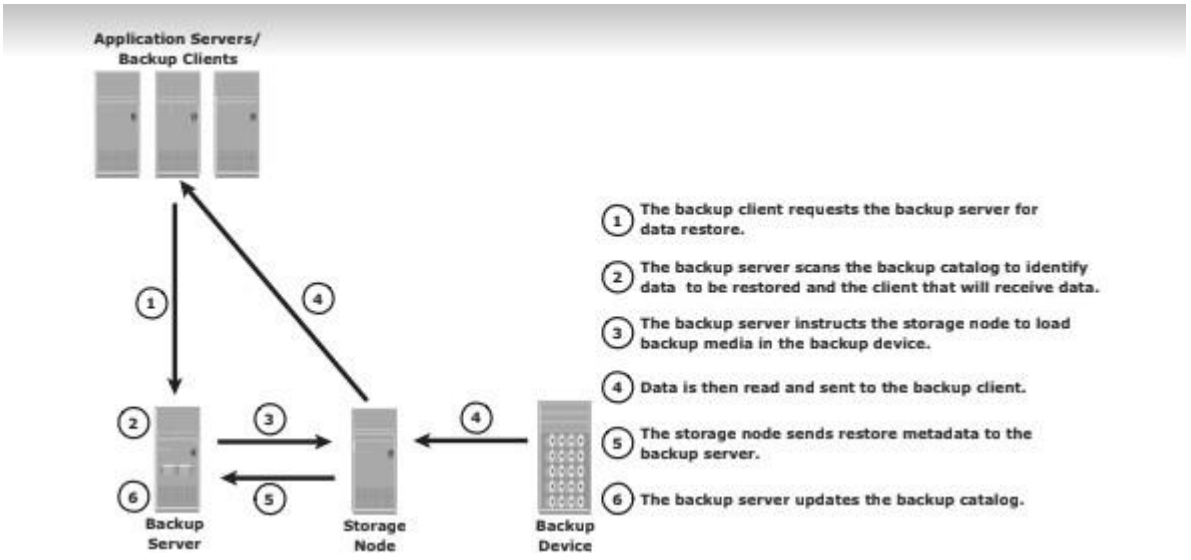


Figure: Restore Operation

Backup Topologies

- Three basic topologies are used in a backup environment:

1. Direct attached backup
2. LAN based backup, and
3. SAN based backup.

- A **mixed topology** is also used by combining LAN based and SAN based topologies.
- In a **direct-attached backup**, a backup device is attached directly to the client. Only the metadata is sent to the backup server through the LAN. This configuration frees the LAN from backup traffic.
- The example shown in Fig 3.7 device is directly attached and dedicated to the backup client. As the environment grows, however, there will be a need for central management of all backup devices and to share the resources to optimize costs. An appropriate solution is to share the backup devices among multiple servers. Network-based topologies (LAN-based and SAN-based) provide the solution to optimize the utilization of backup devices.

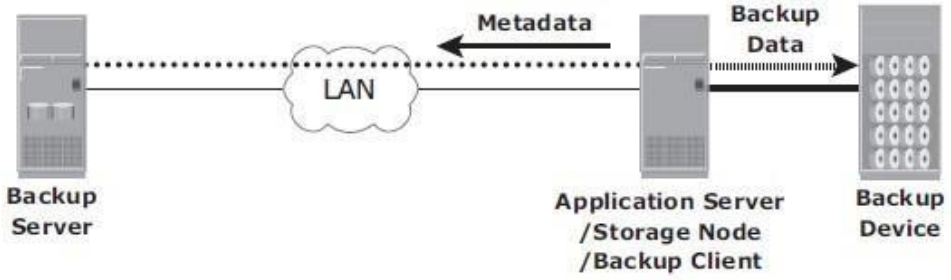


Fig 3.7: Direct-attached backup topology

- In **LAN-based backup**, the clients, backup server, storage node, and backup device are connected to the LAN (see Fig 3.8). The data to be backed up is transferred from the backup client (source), to the backup device (destination) over the LAN, which may affect network performance.
- This impact can be minimized by adopting a number of measures, such as configuring separate networks for backup and installing dedicated storage nodes for some application servers.

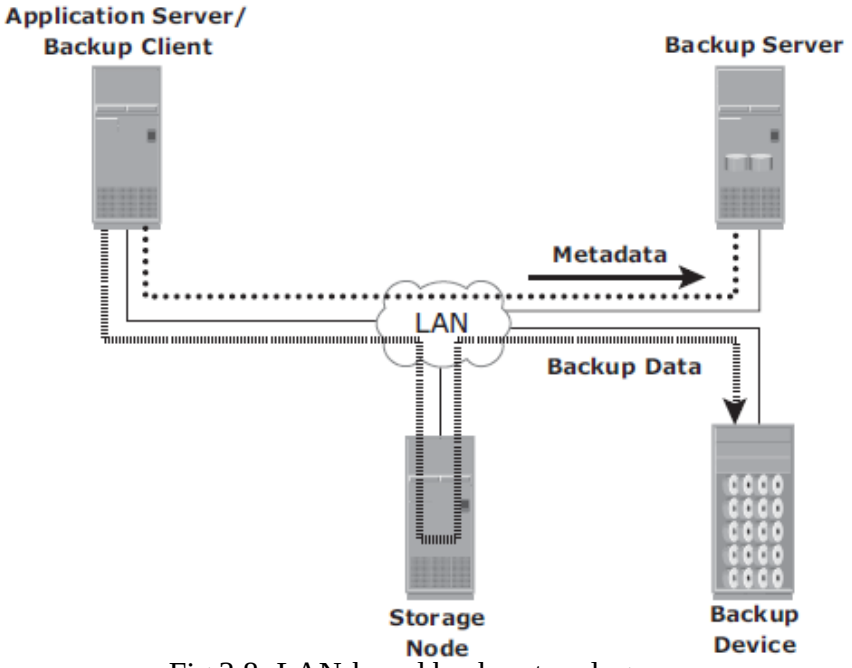


Fig 3.8: LAN-based backup topology

- The **SAN-based backup** is also known as the *LAN-free backup*. Fig 3.9 illustrates a SAN-based backup. The SAN-based backup topology is the most appropriate solution when a backup device needs to be shared among the clients. In this case the backup device and clients are attached to the SAN.
- In the example from Fig 3.9, a client sends the data to be backed up to the backup device over the SAN. Therefore, the backup data traffic is restricted to the SAN, and only the backup metadata is transported over the LAN. The volume of metadata is insignificant when compared to the production data; the LAN performance is not degraded in this configuration.

