Information Storage and Management™
Volume 1 of 2
Student Guide

EMC Education Services
July 2009
Course Introduction

Welcome to Information Storage and Management.
Welcome
Introductions

• Name
• Job
• Background in IT
• Business and/or location
• Expectations of this course
Class Logistics and Operations

- Health & safety
- Lunch & breaks
- Cell phone use and location of other phones
- Systems for outside access
- Schedule
- Materials
Purpose
Course Objectives

Upon completion of this course, you will be able to:

• Describe the challenges found in today’s complex information management environment

• Describe storage technology solutions (such as DAS, NAS, SAN and Virtualization technologies)

• Describe various Business Continuity solutions: Backup and Replication

• Describe common storage management roles and responsibilities

• Describe the processes and technologies for identifying, analyzing, and mitigating security risks in storage infrastructure

The objectives for this course are shown here. Please take a moment to read them.
Storage System

Section 1
Section Objective

Upon completion of this section, you will be able to:

- Describe the challenges in information storage and management
- Describe the core elements in a data center
- Describe RAID and its various levels
- Describe features of intelligent disk storage systems

The objectives for this section are shown here. Please take a moment to read them.
Introduction to Information Storage and Management

Module 1.1
Information is increasingly important in our daily lives. We have become information dependents of the twenty-first century, living in an 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, take pictures and videos through digital cameras, and satisfy many other personal and professional needs. 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.
Module Objective

Upon completion of this module, you will be able to:

• Describe who is creating data and the amount of data being created
• Describe the value of data to business
• Describe storage technology and architecture evolution
• List and explain the core elements of data center
• Describe the ILM strategy

These are the objectives for this module. Please take a moment to review them.
Lesson: Information Storage

Upon completion of this lesson, you will be able to:

• Describe the importance of information to individuals and to businesses
• Define data and information
• Discuss the categories of data
• Describe the storage architectures and their evolution

This lesson defines data, information and storage, it also covers evolution of storage technology over a period of time.
Data is a collection of raw facts from which conclusions may be drawn. Handwritten letters, a printed book, a family photograph, a movie on video tape, printed and duly signed copies of mortgage papers, a bank’s ledgers, and an account holder’s passbooks are all examples of data. Before the advent of computers, the procedures and methods adopted for data creation and sharing were limited to fewer forms, such as paper and film. Today, the same data can be converted into more convenient forms such as an e-mail message, an e-book, a bitmapped image, or a digital movie. This is shown in the figure. Data in this form is called digital data and is accessible by the user only after it is processed by a computer.

Inexpensive and easier ways to create, collect, and store all types of data, coupled with increasing individual and business needs, have led to accelerated data growth, popularly termed the data explosion. Data has different purposes and criticality, so both individuals and businesses have contributed in varied proportions to this data explosion.
Data can be classified as structured or unstructured based on how it is stored and managed. 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).

Data is unstructured if its elements cannot be stored in rows and columns, and is therefore difficult to query and retrieve by business applications. For example, customer contacts may be stored in various forms such as sticky notes, e-mail messages, business cards, or even digital format files such as .doc, .txt, and .pdf. Due its unstructured nature, it is difficult to retrieve using a customer relationship management application. Unstructured data may not have the required components to identify itself uniquely for any type of processing or interpretation. Businesses are primarily concerned with managing unstructured data because over 80 percent of enterprise data is unstructured and requires significant storage space and effort to manage.
Define Information

- What do individuals/businesses do with the data they collect?
  - They turn it into “information”
  - “Information is the intelligence and knowledge derived from data”

- Businesses analyze raw data in order to identify meaningful trends
  - For example:
    - Buying habits and patterns of customers
    - Health history of patients

Data, whether structured or unstructured, does not fulfill any purpose for individuals or businesses unless it is presented in a meaningful form. Businesses need to analyze data for it to be of value. 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.

Information created by individuals gains value when shared with others. When created, information resides locally on devices such as cell phones, cameras, and laptops. To share this information, it needs to be uploaded via networks to data centers. It is interesting to note that while the majority of information is created by individuals, it is stored and managed by a relatively small number of organizations. Figure depicts this virtuous cycle of information.
**Value of Information to a Business**

- Identifying new business opportunities
  - Buying/spending patterns
    - Internet stores, retail stores, supermarkets
  - Customer satisfaction/service
    - Tracking shipments, and deliveries

- Identifying patterns that lead to changes in existing business
  - Reduced cost
    - Just-in-time inventory, eliminating over-stocking of products, optimizing shipment and delivery
  - New services
    - Security alerts for “stolen” credit card purchases
  - Targeted marketing campaigns
    - Communicate to bank customers with high account balances about a special savings plan

- Creating a competitive advantage

Effective data analysis not only extends its benefits to existing businesses, but also creates the potential for new business opportunities by using the information in creative ways. Job portal is an example. In order to reach a wider set of prospective employers, job seekers post their résumés on various websites offering job search facilities. These websites collect the resumes and post them on centrally accessible locations for prospective employers. In addition, companies post available positions on job search sites. Job-matching software matches keywords from résumés to keywords in job postings. In this manner, the job search engine uses data and turns it into information for employers and job seekers.
### Storage

- Data created by individuals/businesses must be stored for further processing
- Type of storage used is based on the type of data and the rate at which it is created and used
- Examples:
  - Individuals: Digital camera, Cell phone, DVD’s, Hard disk
  - Businesses: Hard disk, external disk arrays, tape library
- Storage model: An evolution
  - Centralized: mainframe computers
  - Decentralized: Client –server model
  - Centralized: Storage Networking

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.

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.

The proliferation of departmental servers in an enterprise resulted in unprotected, unmanaged, fragmented islands of information and increased operating cost. Originally, there were very limited policies and processes for managing these servers and the data created. To overcome these challenges, storage technology evolved from non-intelligent internal storage to intelligent networked storage.
Highlights of storage technology and architecture evolution include:

- **Redundant Array of Independent Disks (RAID):** This technology was developed to address the performance and availability requirements of data. It continues to evolve today and is used in all storage architectures such as DAS, SAN, and so on.

- **Direct-attached storage (DAS):** This type of storage connects directly to a server (host) or a group of servers in a cluster. Storage can be either internal or external to the server. External DAS alleviated the challenges of limited internal storage capacity.

- **Storage area network (SAN):** This is a dedicated, high-performance Fibre Channel (FC) network to facilitate block-level communication between servers and storage. Storage is partitioned and assigned to a server for accessing its data. SAN offers scalability, availability, performance, and cost benefits compared to DAS.

- **Network-attached storage (NAS):** This is dedicated storage for file serving applications. Unlike a SAN, it connects to an existing communication network (LAN) and provides file access to heterogeneous clients. Because it is purposely built for providing storage to file server applications, it offers higher scalability, availability, performance, and cost benefits compared to general purpose file servers.

- **Internet Protocol SAN (IP-SAN):** One of the latest evolutions in storage architecture, IP-SAN is a convergence of technologies used in SAN and NAS. IP-SAN provides block-level communication across a local or wide area network (LAN or WAN), resulting in greater consolidation and availability of data.

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. These technologies are discussed in detail later in the course.
Lesson Summary

Key points covered in this lesson:

- Importance of information
- Data, information and storage
- Categories of data
- Storage architectures and their evolution

These are the key points covered in this lesson. Please take a moment to review them.
Lesson: Data Center Infrastructure and Introduction to ILM

Upon completion of this lesson, you will be able to:

• List the five core elements of a data center infrastructure

• Describe the requirements of storage systems for optimally supporting business activities

• Explain the importance of Information Lifecycle Management

• List the activities in developing the ILM strategy

So far we have looked at the importance of data and information to the business (competitive advantage) and the proliferation of data that needs to be stored efficiently.

This lesson presents the elements of the data center and the challenges of managing a complex storage systems environment.
Organizations maintain data centers to provide centralized data processing capabilities across the enterprise. Data centers store and manage large amounts of mission-critical data. The data center infrastructure includes servers, storage systems, network devices, dedicated power backups, and environmental controls (such as air conditioning and fire suppression). Large organizations often maintain more than one data center to distribute data processing workloads and provide backups in the event of a disaster. The storage requirements of a data center are met by a combination of various storage architectures.

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

**Application:** An application is a computer program that provides the logic for computing operations. Applications, such as an order processing system, can be layered on a database, which in turn uses operating system services to perform read/write operations to storage devices.

**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.

**Server and operating system:** A computing platform that runs applications and databases.

**Network:** A data path that facilitates communication between clients and servers or between servers and storage.

**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.

These core elements are typically viewed and managed as separate entities, but all the elements must work together to address data processing requirements.
Uninterrupted operation of data centers is critical to the survival and success of a business. It is necessary to have a reliable infrastructure that ensures data is accessible at all times. While the requirements, shown in above figure, are applicable to all elements of the data center infrastructure, our focus here is on storage systems.

**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.

**Security:** Polices, procedures, and proper integration of the data center core elements that will prevent unauthorized access to information must be established. In addition to the security measures for client access, specific mechanisms must enable servers to access only their allocated resources on storage arrays.

**Scalability:** Data center operations should be able to allocate additional processing capabilities or storage on demand, without interrupting business operations. Business growth often requires deploying more servers, new applications, and additional databases. The storage solution should be able to grow with the business.

**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.

**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.

**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.

**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. Managing a modern, complex data center involves many tasks. Key management activities include Monitoring, Reporting and Provisioning. This activities are covered in detail later in the course.
Challenges in Managing Information

- Exploding digital universe
  - Multifold increase of information growth

- Increasing dependency on information
  - The strategic use of information plays an important role in determining the success of a business

- Changing value of information
  - Information that is valuable today may become less important tomorrow.

In order to frame an effective information management policy, businesses need to consider the following key challenges of information management:

**Exploding digital universe:** The rate of information growth is increasing exponentially. Duplication of data to ensure high availability and repurposing has also contributed to the multifold increase of information growth.

**Increasing dependency on information:** The strategic use of information plays an important role in determining the success of a business and provides competitive advantages in the marketplace.

**Changing value of information:** Information that is valuable today may become less important tomorrow. The value of information often changes over time. Framing a policy to meet these challenges involves understanding the value of information over its lifecycle.
The information lifecycle is the “change in the value of information” over time. When data is first created, it often has the highest value and is used frequently. As data ages, it is accessed less frequently and is of less value to the organization. Understanding the information lifecycle helps to deploy appropriate storage infrastructure, according to the changing value of information. For example, in a sales order application, the value of the information changes from the time the order is placed until the time that the warranty becomes void. The value of the information is highest when a company receives a new sales order and processes it to deliver the product. After order fulfillment, the customer or order data need not be available for real-time access. The company can transfer this data to less expensive secondary storage with lower accessibility and availability requirements unless or until a warranty claim or another event triggers its need. After the warranty becomes void, the company can archive or dispose of data to create space for other high-value information. Today’s business requires data to be protected and available 24 × 7. Data centers can accomplish this with the optimal and appropriate use of storage infrastructure. An effective information management policy is required to support this infrastructure and leverage its benefits.

Information lifecycle management (ILM) is a proactive strategy that enables an IT organization to effectively manage the data throughout its lifecycle, based on predefined business policies. This allows an IT organization to optimize the storage infrastructure for maximum return on investment. An ILM strategy should include the following characteristics:

**Business-centric:** It should be integrated with key processes, applications, and initiatives of the business to meet both current and future growth in information.

**Centrally managed:** All the information assets of a business should be under the purview of the ILM strategy.

**Policy-based:** The implementation of ILM should not be restricted to a few departments. ILM should be implemented as a policy and encompass all business applications, processes, and resources.

**Heterogeneous:** An ILM strategy should take into account all types of storage platforms and operating systems.

**Optimized:** Because the value of information varies, an ILM strategy should consider the different storage requirements and allocate storage resources based on the information’s value to the business.

**Tiered Storage:** Tiered storage is an approach to define different storage levels in order to reduce total storage cost. Each tier has different levels of protection, performance, data access frequency, and other considerations. Information is stored and moved between different tiers based on its value over time. For example, mission-critical, most accessed information may be stored on Tier 1 storage, which consists of high performance media with a highest level of protection. Medium accessed and other important data is stored on Tier 2 storage, which may be on less expensive media with moderate performance and protection. Rarely accessed or event specific information may be stored on lower tiers of storage.
ILM implementation is the process of developing an ILM strategy includes four activities—classifying, implementing, managing, and organizing.

Classifying data and applications on the basis of business rules and policies to enable differentiated treatment of information

Implementing policies by using information management tools, starting from the creation of data and ending with its disposal

Managing the environment by using integrated tools to reduce operational complexity

Organizing storage resources in tiers to align the resources with data classes, and storing information in the right type of infrastructure based on the information’s current value

Implementing ILM across an enterprise is an ongoing process. Following are the steps for implementing ILM enterprise wide:

Initially implement ILM in a limited way across a few enterprise-critical applications. the goal is to implement a storage networking environment. The value of tiered storage platforms can be exploited by allocating appropriate storage resources to the applications based on the value of the information processed. In the next step, ILM provides detailed application/data classification and linkage of the storage infrastructure to business policies. These classifications and the resultant policies can be automatically executed using tools for one or more applications, resulting in better management and optimal allocation of storage resources. Further step in the implementation is to automate more of the application and policy management activities in order to scale to a wider set of enterprise applications.
Implementing an ILM strategy has the following key benefits that directly address the challenges of information management:

* Improved utilization by using tiered storage platforms and increased visibility of all enterprise information.

* Simplified management by integrating process steps and interfaces with individual tools and by increasing automation.

* A wider range of options for backup, and recovery to balance the need for business continuity.

* Maintaining compliance by knowing what data needs to be protected for what length of time.

* Lower Total Cost of Ownership (TCO) by aligning the infrastructure and management costs with information value. As a result, resources are not wasted, and complexity is not introduced by managing low-value data at the expense of high-value data.
Lesson Summary

Key points covered in this lesson:

- The five core elements of a Data Center infrastructure
- Key requirements of storage systems to support business activities, as well as some of the constraints
- ILM strategy
  - Importance
  - Characteristics
  - Activities in developing ILM strategy
  - IML implementation
  - Benefits of ILM

These are the key points covered in this lesson. Please take a moment to review them.
Module Summary

Key points covered in this module:

- Importance of data, information, and storage infrastructure
- Types of data, its value, and key management requirements of a storage system
- Evolution of storage architectures
- Core elements of a data center
- Importance of the ILM strategy

These are the key points covered in this module. Please take a moment to review them.
ILM Video
Check Your Knowledge

- What are the two categories of data?
- What are the five core technology elements of the Data Center Infrastructure?
- What are the seven requirements of storage technology?
- What are the benefits of ILM
Module 1.2
Module Objective

Upon completion of this module, you will be able to:

• List components of storage system environment
  – Host, connectivity and storage

• List physical and logical components of hosts

• Describe key connectivity options

• Describe the physical disk structure

• Discuss factors affecting disk drive performance

The objectives for this module are shown here. Please take a moment to read them.
Lesson: Components of Storage System Environment

Upon completion of this lesson, you will be able to:

- Describe the three components of storage system environment
  - Host, Connectivity and Storage
- Detail Host physical and logical components
- Describe interface protocol
  - PCI, IDE/ATA and SCSI
- Describe storage options
  - Tape, optical and disk drives

The objectives for this lesson are shown here. Please take a moment to read them.
Users store and retrieve data through applications. The computers on which these applications run are referred to as hosts. Hosts can range from simple laptops to complex clusters of servers. A host consists of physical components (hardware devices) that communicate with one another using logical components (software and protocols). Access to data and the overall performance of the storage system environment depend on both the physical and logical components of a host.

A host has three key physical components: Central processing unit (CPU), Storage (such as internal memory and disk devices) and Input/Output (I/O) devices.

I/O devices enable sending and receiving data to and from a host. Communication between various devices takes place in the following way:

**User to host communications:** Handled by basic I/O devices, such as the keyboard, mouse, and monitor. These devices enable users to enter data and view the results of operations.

**Host to host communications:** Enabled using devices such as a Network Interface Card (NIC) or modem.

**Host to storage device communications:** Handled by a Host Bus Adaptor (HBA). HBA is an application-specific integrated circuit (ASIC) board that performs I/O interface functions between the host and the storage, relieving the CPU from additional I/O processing workload. HBAs also provide connectivity outlets known as ports to connect the host to the storage device. A host may have multiple HBAs.
The logical components of a host consist of the software applications and protocols that enable data communication with the user as well as the physical components. Following are the logical components of a host:

- Application
- Operating system
- File system
- Volume manager
- Device drivers
Logical Components of the Host

• Application
  – Interface between user and the host
  – Three-tiered architecture
    ➢ Application UI, computing logic and underlying databases
  – Application data access can be classified as:
    ➢ Block-level access: Data stored and retrieved in blocks, specifying the LBA
    ➢ File-level access: Data stored and retrieved by specifying the name and path of files

• Operating system
  – Resides between the applications and the hardware
  – Controls the environment

An application is a computer program that provides the logic for computing operations. It provides an interface between the user and the host and among multiple hosts. Conventional business applications using databases have a three-tiered architecture — the application user interface forms the front-end tier; the computing logic forms, or the application itself is, the middle tier; and the underlying databases that organize the data form the back-end tier. The application sends requests to the underlying operating system to perform read/write (R/W) operations on the storage devices. Applications can be layered on the database, which in turn uses the OS services to perform R/W operations to storage devices. These R/W operations (I/O operations) enable transactions between the front-end and back-end tiers. Data access can be classified as block-level or file-level depending on whether the application uses a logical block address or the file name and a file record identifier to read from and write to a disk.

Block-level access is the basic mechanism for disk access. In this type of access, data is stored and retrieved from disks by specifying the logical block address. The block address is derived based on the geometric configuration of the disks. Block size defines the basic unit of data storage and retrieval by an application. Databases, such as Oracle and SQL Server, define the block size for data access and the location of the data on the disk in terms of the logical block address when an I/O operation is performed.

File-level access is an abstraction of block-level access. File-level access to data is provided by specifying the name and path of the file. It uses the underlying block-level access to storage and hides the complexities of logical block addressing (LBA) from the application and the DBMS.

An operating system controls all aspects of the computing environment. It works between the application and physical components of the computer system. One of the services it provides to the application is data access. The operating system also monitors and responds to user actions and the environment. It organizes and controls hardware components and manages the allocation of hardware resources. It provides basic security for the access and usage of all managed resources. An operating system also performs basic storage management tasks while managing other underlying components, such as the file system, volume manager, and device drivers.
Logical Components of the Host: LVM

- Responsible for creating and controlling host level logical storage
  - Physical view of storage is converted to a logical view by mapping
  - Logical data blocks are mapped to physical data blocks

- Usually offered as part of the operating system or as third party host software

- LVM Components:
  - Physical Volumes
  - Volume Groups
  - Logical Volumes

Logical Volume Managers (LVMs) introduce a logical layer between the operating system and the physical storage. LVMs have the ability to define logical storage structures that can span multiple physical devices. The logical storage structures appear contiguous to the operating system and applications.

The fact that logical storage structures can span multiple physical devices provides flexibility and additional functionality:

- Dynamic extension of file systems
- Host based mirroring
- Host based striping

The Logical Volume Manager provides a set of operating system commands, library subroutines, and other tools that enable the creation and control of logical storage.
Volume Groups

- One or more Physical Volumes form a Volume Group
- LVM manages Volume Groups as a single entity
- Physical Volumes can be added and removed from a Volume Group as necessary
- Physical Volumes are typically divided into contiguous equal-sized disk blocks
- A host will always have at least one disk group for the Operating System
  - Application and Operating System data maintained in separate volume groups

A Volume Group is created by grouping together one or more Physical Volumes. Physical Volumes can be added or removed from a Volume Group dynamically. It cannot be shared between Volume Groups, the entire Physical Volume becomes part of a Volume Group. Each Physical Volume is partitioned into equal-sized data blocks. The size of a Logical Volume is based on a multiple of the equal-sized data block. The Volume Group is handled as a single unit by the LVM, a Volume Group as a whole can be activated or deactivated. A Volume Group would typically contain related information. For example, each host would have a Volume Group which holds all the OS data, while applications would be on separate Volume Groups.

Logical Volumes are created within a given Volume Group. A Logical Volume can be thought of as a virtual disk partition, while the Volume Group itself can be thought of as a disk. A Volume Group can have a number of Logical Volumes. Logical Volumes (LV) form the basis of logical storage. They contain logically contiguous data blocks (or logical partitions) within the volume group. Each logical partition is mapped to at least one physical partition on a physical volume within the Volume Group. The OS treats an LV like a physical device and accesses it via device special files (character or block). A Logical Volume can only belong to one Volume Group. However, a Volume Group can have multiple LVs and can span multiple physical volumes. It can be made up of physical disk blocks that are not physically contiguous but appears as a series of contiguous data blocks to the OS. LV can contain a file system or be used directly.

Note: 1. There is a one-to-one relationship between LV and a File System.

2. Under normal circumstances, there is a one-to-one mapping between a logical and physical Partition. A one-to-many mapping between a logical and physical partition leads to mirroring of Logical Volumes.
Disk partitioning was introduced to improve the flexibility and utilization of HDDs. In partitioning, an HDD is divided into logical containers called logical volumes (LVs). For example, a large physical drive can be partitioned into multiple LVs to maintain data according to the file system’s and applications’ requirements. The partitions are created from groups of contiguous cylinders when the hard disk is initially set up on the host. The host’s file system accesses the partitions without any knowledge of partitioning and the physical structure of the disk.

Concatenation is the process of grouping several smaller physical drives and presenting them to the host as one logical drive.
**Logical Components of the Host (Cont)**

- **Device Drivers**
  - Enables operating system to recognize the device
  - Provides API to access and control devices
  - Hardware dependent and operating system specific

- **File System**
  - File is a collection of related records or data stored as a unit
  - File system is hierarchical structure of files
    - Examples: FAT 32, NTFS, UNIX FS and EXT2/3

---

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 hard drive. A device driver enables the operating system to recognize the device and to use a standard interface (provided as an *application programming interface*, or *API*) to access and control devices. Device drivers are hardware dependent and operating system specific.

A file is a collection of related records or data stored as a unit with a name. A file system is a hierarchical structure of files. File systems enable easy access to data files residing within a disk drive, a disk partition, or a logical volume. A file system needs host-based logical structures and software routines that control access to files. It provides users with the functionality to create, modify, delete, and access files. Access to the files on the disks is controlled by the permissions given 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. Some of the common file systems are FAT 32 (File Allocation Table) and NT File System (NTFS) for Microsoft Windows, UNIX File System (UFS) for UNIX and Extended File System (EXT2/3) for Linux.

Apart from the files and directories, the file system also includes a number of other related records, which are collectively called the *metadata*. The metadata of a file system has to be consistent in order for the file system to be considered healthy. A metadata contains information about the file system, such as the file system type, creation and modification dates, size and layout, the count of available resources (such as number of free blocks, inodes, etc.), and a flag indicating the mount status of the file system.
A file system block is the smallest “container” of physical disk space allocated for data. Each file system block is a contiguous area on the physical disk. The block size of a file system is fixed at the time of its creation. File system size depends on block size and the total number of blocks of data stored. A file can span multiple file system blocks because most files are larger than the predefined block size of the file system. File system blocks cease to be contiguous (i.e., become fragmented) when new blocks are added or deleted. Over time, as files grow larger, the file system becomes increasingly fragmented.

Shown here is the following process of mapping user files to the disk storage subsystem with an LVM:

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

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

The file system tree starts with the root directory. The root directory has a number of subdirectories. A file system should be mounted before it can be used.
Connectivity

• Interconnection between hosts or between a host and any storage devices

• Physical Components of Connectivity are:
  – Bus, port and cable

Connectivity refers to the interconnection between hosts or between a host and any other peripheral devices, such as printers or storage devices. The discussion here focuses on the connectivity between the host and the storage device. The components of connectivity in a storage system environment can be classified as physical and logical. The *physical components* are the hardware elements that connect the host to storage and the *logical components* of connectivity are the protocols used for communication between the host and storage.

The three physical components of connectivity between the host and storage are Bus, Port, and Cable.
Connectivity Protocol

- Protocol = a defined format for communication between sending and receiving devices

- Tightly connected entities such as central processor to RAM, or storage buffers to controllers (example PCI)
- Directly attached entities connected at moderate distances such as host to storage (example IDE/ATA)
- Network connected entities such as networked hosts, NAS or SAN (example SCSI or FC)

Protocol is a defined format, in this case, for communication between hardware or software components. Communication protocols are defined for systems and components that are:

- Tightly connected entities – such as central processor to RAM, or storage buffers to controllers – use standard BUS technology (e.g. System bus or I/O – Local Bus)
- Directly attached entities or devices connected at moderate distances – such as host to printer or host to storage
- Network connected entities – such as networked hosts, Network Attached Storage (NAS) or Storage Area Networks (SAN)

The popular interface protocol used for the local bus to connect to a peripheral device is Peripheral Component Interconnect (PCI). The interface protocols that connect to disk systems are Integrated Device Electronics/Advanced Technology Attachment (IDE/ATA) and Small Computer System Interface (SCSI).
**Popular Connectivity Options: PCI**

- PCI is used for local bus system within a computer
- It is an interconnection between microprocessor and attached devices
- Has Plug and Play functionality
- PCI is 32/64 bit
- Throughput is 133 MB/sec
- **PCI Express**
  - Enhanced version of PCI bus with higher throughput and clock speed

PCI is a specification that standardizes how PCI expansion cards, such as network cards or modems, exchange information with the CPU. PCI provides the interconnection between the CPU and attached devices. The plug-and-play functionality of PCI enables the host to easily recognize and configure new cards and devices. The width of a PCI bus can be 32 bits or 64 bits. A 32-bit PCI bus can provide a throughput of 133 MB/s. **PCI Express** is an enhanced version of PCI bus with considerably higher throughput and clock speed.
Popular Connectivity Options: IDE/ATA

- Integrated Device Electronics (IDE) / Advanced Technology Attachment (ATA)
  - Most popular interface used with modern hard disks
  - Good performance at low cost
  - Inexpensive storage interconnect
  - Used for internal connectivity

- Serial Advanced Technology Attachment (SATA)
  - Serial version of the IDE /ATA specification
  - Hot-pluggable
  - Enhanced version of bus provides up to 6 Gb/s (revision 3.0)

An Integrated Device Electronics/Advanced Technology Attachment (IDE/ATA) disk supports the IDE protocol. The term IDE/ATA conveys the dual-naming conventions for various generations and variants of this interface. The IDE component in IDE/ATA provides the specification for the controllers connected to the computer’s motherboard for communicating with the device attached. The ATA component is the interface for connecting storage devices, such as CD-ROMs, floppy disk drives, and HDDs, to the motherboard. IDE/ATA has a variety of standards and names, such as ATA, ATA/ATAPI, EIDE, ATA-2, Fast ATA, ATA-3, Ultra ATA, and Ultra DMA. The latest version of ATA—Ultra DMA/133—supports a throughput of 133 MB per second. In a master-slave configuration, an ATA interface supports two storage devices per connector. However, if the performance of the drive is important, sharing a port between two devices is not recommended. A 40-pin connector is used to connect ATA disks to the motherboard, and a 34-pin connector is used to connect floppy disk drives to the motherboard. An IDE/ATA disk offers excellent performance at low cost, making it a popular and commonly used hard disk.

A SATA (Serial ATA) is a serial version of the IDE/ATA specification. SATA is a disk-interface technology that was developed by a group of the industry’s leading vendors with the aim of replacing parallel ATA. A SATA provides point-to-point connectivity and enhance version (version 3.0) enables data transfer at a speed up to 6 Gb/s. A SATA bus directly connects each storage device to the host through a dedicated link, making use of low-voltage differential signaling (LVDS). LVDS is an electrical signaling system that can provide high-speed connectivity over low-cost, twisted-pair copper cables. For data transfer, a SATA bus uses LVDS with a voltage of 250 mV. A SATA bus uses a small 7-pin connector and a thin cable for connectivity. A SATA port uses 4 signal pins, which improves its pin efficiency compared to the parallel ATA that uses 26 signal pins, for connecting an 80-conductor ribbon cable to a 40-pin header connector. SATA devices are hot-pluggable, and permits single-device connectivity. Connecting multiple SATA drives to a host requires multiple ports to be present on the host.
Popular Connectivity Options: SCSI

- Parallel SCSI (Small computer system interface)
  - Most popular hard disk interface for servers
  - Higher cost than IDE/ATA
  - Supports multiple simultaneous data access
  - Used primarily in “higher end” environments
  - SCSI Ultra provides data transfer speeds of 320 MB/s

- Serial SCSI
  - Supports data transfer rate of 3 Gb/s (SAS 300)

SCSI is available in a variety of interfaces.

Parallel SCSI (referred to as SCSI) is one of the oldest and most popular forms of storage interface used in hosts. SCSI is a set of standards used for connecting a peripheral device to a computer and transferring data between them. Often, SCSI is used to connect HDDs and tapes to a host. SCSI can also connect a wide variety of other devices such as scanners and printers. Communication between the hosts and the storage devices uses the SCSI command set. Since its inception, SCSI has undergone rapid revisions, resulting in continuous performance improvements. The oldest SCSI variant, called SCSI-1 provided data transfer rate of 5 MB/s; SCSI Ultra 320 provides data transfer speeds of 320 MB/s.

Serial Attached SCSI (SAS) is the evolution of SCSI beyond SCSI Ultra 320. SAS addresses the scalability, performance, reliability, and manageability requirements of a data center while leveraging a common electrical and physical connection interface with SATA. SAS uses SCSI commands for communication and is pin compatible with SATA. SAS supports data transfer rate of 3 Gb/s (SAS 300). It supports dual porting, full-duplex, device addressing, and uses a simplified protocol to minimize interoperability issues between controllers and drives. It also enables connectivity to multiple devices through expanders and is commonly preferred over SCSI in high-end servers for faster disk access.
The storage device is the most important component in the storage system environment. A storage device uses. Disks, tapes, and diskettes use magnetic media.

*Tapes* are a popular storage media used for backup because of their relatively low cost. However, tape has various limitations; data is stored on the tape linearly along the length of the tape. Search and retrieval of data is done sequentially, invariably taking several seconds to access the data. As a result, random data access is slow and time consuming. This limits tapes as a viable option for applications that require real-time, rapid access to data. 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. On a tape drive, the read/write head touches the tape surface, so the tape degrades or wears out after repeated use. The storage and retrieval requirements of data from tape and the overhead associated with managing tape media are significant. Even with all these limitations, tape is not yet obsolete.

*Optical disk storage* is popular in small, single-user computing environments. Optical disks have limited capacity and speed, which limits the use of optical media as a business data storage solution. The capability to write once and read many (WORM) is one advantage of optical disk storage. A CD-ROM is an example of a WORM device. Optical disks, to some degree, guarantee that the content has not been altered, so they can be used as low-cost alternatives for long-term storage of relatively small amounts of fixed content that will not change after it is created. Collections of optical disks in an array, called *jukeboxes*, are still used as a fixed-content storage solution.

*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. This means that data can be written or retrieved quickly for a large number of simultaneous users or applications. In addition, disks have a large capacity.
Lesson Summary

Key points covered in this lesson:

• Host components
  – Physical and Logical

• Connectivity options
  – PCI, IDE/ATA, SCSI

• Storage options
  – Tape, optical and disk drive

These are the key points covered in this section. Please take a moment to review them.
Lesson: Disk Drive

Upon completion of this lesson, you will be able to:

- List and discuss various disk drive components
  - Platter, spindle, read/write head and actuator arm assembly
- Discuss disk drive geometry
- Describe CHS and LBA addressing scheme
- Disk drive performance
  - Seek time, rotational latency and transfer rate
- Law’s governing disk drive performance
- Enterprise flash drive

The objectives for this lesson are shown here. Please take a moment to read them.
A disk drive uses a rapidly moving arm to read and write data across a flat platter coated with magnetic particles. Data is transferred from the magnetic platter through the R/W head to the computer. Several platters are assembled together with the R/W head and controller, most commonly referred to as a hard disk drive (HDD). Data can be recorded and erased on a magnetic disk any number of times. This section details the different components of the disk, the mechanism for organizing and storing data on disks, and the factors that affect disk performance.

Key components of a disk drive are platter, spindle, read/write head, actuator arm assembly, and controller:

A typical HDD consists of one or more flat circular disks called platters. The data is recorded on these platters in binary codes (0s and 1s). The set of rotating platters is sealed in a case, called a Head Disk Assembly (HDA). 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.

A spindle connects all the platters, as shown in the figure, and is connected to a motor. The motor of the spindle rotates with a constant speed. The disk platter spins at a speed of several thousands of revolutions per minute (rpm). Disk drives have spindle speeds of 7,200 rpm, 10,000 rpm, or 15,000 rpm. Disks used on current storage systems have a platter diameter of 3.5” (90 mm). When the platter spins at 15,000 rpm, the outer edge is moving at around 25 percent of the speed of sound. The speed of the platter is increasing with improvements in technology, although the extent to which it can be improved is limited.
Data on the disk is recorded on *tracks*, which are concentric rings on the platter around the spindle. 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 formatting operation. The number of sectors per track varies according to the specific drive. 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 sector number, head number or platter number, and track number. This information helps the controller to locate the data on the drive, but storing this information consumes space on the disk. Consequently, there is a difference between the capacity of an unformatted disk and a formatted one. Drive manufacturers generally advertise the unformatted capacity—for example, a disk advertised as being 500GB will only hold 465.7GB of user data, and the remaining 34.3GB is used for *metadata*. A cylinder is the set of identical tracks on both surfaces of each drive platter. The location of drive heads is referred to by cylinder number, not by track number.
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 (a), and the host operating system had to be aware of the geometry of each disk being used. *Logical block addressing (LBA)*, shown in figure (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 only needs to know 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 (b), the drive shows eight sectors per track, eight heads, and four cylinders. This means a total of $8 \times 8 \times 4 = 256$ blocks, so the block number ranges from 0 to 255. Each block has its own unique address. Assuming that the sector holds 512 bytes, a 500 GB drive with a formatted capacity of 465.7 GB will have in excess of 976,000,000 blocks.
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** is the time taken by a disk to complete an I/O request. Components that contribute to service time on a disk drive are *seek time, rotational latency*, and *data transfer rate*. 
Disk Drive Performance: Seek Time

- Time taken to position the read/write head
- Lower the seek time, the faster the I/O operation
- Seek time specifications include:
  - Full stroke
  - Average
  - Track-to-track

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 reposition and settle the arm and the head over the correct track. 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 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. For example, a 500 GB disk drive is set up to use only the first 40 percent of the cylinders and is effectively treated as a 200 GB drive. This is known as short-stroking the drive.
**Disk Drive Performance: Rotational Speed/Latency**

- The time taken by platter to rotate and position the data under the R/W head
- Depends on the rotation speed of the spindle
- Average rotational latency
  - One-half of the time taken for a full rotation
  - Appx. 5.5 ms for 5400-rpm drive
  - Appx. 2.0 ms for 15000-rpm drive

Rotational delay (in sec) = \(0.5/(RPM/60)\)

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 is around 5.5 ms for a 5,400-rpm drive, and around 2.0 ms for a 15,000 rpm drive.
Disk Drive Performance: Data Transfer Rate

- Average amount of data per unit time
- Internal Transfer Rate
  - Speed at which data moves from a track to disk internal buffer
- External Transfer Rate
  - The advertised speed of the interface

Transfer time = Block size/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 and write operations in order to calculate data transfer rates. In a read operation, the data first moves from disk platters to R/W heads, and 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 here.

Internal transfer rate is the speed at which data moves from a single track of a platter’s surface to internal buffer (cache) of the disk. Internal transfer rate takes into account factors such as the seek time. External transfer rate is the rate at which data can be moved through the interface to the HBA. 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.
Fundamental Laws Governing Disk Performance

- **Little’s Law**
  - Describes the relationship between the number of requests in a queue and the response time.
  - \( N = a \times R \)
    - “\( N \)” is the total number of requests in the system
    - “\( a \)” is the arrival rate
    - “\( R \)” is the average response time

- **Utilization law**
  - Defines the I/O controller utilization
  - \( U = a \times R_S \)
    - “\( U \)” is the I/O controller utilization
    - “\( R_S \)” is the service time

To understand the laws of disk performance, 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.

**I/O Controller:** Processes I/Os that are 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. The I/O arrival rate, the queue length, and the time taken by the I/O controller to process each request determines the performance of the disk system, which is measured in terms of response time.

**Little’s Law** is a fundamental law describing the relationship between the number of requests in a queue and the response time. The law states the following relation:

\[ N = a \times R \]

Where
- “\( N \)” is the total number of requests in the queuing system (requests in the queue + requests in the I/O controller)
- “\( a \)” is the arrival rate, or the number of I/O requests that arrive to the system per unit of time
- “\( R \)” is the average response time or the turnaround time for an I/O request — the total time from arrival to departure from the system

The *utilization law* is another important law that defines the I/O controller utilization. This law states the relation:

\[ U = a \times R_S \]

Where “\( U \)” is the I/O controller utilization
- “\( R_S \)” is the *service time*, or the average time spent by a request on the controller. \( 1/R_S \) is the *service rate*. 
Consider a disk I/O system in which an I/O request arrives at a rate of 100 I/Os per second. The service time, $R_s$, is 4 ms.

- Utilization of I/O controller ($U = a \times R_s$)
- Total response time ($R = R_s / (1 - U)$)

Calculate the response time at different % of utilization

It can be concluded that by reducing the service time (the sum of seek time, latency, and internal transfer rate) or utilization by half, the response time can be reduced drastically. The relationship between utilization and response time is shown here.

Response time changes are nonlinear as utilization increases. When the average queue sizes are low, response time remains low. Response time increases slowly with added load on the queue, and increases exponentially when utilization exceeds 70 percent.
Application Requirements and Disk Performance Exercise

• Consider an application that requires 1TB of storage capacity and performs 4900 IOPS
  – Application I/O size is 4KB
  – As it is business critical application, response time must be within acceptable range

• Specification of available disk drive:
  – Drive capacity = 73 GB
  – 15000 RPM
  – 5 ms average seek time
  – 40 MB/sec transfer rate

  Calculate the number of disks required?
**Solution**

- Time required to perform one I/O is sum of seek time, rotational delay and transfer time
  
  Therefore, \(5 \text{ ms} + \frac{0.5}{(15000/60)} + \frac{4\text{KB}}{(40\text{MB/sec})} = 7.1 \text{ msec}\)

- Calculate max. number of IOPS a disk can perform
  
  \[\frac{1}{7.1 \text{ ms}} = 140 \text{ IOPS}\]

- For acceptable response time disk controller utilization must be less than 70%
  
  Therefore, \(140 \times 0.7 = 98 \text{ IOPS}\)

- To meet application
  
  - Performance requirement we need \(4900/98 \text{ i.e. 50 disk}\)
  
  - Capacity requirement we need \(1\text{TB}/73 \text{ GB i.e. 14 disk}\)

**Disk required = max (capacity, performance)**

To determine the number of disks required, we need to know two parameters of a disk:

1. Storage Capacity of a disk (capacity requirement)
2. Number of IOPS a disk can perform (performance requirement)

Now let us determine the number of IOPS a disk can perform: First calculate the time required to perform one I/O, which is a sum of seek time, rotational latency and transfer time.

Seek time = 5ms (given)

Rotational latency = One-half of the time taken for a full rotation

Transfer time = Block size/transfer rate

Therefore, \(5 \text{ ms} + \frac{0.5}{(15000/60)} + \frac{4\text{KB}}{(40\text{MB/sec})} = 7.1 \text{ msec}\)

Now calculate maximum number of IO a disk can perform in one sec = \(\frac{1}{7.1\text{ms}} = 140 \text{ IOPS}\)

For acceptable response time disk controller utilization must be less than 70%, Therefore, \(140 \times 0.7 = 98 \text{ IOPS}\)

To meet application

- Performance requirement we need \(4900/98 \text{ i.e. 50 disk}\)
- Capacity requirement we need \(1\text{TB}/73 \text{ GB i.e. 14 disk}\)

**Therefore disk required = max (14, 50) = 50**
Enterprise Flash Drives: A New Generation Drives

Conventional disk drive
- Mechanical Delay associated with conventional drive
  - Seek time
  - Rotational latency
- More power consumption due to mechanical operations
- Low Mean Time Between Failure

Enterprise flash drive
- Highest possible throughput per drive
  - No Spinning magnetic media
  - No Mechanical movement which causes seek and latency
  - Solid State enables consistent I/O performance
- Very low latency per I/O
- Energy efficient storage design
  - Lower power requirement per GB of storage
  - Lower power requirement per IOPS

Hard disk drives use spinning magnetic media to store digital information. In contrast, Enterprise Flash Drives leverage semiconductor-based block storage that behaves as a virtual hard disk drive via a traditional Fibre Channel interface. They are constructed with non-volatile semiconductor memory and fit into a 3.5” disk drive form factor.

When compared with conventional FC drives, EFDs provide better response time, higher throughput and consistent performance, even as the workload increases.

The recent announcement of Enterprise Flash Drives continues the energy saving trend. Enterprise Flash Drives consume significantly less power on a per-GB basis, as well as on a per IOPS basis - when compared to other disk technologies.
**Enterprise Flash Drives – Overview**

- Drive is based on Flash Solid State memory technology
  - High performance and low latency
  - Non volatile memory
  - Uses single layer cell (SLC) or Multi Level cell (MLC) to store data

- Enterprise Flash Drives use a 4Gb FC interface

With the introduction of the Enterprise Flash Drive (EFD), EMC has created a “Tier 0” ultra performance storage tier that transcends the limitations previously imposed by magnetic drives. EMC has optimized Solid State Drive technology to provide low latency and high throughput to break the performance barriers of traditional disk technology. Enterprise Flash Drives provide maximum performance for low latency applications.

Flash drives are constructed with nonvolatile semiconductor memory to support persistent storage and they use either single-level cell (SLC) or multi-level cell (MLC) to store bits on each memory cell. SLC stores one bit per cell and is used in high-performance memory cards. MLC memory cards store more bits per cell and provide slower transfer speeds. The advantage of MLC over SLC memory cards is the lower manufacturing cost.
Enterprise Flash Drives – Benefits

• Faster performance
  – Up to 30 times greater IOPS (benchmarked)
  – Typical applications: 8 – 12X
  – Less than 1 millisecond service time

• More energy efficient
  – 38 percent less per terabyte
  – 98 percent less per IO

• Better reliability
  – No moving parts
  – Faster RAID rebuilds

The benefits of Enterprise Flash Drives are numerous. EFDs offer up to 30 times greater IOs per second in random read workloads compared to FC hard disk drives. Service times for random I/O are far lower than FC: less than 1 millisecond. Enterprise Flash Drives use 38 percent less energy per terabyte and 98 percent less energy per IO. The absence of any moving parts provides for better reliability.
Enterprise Flash Drives – “Tier-0” Application

• Position Enterprise Flash Drives as the high-performance option in demanding environments
  – Low latency applications, also known as “Tier-0” applications

• Standard form-factor and capacity design allows for easier integration

• High performance, low power for a “Green” initiative

• Target Customer/Market Segments:
  – High performance solutions coupled with low power
  – Specifically target Oracle database customers initially
  – Financial trading
  – OLTP databases

Enterprise Flash Drives provide a very low latency, very high IOPS drive solution with lower energy utilization than traditional hard disk drives. This will give a competitive advantage to the organizations in meeting the business requirements of financial and other high transactions.