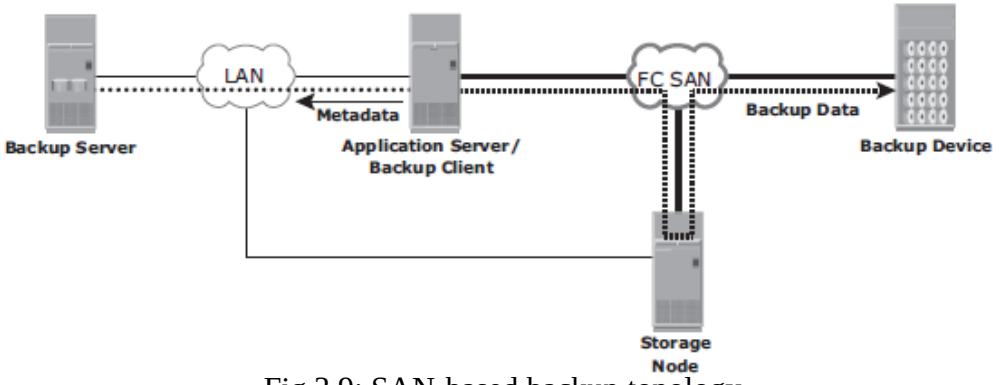


Fig 3.9: SAN-based backup topology

- The emergence of low-cost disks as a backup medium has enabled disk arrays to be attached to the SAN and used as backup devices. A tape backup of these data backups on the disks can be created and shipped offsite for disaster recovery and long-term

retention.

- The mixed topology uses both the LAN-based and SAN-based topologies, as shown in Fig 3.10. This topology might be implemented for several reasons, including cost, server location, reduction in administrative overhead, and performance considerations.

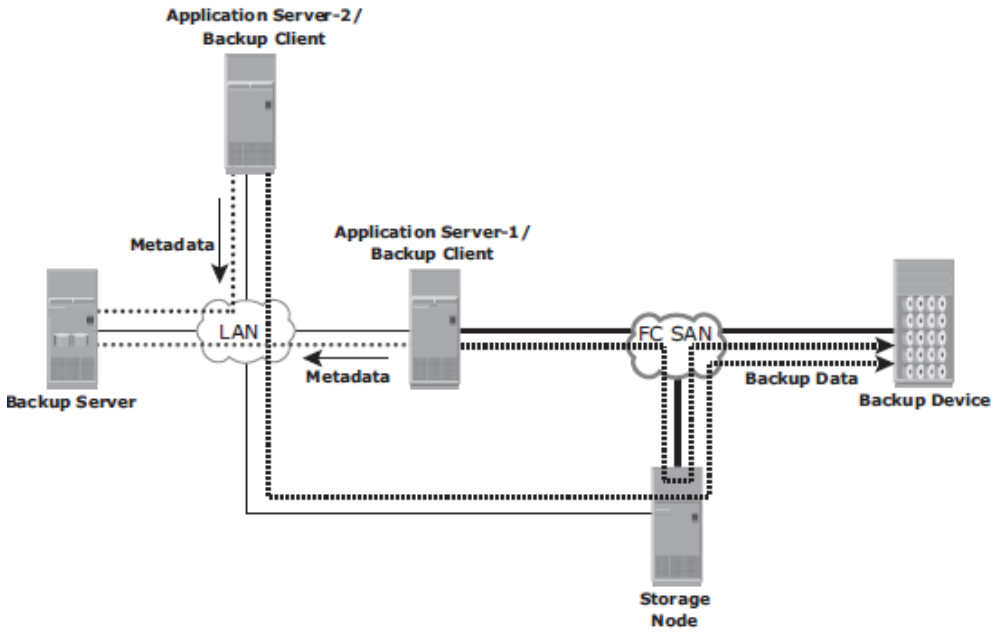


Fig 3.10: Mixed backup topology

Backup in NAS Environments

- The use of a NAS head imposes a new set of considerations on the backup and recovery strategy in NAS environments. NAS heads use a proprietary operating system and file system structure that supports multiple file-sharing protocols.

- In the NAS environment, backups can be implemented in different ways: server based, serverless, or using Network Data Management Protocol (NDMP). Common implementations are NDMP 2-way and NDMP 3-way.

Server-Based and Serverless Backup

- In an *application server-based backup*, the NAS head retrieves data from a storage array over the network and transfers it to the backup client running on the application server.
- The backup client sends this data to the storage node, which in turn writes the data to the backup device. This results in overloading the network with the backup data and using application server resources to move the backup data. Figure 10-11 illustrates server-based backup in the NAS environment.

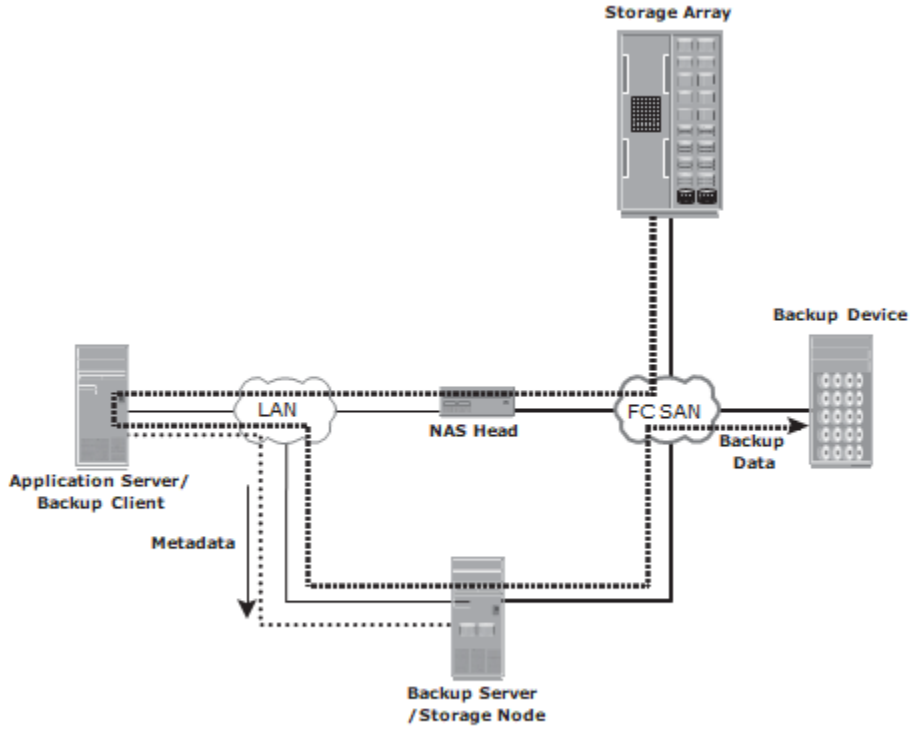


Figure 10-11: Server-based backup in a NAS environment

- In a *serverless backup*, the network share is mounted directly on the storage node. This avoids overloading the network during the backup process and eliminates the need to use resources on the application server.
- Figure 10-12 illustrates serverless backup in the NAS environment. In this scenario, the storage node, which is also a backup client, reads the data from the NAS head and writes it to the backup device without involving the application server.

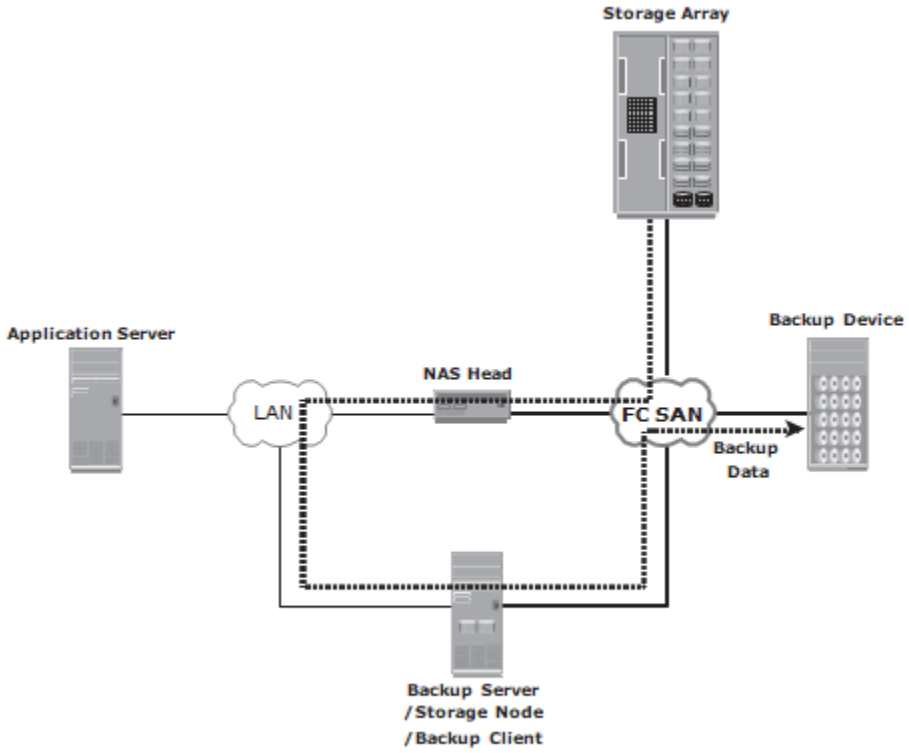


Figure 10-12: Serverless backup in a NAS environment

NDMP-Based Backup

- *NDMP* is an industry-standard TCP/IP-based protocol specifically designed for a backup in a NAS environment.
- It communicates with several elements in the backup environment (NAS head, backup devices, backup server, and so on) for data transfer and enables vendors to use a common protocol for the backup architecture.
- Data can be backed up using NDMP regardless of the operating system or platform. Due to its flexibility, it is no longer necessary to transport data through the application server, which reduces the load on the application server and improves the backup speed.
- NDMP optimizes backup and restore by leveraging the high-speed connec- tion between the backup devices and the NAS head.
- In NDMP, backup data is sent directly from the NAS head to the backup device, whereas metadata is sent to the backup server.
- Figure 10-13 illustrates a backup in the NAS environment using NDMP 2-way. In this model, network traffic is minimized by isolating data movement from the NAS head to the locally attached backup device.
- Only metadata is transported on the network. The backup device is dedicated to the NAS device, and hence, this method does not support centralized management of all backup devices.

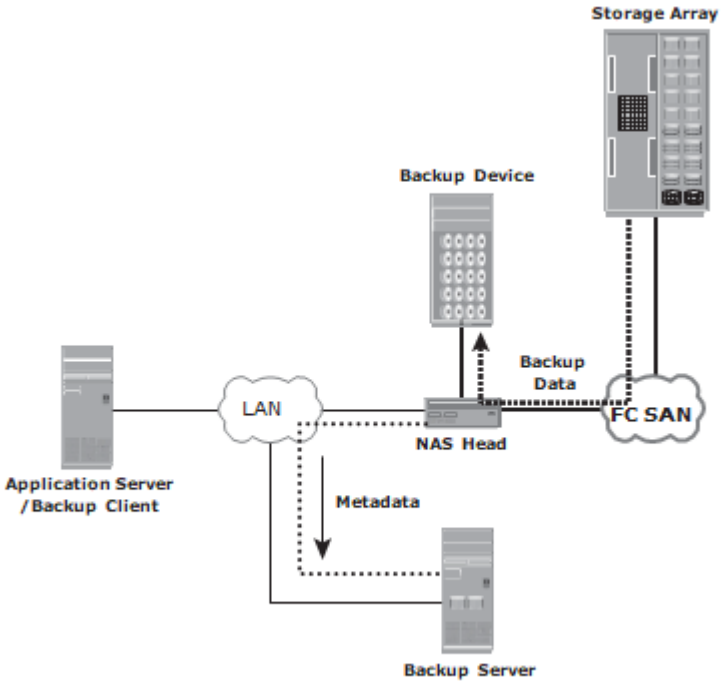


Figure 10-13: NDMP 2-way in a NAS environment

- In the *NDMP 3-way* method, a separate private backup network must be established between all NAS heads and the NAS head connected to the backup device.
- Metadata and NDMP control data are still transferred across the public network. Figure 10-14 shows a NDMP 3-way backup.

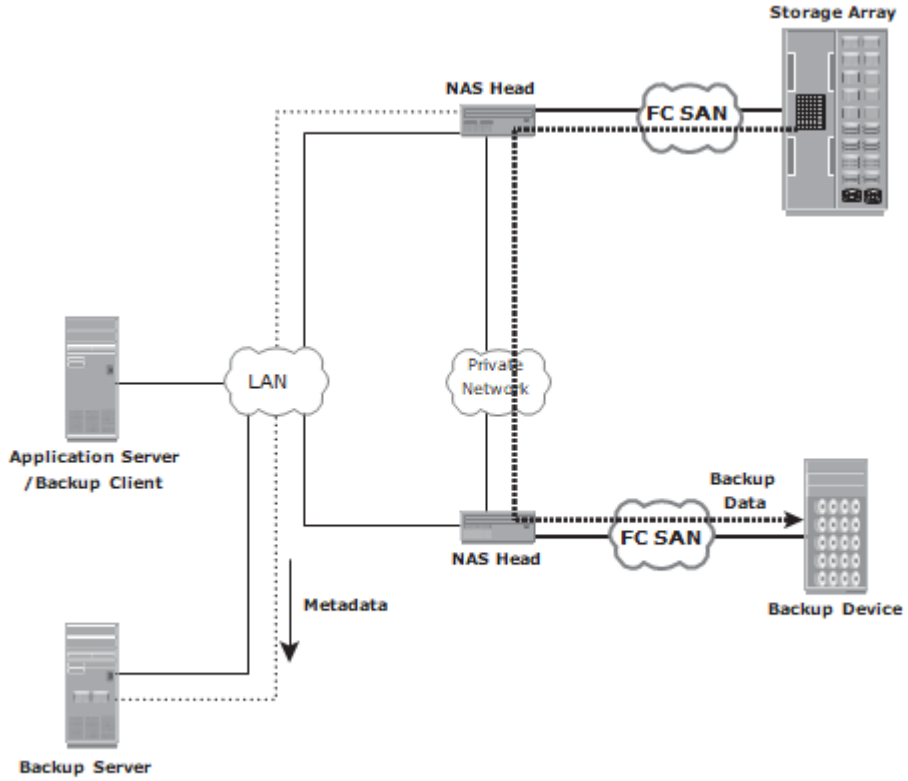


Figure 10-14: NDMP 3-way in a NAS environment

Module 1

STORAGE SYSTEM

1.1 Introduction to Information storage

1.1.1 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.