In applications such as financial trading transaction systems, small increases in performance can give substantial competitive advantage. These applications are dependent on both low latency and high IOPS to get the best possible performance. Enterprise Flash Drives support these customer requirements.
Lesson Summary

Key points covered in this lesson:

• Disk drive components and geometry
• Disk drive addressing scheme
• Disk drive performance
• Convention drive Vs Enterprise Flash Drives
• Enterprise Flash Drives for high performance and low power storage solution

These are the key points covered in this module. Please take a moment to review them.
Module Summary

Key points covered in this module:

• Storage system environment components:
  – Host, connectivity and storage
• Physical disk structure and addressing
• Factors affecting disk performance
• Flash drives benefits

These are the key points covered in this module. Please take a moment to review them.
Check Your Knowledge

• What are some examples of hosts?

• What are the physical and logical components of a host?

• What are the common connectivity protocols used in computing environments?

• What is the difference between seek time and rotational latency?

• What is the difference between internal and external data transfer rates?
Module 1.3 – Data Protection: RAID
Module Objective

After completing this module, you will be able to:

• Describe what is RAID and the needs it addresses
• Describe the concepts upon which RAID is built
• Define and compare RAID levels
• Recommend the use of the common RAID levels based on performance and availability considerations
• Explain factors impacting disk drive performance

The objectives for this module are shown here. Please take a moment to read them.
HDDs are susceptible to failures due to mechanical wear and tear and other environmental factors. An HDD failure may result in data loss. An HDD has a projected life expectancy before it fails. *Mean Time Between Failure (MTBF)* measures (in hours) the average life expectancy of an HDD. Today, data centers deploy thousands of HDDs in their storage infrastructures. Greater the number of HDDs in a storage array, greater the probability of a disk failure in the array.

For example, consider a storage array of 1000 HDDs, each with an MTBF of 750,000 hours. The MTBF of this collection is therefore 750,000/1000 or 750 hours. This means that a HDD in this array is likely to fail at least once in 750 hours.

In 1987, Patterson, Gibson, and Katz at the University of California, Berkeley, published a paper titled “A Case for Redundant Arrays of Inexpensive Disks (RAID).” This paper described the use of small-capacity, inexpensive disk drives as an alternative to large-capacity drives common on mainframe computers. The term RAID has been redefined to refer to independent disks, to reflect advances in the storage technology. RAID storage has now grown from an academic concept to an industry standard.

RAID is an enabling technology that leverages multiple disks as part of a set, which provides data protection against HDD failures. In general, RAID implementations also improve the I/O performance of storage systems by storing data across multiple HDDs.
A RAID array is an enclosure that contains a number of HDDs and the supporting hardware and software to implement RAID. HDDs inside a RAID array are usually contained in smaller sub-enclosures. These sub-enclosures, or physical arrays, hold a fixed number of HDDs, and may also include other supporting hardware, such as power supplies. A subset of disks within a RAID array can be grouped to form logical associations called logical arrays, also known as a RAID set or a RAID group.

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

- **Hardware** (usually a specialized disk controller card)
  - Controls all drives attached to it
  - Array(s) appear to host operating system as a regular disk drive
  - Provided with administrative software

- **Software**
  - Runs as part of the operating system
  - Performance is dependent on CPU workload
  - Does not support all RAID levels

In *hardware RAID* implementations, a specialized hardware controller is implemented either on the host or on the array. These implementations vary in the way the storage array interacts with the host.

*Controller card RAID* is host-based hardware RAID implementation in which a specialized RAID controller is installed in the host and HDDs are connected to it. The RAID Controller interacts with the hard disks using a PCI bus. Manufacturers also integrate RAID controllers on motherboards. This integration reduces the overall cost of the system, but does not provide the flexibility required for high-end storage systems.

The external RAID controller is an array-based hardware RAID. It acts as an interface between the host and disks. It presents storage volumes to the host, which manage the drives using the supported protocol. Key functions of RAID controllers are:

- Management and control of disk aggregations
- Translation of I/O requests between logical disks and physical disks
- Data regeneration in the event of disk failures

*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. Software RAID implementations offer cost and simplicity benefits when compared with hardware RAID. However, they have the following limitations:

- **Performance**: Software RAID affects overall system performance. This is due to the additional CPU cycles required to perform RAID calculations. The performance impact is more pronounced for complex implementations of RAID.

- **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 Levels

- 0 Striped array with no fault tolerance
- 1 Disk mirroring
- Nested RAID (i.e., 1 + 0, 0 + 1, etc.)
- 3 Parallel access array with dedicated parity disk
- 4 Striped array with independent disks and a dedicated parity disk
- 5 Striped array with independent disks and distributed parity
- 6 Striped array with independent disks and dual distributed parity

RAID levels are defined on the basis of striping, mirroring, and parity techniques. These techniques determine the data availability and performance characteristics of an array. Some RAID arrays use one technique, whereas others use a combination of techniques. Application performance and data availability requirements determine the RAID level selection.
A RAID set is a group of disks. Within each disk, a predefined number of contiguously addressable disk blocks are defined as *strips*. The set of aligned strips that spans across all the disks within the RAID set is called a *stripe*. Figure shows 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 HDD in the set before the next HDD is accessed, assuming that the accessed data starts at the beginning of the strip. Note that all strips in a stripe have the same number of blocks, and decreasing strip size means that data is broken into smaller pieces when spread across the disks.

Stripe size is a multiple of strip size by the number of HDDs in the RAID set. *Stripe width* refers to the number of data strips in a stripe.

Striped RAID does not protect data unless parity or mirroring is used. However, striping may significantly improve I/O performance. Depending on the type of RAID implementation, the RAID controller can be configured to access data across multiple HDDs simultaneously.

Figure shows striping in which a stripe of 192 KB is distributed over three disks with a strip size of 64 KB each. The controller writes 64 KB of data on each of the three disks, totaling 192 KB.
In a RAID 0 configuration, data is striped across the HDDs in a RAID set. It utilizes the full storage capacity by distributing strips of data over multiple HDDs in a RAID set. To read data, all the strips are put back together by the controller. The stripe size is specified at a host level for software RAID and is vendor specific for hardware RAID. When the number of drives in the array increases, performance improves because more data can be read or written simultaneously. RAID 0 is used in applications that need high I/O throughput. However, if these applications require high availability, RAID 0 does not provide data protection and availability in the event of drive failures.
**Mirroring** is a technique whereby data is stored on two different HDDs, yielding two copies of data. In the event of one HDD failure, the data is intact on the surviving HDD and the controller continues to service the host’s data requests from the surviving disk 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.

In addition to providing complete data redundancy, mirroring enables faster recovery from disk failure. However, disk mirroring provides only data protection and is not a substitute for data backup. Mirroring constantly captures changes in the data, whereas a backup captures point-in-time images of data.

Mirroring involves duplication of data — the amount of storage capacity needed is twice the amount of data being stored. Therefore, mirroring is considered expensive and is preferred for mission-critical applications that cannot afford data loss. Mirroring improves read performance because read requests can be serviced by both disks. However, write performance deteriorates, as each write request manifests as two writes on the HDDs. In other words, mirroring does not deliver the same levels of write performance as a striped RAID. Mirroring can be implemented with striped RAID by mirroring entire stripes of disks to stripes on other disks. This is known as **nested RAID**.
Most data centers require data redundancy and performance from their RAID arrays. RAID 0+1 and RAID 1+0 combine the performance benefits of RAID 0 with the redundancy benefits of RAID 1. They use striping and mirroring techniques and combine their benefits. These types of RAID require an even number of disks, the minimum being four.

RAID 0+1 is also called *mirrored stripe*. The basic element of RAID 0+1 is a stripe. This means that the process of striping data across HDDs is performed initially and then the entire stripe is mirrored.
In the event of a single drive failure, the entire stripe set is faulted. Normal processing can continue with the mirrors. However, rebuild of the failed drive will involve copying data from the mirror to the entire stripe set. This will result in increased rebuild times as compared to RAID 1+0 solution. This makes RAID 0+1 implementation less common than RAID 1+0.
RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0. Similarly, RAID 0+1 is also known as RAID 01 or RAID 0/1. RAID 1+0 performs well for workloads that use small, random, write intensive I/O.

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

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

A common misconception is that RAID 1+0 and RAID 0+1 are the same. Under normal conditions, RAID levels 1+0 and 0+1 offer identical benefits. However, rebuild operations in the case of disk failure differ between the two. RAID 1+0 is also called striped mirror. The basic element of RAID 1+0 is a mirrored pair, which means that data is first mirrored and then both copies of data are striped across multiple HDDs in a RAID set. When replacing a failed drive, only the mirror is rebuilt. In other words, the disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation. Data from the surviving disk is copied to the replacement disk.
In the event of a drive failure, normal processing can continue with the surviving mirror. Only the data on the failed drive has to be copied over from the mirror for the rebuild, as opposed to rebuilding the entire stripe set in RAID 0+1. This results in faster rebuild times for RAID 1+0 and makes it a more common solution than RAID 0+1.

Note that under normal operating conditions both RAID 0+1 and RAID 1+0 provide the same benefits. These solutions are still aimed at protecting against a single drive failure and not against multiple drive failures.
Parity is a method of protecting striped data from HDD failure without the cost of mirroring. An additional HDD is added to the stripe width to hold parity, a mathematical construct that allows re-creation of the missing data. Parity is a redundancy check that ensures full protection of data without maintaining a full set of duplicate data. Parity RAID is less expensive than mirroring because parity overhead is only a fraction of the total capacity.

Parity information can be stored on separate, dedicated HDDs or distributed across all the drives in a RAID set. Slide shows a parity RAID. The first four disks contain the data. The fifth disk stores the parity information, which in this case is the sum of the elements in each row. Think of parity as the sum of the data on the other disks in the RAID set. Each time data is updated, the parity is updated as well, so that it always reflects the current sum of the data on the other disks.

In slide, the computation of parity is represented as a simple arithmetic operation on the data. However, parity calculation is a bitwise XOR operation. Calculation of parity is a function of the RAID controller.

Note: While parity is calculated on a per stripe basis, the diagram omits this detail for the sake of simplification. Now, if one of the disk fails, the value of its data is calculated by using the parity information and the data on the surviving disks. (In this case by subtracting the sum of the rest of the elements from the parity value). If the parity disk fails, the value of its data is calculated by using the data disks. Parity will only need to be recalculated, and saved, when the failed disk is replaced with a new disk.

Compared to mirroring, parity implementation considerably reduces the cost associated with data protection. Consider a RAID configuration with five disks. Four of these disks hold data, and the fifth holds parity information. Parity requires 25 percent extra disk space compared to mirroring, which requires 100 percent extra disk space. If the number of disks is increased from 5 to 10, the parity disk requirement still remains 1. Therefore, the cost overhead is further reduced to 10 percent.

However, there are some disadvantages of using parity. Parity information is generated from data on the data disk. Therefore, parity is recalculated every time there is a change in data. This recalculation is time-consuming and affects the performance of the RAID controller.

In the event of a disk failure, each request for data from the failed disk requires recalculcation of data before sent to the host. This recalculation is time-consuming, and decreases the performance of the RAID set. Hot spare drives, introduced later, provide a way to minimize the disruption caused by a disk failure.
RAID 3 stripes data for high performance and uses parity for improved fault tolerance. Parity information is stored on a dedicated drive so that data can be reconstructed if a drive fails. For example, of five disks, four are used for data and one is used for parity. Therefore, the total disk space required is 1.25 times the size of the data disks. RAID 3 always reads and writes complete stripes of data across all disks, as the drives operate in parallel. There are no partial writes that update one out of many strips in a stripe. Figure illustrates the RAID 3 implementation.

RAID 3 provides good bandwidth for the transfer of large volumes of data. RAID 3 is used in applications that involve large sequential data access, such as video streaming.
Similar to RAID 3, RAID 4 stripes data for high performance and uses parity for improved fault tolerance. Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails.

Unlike RAID 3, data disks in RAID 4 can be accessed independently so that the specific data elements can be read or written on a single disk without read or write of an entire stripe. RAID 4 provides good read throughput and reasonable write throughput.
RAID 5 is a very versatile RAID implementation. Unlike RAID 3, in RAID 5 the drives (strips) are independently accessible. It is similar to RAID 4 because it uses striping and the drives (strips) are independently accessible. The difference between RAID 4 and RAID 5 is the parity location. In RAID 4, parity is written to a dedicated drive, creating a write bottleneck for the parity disk.

In RAID 5, parity is distributed across all disks. The distribution of parity in RAID 5 overcomes the write bottleneck. Figure illustrates the RAID 5 implementation. In RAID 5, write I/O operations suffer performance degradation because of the write penalty that manifests with a parity RAID implementation, as explained later in this chapter. The performance degradation also occurs during recovery and reconstruction operations in the event of a disk failure. In addition, multiple disk failures within the array may result in data loss.

RAID 5 is preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations in which database administrators (DBAs) optimize data access.
RAID 6 – Dual Parity RAID

- Two disk failures in a RAID set leads to data unavailability and data loss in single-parity schemes, such as RAID-3, 4, and 5

- Increasing number of drives in an array and increasing drive capacity leads to a higher probability of two disks failing in a RAID set

- RAID-6 protects against two disk failures by maintaining two parities
  - Horizontal parity which is the same as RAID-5 parity
  - Diagonal parity is calculated by taking diagonal sets of data blocks from the RAID set members

- Even-Odd, and Reed-Solomon are two commonly used algorithms for calculating parity in RAID-6

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

<table>
<thead>
<tr>
<th>RAID</th>
<th>Min Disks</th>
<th>Storage Efficiency %</th>
<th>Cost</th>
<th>Read Performance</th>
<th>Write Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>100</td>
<td>Low</td>
<td>Very good for both random and sequential read</td>
<td>Very good</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>50</td>
<td>High</td>
<td>Good Better than a single disk</td>
<td>Good Slower than a single disk, as every write must be committed to two disks</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>(n-1)*100/n where n= number of disks</td>
<td>Moderate</td>
<td>Good for random reads and very good for sequential reads</td>
<td>Poor to fair for small random reads Good for large, sequential writes</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>(n-1)*100/n where n= number of disks</td>
<td>Moderate</td>
<td>Very good for random reads Good for sequential reads</td>
<td>Fair for random write Slower due to parity overhead Fair to good for sequential writes</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>(n-2)*100/n where n= number of disks</td>
<td>Moderate but more than RAID 5</td>
<td>Very good for random reads Good for sequential reads</td>
<td>Good for small, random writes (has write penalty)</td>
</tr>
<tr>
<td>1+0 and 0+1</td>
<td>4</td>
<td>50</td>
<td>High</td>
<td>Very good</td>
<td>Good</td>
</tr>
</tbody>
</table>

© 2009 EMC Corporation. All rights reserved.
When choosing a RAID type, it is imperative to consider the impact to disk performance and application IOPS. In both mirrored and parity RAID 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 while 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. Figure illustrates a single write operation on RAID 5 that contains a group of five disks. Four of these disks are used for data and one is used for parity.

The parity (P) at the controller is calculated as follows:

\[ E_p = E_1 + E_2 + E_3 + E_4 \]  

Here, D1 to D4 is striped data across the RAID group of five disks. Whenever the controller performs a write I/O, parity must be computed by reading the old parity (Ep old) and the old data (E4 old) from the disk, which means two read I/Os. The new parity (Ep new) is computed as follows:

\[ E_{p\text{ new}} = E_{p\text{ old}} - E_{4\text{ old}} + E_{4\text{ new}} \]  

After computing the new parity, the controller completes the write I/O by writing the new data and the new parity onto the disks, amounting to two write I/Os. Therefore, the controller performs two disk reads and two disk writes for every write operation, and the write penalty in RAID 5 implementations is 4. In RAID 6, which maintains dual parity, a disk write requires three read operations: for Ep1 old, Ep2 old, and E4 old. After calculating Ep1 new and Ep2 new, the controller performs three write I/O operations for Ep1 new, Ep2 new and E4 new. 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 Penalty Exercise**

- Total IOPS at peak workload is 1200
- Read/Write ratio 2:1
- Calculate IOPS requirement at peak activity for
  - RAID 1/0
  - RAID 5

Total IOPS = 1200

Read / Write ratio 2:1

For RAID 1/0:

\[
(1200 \times \frac{2}{3}) + (1200 \times \frac{1}{3} \times 2) = 800 + 800 = 1600 \text{ IOPS}
\]

For RAID 5:

\[
(1200 \times \frac{2}{3}) + (1200 \times \frac{1}{3} \times 4) = 800 + 1600 = 2400 \text{ IOPS}
\]
A hot spare refers to a spare HDD in a RAID array that temporarily replaces a failed HDD of a RAID set. A hot spare takes the identity of the failed HDD in the array. One of the following methods of data recovery is performed depending on the RAID implementation:

- If parity RAID is used, then the data is rebuilt onto the hot spare from the parity and the data on the surviving HDDs in the RAID set.
- If mirroring is used, then the data from the surviving mirror is used to copy the data.
- When the failed HDD is replaced with a new HDD, one of the following takes place:
  - The hot spare replaces the new HDD permanently. This means that it is no longer a hot spare, and a new hot spare must be configured on the array.
  - When a new HDD is added to the system, data from the hot spare is copied to it. The hot spare returns to its idle state, ready to replace the next failed drive.

A hot spare should be large enough to accommodate data from a failed drive. Some systems implement multiple hot spares to improve data availability. A hot spare can be configured as automatic or user initiated, which specifies how it will be used in the event of disk failure. In an automatic configuration, when the recoverable error rates for a disk exceed a predetermined threshold, the disk subsystem tries to copy data from the failing disk to the hot spare automatically. If this task is completed before the damaged disk fails, then the subsystem switches to the hot spare and marks the failing disk as unusable. Otherwise, it uses parity or the mirrored disk to recover the data. In the case of a user-initiated configuration, the administrator has control of the rebuild process. For example, the rebuild could occur overnight to prevent any degradation of system performance. However, the system is vulnerable to another failure if a hot spare is unavailable.
Module Summary

Key points covered in this module:

• What RAID is and the needs it addresses
• The concepts upon which RAID is built
• Some commonly implemented RAID levels

These are the key points covered in this module. Please take a moment to review them.
Exercise 1

Business Profile:

Acme Telecom is involved in mobile wireless services across the United States and has about 5000 employees worldwide. This company is Chicago based and has 7 regional offices across the country. Although Acme is doing well financially, they continue to feel competitive pressure. As a result, the company needs to ensure that the IT infrastructure takes advantage of fault tolerant features.

Current Situation/Issues:

• The company uses a number of different applications for communication, accounting, and management. All the applications are hosted on individual servers with disks configured as RAID 0.

• All financial activity is managed and tracked by a single accounting application. It is very important for the accounting data to be highly available.

• The application performs around 15% write operations, and the remaining 85% are reads.

• The accounting data is currently stored on a 5-disk RAID 0 set. Each disk has an advertised formatted capacity of 200 GB, and the total size of their files is 730 GB.

• The company performs nightly backups and removes old information—so the amount of data is unlikely to change much over the next 6 months.

The company is approaching the end of the financial year and the IT budget is depleted. Buying even one new disk drive will not be possible.

How would you suggest that the company restructure their environment? You will need to justify your choice based on cost, performance, and availability of the new solution.

RAID level to use:

Advantages:

Disadvantages:
Exercise 2

Business Profile:
Acme Telecom is involved in mobile wireless services across the United States and has about 5000 employees worldwide. This company is Chicago based and has 7 regional offices across the country. Although Acme is doing well financially, they continue to feel competitive pressure. As a result, the company needs to ensure that the IT infrastructure takes advantage of fault tolerant features.

Current Situation/Issues:
• The company uses a number of different applications for communication, accounting, and management. All the applications were hosted on individual servers with disks configured as RAID 0.
• The company changed the RAID level of their accounting application based on your recommendations 6 months ago.
• It is now the beginning of a new financial year and the IT department has an increased budget. You are called in to recommend changes to their database environment.
• You investigate their database environment closely, and observe that the data is stored on a 6-disk RAID 0 set. Each disk has an advertised formatted capacity of 200 GB and the total size of their files is 900 GB. The amount of data is likely to change by 30% over the next 6 months and your solution must accommodate this growth.
• The application performs around 40% write operations, and the remaining 60% are reads. The average size of a read or write is small, at around 2 KB.

How would you suggest that they restructure their environment? A new 200 GB disk drive costs $1000. The controller can handle all commonly used RAID levels, so will not need to be replaced. What is the cost of the new solution?

Justify your choice based on cost, performance, and data availability of the new solution.