1.1.2 Information Storage

Businesses use data to derive information that is critical to their day-to-day operations. Storage is a repository that enables users to store and retrieve this digital data.

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.

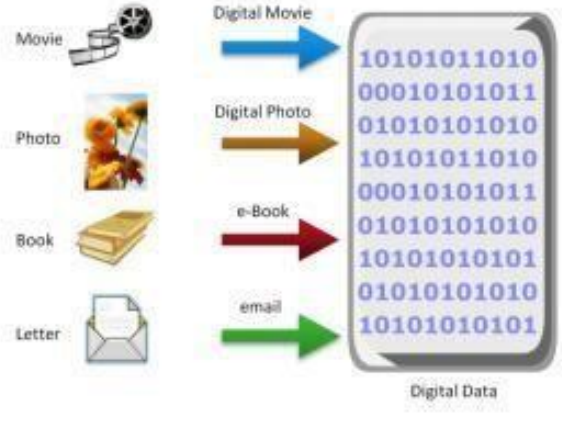


Fig 1.1: Digital data

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 digitaldevices, along with smart applications, have significantly contributed to the generation of digital content.

1.1.3 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.

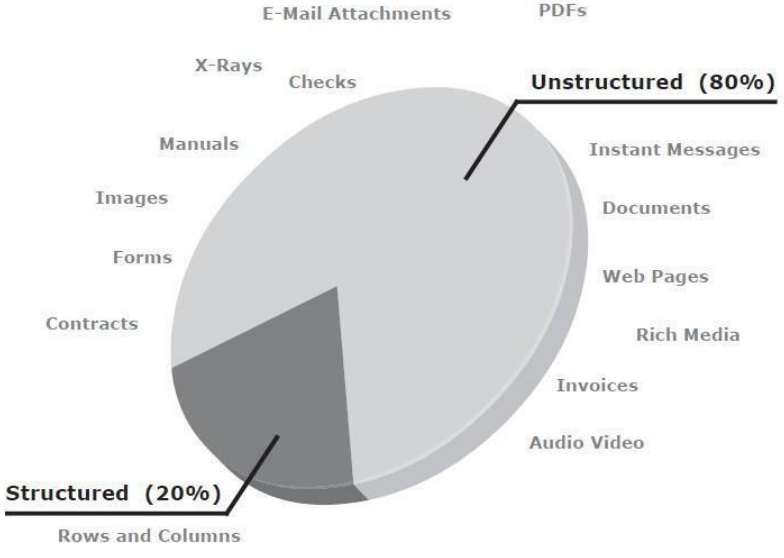


Fig 1.2: Types of data

1.1.4 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.
- 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 .

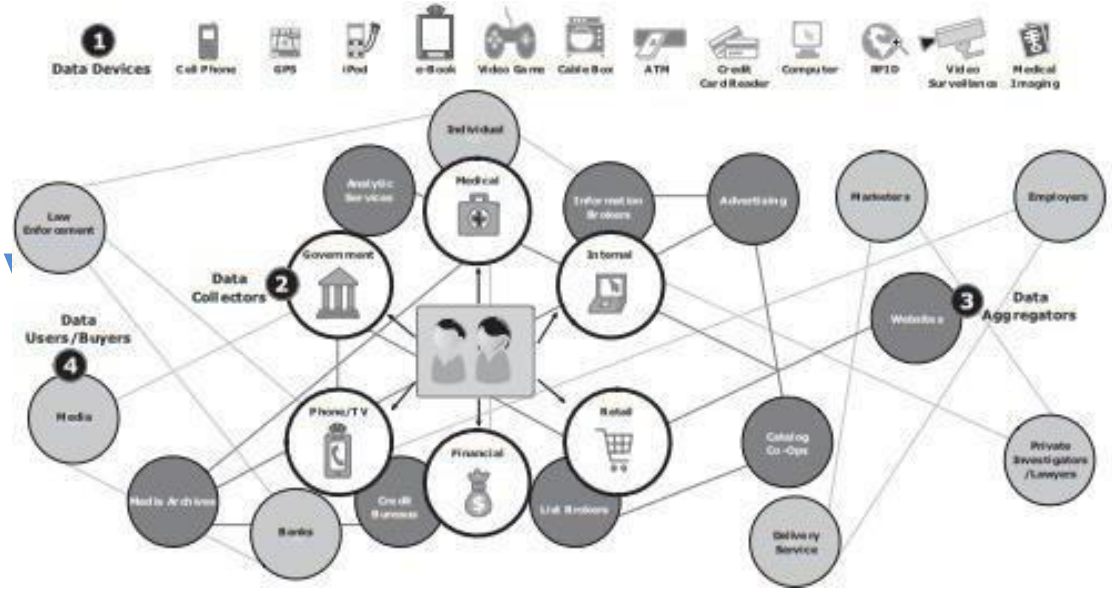


Fig 1.3: Big data Ecosystem

- 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.

1.1.5 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.
- Because information is critical to the success of a business, there is an ever present concern about its availability and protection.

1.1.6 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]).

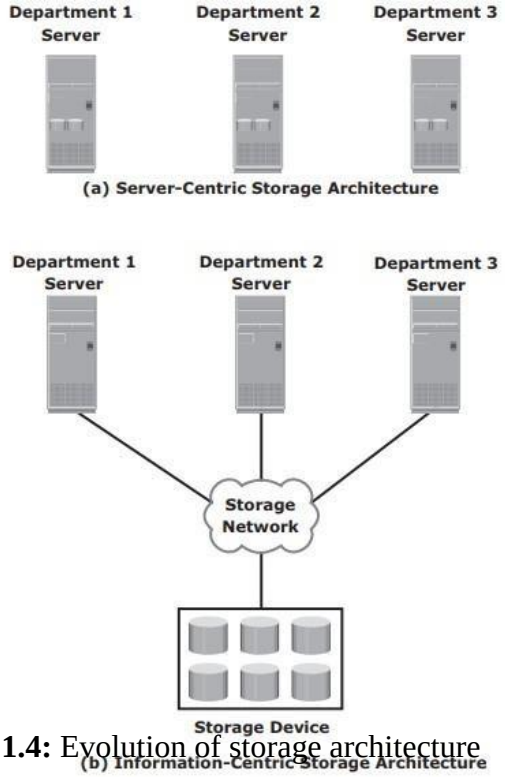


Fig 1.4: Evolution of storage architecture

- 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 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.

- 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.