RAID level to use:
Advantages:
Disadvantages:
Check Your Knowledge

• What is a RAID array?
• What benefits do RAID arrays provide?
• What methods can be used to provide higher data availability in a RAID array?
• What is the primary difference between RAID 3 and RAID 5?
• What is advantage of using RAID 6?
• What is a hot spare?
Module 1.4 – Intelligent Storage Systems
Module objective

After completing this module, you will be able to:

- Describe components of intelligent storage system
- List benefits of intelligent storage system
- Explain intelligent cache algorithms and protection
- Describe intelligent storage array implementation
  - High-end storage array
  - Mid-range storage array

The objectives for this module are shown here. Please take a moment to read them.
What is an Intelligent Storage System

Intelligent Storage Systems are RAID arrays that are:

- Highly optimized for I/O processing
- Have large amounts of cache for improving I/O performance
- Have operating environments that provide:
  - Intelligence for managing cache
  - Array resource allocation
  - Connectivity for heterogeneous hosts
  - Advanced array based local and remote replication options

Business-critical applications require high levels of performance, availability, security, and scalability. A hard disk drive is a core element of storage that governs the performance of any storage system. Some of the older disk array technologies could not overcome performance constraints due to the limitations of a hard disk and its mechanical components. RAID technology made an important contribution to enhancing storage performance and reliability, but hard disk drives even with a RAID implementation could not meet performance requirements of today’s applications. With advancements in technology, a new breed of storage solutions known as an intelligent storage system has evolved. The intelligent storage systems detailed in this chapter are the feature-rich RAID arrays that provide highly optimized I/O processing capabilities. These arrays have an operating environment that controls the management, allocation, and utilization of storage resources. These storage systems are configured with large amounts of memory called cache and use sophisticated algorithms to meet the I/O requirements of performance sensitive applications.
Benefits of an Intelligent Storage System

Intelligent storage system provides several benefits over a collection of disks in an array (JBOD) or even a RAID arrays:

– Increased capacity
– Improved performance
– Easier data management
– Improved data availability and protection
– Enhanced Business Continuity support
– Improved security and access control

Intelligent storage systems, a collection of disks in an array, and RAID arrays, all provide increased data storage capacity. However, intelligent storage systems provide more benefits, as listed in the slide.
An intelligent storage system consists of four key components: front end, cache, back end, and physical disks. Figure illustrates these components and their interconnections. 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.
The front end provides the interface between the storage system and the host. It consists of two components: front-end ports and front-end controllers. The front-end ports enable hosts to connect to the intelligent storage system. Each front-end port has processing logic that executes the appropriate transport protocol, such as SCSI, Fibre Channel, or iSCSI, for storage connections. Redundant ports are provided on the front end for high availability. Front-end controllers route data to and from cache via the internal data bus. When cache receives write data, the controller sends an acknowledgment message back to the host. Controllers optimize I/O processing by using command queuing algorithms.
Command queuing is a technique implemented on front-end controllers. It determines the execution order of received commands and can reduce unnecessary drive head movements and improve disk performance. When a command is received for execution, the command queuing algorithms assigns a tag that defines a sequence in which commands should be executed. With command queuing, multiple commands can be executed concurrently based on the organization of data on the disk, regardless of the order in which the commands were received. The most commonly used command queuing algorithms are as follows:

• **First In First Out (FIFO):** This is a default algorithm where commands are executed in the order in which they are received. There is no reordering of requests for optimization; therefore, it is inefficient in terms of performance.

• **Seek Time Optimization:** Commands are executed based on optimizing read/write head movements, which may result in reordering of commands. Without seek time optimization, the commands are executed in the order they are received. For example, as shown in Figure, the commands are executed in the order A, B, C and D. The radial movement required by the head to execute C immediately after A is less than what would be required to execute B. With seek time optimization, the command execution sequence would be A, C, B and D, as shown in Figure.

• **Access Time Optimization:** Commands are executed based on the combination of seek time optimization and an analysis of rotational latency for optimal performance.

Command queuing is also implemented on disk controllers and this may further supplement the command queuing implemented on the front-end controllers. Some models of SCSI and Fibre Channel drives have command queuing implemented on their controllers.
Cache is an important component that enhances the I/O performance in an intelligent storage system. Cache is semiconductor memory where data is placed temporarily to reduce the time required to service I/O requests from the host. Cache improves storage system performance by isolating hosts from the mechanical delays associated with physical disks, which are the slowest components of an intelligent storage system. Accessing data from a physical disk usually takes a few milliseconds because of seek times and rotational latency. If a disk has to be accessed by the host for every I/O operation, requests are queued, which results in a delayed response. Accessing data from cache takes less than a millisecond. Write data is placed in cache and then written to disk. After the data is securely placed in cache, the host is acknowledged immediately.
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-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.

- **Write-back cache**: Data is placed in cache and an acknowledgment is sent to the host immediately. Later, data from several writes are committed (de-staged) to the disk. Write response times are much faster, as the write operations are isolated from the mechanical delays of the disk. However, uncommitted data is at risk of loss in the event of cache failures.

Cache can be bypassed under certain conditions, such as very 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 area. This is particularly useful in an environment where cache resources are constrained and must be made available for small random I/Os.
When a host issues a read request, the front-end controller determines 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. This provides a fast response time to the host (about a millisecond). If the requested data is not found in cache, it is called a cache miss and the data must be read from the disk. The back-end controller accesses the appropriate disk and retrieves the requested data. Data is then placed in cache and is finally sent to the host through the front-end controller. Cache misses increase I/O response time. A pre-fetch, or read-ahead, algorithm is used when read requests are sequential. In a sequential read request, a contiguous set of associated blocks is retrieved. Several other blocks that have not yet been requested by the host can be read from the disk and placed into cache in advance. When the host subsequently requests these blocks, the read operations will be read hits. This process significantly improves the response time experienced by the host. The intelligent storage system offers fixed and variable pre-fetch sizes. In fixed pre-fetch, the intelligent storage system pre-fetches a fixed amount of data. It is most suitable when I/O sizes are uniform. In variable pre-fetch, the storage system pre-fetches an amount of data in multiples of the size of the host request. Maximum pre-fetch limits the number of data blocks that can be pre-fetched to prevent the disks from being rendered busy with pre-fetch at the expense of other I/O. Read performance is measured in terms of the read hit ratio, or the hit rate, usually expressed as a percentage. This ratio is the number of read hits with respect to the total number of read requests. A higher read hit ratio improves the read performance.
Cache Management: Algorithms

- **Least Recently Used (LRU)**
  - Discards least recently used data

- **Most Recently Used (MRU)**
  - Discards most recently used data

Cache is a finite and expensive resource that needs proper management. Even though intelligent storage systems can be configured with large amounts 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:

Least Recently Used (LRU): A 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. However, if a page contains write data that has not yet been committed to disk, data will first be written to disk before the page is reused.

Most Recently Used (MRU): An algorithm that is the converse of LRU. In MRU, the pages that have been accessed most recently are freed up or marked for reuse. This algorithm is based on the assumption that recently accessed data may not be required for a while.
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) in order to manage its availability. Flushing is the process of committing data from cache to the disk. On the basis of the I/O access rate and pattern, high and low levels called 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 the high-speed or forced flushing and returns to idle flush behavior. The cache utilization level, as shown in Figure, 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 to flushing. This type of flushing has minimal impact on host I/O processing.

- **Forced flushing**: Occurs in the event of a large I/O burst when cache reaches 100 percent of its capacity, which significantly affects the I/O response time. In forced flushing, dirty pages are forcibly flushed to disk.

Some of the things that improve performance include:

- Manage peak I/O requests by absorbing large groups of writes—called bursts—without becoming bottlenecked by the speed of a physical disk. This is known as burst smoothing.

- Merging several writes to the same area into a single operation

The algorithms that manage cache should adapt to changing data access patterns. The actual algorithms used are vendor-specific.
Cache Data Protection

• Protecting cache data against failure:
  – Cache mirroring
    ➢ Each write to the cache is held in two different memory locations on two independent memory cards
  – Cache vaulting
    ➢ Cache is exposed to the risk of uncommitted data loss due to power failure
    ➢ In the event of power failure, uncommitted data is dumped to a dedicated set of drives called vault drives

Cache is volatile memory, so a power failure or any kind of cache failure will cause the loss of data not yet committed to the disk. This risk of losing uncommitted data held in cache can be mitigated using cache mirroring and cache vaulting:

• **Cache mirroring:** Each write to cache is held in two different memory locations on two independent memory cards. In the event of a cache failure, the write data will still be safe in the mirrored location and can be committed to the disk. 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. As only writes are mirrored, this method results in better utilization of the available cache. 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:** Cache is exposed to the risk of uncommitted data loss due to power failure. This problem can be addressed in various ways: powering the memory with a battery until AC power is restored or using battery power to write the cache content to the disk. In the event of extended power failure, using batteries is not a viable option because in intelligent storage systems, large amounts of data may need to be committed to numerous disks and batteries may not provide power for sufficient time to write each piece of data to its intended disk. Therefore, 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.
The back end provides an interface between cache and the physical disks. It consists of two components: back-end ports and back-end controllers. The back end controls data transfers between cache and the physical disks. From cache, data is sent to the back end and then routed to the destination disk. Physical disks are connected to ports on the back end. The back end controller communicates with the disks when performing reads and writes and also provides additional, but limited, temporary data storage. The algorithms implemented on back-end controllers provide error detection and correction, along with RAID functionality. For high data protection and availability, storage systems are configured with dual controllers with multiple ports. Such configurations provide an alternate path to physical disks in the event of a controller or port failure. This reliability is further enhanced if the disks are also dual-ported. In that case, each disk port can connect to a separate controller. Multiple controllers also facilitate load balancing.
A physical disk stores data persistently. Disks are connected to the back-end with either SCSI or a Fibre Channel interface. An intelligent storage system enables the use of a mixture of SCSI or Fibre Channel drives and IDE/ATA drives.
Physical drives or groups of RAID protected drives can be logically split into volumes known as logical volumes, commonly referred to as *Logical Unit Numbers* (LUNs). The use of LUNs improves disk utilization. For example, without the use of LUNs, a host requiring only 200 GB could be allocated an entire 1TB physical disk. Using LUNs, only the required 200 GB would be allocated to the host, allowing the remaining 800 GB to be allocated to other hosts.

In the case of RAID protected drives, these logical units are slices of RAID sets and are spread across all the physical disks belonging to that set. The logical units can also be seen as a logical partition of a RAID set that is presented to a host as a physical disk. For example, Figure shows a RAID set consisting of five disks that have been sliced, or partitioned, into several LUNs. LUNs 0 and 1 are shown in the figure.

Note how a portion of each LUN resides on each physical disk in the RAID set. LUNs 0 and 1 are presented to hosts 1 and 2, respectively, as physical volumes for storing and retrieving data. Usable capacity of the physical volumes is determined by the RAID type of the RAID set. A host will see a LUN as if it were a single disk device. The host is not aware that this LUN is only a part of a larger physical drive. The host assigns logical device names to the LUNs; the naming conventions vary by platform/OS.

The capacity of a LUN can be expanded by aggregating other LUNs with it. The result of this aggregation is a larger capacity LUN, known as a *meta-LUN*. The mapping of LUNs to their physical location on the drives is managed by the operating environment of an intelligent storage system.
LUN Masking

- LUN masking is access control mechanism
- Process of masking LUNs from unauthorized access
- Implemented on storage arrays

*LUN masking* is a process that provides data access control by defining which LUNs a host can access. LUN masking function is typically implemented at the front end controller. This ensures that volume access by servers is controlled appropriately, preventing unauthorized or accidental use in a distributed environment. For example, consider a storage array with two LUNs that store data of the sales and finance departments. Without LUN masking, both departments can easily see and modify each other’s data, posing a high risk to data integrity and security. With LUN masking, LUNs are accessible only to the designated hosts.
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. To address the enterprise storage needs, these arrays provide the following capabilities:

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

In addition to these features, high-end arrays possess some unique features and functional that are required for mission-critical applications in large enterprises.
Midrange Storage Systems

- Also referred as Active-passive arrays
  - Host can perform I/Os to LUNs only through active paths
  - Other paths remain passive till active path fails
- Midrange array have two controllers, each with cache, RAID controllers and disks drive interfaces
- Designed for small and medium enterprises
- Less scalable as compared to high-end array

Midrange storage systems are also referred to as *active-passive arrays* and they are best suited for small- and medium-sized enterprises. In an active-passive array, a host can perform I/Os to a LUN only through the paths to the owning controller of that LUN. These paths are called *active paths*. The other paths are passive with respect to this LUN. As shown in Figure, 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 enterprises; therefore, they host less storage capacity and global cache than active-active arrays. There are also fewer front-end ports for connection to servers. However, they ensure high redundancy and high performance for applications with predictable workloads. They also support array-based local and remote replication.
Module Summary

Key points covered in this module:

- Intelligent Storage Systems features
- Components of Intelligent Storage Systems
- Cache management algorithms
- Intelligent Storage System implementation
  - High-end storage array
  - Mid range storage array

This concludes the module. Key points covered in this module are shown here. Please take a moment to review them.
Concept in Practice: EMC CLARiiON CX-4

- Support for UltraFlex technology
- Scalable from up to 960 disks
- Supports flash drives
- Supports different types and sizes of drives, and RAID types (0, 1, 1+0, 3, 5, 6)
- Supports up to 16 GB of available cache memory per controller (Storage Processor)
- Enhances availability with non-disruptive upgrade and failover
- Ensures data protection through mirrored write cache and cache vaulting
- Supports storage-based local and remote data replication
  - Through SnapView and MirrorView software

It is the EMC midrange networked storage offering that delivers enterprise-quality features and functionality. It is ideally suited for applications with predictable workloads that need moderate to high response time and throughput. CLARiiON is built with modular building blocks and no single point of failure. The CX4 series is the fourth generation CLARiiON CX storage platform. The features of CLARiiON CX-4 960 storage array are as follows:

- UltraFlex technology for dual protocol connectivity, online expansion via IO modules, and readiness for future technologies—such as 8 Gb/s Fibre Channel and 10 Gb/s iSCSI.
- Scalable from up to 960 disks
- Supports flash drives with 30 times more IOPS capability.
- Supports different types and sizes of drives, and RAID types (0, 1, 1+0, 3, 5, 6)
- Supports up to 16 GB of available cache memory per controller (Storage Processor)
- Enhances availability with non-disruptive upgrade and failover
- Ensures data protection through mirrored write cache and cache vaulting
- Provides data integrity through disk scrubbing. The background verification process runs continually and reads all sectors of all the disks. If a block is unreadable, the back-end error handling recovers the bad sectors from parity or mirror data.
- Supports storage-based local and remote data replication for backup and disaster recovery through SnapView and MirrorView software.
The Storage Processor Enclosure (SPE) and the Disk Array Enclosure (DAE) are the key modular building blocks of a CLARiiON. A DAE enclosure contains up to 15 disk drives, two link control cards (LCCs), two power supplies, and two fan modules. An SPE contains two storage processors, each consisting of one CPU module and slots for I/O modules. The CLARiiON architecture supports fully redundant, hot swappable components. This means that the system can survive with a failed component, which can be replaced without powering down the system. The important components of the CLARiiON storage system include the following:

- **Intelligent storage processor (SP):** Intelligent SP is the main component of the CLARiiON architecture. SP are configured in pairs for maximum availability. SP provide both front-end and back-end connectivity to the host and the physical disk, respectively. SP also include memory, most of which is used for cache. Depending on the model, each SP includes one or two CPUs.

- **CLARiiON Messaging Interface (CMI):** The SPs communicate to each other over the CLARiiON Messaging Interface, which transports commands, status information, and data for write cache mirroring between the SPs. CLARiiON uses PCI-Express as the high-speed CMI. The PCI Express architecture delivers high bandwidth per pin, has superior routing characteristics, and provides improved reliability.

- **Standby Power Supply (SPS):** In the event of a power failure, the SPS maintains a power supply to the cache for long enough to allow the content to be copied to the vault.

- **Link Control Card (LCC):** The LCC provides services to the drive enclosure, which includes the capability to control enclosure functionalities and monitor environmental status. Each drive enclosure has two LCCs. The other functions performed by LCCs are loop configuration control, failover control, marker LED control, individual disk port control, drive presence detection, and voltage status information.

- **FLARE Storage Operating Environment:** FLARE is a special software designed for EMC CLARiiON. Each storage system ships with a complete copy of the FLARE operating system installed on its first four disks. When CLARiiON is powered up, each SP boots and runs the FLARE operating system. FLARE performs resource allocation and other management tasks in the array.
Concept in Practice – CLARiiON Video

CLARiiON Video
The EMC Symmetrix establishes the highest standards for performance and capacity for an enterprise information storage solution and is recognized as the industry’s most trusted storage platform. EMC Symmetrix uses the Direct Matrix Architecture and incorporates a fault tolerant design. Some of the important features of Symmetrix DMX-4 storage array are as follows:

- Incrementally scalable to 2,400 disks
- Supports Flash-based solid-state drives
- Dynamic global cache memory (16 GB–512 GB)
- Advanced processing power (up to 130 PowerPC)
- Direct matrix Architecture
- High data processing bandwidth (up to 128 GB/s)
- Data protection with RAID 1, 1+0 (also known as 10 for mainframe), 5, and 6
- Storage-based local and remote replication
  - Through TimeFinder and SRDF software
Cooling memory, and the back end. Key components of Symmetrix DMX are as follows:

- **Front end**: The host connects to Symmetrix via a front-end port on the channel director. Multiple directors are configured, each with multiple ports to provide host connectivity and redundancy. Each Symmetrix channel director supports eight internal links to global memory. Data transfers between host and global memory can execute concurrently across multiple ports on a director.
- **Back end**: Back end disk directors manage the interface to the disk drives and are responsible for data movement between the disk drives and global memory. Each disk director on a Symmetrix system supports 8 internal links to global memory.
- **Global Memory**: The Symmetrix global memory is its most important component. All read and write operations are performed through global memory. Host I/Os are received at the front end and processed through global memory at much greater electronic speeds than transfers involving disks. The global memory directors work in pairs. The hardware writes to the one global memory director first and then writes are mirrored to the secondary global memory director, for data protection. Each global memory director has 16 ports with full-duplex serial connections between the global memory director and the channel or disk directors (a total of 16 directors) through the direct matrix. Each of the 8 director ports on the 16 directors connects to one of the 16 memory ports on each of the 8 global memory directors. These 128 individual point-to-point connections facilitate up to 128 concurrent cache operations in the system, providing ultra-high bandwidth for I/O processing.
- **XCM**: The a communication agent between the service processor and all the processing nodes (channel, disk, and memory director) within the system. External connections to the service processor provide dial-home capability for remote monitoring and diagnostics.
- **Symmetrix Enginuity**: This is the operating environment for EMC Symmetrix. Enginuity manages and ensures the optimal flow and integrity of information through the various hardware components of the Symmetrix system.
Concept in Practice – Symmetrix Video

Symmetrix Video
The Symmetrix VMAX system with the Enginuity operating environment is a new enterprise class storage array that is built on the strategy of simple, intelligent modular storage. The Symmetrix VMAX series includes the Symmetrix VMAX and the Symmetrix VMAX SE models, and meets a wide range of high-end requirements for scalability, performance, and cost.

The Symmetrix VMAX is ideal for high-end configurations that require performance and the scaling capability to start as small as one engine pair and 96 drives, and grow to a maximum of eight engines and 2,400 drives. Symmetrix VMAX’s incremental scalability allows to meet the growth requirements by adding VMAX engines and drives nondisruptively to the existing frame.

The Symmetrix VMAX SE is an ideal entry point for high-end configurations requiring one engine and between 48 and 360 drives. The same functionality, storage interoperability, and operational efficiency is maintained across the entire Symmetrix VMAX series. And, the Symmetrix VMAX SE supports both open systems and mainframe connectivity. Symmetrix VMAX supports single-phase power, extended drive loop configurations and 8 Gb/s connectivity.
The Virtual Matrix Architecture replaces individual, function-specific directors with Symmetrix VMAX Engines, each containing a portion of Global Memory and two directors capable of managing Front End, Back End, and remote connections simultaneously. The slide reviews the Virtual Matrix architecture in more detail. The system starts with a VMAX Engine as a high-availability base configuration. The VMAX Engine is connected to the Virtual Matrix and allows all system resources — including CPU, memory, drives, and host ports — to be dynamically accessed and shared by any application. Additional VMAX engines are added nondisruptively to efficiently scale system resources. The Virtual Matrix supports a total of eight VMAX engines in a single system. As shown, the new architecture enables significant increases in scalability relative to the DMX platform. Scalability has improved in all aspects: Front End connectivity, Global Memory, Back End connectivity, and usable capacity. The Virtual Matrix is redundant and dual active and supports all Global Memory references, all messaging, and all management operations including internal discovery and initialization, path management, load balancing, fail over, and fault isolation within the array. The Symmetrix VMAX array is comprised of 1 to 8 VMAX Engines.
✓ Check Your Knowledge

• What are the parts of an Intelligent Storage System?
• What are the differences between a high-end and midrange storage array?
• What is the difference between a read cache hit and a read cache miss?
• What is the difference between Least Recently Used and Most Recently Used algorithms?
• What is the difference between Write-through and Write-back cache?
Information Storage and Management
Section 1 – Quiz

1. What are the two [2] primary reasons why businesses collect and store data over long periods of time?
   a. Availability of high capacity storage systems
   b. Availability of high performance servers
   c. Potential for new business initiatives
   d. Regulations and compliance
   e. Availability of low cost storage system

2. What are two [2] advantages of centralized networked storage?
   a. Consolidated data storage and retrieval
   b. Islands of data
   c. Improved storage performance
   d. Multiple dedicated storage arrays
   e. Sharing storage resources and data between different departments in an organization

3. Which storage systems key requirement refers to the ability of the storage solution to grow with the business?
   a. Availability
   b. Capacity
   c. Data Integrity
   d. Scalability

4. What is Read Hit Ratio?
   a. Number of misses with respect to the total read request
   b. Number of hits with respect to the total write request
   c. Number of hits with respect to the total read request
   d. Number of misses with respect to the total write request

5. Which statement best describes little’s law?
   a. Arrival rate is directly proportional to response time
   b. Arrival rate is inversely proportional to total number of request in the system
   c. Response time is inversely proportional to total number of request in the system
   d. Arrival rate is inversely proportional to response time

6. What two [2] factors or conditions determine the storage capacity of a Head Disk Assembly (HDA)?
   a. Amount of data that can be stored on each platter
   b. Cache size of the platters
   c. Connectivity of platters to the HDA
   d. Number of platters
   e. Size of HDA

7. What application I/O characteristic is most suitable for use of RAID 3?
   a. Small random reads
   b. Large sequential reads
   c. Large random writes
   d. Small sequential writes
8. What is the primary benefit of data striping?
   a. Independent disks allow multiple reads/writes simultaneously
   b. Improves security by creating multiple identical copies of data
   c. More efficient allocation of sequential reading/writing
   d. More efficient allocation of random reading/writing

9. Which term is used to describe an idle component in a RAID array that becomes a temporary replacement for a failed component?
   a. Hot disk
   b. Hot spare
   c. Hot swap
   d. RAID controller

10. What is the role of the front-end in an intelligent storage system?
    a. Improves system performance by isolating disks from the mechanical delays associated with physical disks
    b. Provides temporary storage of data before committing to disk
    c. Provides connectivity from the host to the storage array
    d. Provides communication with the disks for read/write operations

11. Which component of an intelligent storage system improves performance of reads and writes?
    a. Back-end
    b. Cache
    c. Host
    d. Physical Disks

12. Which activity represents ILM strategy?
    a. Automate backup and recovery process
    b. Implement storage tiering
    c. Migrate from DAS to SAN
    d. Implement end-to-end storage management

13. What is the purpose of cache vaulting?
    a. Protect uncommitted data due to cache failure
    b. Backup data at offsite location
    c. Protect uncommitted data due to power failure
    d. Provides better cache management

14. Which key parameter decides the number of disks in a RAID group?
    a. Capacity required
    b. RAID type
    c. Number of IOPS
    d. Stripe depth

15. A host generates 1000 IOPS of which 60% are writes. In a RAID 5 configuration, what is the number of IOPS on the disk?
    a. 600 IOPS
    b. 1600 IOPS
    c. 2200 IOPS
    d. 2800 IOPS
Section 2
Section Objective

Upon completion of this section, you will be able to:

• Describe the elements, connectivity, and management of:
  – Direct Attached Storage (DAS)
  – Network Attached Storage (NAS)
  – FC and IP Storage Area Networks (SAN)
  – Content Addressed Storage (CAS) and

• Discuss the benefits and challenges of each of the storage models

• Describe various virtualization technologies

In the previous section, we looked at the components of a storage system. In this section, the emphasis will be on different storage system models-- starting with a basic storage model (DAS) and focusing on networked storage models such as SAN, NAS, and CAS.