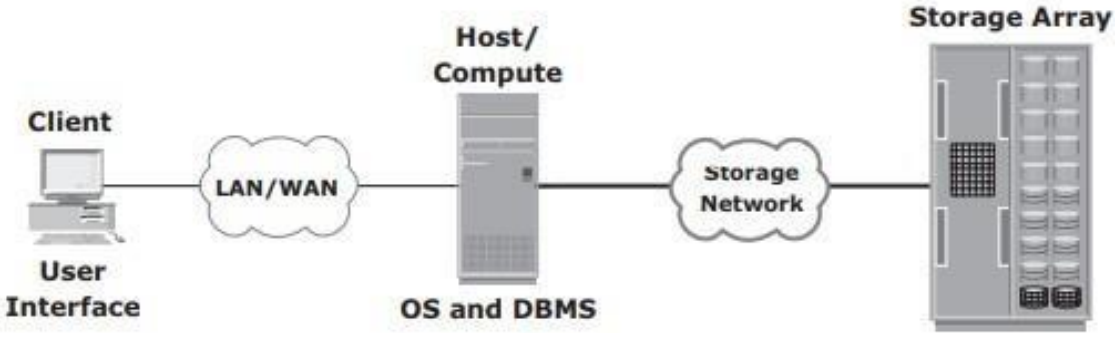


Fig 1.5: Example of an online order transaction system

Figure 1-5: Example of an online order transaction system

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.

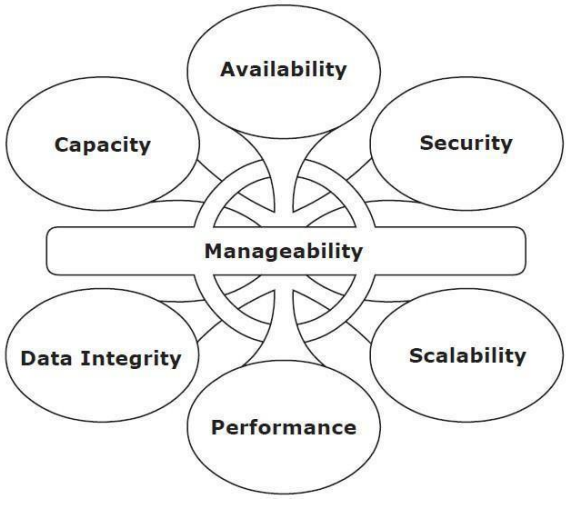


Fig 1.6: Key characteristics of data center elements

1.4 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.

1.5 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.

1.6 Key Data center Elements

1.7.1 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

1.7.2. 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.

1.7.3 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.

1.7.3.1 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.

1.7.3.2 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.

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)

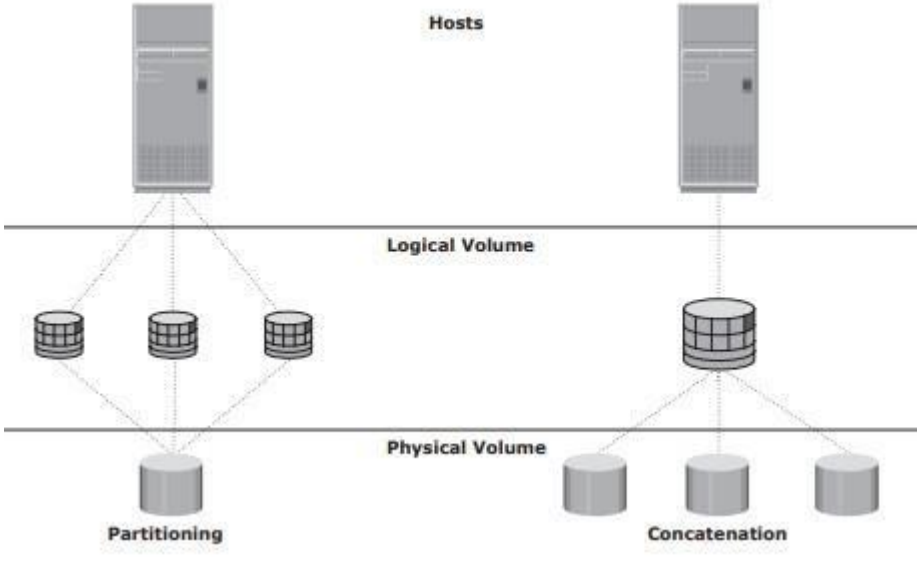


Fig 1.7: Disk Partitioning and concatenation

- 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.

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.

- 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 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.

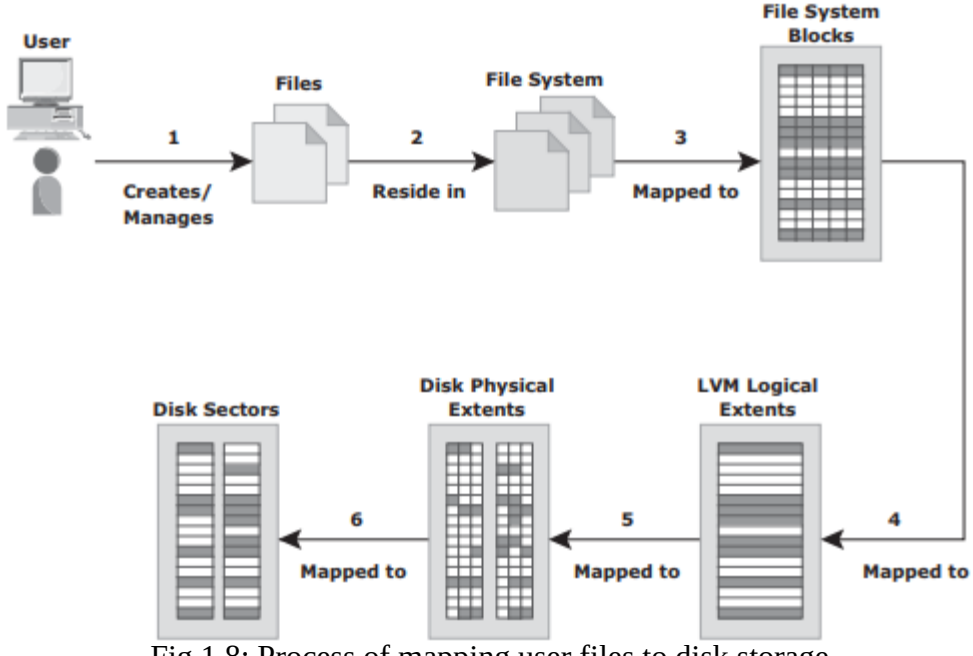


Fig 1.8: Process of mapping user files to disk storage

- 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.

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)).

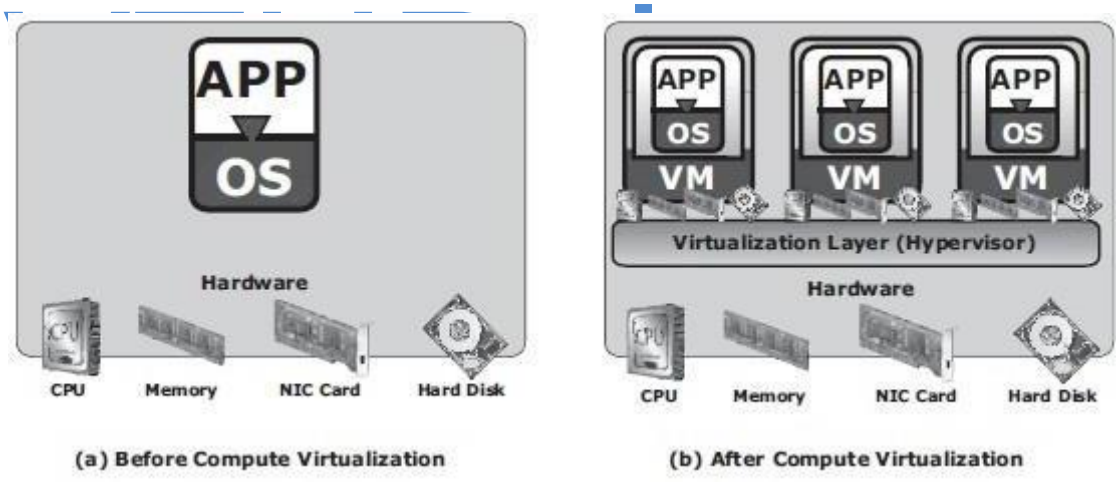


Fig 1.9: Server Virtualization

- 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. This is required for maintenance activities

1.7 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.

1.8.1 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 1.10):
  - ✓ the host interface device
  - ✓ port
  - ✓ cable.

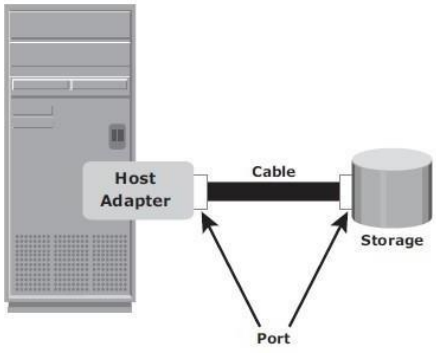


Fig 1.10: Physical components of connectivity

- 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.

1.8.2 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)
  - v.

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.

1.8 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.

1.9 Disk Drive Components

- The key components of a hard disk drive are platter, spindle, read-write head, actuator arm assembly, and controller board .
- 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

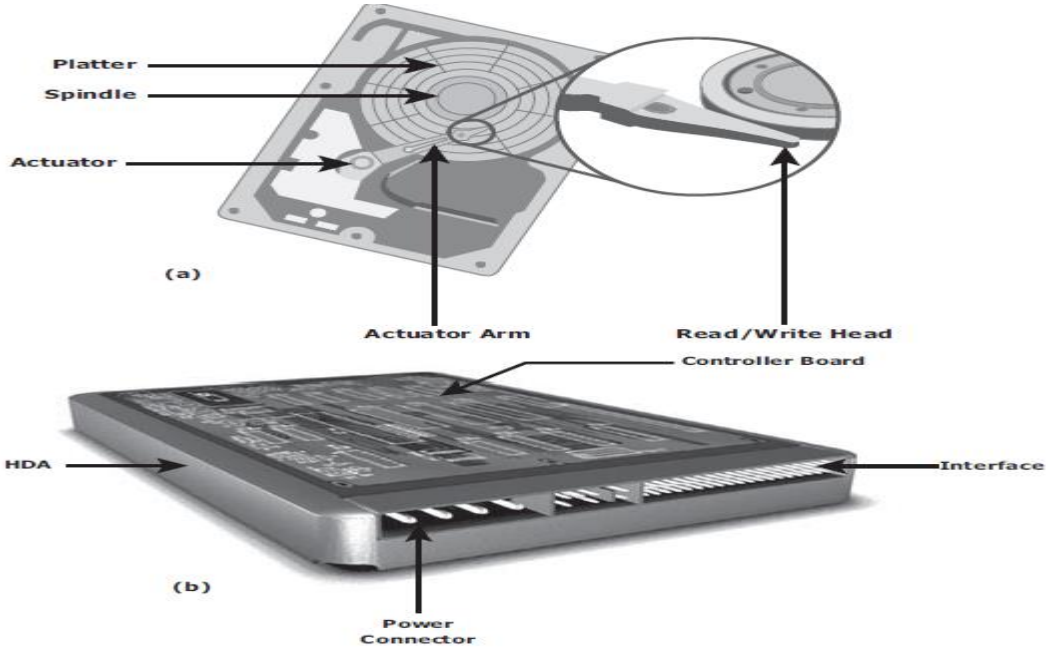


Figure 2-5: Disk drive components

Platter

- 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

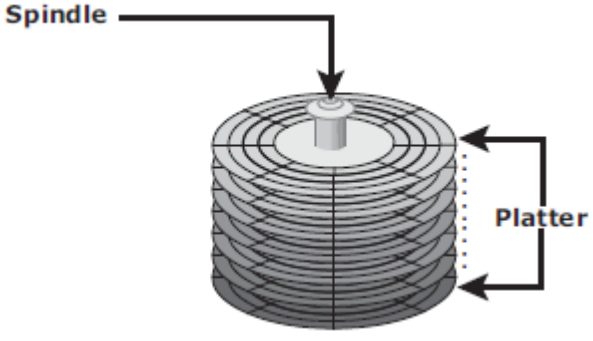


Figure 2-6: Spindle and platter

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.

Read/Write Head

- Read/Write Head 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

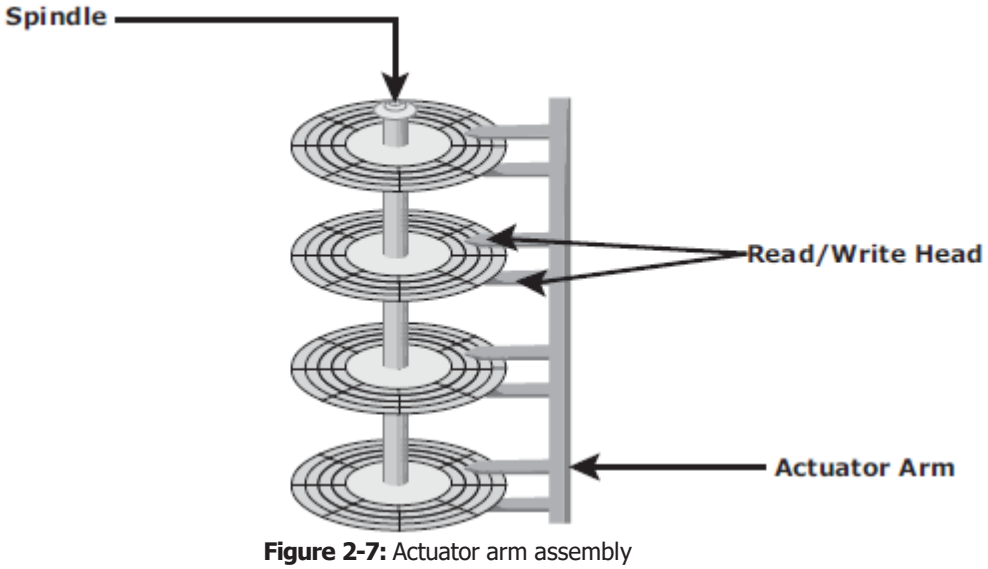


Figure 2-7: Actuator arm assembly

- 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 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 (as shown in 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.

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.

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.
- The number of sectors per track varies according to the drive type.
- The first personal computer disks had 17 sectors per track. Recent disks have a much larger number of sectors on a single track.