When effectively designed and implemented, Networked Storage enables more effective utilization of available storage, increased flexibility, and better availability of data.
Module 2.1 – Direct Attached Storage and Introduction to SCSI
Module Objective

Upon completion of this module, you will be able to:

• Discuss the benefits and challenges of DAS
• Discuss DAS management options
• Discuss evolution of SCSI
• Describe SCSI – 3 architecture
• Discuss SCSI addressing and communication model

The objectives for this module are shown here. Please take a moment to read them.
Lesson: Direct Attached Storage

Upon completion of this lesson, you will be able to:

- Discuss the benefits of DAS
- Describe the elements of DAS
- Discuss DAS management considerations
- Discuss DAS challenges

The objectives for this lesson are shown here. Please take a moment to read them.
Direct-Attached Storage (DAS) is an architecture where storage connects directly to servers. Applications access data from DAS using block-level access protocols. The internal HDD of a host, tape libraries, and directly connected external HDD packs are some examples of DAS. DAS is classified as internal or external, based on the location of the storage device with respect to the host.

**Internal DAS:**

In internal DAS architectures, the storage device is internally connected to the host by a serial or parallel bus. The physical bus has distance limitations and can only be sustained over a shorter distance for high-speed 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.

**External DAS:**

In external DAS architectures, the server connects directly to the external storage device. In most cases, communication between the host and the storage device takes place over 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.
DAS Benefits

- Ideal for local data provisioning
- Quick deployment for small environments
- Simple to deploy
- Reliability
- Low capital expense
- Low complexity

DAS requires a relatively lower initial investment than storage networking. Storage networking architectures are discussed later in this book. DAS configuration is simple and can be deployed easily and rapidly. Setup is managed using host-based tools, such as the host OS, which makes storage management tasks easy for small and medium enterprises. DAS is the simplest solution when compared to other storage networking models and requires fewer management tasks, and less hardware and software elements to set up and operate.
**DAS Connectivity Options**

- **ATA (IDE) and SATA**
  - Primarily for internal bus

- **SCSI**
  - Parallel (primarily for internal bus)
  - Serial (external bus)

- **FC**
  - High speed network technology

- **Buss and Tag**
  - Primarily for external mainframe
  - Precursor to ESCON and FICON

---

The host and the storage device in DAS communicate with each other by using predefined protocols such as IDE/ATA, SATA, SAS, SCSI, and FC. These protocols are implemented on the HDD controller. Therefore, a storage device is also known by the name of the protocol it supports.

Note: FC is detailed in next module.
DAS Management

• Internal
  – Host provides:
    ➢ Disk partitioning (Volume management)
    ➢ File system layout
  – Direct Attached Storage managed individually through the server and the OS

• External
  – Array based management
  – Lower TCO for managing data and storage Infrastructure

Internal DAS is generally managed through the host and OS, or by some third party software. Device management through host provides many features including:
  • Disk/volume partitioning/management
  • File system specific layouts for the OS

A key feature of external DAS management is that the host OS is not directly responsible for any fundamental management of the resources (e.g. LUN creation, filesystem layout, and data addressing). Multi-vendor internal DAS storage must be managed individually and by placing the majority of storage externally on single vendor arrays, management becomes more centralized and skill sets required for multi-vendor management can be reduced.
DAS Challenges

- Scalability is limited
  - Number of connectivity ports to hosts
  - Number of addressable disks
  - Distance limitations

- Downtime required for maintenance with internal DAS

- Limited ability to share resources
  - Array front-end port, storage space
  - Resulting in islands of over and under utilized storage pools

DAS does not scale well. A storage device has a limited number of ports, which restricts the number of hosts that can directly connect to the storage. A limited bandwidth in DAS restricts the available I/O processing capability. When capacities are being reached, the service availability may be compromised, and this has a ripple effect on the performance of all hosts attached to that specific device or array.

DAS does not make optimal use of resources due to its limited ability to share front end ports. In DAS environments, unused resources cannot be easily re-allocated, resulting in islands of over-utilized and under-utilized storage pools. Disk utilization, throughput, and cache memory of a storage device, along with virtual memory of a host govern the performance of DAS. RAID-level configurations, storage controller protocols, and the efficiency of the bus are additional factors that affect the performance of DAS. The absence of storage interconnects and network latency provide DAS with the potential to outperform other storage networking configurations.
Lesson Summary

Key points covered in this lesson:

- Internal and External DAS
- DAS Benefit
- DAS Management Options
- DAS Limitations

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Lesson: Introduction to SCSI

Upon completion of this module, you will be able to:

• Describe SCSI-3 architecture
• Discuss SCSI device models with different port configurations
• Describe SCSI Addressing

Prior to the development of SCSI, the interfaces used to communicate with devices varied with each device. For example, an HDD interface could only be used with a hard disk drive. SCSI was developed to provide a device-independent mechanism for attaching to and accessing host computers. SCSI also provided an efficient peer-to-peer I/O bus that supported multiple devices. Today, SCSI is commonly used as a hard disk interface. However, SCSI can be used to add devices, such as tape drives and optical media drives, to the host computer without modifying the system hardware or software.
Evolution of Parallel SCSI

- Developed by Shugart Associates & named as SASI
- ANSI acknowledged SCSI as an industry standard
- SCSI versions
  - SCSI-1
    - Defined cable length, signaling characteristics, commands, & transfer modes
    - Used 8-bit narrow bus with maximum data transfer rate of 5 MB/s
  - SCSI-2
    - Defined Common command Set (CCS)
    - Improved performance, reliability, and added additional features
  - SCSI-3
    - Latest version of SCSI,
    - Comprised different but related standards, rather than one large document.

Shugart Associates and NCR developed a system interface in 1981 and named it Shugart Associates System Interface (SASI). SASI was developed to build a proprietary, high-performance standard primarily for use by these two companies. However, to increase the acceptance of SASI in the industry, the standard was updated to a more robust interface and renamed SCSI. In 1986, the American National Standards Institution (ANSI) acknowledged the new SCSI as an industry standard. SCSI, first developed for hard disks, is often compared to IDE/ATA. SCSI offers improved performance and expandability and compatibility options, making it suitable for high-end computers. However, the high cost associated with SCSI limits its popularity among home or business desktop users.

Over the years, SCSI has undergone radical changes and has evolved into a robust industry standard. Various SCSI standards are detailed here:

SCSI-1, renamed to distinguish it from other SCSI versions, is the original standard that the ANSI approved. SCSI-1 defined the basics of the first SCSI bus, including cable length, signaling characteristics, commands, and transfer modes. SCSI-1 devices supported only single-ended transmission and passive termination. SCSI-1 used a narrow 8-bit bus, which offered a maximum data transfer rate of 5 MB/s.

To control the various problems caused by the nonstandard implementation of the original SCSI, a working paper was created to define a set of standard commands for a SCSI device. This set of standards, called the common command set (CCS), formed the basis of the SCSI-2 standard. SCSI-2 was focused on improving performance, enhancing reliability, and adding additional features to the SCSI-1 interface, in addition to standardizing and formalizing the SCSI commands. In 1993, work began on developing the next version of the SCSI standard, SCSI-3. Unlike SCSI-2, the SCSI-3 standard document is comprised different but related standards, rather than one large document.
The SCSI-3 architecture defines and categorizes various SCSI-3 standards and requirements for SCSI-3 implementations. (For more information, see Technical Committee T10 “SCSI Architecture Model-3 (SAM-3)” document from www.t10.org.) The SCSI-3 architecture was approved and published as the standard X.3.270-1996 by the ANSI. This architecture helps developers, hardware designers, and users to understand and effectively utilize SCSI. The three major components of a SCSI architectural model are as follows:

**SCSI-3 command protocol:**
This consists of primary commands that are common to all devices as well as device-specific commands that are unique to a given class of devices.

**Transport layer protocols:**
These are a standard set of rules by which devices communicate and share information.

**Physical layer interconnects:**
These are interface details such as electrical signaling methods and data transfer modes. Common access methods are the ANSI software interfaces for SCSI devices. Figure shows the SCSI-3 standards architecture with interrelated groups of other standards within SCSI-3.
SCSI Device Model

SCSI communication involves:
- **SCSI initiator device**
  - Issues commands to SCSI target devices
  - Example: SCSI host adaptor
- **SCSI target device**
  - Executes commands issued by initiators
  - Examples: SCSI peripheral devices
- **Device requests contain Command Descriptor Block (CDB)**

SCSI-3 architecture derives its base from the client-server relationship, in which a client directs a service request to a server, which then fulfills the client’s request. In a SCSI environment, an initiator-target concept represents the client-server model. In a SCSI-3 client-server model, a particular SCSI device acts as a SCSI target device, a SCSI initiator device, or a SCSI target/initiator device. Each device performs the following functions:

**SCSI initiator device:** Issues a command to the SCSI target device, to perform a task. A SCSI host adaptor is an example of an initiator.

**SCSI target device:** Executes commands to perform the task received from a SCSI initiator. Typically a SCSI peripheral device acts as a target device. However, in certain implementations, the host adaptor can also be a target device.

Figure displays the SCSI-3 client-server model, in which a SCSI initiator, or a client, sends a request to a SCSI target, or a server. The target performs the tasks requested and sends the output to the initiator, using the protocol service interface. A SCSI target device contains one or more logical units. A logical unit is an object that implements one of the device functional models as described in the SCSI command standards. The logical unit processes the commands sent by a SCSI initiator. A logical unit has two components, a *device server* and a *task manager*. The device server addresses client requests, and the task manager performs management functions. The SCSI initiator device is comprised of an application client and task management function, which initiates device service and task management requests.
SCSI Device Model (Cont.)

- **CDB structure**
  - 8 bit structure
  - Contain operation code, command specific parameter and control parameter

- **SCSI Ports**
  - SCSI device may contain initiator port, target port, target/initiator port
  - Based on port combination device is classified
  - For example
    - Target/initiator device contain target/initiator port and can switch orientations depending on the role it plays while participating in an I/O operation
    - To cater to service requests from multiple devices, a SCSI device may also have multiple ports

Each device service request contains a *Command Descriptor Block (CDB)*. The CDB defines the command to be executed and lists command-specific inputs and other parameters specifying how to process the command.

SCSI ports are the physical connectors that the SCSI cable plugs into for communication with a SCSI device. A SCSI device may contain target ports, initiator ports, target/initiator ports, or a target with multiple ports. Based on the port combinations, a SCSI device can be classified as an initiator model, a target model, a combined model, or a target model with multiple ports. In an initiator model, the SCSI initiator device has only initiator ports. Therefore, the application client can only initiate requests to the service delivery subsystem and receive confirmation. This device cannot serve any requests, and therefore does not contain a logical unit. Similarly, a SCSI target device with only a target port can serve requests but cannot initiate them. The SCSI target/initiator device has a target/initiator port that can switch orientations depending on the role it plays while participating in an I/O operation. To cater to service requests from multiple devices, a SCSI device may also have multiple ports of the same orientation (target).
SCSI Addressing

- Initiator ID - a number from 0 to 15 with the most common value being 7.
- Target ID - a number from 0 to 15
- LUN - a number that specifies a device addressable through a target.

The SCSI devices are identified by a specific number called a SCSI ID. In narrow SCSI (bus width=8), the devices are numbered 0 through 7; in wide (bus width=16) SCSI, the devices are numbered 0 through 15. These ID numbers set the device priorities on the SCSI bus. In narrow SCSI, 7 has the highest priority and 0 has the lowest priority. In wide SCSI, the device IDs from 8 to 15 have the highest priority, but the entire sequence of wide SCSI IDs has lower priority than narrow SCSI IDs. Therefore, the overall priority sequence for a wide SCSI is 7, 6, 5, 4, 3, 2, 1, 0, 15, 14, 13, 12, 11, 10, 9, and 8. When a device is initialized, SCSI allows for automatic assignment of device IDs on the bus, which prevents two or more devices from using the same SCSI ID.
SCSI Addressing Example

In the Parallel SCSI Initiator-Target communication, an initiator ID uniquely identifies the initiator and is used as an originating address. This ID is in the range of 0 to 15, with the range 0 to 7 being the most common. A target ID uniquely identifies a target and is used as the address for exchanging commands and status information with initiators. The target ID is in the range of 0 to 15.

SCSI addressing is used to identify hosts and devices. In this addressing, the UNIX naming convention is used to identify a disk and the three identifiers—initiator ID, target ID, and a LUN—in the $c_n|t_n|d_n$ format, which is also referred as $ctd$ addressing. Here, $C_n$ is the initiator ID, commonly referred to as the controller ID; $T_n$ is the target ID of the device, such as $t_0$, $t_1$, $t_2$, and so on; and $D_n$ is the device number reflecting the actual address of the device unit, such as $d_0$, $d_1$, and $d_2$. A LUN identifies a specific logical unit in a target. The implementation of SCSI addressing may differ from one vendor to another. Figure shows $ctd$ addressing in the SCSI architecture.
Lesson Summary

Key points covered in this lesson:

• SCSI – 3 Architecture
• SCSI device model
• SCSI addressing

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Module Summary

Key points covered in this module:

- DAS can be internal or external
- Multiple hosts cannot share same storage ports
- DAS is made up of a CPU, connectivity, and storage devices
- DAS connectivity uses block-level access protocols
- SCSI – 3 architecture
- Parallel SCSI addressing
- SCSI Command model

This concludes the module. Key points covered in this module are shown here. Please take a moment to review them.
Check Your Knowledge

• What are the physical elements of DAS?
• Give an example of when DAS is a good solution.
• Describe internal DAS connectivity.
• Describe external DAS connectivity.
• List SCSI Device Models with Different Port Configurations.
• How many devices SCSI can support?
• Which SCSI ID has highest priority?
Module 2.2 – Storage Area Network
The objectives for this module are shown here. Please take a moment to read them.
Lesson: Fibre Channel SAN

Upon completion of this lesson, you will be able to:

• Define Storage Area Network and its benefits
• Define Fibre Channel
• List the component of SAN
• Describe three FC interconnectivity options
• List different FC port types

The objectives for this lesson are shown here. Please take a moment to read them.
### Business Needs and Technology Challenges

- Just-in-time information to business users
- Integration of information infrastructure with business processes
- Flexible and resilient storage architecture
- DAS is inefficient to meet these challenges
  - Storage Networking emerged as a solution
    - FC SAN
    - NAS
    - IP SAN

**Just-in-time information to business users:** Information must be available to business users when they need it. The explosive growth in online storage, proliferation of new servers and applications, spread of mission-critical data throughout enterprises, and demand for 24 × 7 data availability are some of the challenges that need to be addressed.

**Integration of information infrastructure with business processes:** The storage infrastructure should be integrated with various business processes without compromising its security and integrity.

**Flexible and resilient storage architecture:** The storage infrastructure must provide flexibility and resilience that aligns with changing business requirements. Storage should scale without compromising performance requirements of the applications and, at the same time, the total cost of managing information must be low.
**What is a SAN?**

- Dedicated high speed network of servers and shared storage devices
- Provide block level data access
- Resource Consolidation
  - Centralized storage and management
- Scalability
  - Theoretical limit: Appx. 15 million devices
- Secure Access

A *storage area network (SAN)* is a dedicated high speed network for block level data access. It carries data between servers (also known as *hosts*) and storage devices through Fibre Channel switches. SAN enables storage consolidation and allows storage to be shared across multiple servers. SAN provides the physical communication infrastructure and enables secure and robust communication between host and storage devices.
Understanding Fibre Channel

- Fibre Channel is a high-speed network technology uses:
  - Optical fiber cables (for front end connectivity)
  - Serial copper cables (for back end connectivity)

- Latest FC implementations support 8Gb/s

- Servers are attached to 2 distinct networks
  - Back-end
  - Front-end

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.

Higher data transmission speeds are an important feature of the FC networking technology. The initial implementation offered throughput of 100 MB/s (equivalent to raw bit rate of 1Gb/s i.e. 1.0625 Mb/s in Fibre Channel), which was greater than the speeds of Ultra SCSI (20 MB/s) commonly used in DAS environments. FC in full-duplex mode could sustain throughput of 200 MB/s. In comparison with Ultra-SCSI, FC is a significant leap in storage networking technology. Latest FC implementations of 8 GFC (Fibre Channel) offers throughput of 1600 MB/s (raw bit rates of 8 Gb/s), whereas Ultra320 SCSI is available with a throughput of 320 MB/s. The FC architecture is highly scalable and theoretically a single FC network can accommodate approximately 15 million nodes.

When looking at an overall IT infrastructure, the SAN and LAN are mutually exclusive but serve similar purposes. The LAN allows clients, such as desktop work-stations, to request data from servers. This could be considered the front-end network. This is where the average user would connect typically across an Ethernet network.

The SAN, or back-end network also connects to servers, but in this case, the servers are acting as clients. They are requesting data from the storage arrays. These connections are accomplished via a Fibre Channel network. (Note: FibRE refers to the protocol versus fibER which refers to a media!). By combining the two networks together, with the servers as the common thread, the end-user is supplied with any data they may need.
In its earliest implementation, the SAN was a simple grouping of hosts and the associated storage that was connected to a network using a hub as a connectivity device. This configuration of a 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. The switched fabric topologies improved connectivity and performance, which enabled SANs to be highly scalable. This enhanced data accessibility to applications across the enterprise. FC-AL has been abandoned for SANs due to its limitations, but still survives as a disk-drive interface. Slide illustrates the FC SAN evolution from FC-AL to enterprise SANs.
A SAN consists of three basic components: servers, network infrastructure, and storage. These components can be further broken down into the following key elements: node ports, cabling, interconnecting devices (such as FC switches or hubs), storage arrays, and SAN management software.

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. These ports are integral components of an HBA and the storage front-end adapters. A port operates in full-duplex data transmission mode with a transmit (Tx) link and a receive (Rx) link.
Components of SAN: Cabling

- SAN implementation uses:
  - Copper cables for short distance
  - Optical fiber cables for long distance

- Two types of optical cables
  - Single-mode
    - Can carry single beams of light
    - Distance up to 10 KM
  - Multi-mode
    - Can carry multiple beams of light simultaneously
    - Distance up to 500 meters

SAN implementations use optical fiber cabling. Copper can be used for shorter distances for back-end connectivity, as it provides a better signal-to-noise ratio for distances up to 30 meters. Optical fiber cables carry data in the form of light. There are two types of optical cables, multi-mode and single-mode.

Multi-mode fiber (MMF) cable carries multiple beams of light projected at different angles simultaneously onto the core of the cable. Based on the bandwidth, multimode fibers are classified as OM1 (62.5μm), OM2 (50μm) and laser optimized OM3 (50μm). 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. An MMF cable is usually used for distances of up to 500 meters because of signal degradation (attenuation) due to modal dispersion.

Single-mode fiber (SMF) carries a single ray of light projected at the center of the core. These cables are available in diameters of 7–11 microns; the most common size is 9 microns. 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.
Components of SAN: Cabling (Connectors)

Node Connectors:
- SC Duplex Connectors
- LC Duplex Connectors

Patch panel Connectors
- ST Simplex Connectors

A Standard connector (SC) and a Lucent connector (LC) 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 8 Gb/s. A Straight Tip (ST) is a fiber optic connector with a plug and a socket that is locked with a half-twisted bayonet lock. In the early days of FC deployment, fiber optic cabling predominantly used ST connectors. This connector is often used with Fibre Channel patch panels.

The Small Form-factor Pluggable (SFP) is an optical transceiver used in optical communication. The standard SFP+ transceivers support data rates up to 10 Gb/s.
Components of SAN: Interconnecting devices

- Basis for SAN communication
  - Hubs
  - Switches and
  - Directors

Hubs, switches, and directors are the interconnect devices commonly used in SAN. **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.
The fundamental purpose of a SAN is to provide host access to storage resources. The large storage capacities offered by modern storage arrays have been exploited in SAN environments for storage consolidation and centralization. SAN implementations complement the standard features of storage arrays by providing high availability and redundancy, improved performance, business continuity, and multiple host connectivity.
Components of SAN: SAN management software

- A suite of tools used in a SAN to manage the interface between host and storage arrays
- Provides integrated management of SAN environment
- Web based GUI or CLI

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. In addition, the software provides management of typical SAN components such as HBAs, storage components, and interconnecting devices.
The FC architecture supports three basic interconnectivity options: point-to-point, arbitrated loop (FC-AL), and fabric connect.