- There can be thousands of tracks on a platter, depending on the physical dimensions and recording density of the platter.
- 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.
- This information helps the controller to locate the data on the drive. 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.

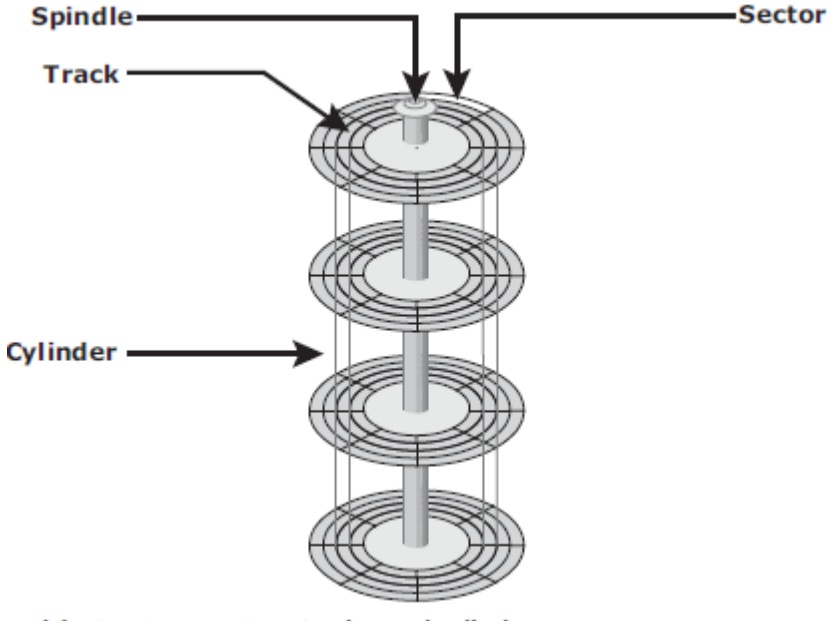


Figure 2-8: Disk structure: sectors, tracks, and cylinders

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 efficiently.

- 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.

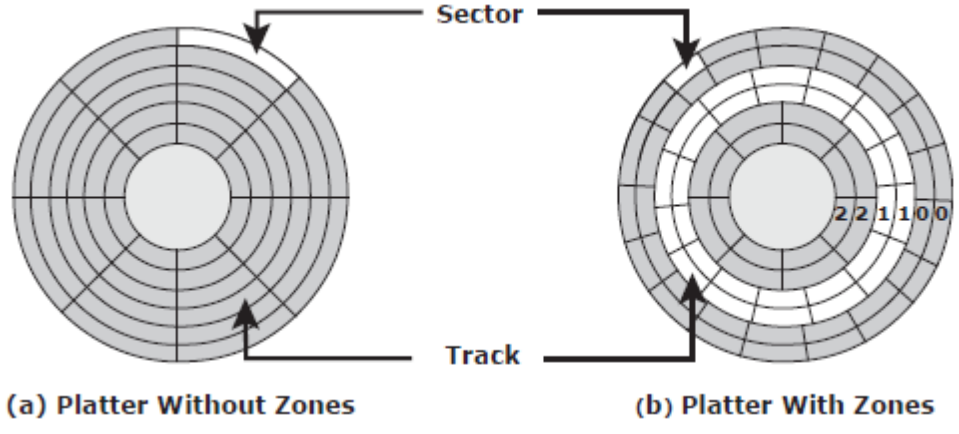


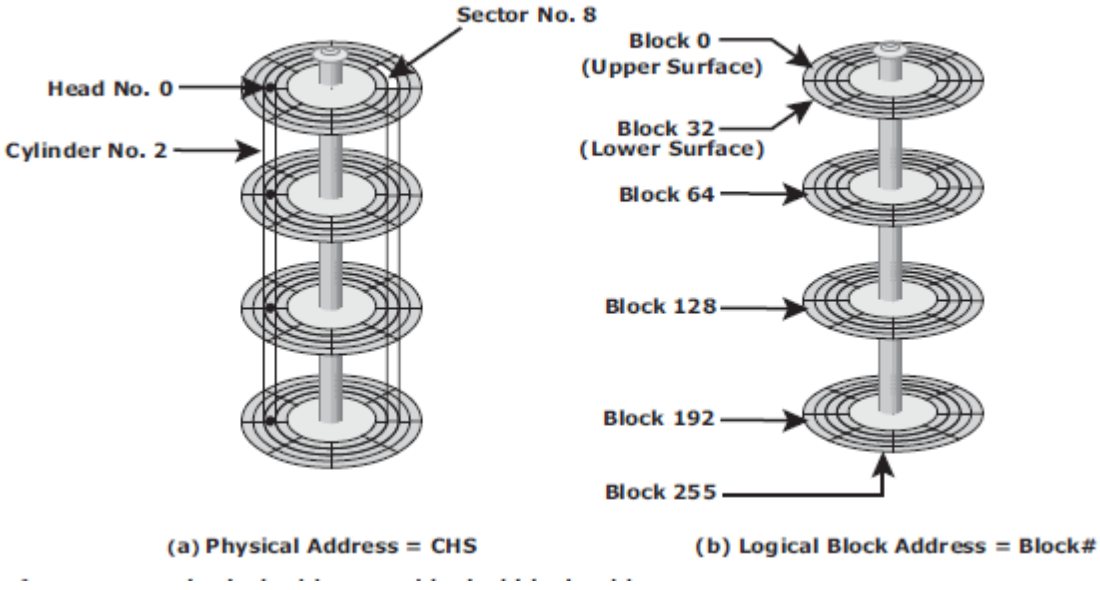
Figure 2-9: Zoned bit recording

Logical Block Addressing

- Earlier drives used physical addresses consisting of the cylinder, head, and sector (CHS) number to refer to specific 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.

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

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.

  - Each of these specifications is measured in milliseconds. The seek time of a disk is typically specified by the drive manufacturer.
  - The average seek time on a modern disk is typically in the range of 3 to 15 milliseconds. Seek time has more impact on the read operation of random tracks rather than adjacent tracks.

➢ To minimize the seek time, data can be written to only a subset of the available cylinders. This results in lower usable capacity than the actual capacity of the drive.

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. Similar to the seek time, rotational latency has more impact on the reading/writing of random sectors on the disk than on the same operations on adjacent sectors.
- 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.
- It is important to first understand the process of read/write operations to calculate data transfer rates.
- 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

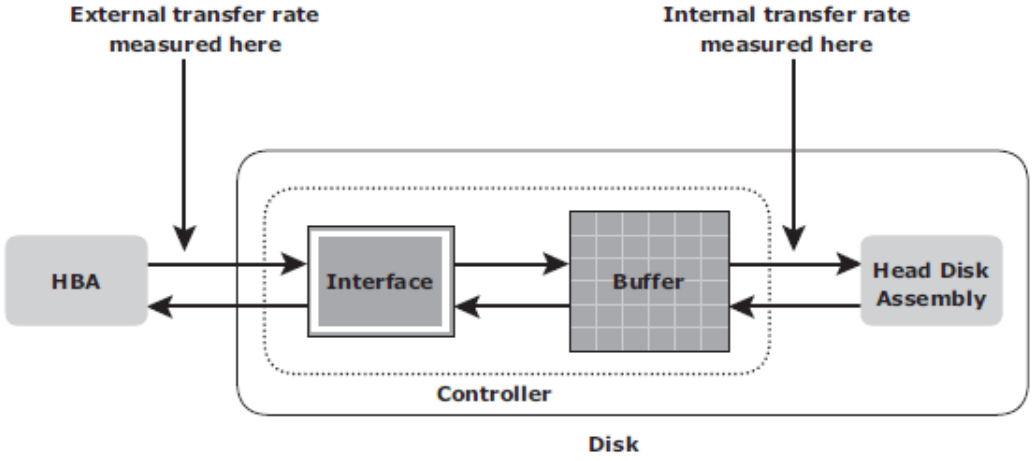


Figure 2-11: Data transfer rate

Disk I/O Controller Utilization

- Utilization of a disk I/O controller has a significant 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 Average response time (T_R) = Service time (T_S) / (1 – 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.

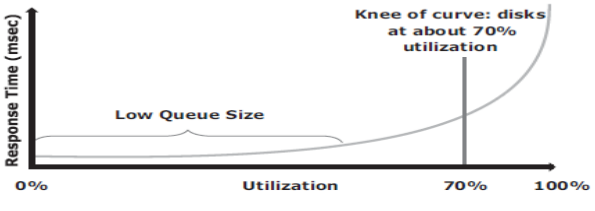


Figure 2-13: Utilization versus response time

- 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.

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.
- The storage device can be internal and (or) external to the host. In either case, the host controller card accesses the storage devices using predefi ned 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 file system (or operating system) by specifying the filename and location. The file system maps the file 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 file 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 file system.
- In a file-level access, the file system is created on a separate file server or at the storage side, and the fi le-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.

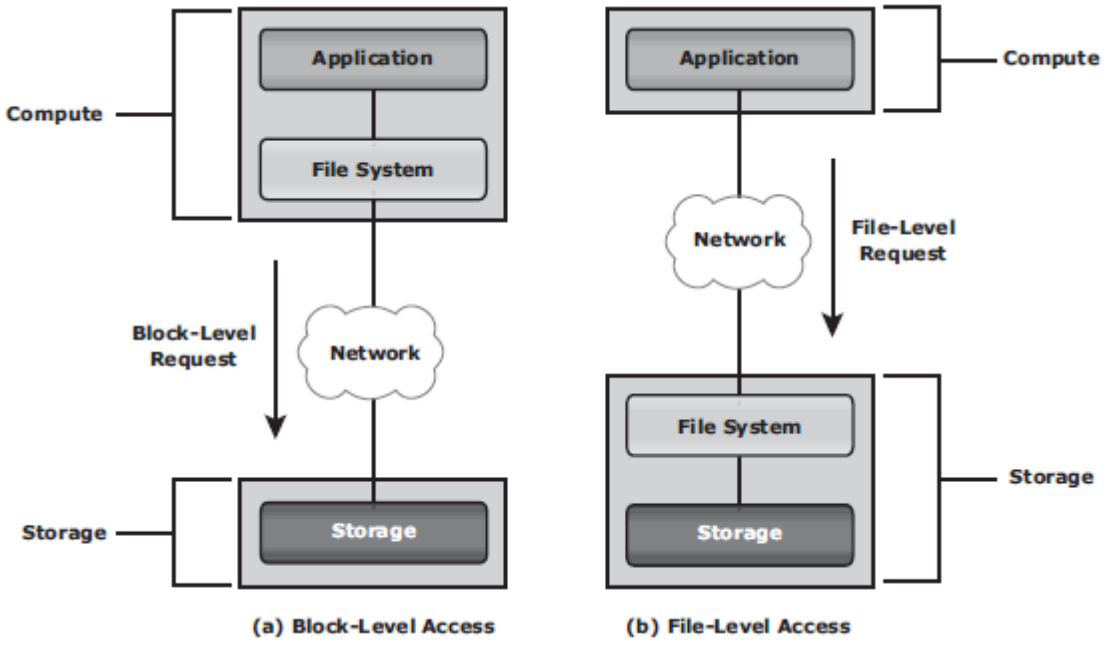
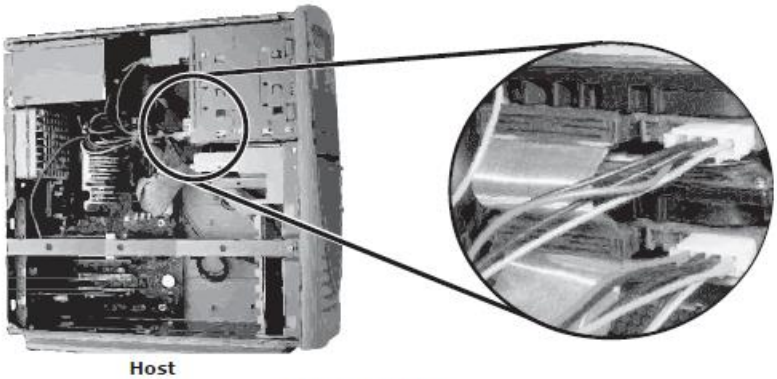


Figure 2-14: Host access to storage

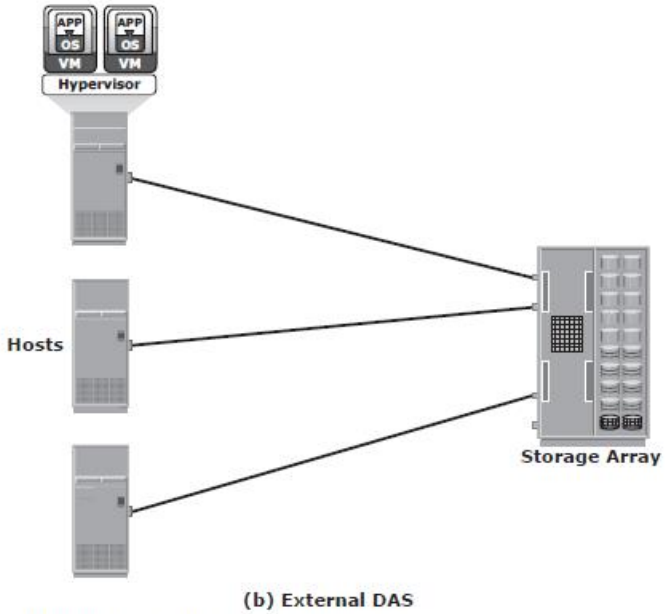
Direct-Attached Storage

- 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.
- Although the implementation of storage networking technologies is gaining popularity, DAS has remained suitable for localized data access in a small environment, such as personal computing and workgroups.
- 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.
- 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

DAS Benefits and Limitations

- 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.
- 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.

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 fi le 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 (TS) is the sum of the seek time (T), rotational latency (L), and internal transfer time (X):

$$T_S = T + L + X$$

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).

- The application response time, R, increases with an increase in disk controller utilization.
- 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/O 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.
- The total number of disks required (D_R) for an application is computed as follows: DR = 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.