**Point-to-point** is the simplest FC configuration — two devices are connected directly to each other. This configuration provides a dedicated connection for data transmission between nodes. However, the point-to-point configuration offers limited connectivity, as only two devices can communicate with each other at a given time. Moreover, it cannot be scaled to accommodate a large number of network devices. Standard DAS uses point-to-point connectivity.
SAN Interconnectivity Options: FC-AL

• Fibre Channel Arbitrated Loop (FC-AL)
  – Devices must arbitrate to gain control
  – Devices are connected via hubs
  – Supports up to 127 devices

In the *FC-AL configuration*, devices are attached to a shared loop. FC-AL has the characteristics of a token ring topology and a physical star topology. In FC-AL, each device contends with other devices to perform I/O operations. Devices on the loop must “arbitrate” to gain control of the loop. At any given time, only one device can perform I/O operations on the loop.

As a loop configuration, FC-AL can be implemented without any interconnecting devices by directly connecting one device to another in a ring through cables.

However, FC-AL implementations may also use hubs whereby the arbitrated loop is physically connected in a star topology. The FC-AL configuration has the following limitations in terms of scalability:

• FC-AL shares the bandwidth in the loop. Only one device can perform I/O operations at a time. Because each device in a loop has to wait for its turn to process an I/O request, the speed of data transmission is low in an FC-AL topology.
• FC-AL uses 8-bit addressing. It can support up to 127 devices on a loop.
• Adding or removing a device results in loop re-initialization, which can cause a momentary pause in loop traffic.
When a node in the FC-AL topology attempts to transmit data, the node sends an *arbitration (ARB)* frame to each node on the loop. If two nodes simultaneously attempt to gain control of the loop, the node with the highest priority is allowed to communicate with another node. This priority is determined on the basis of Arbitrated Loop Physical Address (AL-PA) and Loop ID. When the initiator node receives the ARB request it sent, it gains control of the loop. The initiator then transmits data to the node with which it has established a virtual connection. The slide illustrates the process of data transmission in an FC-AL configuration.

Let's understand how arbitration is done in FC-AL. Consider Node A want to communicate with Node B, the steps are as follows:

1. High priority initiator, Node A inserts the ARB frame in the loop.
2. ARB frame is passed to the next node (Node D) in the loop.
3. Node D receives high priority ARB, therefore remains idle.
4. ARB is forwarded to next node (Node C) in the loop.
5. Node C receives high priority ARB, therefore remains idle.
6. ARB is forwarded to next node (Node B) in the loop.
7. Node B receives high priority ARB, therefore remains idle and
8. ARB is forwarded to next node (Node A) in the loop.
9. Node A receives ARB back; now it gains control of the loop and can start communicating with target Node B.
SAN Interconnectivity Options: FC-SW

- Fabric connect (FC-SW)
  - Dedicated bandwidth between devices
  - Support up to 15 million devices
  - Higher availability than hubs

Unlike a loop configuration, a Fibre Channel switched fabric (FC-SW) network provides interconnected devices, dedicated bandwidth, and scalability. The addition or removal of a device in a switched fabric is minimally disruptive; it does not affect the ongoing traffic between other devices. FC-SW is also referred to as fabric connect. A fabric is a logical space in which all nodes communicate with one another in a network. This virtual space can be created with a switch or a network of switches. Each switch in a fabric contains a unique domain identifier, which is part of the fabric’s addressing scheme. In FC-SW, nodes do not share a loop; instead, data is transferred through a dedicated path between the nodes. Each port in a fabric has a unique 24-bit fibre channel address for communication.
FC-SW uses switches that are intelligent devices. They can switch data traffic from an initiator node to a target node directly through switch ports. Frames are routed between source and destination by the fabric. As shown in slide, if node B wants to communicate with node D, nodes should individually login first and then transmit data via the FC-SW. This link is considered a dedicated connection between the initiator and the target.
Ports are the basic building blocks of an FC network. Ports on the switch can be one of the following types:

- **N_port**: An end point in the fabric. This port is also known as the node port. Typically, it is a host port (HBA) or a storage array port that is connected to a switch in a switched fabric.

- **NL_port**: A node port that supports the arbitrated loop topology. This port is also known as the node loop port.

- **E_port**: An FC port that forms the connection between two FC switches. This port is also known as the expansion port. The E_port on an FC switch connects to the E_port of another FC switch in the fabric through a link, which is called an InterSwitch Link (ISL). ISLs are used to transfer host-to-storage data as well as the fabric management traffic from one switch to another. ISL is also one of the scaling mechanisms in SAN connectivity.

- **F_port**: A port on a switch that connects an N_port. It is also known as a fabric port and cannot participate in FC-AL.

- **FL_port**: A fabric port that participates in FC-AL. This port is connected to the NL_ports on an FC-AL loop. A FL_port also connects a loop to a switch in a switched fabric. As a result, all NL_ports in the loop can participate in FC-SW. This configuration is referred to as a public loop. In contrast, an arbitrated loop without any switches is referred to as a private loop. A private loop contains nodes with NL_ports, and does not contain FL_port.

- **G_port**: A generic port that can operate as an E_port or an F_port and determines its functionality automatically during initialization.

Slide shows various FC ports located in the fabric.
**Inter Switch Links (ISL)**

- ISL connects two or more FC switches to each other using E-Ports
- ISLs are used to transfer host-to-storage data as well as the fabric management traffic from one switch to another
- ISL is also one of the scaling mechanisms in SAN connectivity

*Multimode Fiber*

- 1Gb=500m
- 2Gb=300m

*Single-mode Fiber*

- up to 10 km

---

An ISL (Inter-Switch Link) allows for two or more Fibre Channel switches to be connected together to form a single, but larger, fabric. Expansion ports (E_Ports) on an FC switch provide interswitch link (ISL) connectivity to fabric directors and switches.

ISLs are used to transfer host-to-storage data as well as the fabric management traffic from one switch to another. ISL is also one of the scaling mechanisms in SAN connectivity.

By using inter-switch links, a switched fabric can be expanded to connect hundreds of nodes.
Login Types in a Switched Network

Extended Link Services that are defined in the standards:

- **FLOGI - Fabric login**
  - Between N_Port to F_Port

- **PLOGI - Port login**
  - Between N_Port to N_Port
  - N_Port establishes a session with another N_Port

- **PRLI - Process login**
  - Between N_Port to N_Port
  - To share information about the upper layer protocol type in use
  - And recognizing device as the SCSI initiator, or target

Fabric services define three login types:

- **Fabric login (FLOGI)** is performed between an N_port and an F_port. To log on to the fabric, a device sends a FLOGI frame with the World Wide Node Name (WWNN) and World Wide Port Name (WWPN) parameters to the login service at the well-known FC address FFFFFFFE. In turn, the switch accepts the login and returns an Accept (ACC) frame with the assigned FC address for the device. Immediately after the FLOGI, the N_port registers itself with the local name server on the switch, indicating its WWNN, WWPN, and assigned FC address.

- **Port login (PLOGI)** is performed between an N_port and another N_port to establish a session. The initiator N_port sends a PLOGI request frame to the target N_port, which accepts it. The target N_port returns an ACC to the initiator N_port. Next, the N_ports exchange service parameters relevant to the session.

- **Process login (PRLI)** is also performed between an N_port and another N_port. This login relates to the FC upper layer protocols (ULP) such as SCSI. N_ports exchange SCSI-3-related service parameters. N_ports share information about the ULP type in use, the SCSI initiator, or the target.
Lesson Summary

Key topics covered in this lesson:

• FC SAN and its components
• SAN Interconnectivity Options
• Port types and inter switch links
Lesson: Fibre Channel Architecture

Upon completion of this lesson, you will be able to:

- Describe layers of FC
- Describe FC protocol stack
- Discuss FC addressing
- Define WWN addressing
- Discuss structure and organization of FC Data

The objectives for this lesson are shown here. Please take a moment to read them.
**FC Architecture Overview**

- FC uses channel technology
- Provide high performance with low protocol overheads
- FCP is SCSI-3 over FC network
  - Sustained transmission bandwidth over long distances
  - Provides speeds up to 8 Gb/s (8 GFC)
- FCP has five layers:
  - FC-4
  - FC-2
  - FC-1
  - FC-0
  - *FC-3 is not yet implemented

The FC architecture represents true channel/network integration with standard interconnecting devices. Connections in a SAN are accomplished using FC. Traditionally, transmissions from host to storage devices are carried out over channel connections such as a parallel bus. Channel technologies provide high levels of performance with low protocol overheads. Such performance is due to the static nature of channels and the high level of hardware and software integration provided by the channel technologies. However, these technologies suffer from inherent limitations in terms of the number of devices that can be connected and the distance between these devices. *Fibre Channel Protocol (FCP)* is the implementation of serial SCSI-3 over an FC network. In the FCP architecture, all external and remote storage devices attached to the SAN appear as local devices to the host operating system. The key advantages of FCP are as follows:

- Sustained transmission bandwidth over long distances.
- Support for a larger number of addressable devices over a network.
- Exhibits the characteristics of channel transport and provides speeds up to 8 Gb/s (8 GFC).
### Fibre Channel Protocol Stack

<table>
<thead>
<tr>
<th>FC layer</th>
<th>Function</th>
<th>SAN relevant features specified by FC layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-4</td>
<td>Mapping interface</td>
<td>Mapping upper layer protocol (e.g. SCSI-3 to FC transport)</td>
</tr>
<tr>
<td>FC-3</td>
<td>Common services</td>
<td>Not implemented</td>
</tr>
<tr>
<td>FC-2</td>
<td>Routing, flow control</td>
<td>Frame structure, ports, FC addressing, buffer credits</td>
</tr>
<tr>
<td>FC-1</td>
<td>Encode/decode</td>
<td>8b/10b encoding, bit and frame synchronization</td>
</tr>
<tr>
<td>FC-0</td>
<td>Physical layer</td>
<td>Media, cables, connector</td>
</tr>
</tbody>
</table>

FCP defines the communication protocol in five layers: FC-0 through FC-4 (except FC-3 layer, which is not implemented). In a layered communication model, the peer layers on each node talk to each other through defined protocols.

**FC-4 Upper Layer Protocol:**

FC-4 is the uppermost layer in the FCP stack. This layer defines the application interfaces and the way Upper Layer Protocols (ULPs) are mapped to the lower FC layers. The FC standard defines several protocols that can operate on the FC-4 layer. Some of the protocols include SCSI, HIPPI Framing Protocol, Enterprise Storage Connectivity (ESCON), ATM, and IP.

**FC-2 Transport Layer:**

The FC-2 is the transport layer that contains the payload, addresses of the source and destination ports, and link control information. The FC-2 layer provides Fibre Channel addressing, structure, and organization of data (frames, sequences, and exchanges). It also defines fabric services, classes of service, flow control, and routing.

**FC-1 Transmission Protocol:**

This layer defines the transmission protocol that includes serial encoding and decoding rules, special characters used, and error control. At the transmitter node, an 8-bit character is encoded into a 10-bit transmissions character. This character is then transmitted to the receiver node. At the receiver node, the 10-bit character is passed to the FC-1 layer, which decodes the 10-bit character into the original 8-bit character.

**FC-0 Physical Interface:**

FC-0 is the lowest layer in the FCP stack. This layer defines the physical interface, media, and transmission of raw bits. The FC-0 specification includes cables, connectors, and optical and electrical parameters for a variety of data rates. The FC transmission can use both electrical and optical media.
Fibre Channel Addressing

- FC Address is assigned during Fabric Login
  - Used to communicate between nodes within SAN
  - Similar in functionality to an IP address on NICs

- Address Format:
  - 24 bit address, dynamically assigned
  - Contents of the three bytes depend on the type of N-Port
  - For an N_Port or a public NL_Port:
    - switch maintains mapping of WWN to FC-Address via the Name Server

Fibre Channel Address is dynamically assigned during fabric login. It has a distinct format depending on the type of node port as shown in the diagram.

The address format depends upon the type of node in the fabric: N_Port, NL_Port in a Public Loop, or NL_Port in a private loop.

**FC Address of an N_Port:**

The first field contains the domain ID of the switch (the domain ID is discussed in more detail in the next lesson). Although this is an 8-bit field, there are only 239 available addresses for domain ID, since some addresses are deemed special and reserved for fabric management services. The area ID is used to identify a group of F_Ports. An example of a group of F_Ports would be a card on the switch with more than one port on it. The last field identifies the F_Port within the group.

The maximum possible number of N_Ports in a switched fabric is thus:

\[(239 \text{ domains} \times 256 \text{ areas} \times 256 \text{ ports}) = 15,663,104 \text{ Fibre Channel addresses.}\]

**FC Address of an NL_Port:**

If no switch connection exists to an arbitrated loop, all NL_Ports are in a private loop and the two upper bytes of their FC addresses are zeroes.

If the loop is attached to a fabric and an NL_Port supports a fabric login, the two upper bytes are assigned a positive value – called the loop identifier - by the switch. We call this mode a public loop. The loop identifier is the same for all NL_Ports on a given loop.

In public and private arbitrated loops, the last byte of the 24-bit port address refers to the arbitrated loop physical address (AL_PA). There are 127 allowable AL_PA addresses within a loop. In a fabric-attached or public loop there may be no more than 126 NL_Ports, with one address reserved for the FL_Port on the switch.
**World Wide Names**

- Unique 64 bit identifier
- Static to the port
  - Used to physically identify ports or nodes within SAN
  - Similar to NIC’s MAC address

<table>
<thead>
<tr>
<th>World Wide Name - Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 0 0 0 6 0 1 6 0 0 0 6 0 0 1 6 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1</td>
</tr>
<tr>
<td>0101 0000 0000 0110 0000 0001 0110 0000 0000 0110 0000 0000 0001 1011 0010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>World Wide Name - HBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0 0 0 0 0 0 0 0 c 9 2 0 0 d c 4 0</td>
</tr>
<tr>
<td>0101 0000 0000 0110 0000 0001 0110 0000 0000 0110 0000 0000 0001 1011 0010</td>
</tr>
</tbody>
</table>

Each device in the FC environment is assigned a 64-bit unique identifier called the World Wide Name (WWN). The Fibre Channel environment uses two types of WWNs: World Wide Node Name (WWNN) and World Wide Port Name (WWPN). Unlike an FC address, which is assigned dynamically, a WWN is a static name for each device on an FC network. WWNs are similar to the Media Access Control (MAC) addresses used in IP networking. WWNs are burned into the hardware or assigned through software. Several configuration definitions in a SAN use WWN for identifying storage devices and HBAs. The name server in an FC environment keeps the association of WWNs to the dynamically created FC addresses for nodes. Figure illustrates the WWN structure for an array and the HBA.
Structure and Organization of FC Data

- FC data is organized as:
  - Exchange operations
    - Enables two N_ports to identify and manage a set of information units
    - Maps to sequence
  - Sequence
    - Contiguous set of frames sent from one port to another
  - Frames
    - Fundamental unit of data transfer
    - Each frame can contain up to 2112 bytes of payload

In an FC network, data transport is analogous to a conversation between two people, whereby a frame represents a word, a sequence represents a sentence, and an exchange represents a conversation.

- **Exchange operation**: An exchange operation enables two N_ports to identify and manage a set of information units. This unit maps to a sequence. Sequences can be both unidirectional and bidirectional depending upon the type of data sequence exchanged between the initiator and the target.

- **Sequence**: A sequence refers to a contiguous set of frames that are sent from one port to another. A sequence corresponds to an information unit, as defined by the ULP.

- **Frame**: A frame is the fundamental unit of data transfer at Layer 2. Each frame can contain up to 2,112 bytes of payload.
Lesson Summary

Key topics covered in this lesson:

• Fibre Channel Protocol Stack
• Fibre Channel Addressing
• Data Organization: Frame, Sequence and Exchange
Lesson: FC Topologies and Management

Upon completion of this lesson, you will be able to:

- Define FC fabric topologies
- Describe different types of zoning

The objectives for this lesson are shown here. Please take a moment to read them.
A fabric topology can be described by the number of tiers it contains. The number of tiers in a fabric is based on the number of switches traversed between two points that are farthest from each other. However, note that this number is based on the infrastructure constructed by the fabric topology; it disregards how the storage and server are connected across the switches.

In the core-edge fabric topology, there are two types of switch tiers in this fabric. The edge tier usually comprises switches and offers an inexpensive approach to adding more hosts in a fabric. The core tier usually comprises enterprise directors that ensure high fabric availability. Additionally all traffic has to either traverse through or terminate at this tier. In a two-tier configuration, all storage devices are connected to the core tier. The host-to-storage traffic has to traverse one and two ISLs in a two-tier and three-tier configuration, respectively. Hosts used for mission-critical applications can be connected directly to the core tier and consequently avoid traveling through the ISLs to process I/O requests from these hosts.

The core-edge fabric topology increases connectivity within the SAN while conserving overall port utilization. If expansion is required, an additional edge switch can be connected to the core. This topology can have different variations.

In a single-core topology, all hosts are connected to the edge tier and all storage is connected to the core tier. A dual-core topology can be expanded to include more core switches. However, to maintain the topology, it is essential that new ISLs are created to connect each edge switch to the new core switch that is added.
**Fabric Topology: Mesh**

- Can be either partial or full mesh
- All switches are connected to each other
- Host and Storage can be located anywhere in the fabric
- Host and Storage can be localized to a single switch

In a *mesh topology*, each switch is directly connected to other switches by using ISLs. This topology promotes enhanced connectivity within the SAN. When the number of ports on a network increases, the number of nodes that can participate and communicate also increases. A mesh topology may be one of the two types: full mesh or partial mesh. In a *full mesh*, every switch is connected to every other switch in the topology. Full mesh topology may be appropriate when the number of switches involved is small. A typical deployment would involve up to four switches or directors, with each of them servicing highly localized host-to-storage traffic. In a full mesh topology, a maximum of one ISL or hop is required for host-to-storage traffic.

In a *partial mesh* topology, several hops or ISLs may be required for the traffic to reach its destination. Hosts and storage can be located anywhere in the fabric, and storage can be localized to a director or a switch in both mesh topologies. A full mesh topology with a symmetric design results in an even number of switches, whereas a partial mesh has an asymmetric design and may result in an odd number of switches.
Zoning is an FC switch function that enables nodes within the fabric to be logically segmented into groups that can communicate with each other. When a device (host or storage array) logs onto a fabric, it is registered with the name server. When a port logs onto the fabric, it goes through a device discovery process with other devices registered in the name server. The zoning function controls this process by allowing only the members in the same zone to establish these link-level services.
Multiple zone sets may be defined in a fabric, but only one zone set can be active at a time. A zone set is a set of zones and a zone is a set of members. A member may be in multiple zones. Members, zones, and zone sets form the hierarchy defined in the zoning process. Members are nodes within the SAN that can be included in a zone. Zones comprise a set of members that have access to one another. A port or a node can be a member of multiple zones. Zone sets comprise a group of zones that can be activated or deactivated as a single entity in a fabric. Only one zone set per fabric can be active at a time. Zone sets are also referred to as zone configurations.
Zoning can be categorized into three types:

- **Port zoning**: It uses the FC addresses of the physical ports to define zones. In port zoning, access to data is determined by the physical switch port to which a node is connected. The FC address is dynamically assigned when the port logs on to the fabric. Therefore, any change in the fabric configuration affects zoning. Port zoning is also called hard zoning. Although this method is secure, it requires updating of zoning configuration information in the event of fabric reconfiguration.

- **WWN zoning**: It uses World Wide Names to define zones. WWN zoning is also referred to as soft zoning. A major advantage of WWN zoning is its flexibility. It allows the SAN to be re-cabled without reconfiguring the zone information. This is possible because the WWN is static to the node port.

- **Mixed zoning**: It combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific port to be tied to the WWN of a node.

Zoning is used in conjunction with LUN masking for controlling server access to storage. However, these are two different activities. Zoning takes place at the fabric level and LUN masking is done at the array level.
Lesson Summary

Key topics covered in this lesson:

• FC SAN Topologies
  – Core-Edge
  – Mesh

• Fabric management by zoning
Module Summary

Key topics covered in this module:

• SAN features and benefits
• SAN connectivity options
• Port types and inter switch links
• FC protocol stack and addressing
• FC fabric topologies
• Fabric management by zoning

These are the key points covered in this module. Please take a moment to review them.
Exercise

Business Profile:
A mid-sized publishing house has a centralized IT department located in San Francisco, CA, connected to 3 branch offices on the west coast. Recent analysis of the storage environment suggests that they are getting poor ROI in their storage infrastructure.

Current Situation/Issue:
The company’s current infrastructure consists of several storage arrays direct-attached to a heterogeneous mix of 90 servers. All servers are dual-attached to the arrays for reliability and redundancy.

Since each storage device has 32 connectivity ports, each could support a maximum of 16 servers. The company sees the 32-ports as a limitation. Each storage device has the disk capacity to support more than 16 servers. However, there was no way to add a 17th server to make use of that capacity and support future growth.

Proposal:
How would you suggest that the company restructure their environment? List the advantages and disadvantages of your proposed solution. (Note: You will need to justify your choice based on scalability, performance, and availability of the new solution.)
FC switches and directors are key components of the SAN environment. EMC offers Enterprise directors, Departmental switches, and Multi-protocol routers connectivity products under the Connectrix brand.

Enterprise directors are ideal for large enterprise connectivity. They offer high port density and high component redundancy. Enterprise directors are deployed in high-availability or large-scale environments. Connectrix directors offer several hundred ports per domain. Departmental switches are best suited for workgroup, mid-tier environments. Multi-protocol routers support mixed iSCSI and FC environments. They can bridge FC SAN and IP SAN, a feature that enables these routers to provide connectivity between iSCSI host initiators and FC storage targets. They can extend FC SAN over long distances through IP networks. In addition to FC ports, Connectrix switches and directors have Ethernet ports and serial ports for communication and switch management functions. Connectrix management software enables configuration, monitoring, and management of Connectrix switches.

EMC ControlCenter SAN Manager provides a single interface for managing SAN. With SAN Manager one can discover, monitor, manage, and configure complex heterogeneous SAN environments faster and easier. It streamlines and centralizes SAN management operations across multi-vendor storage networks and storage devices.
Connectrix Video

Connectrix Video
Check Your Knowledge

- Name three key features of a SAN implementation.
- What is a fabric?
- Describe how a SAN can be connected?
- What is a purpose of ISL
- What is a function of layer-2 in FC architecture
- Define the purpose of zoning?
- What is a Core-Edge Fabric?
Network-Attached Storage

Module 2.3
Module Objective

After completing this module, you will be able to:

- Describe NAS, its benefits and components
- Discuss different NAS implementations
- Describe NAS file-sharing protocols
- Discuss NAS management options

The objectives for this module are shown here. Please take a moment to review them.
File Sharing Environment

- File system is structured way of storing and organizing data files

- File Sharing
  - Storing and accessing data files over network
  - FS must be mounted in order to access files

- *Client/server model* is implemented with file-sharing protocols for remote file sharing
  - Example: FTP, CIFS and NFS

File sharing refers to storing and accessing files over a network. In a file sharing environment, a user who creates the file (the creator or owner of a file) determines the type of access to be given to other users (read, write, execute, append, delete, and list) and controls changes to the file. When multiple users try to access a shared file at the same time, a protection scheme is required to maintain data integrity and, at the same time, make this sharing possible. FTP (file transfer protocol) and Distributed File Systems (DFS), that uses a file-sharing protocol are some examples of implementations of file-sharing environments. 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. A DFS (or NFS) 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. In a client/server model that uses DFS, the clients mount remote file systems that are available on dedicated file servers. Example of standard client/server file-sharing protocols are NFS for UNIX and CIFS for Windows. 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. In addition, a name service, such as Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), and Network Information Services (NIS), helps users identify and access a unique resource over the network.
In the past, floppy drives with capacities in mere KB’s were widely used to share data files. Over time the need for larger capacity has emerged due to growing need for data to be shared across organizations. Removable storage media, such as flash drives, capable of storing gigabytes (GB) of data, have now complimented the traditional removable media drives.

Businesses not only need the capacity to handle huge data storage requirements, the need to share their data has made Network Attached Storage (NAS) an attractive option. NAS systems use external storage for server-hosts, adding flexibility to network storage. NAS works at the file level, rather than the block level. This enables widespread access to the data over the network, based upon the file system client loaded.
Network-attached storage (NAS) is an IP-based file-sharing device attached to a local area network. NAS provides the advantages of server consolidation by eliminating the need for multiple file servers. It provides storage consolidation through file-level data access and sharing. NAS is a preferred storage solution that enables clients to share files quickly and directly with minimum storage management overhead. NAS also helps to eliminate bottlenecks that users face when accessing files from a general-purpose server. NAS uses network and file-sharing protocols to perform filing and storage functions. These protocols include TCP/IP for data transfer and CIFS and NFS for remote file service. NAS enables both UNIX and Microsoft Windows users to share the same data seamlessly. To enable data sharing, NAS typically uses NFS for UNIX, CIFS for Windows, and File Transfer Protocol (FTP) and other protocols for both environments. Recent advancements in networking technology have enabled NAS to scale up to enterprise requirements for improved performance and reliability in accessing data. A NAS device is a dedicated, high-performance, high-speed, single-purpose file serving and storage system. NAS serves a mix of clients and servers over an IP network. Most NAS devices support multiple interfaces and networks. A NAS device uses its own operating system and integrated hardware, software components to meet specific file service needs. Its operating system is optimized for file I/O and, therefore, performs file I/O better than a general purpose server. As a result, a NAS device can serve more clients than traditional file servers, providing the benefit of server consolidation.
A NAS device is optimized for file-serving functions such as storing, retrieving, and accessing files for applications and clients. As shown in Figure, a general-purpose server can be used to host any application, as it runs a generic operating system. Unlike a general-purpose server, a NAS device is dedicated to file-serving. It has a real-time operating system dedicated to file serving by using open-standard protocols. Some NAS vendors support features such as built-in native clustering for high availability.
Benefits of NAS

- Improves efficiency
- Improved flexibility
- Centralized storage
- Simplifies management
- Scalability
- High availability – through native clustering
- Provides security integration to environment (user authentication and authorization)

NAS offers the following benefits:

- **Improved efficiency**: Eliminates bottlenecks that occur during file access from a general-purpose file server because NAS uses an operating system specialized for file serving. It improves the utilization of general-purpose servers by relieving them of file-server operations.
- **Improved flexibility**: Compatible for 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, simplify data management, and ensures greater data protection.
- **Simplified management**: Provides a centralized console that makes it possible to manage file systems efficiently.
- **Scalability**: Scales well in accordance 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 networking components that provide maximum connectivity options. A NAS device can use clustering technology for failover.
- **Security**: Ensures security, user authentication, and file locking in conjunction with industry-standard security schemas.
A NAS has the following components:

- NAS head (CPU and Memory)
- One or more network interface cards (NICs), which provide connectivity to the network. NIC uses technologies such as Gigabit Ethernet, Fast Ethernet, ATM, and Fiber Distributed Data Interface (FDDI).
- An optimized operating system for managing NAS functionality
- NFS and CIFS protocols for stack file sharing
- Industry-standard storage protocols to connect and manage physical disk resources, such as ATA, SCSI, or FC
- Storage Array

The NAS environment includes clients accessing a NAS device over an IP network using standard protocols.
Most NAS devices support multiple file service protocols to handle file I/O requests to a remote file system. As mentioned earlier, NFS and CIFS are the common protocols for file sharing. NFS is predominantly used in UNIX-based operating environments; CIFS is used in Microsoft Windows–based operating environments. These file sharing protocols enable users to share file data across different operating environments and provide a means for users to migrate transparently from one operating system to another.
Network File System (NFS)

- Client/server application
- Uses RPC mechanisms over TCP protocol
- Mount points grant access to remote hierarchical file structures for local file system structures
- Access to the mount can be controlled by permissions
  - NFS v2 was stateless and uses UDP as the transport layer protocol
  - NFS v3 added TCP as an option for transport layer protocol
  - NFS v4 uses stateful protocol
    - Aimed to provide support to cluster server deployment
    - Allow scalable parallel access to the files distributed among multiple servers

NFS is a client/server application that enables a computer user view, and optionally stores and update files on a remote computer as though they were on the user's own computer. It uses Remote Procedure Calls (RPC) to communicate between computers.

The user's system requires an NFS client to connect to the NFS server. Since the NFS server and client use TCP/IP to transfer files, TCP/IP must be installed on both systems.

Using NFS, the user or system administrator can mount all or a portion of a file system (which is a portion of the hierarchical tree in any file directory and subdirectory). The portion of the file system that is mounted (designated as accessible) can be controlled using permissions (e.g., read-only or read-write). NFS uses Network Information Service for domain name resolution.
**NAS File Sharing - CIFS**

- **Common Internet File System**
  - Developed by Microsoft in 1996
  - An enhanced version of the Server Message Block (SMB) protocol
  - Stateful Protocol
    - Can automatically restore connections and reopen files that were open prior to interruption
  - Operates at the Application/Presentation layer of the OSI model
  - Most commonly used with Microsoft operating systems, but is platform-independent
  - CIFS runs over TCP/IP and uses DNS (Domain Naming Service) for name resolution

CIFS is a client/server application protocol that enables client programs to make requests for files and services on remote computers over TCP/IP. It is a public, or open, variation of Server Message Block (SMB) protocol. The CIFS protocol enables remote clients to gain access to files that are on a server. CIFS enables file sharing with other clients by using special locks. File names in CIFS are encoded using unicode characters. CIFS provides the 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 runs over TCP
- It supports fault tolerance and can automatically restore connections and reopen files that were open prior to interruption. The fault tolerance features of CIFS depend on whether an application is written to take advantage of these features. Moreover, CIFS is a stateful protocol because the CIFS server maintains connection information regarding every connected client. In the event of a network failure or CIFS server failure, 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 has to take steps to reestablish the CIFS connection.

Users refer to remote file systems with an easy-to-use file naming scheme:

```
\server\share or \servername.domain.suffix\share.
```
The NFS and CIFS protocols handle file I/O requests to a remote file system, which is managed by the NAS device. The process of NAS I/O is as follows:

1. The requestor 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 against the physical storage pool.

3. When the data is returned from the physical storage pool, the NAS device 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.
There are two types of NAS implementations: integrated and gateway. The *integrated NAS* device has all of its components and storage system in a single enclosure. In *gateway* implementation, NAS head shares its storage with SAN environment.
An integrated NAS device has all the components of NAS, such as the NAS head and storage, in a single enclosure, or frame. This makes the integrated NAS a self-contained environment. The NAS head connects to the IP network to provide connectivity to the clients and service the file I/O requests. The storage consists of a number of disks that can range from low-cost ATA to high throughput FC disk drives. Management software manages the NAS head and storage configurations. An integrated NAS solution ranges from a low-end device, which is a single enclosure, to a high-end solution that can have an externally connected storage array.

A low-end appliance-type NAS solution is suitable for applications that a small department may use, where the primary need is consolidation of storage, rather than high performance or advanced features such as disaster recovery and business continuity. This solution is fixed in capacity and might not be upgradable beyond its original configuration. To expand the capacity, the solution must be scaled by deploying additional units, a task that increases management overhead because multiple devices have to be administered. In a high-end NAS solution, external and dedicated storage can be used. This enables independent scaling of the capacity in terms of NAS heads or storage. However, there is a limit to scalability of this solution.

An integrated solution is self-contained and can connect into a standard IP network. Although the specifics of how devices are connected within a NAS implementation vary by vendor and model. In some cases, storage is embedded within a NAS device and is connected to the NAS head through internal connections, such as ATA or SCSI controllers. In others, the storage may be external but connected by using SCSI controllers. In a high-end integrated NAS model, external storage can be directly connected by FC HBAs or by dedicated FC switches. In the case of a low-end integrated NAS model, backup traffic is shared on the same public IP network along with the regular client access traffic. In the case of a high-end integrated NAS model, an isolated backup network can be used to segment the traffic from impeding client access. More complex solutions may include an intelligent storage subsystem, enabling faster backup and larger capacities while simultaneously enhancing performance.
A gateway NAS device consists of an independent NAS head and one or more storage arrays. The NAS head performs the same functions that it does in the integrated solution; while the storage is shared with other applications that require block-level I/O. Management functions in this type of solution are more complex than those in an integrated environment because there are separate administrative tasks for the NAS head and the storage. In addition to the components that are explicitly tied to the NAS solution, a gateway solution can also utilize the FC infrastructure, such as switches, directors, or direct-attached storage arrays. The gateway NAS is the most scalable because NAS heads and storage arrays can be independently scaled up when required. Adding processing capacity to the NAS gateway is an example of scaling. When the storage limit is reached, it can scale up, adding capacity on the SAN independently of the NAS head. Administrators can increase performance and I/O processing capabilities for their environments without purchasing additional interconnect devices and storage. Gateway NAS enables high utilization of storage capacity by sharing it with SAN environment.

In a gateway solution, front-end connectivity is similar to that in an integrated solution. An integrated environment has a fixed number of NAS heads, making it relatively easy to determine IP networking requirements. In contrast, networking requirements in a gateway environment are complex to determine due to scalability options. Adding more NAS heads may require additional networking connectivity and bandwidth. Communication between the NAS gateway and the storage system in a gateway solution is achieved through a traditional FC SAN. To deploy a stable NAS solution, factors such as multiple paths for data, redundant fabrics, and load distribution must be considered.

Implementation of a NAS gateway solution requires analysis of current SAN environment. This analysis is required to determine the feasibility of introducing a NAS workload to the existing SAN. Analyze the SAN to determine whether the workload is primarily read or write, or random or sequential. Determine the predominant I/O size in use. In general, sequential workloads have large I/Os. Typically, NAS workloads are random with small I/O size. Introducing sequential workload with random workloads can be disruptive to the sequential workload. Therefore, it is recommended to separate the NAS and SAN disks. Also, determine whether the NAS workload performs adequately with the configured cache in the storage subsystem.
Hosting and Accessing Files on the NAS

Steps to host a file system:

- Create an array volume
- Assign volume to NAS device
- Create a file system on the volume
- Mount the file system
- Access the file system
  - Use NFS in UNIX environment
    - Execute mount/nfsmount command
  - Use CIFS in windows environment
    - Map the network drive as: \Account1\Act_Rep

Following are the steps required to host files and permit users to access the hosted files on a NAS device:

1. **Create storage array volumes:** Create volumes on the storage array and assign Logical Unit Numbers (LUN) to the volumes. Present the newly created volumes to the NAS device.

2. **Create NAS Volumes:** Perform a discovery operation on the NAS device, to recognize the new array-volumes and create NAS Volumes (logical volumes). Multiple volumes from the storage array may be combined to form large NAS volumes.

3. **Create NAS file systems:** Create NAS file systems on the NAS volumes.

4. **Mount file systems:** Mount the created NAS file system on the NAS device.

5. **Access the file systems:** Publish the mounted file systems on the network using NFS or CIFS for client access.
Most NAS devices ship with vendor supplied management software, which typically provides a means to configure a NAS device. Select devices even provide basic device monitoring and performance capabilities. NAS devices have several unique management issues, such as device availability, backup and recovery, in addition to traditional storage management issues, such as: resource management, space management, capacity, and performance analysis. Monitoring CPU utilization, memory utilization along with IP traffic are also factors. All are essential to ensure service levels are met.
Managing NAS Environments

- Managing an Integrated System
  - Both NAS component and the storage array are managed via NAS management software

- Managing a Gateway System
  - NAS component managed via NAS management software
  - Storage array managed via array management software

Managing an Integrated System

Since the storage array is dedicated to the NAS functionality, the NAS management software is responsible for the managing both the NAS components and the backend storage array.

Managing a Gateway System

Gateway NAS Systems use shared storage, which means that traditional SAN hosts can also utilize the same array. Therefore both the NAS hardware and the array are individually managed by their own specialized management software:

- The NAS component is managed via specialized NAS management software.
- The storage array is managed via its native array management software Supports standard host connectivity where integrated does not (enabling the array to be multipurpose).
Traditional File Server Environment – Example 1

- Internal Users
- Business Clients
- Surfers, Shoppers
- Web Database Transaction Mission Critical Servers
- UNIX File Server
- FC SAN
- Windows File Server
- IP

Internal Users
Business Clients
Surfers, Shoppers
Web Database Transaction Mission Critical Servers
UNIX File Server
FC SAN
Windows File Server
IP
Storage consolidation benefits:
- Increases performance throughput (service level) to end users
- Minimizes investment in additional servers
- Provides storage pooling
- Provides heterogeneous file servings
- Uses existing infrastructure, tools, and processes
In many companies, the need for two differing environments maintains the separation of two technologies using the same infrastructure for the same purpose, but in different ways. Access to networked files for UNIX (NFS) and Microsoft (CIFS) are traditionally housed on separate server infrastructures.
By implementation of NAS, these same file structures can be housed together, while still maintaining their integrity. Within NAS deployments, the same file system can be accessed by the same user via different technologies, either NFS or CIFS, and still maintain the integrity of the data and security structures, as long as the applications used for both methodologies understand the data structures presented.

Benefits of this solution:
- Provides continuous availability to files
- Heterogeneous file sharing
- Reduces cost for additional OS dependent servers
- Adds storage capacity non-disruptively
- Consolidates storage management
- Lowers Total Cost of Ownership
Module Summary

- Key topics covered in this module:
  - NAS Benefits
  - NAS Components
  - NAS File Sharing Protocols
    - CIFS and NFS
  - NAS Implementation
    - Integrated NAS
    - Gateway NAS
  - NAS Management
Concept in Practice – EMC Celerra

- Celerra is:
  - Dedicated and high-performance infrastructure for file level I/Os

- Consists of:
  - Data Movers
  - Control Station
  - Specialized O/S (DART)

EMC offers NAS solutions with the Celerra family of products. EMC Celerra provides a dedicated, high-performance, high-speed communication infrastructure for file level I/Os.

Celerra Architecture

Celerra consists of a Control Station and NAS heads (Data Movers). The Data Mover is a network and storage interface device and the control station is a management interface device. The Data Mover is an independent, autonomous file server that transfers requested files to clients. DART (data access at real time) is Celerra’s specialized operating system, which runs on the Data Mover. This operating system is optimized for performing file operations to move data from the storage array to the network. DART supports standard network file access protocols, such as NFS, CIFS, and FTP.

The control station provides dedicated processing capabilities to control, manage, and configure a NAS solution. The control station hosts the Linux operating system that is used to install, manage, and configure Data Movers and monitor the environmental conditions and performance of all components. The control station also provides high-availability features such as fault monitoring, fault recovery, fault reporting, call home, and remote diagnostics. Administrative functions are also accessible through the local console, SSH, or a Web browser.
For an Integrated configuration, the system is assigned to dedicated NAS storage. This means that there are no other SAN hosts connected to the storage and therefore the whole array is dedicated solely to NAS provisioning.

In the case of EMC NAS, the Control Station is connected to the Data Movers (client network data provider) via both an internal Ethernet and serial interface connection. The Data Mover has two internal network connections for redundancy.

Each Data Mover is directly, dual connected to the Storage array via Fibre Channel connections in the Integrated configuration.

Once the storage array has been connected and the specialized operating system is loaded, the storage array physical disks are partitioned via commands from the Control Station to create the system volumes and the data volumes.
For a Gateway configuration, the NAS system is assigned separately apportioned storage within the array. This means that any capacity remaining within the array can be assigned to conventional SAN hosts connected to the Fabric Switch, once the appropriate zoning and LUN masking has been performed.
Check your Knowledge

• What is the difference between an Integrated and Gateway NAS solution?

• Which NAS type would you deploy if you already have a SAN and storage array?

• Which NAS type is the simplest to deploy?

• Which file serving environments typically uses CIFS and NFS?
Module 2.4 – IP SAN
The objectives for this module are shown here. Please take a moment to read them.
The objectives for this lesson are shown here. Please take a moment to read them.
**Driver for IP SAN Internetworking**

- In FC SAN transfer of block level data takes place over Fibre Channel
- Emerging technologies provide the transfer of block-level data over an existing IP network infrastructure
- IP is being positioned as a storage transport because:
  - Easier management
  - Existing network infrastructure can be leveraged
  - Reduced cost compared to new SAN hardware and software
  - Supports multi-vendor interoperability
  - Many long-distance disaster recovery solutions already leverage IP-based networks
  - Many robust and mature security options are available for IP networks

Traditional SAN environments allow block I/O over Fibre Channel, whereas NAS environments allow file I/O over IP-based networks. Organizations need the performance and scalability of SAN plus the ease of use and lower TCO of NAS solutions. The emergence of IP technology that supports block I/O over IP has positioned IP for storage solutions.

IP offers easier management and better interoperability. When block I/O is run over IP, the existing network infrastructure can be leveraged, which is more economical than investing in new SAN hardware and software. Many long-distance, disaster recovery (DR) solutions are already leveraging IP-based networks. In addition, many robust and mature security options are now available for IP networks. With the advent of block storage technology that leverages IP networks (the result is often referred to as IP SAN), organizations can extend the geographical reach of their storage infrastructure.

IP SAN technologies can be used in a variety of situations. Disaster recovery solutions can also be implemented using both of these technologies.
Two primary protocols that leverage IP as the transport mechanism are iSCSI and Fibre Channel over IP (FCIP).

iSCSI is the host-based encapsulation of SCSI I/O over IP using an Ethernet NIC card or an iSCSI HBA in the host. As illustrated in Figure, IP traffic is routed over a network either to a gateway device that extracts the SCSI I/O from the IP packets or to an iSCSI storage array. The gateway can then send the SCSI I/O to an FC-based external storage array, whereas an iSCSI storage array can handle the extraction and I/O natively.

FCIP uses a pair of bridges (FCIP gateways) communicating over TCP/IP which is the transport protocol. FCIP is used to extend FC networks over distances using an existing IP-based infrastructure, as illustrated in Figure. Today, iSCSI is widely adopted for connecting servers to storage because it is relatively inexpensive and easy to implement, especially in environments where an FC SAN does not exist. FCIP is extensively used in disaster-recovery implementations, where data is duplicated on disk or tape to an alternate site.
**What is iSCSI?**

- IP based protocol used to connect host and storage
- Carries block-level data over IP-based network
- Encapsulate SCSI commands and transport as TCP/IP packet

iSCSI is an IP-based protocol that establishes and manages connections between storage, hosts, and bridging devices over IP. iSCSI carries block-level data over IP-based networks, including Ethernet networks and the Internet. iSCSI is built on the SCSI protocol by encapsulating SCSI commands and data in order to allow these encapsulated commands and data blocks to be transported using TCP/IP packets.
Components of iSCSI

- iSCSI host initiators
  - Host computer using a NIC or iSCSI HBA to connect to storage
  - iSCSI initiator software may need to be installed

- iSCSI targets
  - Storage array with embedded iSCSI capable network port
  - FC-iSCSI bridge

- LAN for IP storage network
  - Interconnected Ethernet switches and/or routers

Host (initiators), targets, and an IP-based network are the principal iSCSI components. The simplest iSCSI implementation does not require any FC components. If an iSCSI-capable storage array is deployed, a host itself can act as an iSCSI initiator, and directly communicate with the storage over an IP network. However, in complex implementations that use an existing FC array for iSCSI connectivity, iSCSI gateways or routers are used to connect the existing FC SAN. These devices perform protocol translation from IP packets to FC packets and vice-versa, thereby bridging connectivity between the IP and FC environments.
**iSCSI - Host Connectivity Options**

- Three options for iSCSI configuration:
  - **Software Initiators**
    - Code that can be loaded onto a host to provide the translation between the storage I/O calls and the network interface
  - **TCP Offload Engine (TOE)**
    - Moves the TCP processing load off the host CPU onto the NIC card, to free up processing cycles for application execution
  - **iSCSI HBA**
    - A network interface adapter with an integrated SCSI ASIC (application-specific integrated circuit)
    - Simplest option for boot from SAN

iSCSI host connectivity requires a hardware component, such as a NIC with a software component (iSCSI initiator) or an iSCSI HBA. In order to use the iSCSI protocol, a software initiator or a translator must be installed to route the SCSI commands to the TCP/IP stack. A standard NIC, a TCP/IP offload engine (TOE) NIC card, and an iSCSI HBA are the three physical iSCSI connectivity options.

A standard NIC 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. However, the NIC provides no external processing power, which places additional overhead on the host CPU because it is required to perform all the TCP/IP and iSCSI processing.

If a standard NIC is used in heavy I/O load situations, the host CPU may become a bottleneck. TOE NIC help alleviate this burden. A TOE NIC offloads the TCP management functions from the host and leaves 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 initiator, requiring host CPU cycles.

An **iSCSI HBA** is capable of providing performance benefits, as it offloads the entire iSCSI and TCP/IP protocol stack from the host processor. Use of an iSCSI HBA is also the simplest way for implementing a boot from SAN environment via iSCSI. If there is no iSCSI HBA, modifications have to 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 very similar to the functionality of an FC HBA, but it is the most expensive option.
The architecture of SCSI is based on the client/server model. Slide displays a model of the iSCSI protocol layers and depicts the encapsulation order of SCSI commands for their delivery through a physical carrier. SCSI is the command protocol that works at the application layer of the 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 initiators and targets.

iSCSI is the session-layer protocol that initiates a reliable session between a device that recognizes 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 service.

TCP is used to control 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 for each hop through a separate physical network.
The topologies used to implement iSCSI can be categorized into two classes: native and bridged. Native topologies do not have any FC components; they perform all communication over IP. The initiators may be either directly attached to targets or connected using standard IP routers and switches. Bridged topologies enable the co-existence 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 SAN.
Native iSCSI Connectivity

- No FC components
- Each iSCSI port on the array is configured with an IP address and port number
  - iSCSI Initiators Connect directly to the Array

If an iSCSI-enabled array is deployed, FC components are not needed for iSCSI connectivity in the native topology. In the example shown in Figure, the array has one or more Ethernet NICs that are connected to a standard Ethernet switch and configured with an IP address and listening port. Once a client/initiator is configured with the appropriate target information, it connects to the array and requests a list of available LUNs. A single array port can service multiple hosts or initiators as long as the array can handle the amount of storage traffic that the hosts generate.

Many arrays provide more than one interface so that they can be configured in a highly available design or have multiple targets configured on the initiator. Some NAS devices are also capable of functioning as iSCSI targets, enabling file-level and block-level access to centralized storage. This offers additional storage options for environments with integrated NAS devices or environments that don’t have an iSCSI/FC bridge.
Bridged iSCSI Connectivity

- Bridge device translates iSCSI/IP to FCP
  - Standalone device
  - Integrated into FC switch (multi-protocol router)
- iSCSI initiator/host configured with bridge as target
- Bridge generates virtual FC initiator

A bridged iSCSI implementation includes FC components in its configuration. Figure illustrates an existing FC storage array used to service hosts connected through iSCSI. The array does not have any native iSCSI capabilities—that is, it does not have any Ethernet ports. Therefore, an external device, called a bridge, router, gateway, or a multi-protocol router, must be used to bridge the communication from the IP network to the FC SAN. These devices can be a stand-alone unit, or in many cases are integrated with an existing FC switch. In this configuration, the bridge device has Ethernet ports connected to the IP network, and FC ports connected to the storage. These ports are assigned IP addresses, similar to the ports on an iSCSI-enabled array. The iSCSI initiator/host is configured with the bridge’s IP address as its target destination. The bridge is also configured with an FC initiator or multiple initiators. These are called virtual initiators because there is no physical device, such as an HBA, to generate the initiator record.
Combining FCP and Native iSCSI Connectivity

- Array provides FC and iSCSI connectivity natively
- No bridge devices needed

A combination topology can also be implemented. In this case, a storage array capable of connecting the FC and iSCSI hosts without the need for external bridging devices is needed. These solutions reduce complexity, as they remove the need for configuring bridges. However, additional processing requirements are placed on the storage array because it has to accommodate the iSCSI traffic along with the standard FC traffic.
For iSCSI communication an initiator must discover the location of the target on a 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** and **internet Storage Name Service (iSNS)**.

In SendTargets discovery, the initiator is manually configured with the target’s network portal, which it uses to establish a discovery session with the iSCSI service on the target. The initiator issues the SendTargets command, and the target responds with the names and addresses of the targets available to the host. iSNS enables the 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 know the targets that it can access, it can query the iSNS server for a list of available targets.
Due to the shared nature of the connectivity to the storage target, an administrator has two options for connectivity management:

Each initiator can be individually configured with a list of its authorized targets and each target can be configured with is list of initiators and access controls. An iSNS server can be configured with a list of initiators and targets, which can be done dynamically. Additional information can also be stored such as: Discovery Domains which organize the resources into manageable groups. Discovery domains are equivalent to zone sets in FC SAN Change State Services which provide notification of storage node change of state e.g. going off line, domain membership changes, IP link status, etc. Fibre Channel and iSCSI device mappings.
iSCSI Names

• All initiators and targets require a unique iSCSI identifier

• Two types of iSCSI names
  – iqn.: iSCSI Qualified Name
    ➢ iqn.2008-02.com.example:optional_string
  – eui.: Extended Unique Identifier
    ➢ eui.0300732A32598D26

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

• iSCSI Qualified Name (IQN): An organization must own a registered domain name in order to generate iSCSI Qualified Names. This domain name does not have 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 transfer of domain names; the organization is required to have owned the domain name on that date. 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 of the storage device identifiers.

• Extended Unique Identifier (EUI): An EUI is a globally unique identifier based on the IEEE EUI-64 naming standard. An EUI comprises the eui prefix followed by a 16-character hexadecimal name, such as eui.0300732A32598D26. The 16-character part of the name includes 24 bits for the company name assigned by IEEE and 40 bits for a unique ID, such as a serial number. This allows for a more streamlined, although less user-friendly, name string because the resulting iSCSI name is simply eui followed by the hexadecimal WWN. In either format, the allowed special characters are dots, dashes, and blank spaces. The iSCSI Qualified Name enables storage administrators to assign meaningful names to storage devices, and therefore manage those devices more easily.
What is FCIP (Fibre Channel over IP)

- FCIP is an IP-based storage networking technology
- Combines advantages of Fibre Channel and IP
- Creates virtual FC links that connect devices in a different fabric
- FCIP is a distance extension solution
  - Used for data sharing over geographically dispersed SAN

Organizations are now looking for new ways to transport data throughout the enterprise, locally over the SAN as well as over longer distances, to ensure that data reaches all the users who need it. One of the best ways to achieve this goal is to interconnect geographically dispersed SANs through reliable, high-speed links. This approach involves transporting FC block data over the existing IP infrastructure used throughout the enterprise. The FCIP standard has rapidly gained acceptance as a manageable, cost effective way to blend the best of two worlds: FC block-data storage and the proven, widely deployed IP infrastructure. FCIP is a tunneling protocol that enables distributed FC SAN islands to be transparently interconnected over existing IP-based local, metropolitan, and wide-area networks. As a result, organizations now have a better way to protect, store, and move their data while leveraging investments in existing technology.
FCIP uses TCP/IP as its underlying protocol. In FCIP, the FC frames are encapsulated onto the IP payload, as shown in Figure. FCIP does not manipulate FC frames (translating FC IDs for transmission). When SAN islands are connected using FCIP, each interconnection is called an FCIP link. A successful FCIP link between two SAN islands results in a fully merged FC fabric.

An FCIP gateway router is connected to each fabric via a standard FC connection. The fabric treats these routers like layer 2 fabric switches. The other port on the router is connected to an IP network and an IP address is assigned to that port. Once IP connectivity is established, the two independent fabrics are merged into a single fabric.
Lesson Summary

Key points covered in this lesson:

- iSCSI components
- iSCSI frame structure and topologies
- iSNS operation
- iSCSI error handling and security
- Architecture of FCIP

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Lesson: Fibre Channel over Ethernet (FCoE)

Upon completion of this lesson, you will be able to:

• Discuss the FCoE and its benefits
• Describe how FCoE works
• Describe FCoE physical and logical elements
• Compare different protocol stack

These are the objectives for this lesson. Please take a moment to read them.
Fibre Channel over Ethernet (FCoE)

• A new protocol that maps Fibre Channel protocol natively over Ethernet

• Based on two new standards that are currently in active development:
  – FCoE standard, being developed by T11 Fibre Channel Interfaces Technical Committee
  – Enhanced Ethernet standard, being developed by the Ethernet IEEE Data Center Bridging Task Group
  – Both standards are expected to be ratified in 2009

• Enables the consolidation of SAN traffic and Ethernet traffic onto a common 10 Gigabit network infrastructure

Fibre Channel over Ethernet, FCoE is a proposed standard for using the Fibre Channel protocol over Ethernet networks. FCoE enables SAN traffic to be natively transported over Ethernet networks, while protecting and extending the investment that enterprises have made in storage networks. FCoE basically would enable organizations to continue to run Fibre Channel over the same wires as their data networks.

Unlike other storage networking protocols that use Ethernet, the emerging FCoE standard will utilize a new version of the Ethernet standard that makes it more reliable. The new “Enhanced” Ethernet known as Converged Enhanced Ethernet (or CEE), will protect Fibre Channel data traveling over Ethernet by eliminating frame drops, thus making it a lossless fabric. DCE (Data Center Ethernet) is the Cisco marketing term for CEE.

FCoE combined with 10 Gigabit Ethernet (10 Gbps) fabrics will grant organizations the ability to consolidate their I/O, cables and adapters while at the same time increase the utilization of their servers. It combines LAN and SAN Traffic over a single 10Gb Ethernet connection.
FCoE Benefits

- Lower capital expenditure
  - Dramatic reduction in the number of adapters, switch ports and cables required

- Reduced power and cooling requirement

- Enabler for consolidated network infrastructure
  - Potentially lower administration cost, with convergence of LAN and SAN
  - Effective sharing of high-bandwidth links

- Lower Total Cost of Ownership (TCO)

The benefits of FCoE include lower capital and operating costs, lower cooling requirements and power savings. This results in lower total cost of ownership.

FCoE enables input/output consolidation by allowing LAN and SAN traffic to converge on a single cable or link. It reduces the number of server cables, adapters and switch ports in the data center and greatly simplifies the physical infrastructure. It also reduces the administrative overhead and complexity associated with managing the data center.
Today, servers often use four, six or even eight network adapters. These adapters can be two Fibre Channel host bus adapters, plus a pair of Network Interface Cards.

FCoE allows the use of multi-function network/storage adapters called Converged Network Adapters (CNA) consolidating both network and storage traffic. With CNAs, there is no need to deploy separate hardware and cables based upon different traffic types, thereby reducing the number of server slots and switch ports, as well as power required for I/O and necessary cooling.
FCoE - Physical Elements

- Host Interface: CNA (Converged Network Adapter)
  - PCIe card on host consolidates NICs and HBAs
  - Provides a 10 Gigabit Ethernet link that carries consolidated traffic

- 10 Gbps connectivity options: Host to FCoE switch
  - Option 1: Copper-based
    - Cost effective option
  - Option 2: Standard optical

The CNA (Converged Network Adaptor) consolidates the data networking of a NIC card with the storage networking of a Fibre Channel HBA onto a single adapter. It eliminates the need of separate interface cards for FC, IP network and it consolidates the I/O onto a single 10 Gigabit Ethernet link. The internals of the first generation CNAs include separate 10 Gigabit Ethernet, Fibre Channel and Menlo ASICs. The Menlo ASIC is used to encapsulate Fibre Channel frames into Ethernet frames.

There are two connectivity options: copper based Twinax and standard optical. Copper-based Twinax cables are the cost effective option. A Twinax cable is composed of two pairs of copper cables that are covered with a shielded casing. Each end of the Twinax cable is terminated with an SFP and a GBIC.
Shown is the topology of the current infrastructure where LAN and SAN are two different networks. Storage resources are accessible using HBAs and network resources that are accessible using NICs by hosts.
Shown is the I/O consolidation with FCoE using FCoE switches and CNAs. The FCoE switch passes Fibre Channel traffic to the SAN, and the Ethernet traffic to an attached Ethernet network.

With FCoE, the cable requirements from host to FCoE switches can be substantially reduced, which in turn reduces the cooling costs, management requirements, and the overall operational cost.
Frame size is also an important factor in FCoE. A typical Fibre Channel data frame has a 2112 byte payload, a header and a CRC. A classical Ethernet frame is typically 1.5 KB or less. FCoE requires jumbo frames (2180-byte) support to prevent a Fibre Channel frame from being split into two Ethernet frames.

FCoE can not run on the existing customers production Ethernet network, like iSCSI. It requires specific Ethernet extensions and Jumbo Frame support.
**Lossless Ethernet**

- No frame drop due to congestion or buffer overflow
  - Fibre Channel manages congestion through link level, credit based flow control
  - Ethernet uses drop flow control method which is not lossless
- PAUSE capability of Ethernet is used to achieve the lossless fabric
  - Busy receive port can send the control frame to the transmit port for pause in transmission

To support Fibre Channel frames over Ethernet, no frames can be dropped throughout the entire transmission. Congestion needs to be managed in a lossless network. Fibre channel manages congestion through a link level, credit flow control based mechanism. Typical Ethernet, coupled with TCP/IP, uses a packet drop flow control mechanism to manage the traffic. Packet drop flow control is not lossless making it unacceptable for use with storage traffic.

Ethernet has a PAUSE capability so that a busy receive port can send a control frame to the transmit port requesting a pause in transmission. Use of this optional IEEE 802.3x Ethernet feature enables Fibre Channel traffic to pass on an Ethernet network in a lossless fabric.
Several protocols exist to run storage on Ethernet in the data center, most notably iSCSI and FCIP. They both require the use of TCP/IP, adding unnecessary overhead in a data center environment. FCoE bypasses TCP/IP.

FCoE would provide an efficient storage protocol for use on a consolidated Ethernet Fabric within the data center as shown.
Lesson Summary

Key points covered in this lesson:

• Basics and benefits of FCoE
• Storage infrastructure with FCoE
• FCoE Architecture and stack comparison

These are the key points covered in this lesson. Please take a moment to review them.
Module Summary

Key topics covered in this module:

- Benefits of IP SAN
- IP convergence in the SAN and its implications
- Architecture and implementation of iSCSI
- Architecture and implementation of FCIP
- Convergence of new protocol FCoE

This concludes the module. Key points covered in this module are shown here. Please take a moment to review them.
FCoE Video
Check Your Knowledge

- What is the difference between a native and bridged iSCSI implementation?

- Explain the benefits and drawbacks of using: NIC, TOE and iSCSI HBA

- Name two iSCSI discovery mechanisms

- What are two types of iSCSI names, and which one is similar to a Fibre Channel name?

- What are the physical elements of FCoE?

- Explain the benefit of using CNA card
Module 2.5 – Content Addressed Storage
The objectives for this module are shown here. Please take a moment to read them.
Lesson: CAS Overview

Upon completion of this lesson, you be able to:

• Define fixed content
• Describe traditional archival solutions and its shortcoming
• Define Content Addressed Storage (CAS)
• List benefits of CAS

The objectives for this lesson are shown here. Please take a moment to read them.
Data that does not change is referred to as fixed content. It is any informational object retained for future reference and/or business value including electronic documents and many types of newly digitized information. Unlike transactions or files, it is typically unchanged once created.
Challenges of Storing Fixed Content

• Fixed content is growing at more than 90% annually
  – Significant amount of newly created information falls into this category
  – New regulations require retention and data protection
• Often, long-term preservation is required (years-decades)
• Simultaneous multi-user online access is preferable to offline storage
• Need faster access to fixed content
• Need for location independent data, enabling technology refresh and migration
• Traditional storage methods are inadequate

Currently, fixed content data is the fastest growing sector of the data storage market. Assets such as X-rays, MRIs, broadcast content, CAD/CAM designs, surveillance video, MP3s and financial documents are just a few examples of an important class of data that is growing at over 90% annually.

• User access to fixed content is changing also. Simultaneous, rapid access is key and thus, storage cannot be offline.
• The increase in fixed content is driven by regulatory needs and by advances in technology.
• Because of this explosive growth and changes to user requirements, traditional storage technologies are inadequate.
Traditional storage solutions for Archive

- Three categories of archival solution are:
  - Online, nearline, and offline based on the means of access

- Traditional archival solution were offline
  - Traditional archival process used optical disks and tapes as media for archival
  - An archive is often stored on a Write Once Read Many (WORM) device, such as a CD-ROM

An electronic data archive is a repository for data that has fewer access requirements. It can be implemented as online, nearline, or offline based on the means of access:

- **Online archive**: the storage device is directly connected to the host to make the data immediately available. This is best suited for active archives.

- **Nearline archive**: the storage device is connected to the host and information is local, but the device must be mounted or loaded to access the information.

- **Offline archive**: the storage device is not directly connected, mounted, or loaded. Manual intervention is required to provide this service before information can be accessed.

An archive is often stored on a *write once read many (WORM)* device, such as a CD-ROM. These devices protect the original file from being overwritten. These devices are inexpensive, they involve operational, management, and maintenance overhead.
Shortcomings of Traditional Archiving Solutions

- Tape is slow, and standards are always changing
- Optical is expensive, and requires vast amounts of media
- Recovering files from tape and optical is often time consuming
- Data on tape and optical is subject to media degradation
- Both solution require sophisticated media management

CAS has emerged as an alternative to traditional archiving solutions

Archives implemented using tape devices and optical disks involve many hidden costs. The traditional archival process using optical disks and tapes is not optimized to recognize the content, so the same content could be archived several times. Additional costs are involved in offsite storage of media and media management. Tapes and optical media are also susceptible to wear and tear. Frequent changes in these device technologies lead to the overhead of converting the media into new formats to enable access and retrieval.

Traditional archive solutions store fixed content offline where you can not readily access it; for this reason CAS emerged. It simplifies fixed content storage and retrieval. With CAS, you get fast, affordable online access to your fixed content assets and benefit from lessons learned from the shortcomings of traditional solutions, such as tape and optical.

CAS has emerged as an alternative to tape and optical solutions because it overcomes many of their obvious deficiencies. CAS also meets the demand to improve data accessibility and to properly protect, dispose of, and ensure service-level agreements for archived data.
What is Content Addressed Storage (CAS)

- Object-oriented, location-independent approach to data storage
- Repository for the “Objects”
- Access mechanism to interface with repository
- Globally unique identifiers provide access to objects

In the life cycle of information, data is actively created, accessed, edited, and changed. As data ages, it becomes less likely to change and eventually becomes “fixed” but continues to be accessed by multiple applications and users. This data is called fixed content.

Traditionally, fixed content was not treated as a specialized form of data and was stored using a variety of storage media, ranging from optical disks to tapes to magnetic disks. Accumulations of fixed content throughout an organization have resulted in an unprecedented growth in the amount of data and hence introduces a challenge of managing fixed content. Furthermore, users demand assurance that stored content has not changed and require an immediate online access to fixed content. These requirements resulted in the development of Content-Addressed Storage (CAS).

CAS is an object-based system that has been purposely built for storing fixed content data. It is designed for secure online storage and retrieval of fixed content. Unlike file-level and block-level data access that use file names and the physical location of data for storage and retrieval, CAS stores user data and its attributes as separate objects. The stored object is assigned a globally unique address known as a content address (CA). This address is derived from the object’s binary representation. CAS provides an optimized and centrally managed storage solution that can support single-instance storage (SiS) to eliminate multiple copies of the same data.
The benefits of CAS include the following:

- **Content authenticity**: It assures the genuineness of stored content. This is achieved by generating a unique content address and automating the process of continuously checking and recalculating the content address for stored objects. Content authenticity is assured because the address assigned to each piece of fixed content is as unique as a fingerprint. Every time an object is read, CAS uses a hashing algorithm to recalculate the object’s content address as a validation step and compares the result to its original content address. If the object fails validation, it is rebuilt from its mirrored copy.

- **Content integrity**: Refers to the assurance that the stored content has not been altered. Use of hashing algorithm for content authenticity also ensures content integrity in CAS. If the fixed content is altered, CAS assigns a new address to the altered content, rather than overwrite the original fixed content, providing an audit trail and maintaining the fixed content in its original state. Every object in a CAS system is systematically checked in the background. Over time, every object is tested, guaranteeing content integrity even in the case of hardware failure, random error, or attempts to alter the content with malicious intent.

- **Location independence**: CAS uses a unique identifier that applications can leverage to retrieve data rather than a centralized directory, path names, or URLs. Using a content address to access fixed content makes the physical location of the data irrelevant to the application requesting the data. Therefore, the location from which the data is accessed is transparent to the application. This yields complete content mobility to applications across locations.
Lesson Summary

Key points covered in this lesson:

- CAS Definition
- Challenges of Storing Fixed Content
- Shortcomings of Traditional Archiving Solutions
- Benefits of CAS

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Lesson: CAS Architecture

Upon completion of this lesson, you will be able to:

- Describe CAS architecture
- Describe Physical and logical elements of CAS
- Describe data storage and retrieval process in CAS environment
- CAS examples

The objectives for this lesson are shown here. Please take a moment to read them.
Physical Elements of CAS

- Storage devices (CAS Based)
  - Storage node
  - Access node

- Servers (to which storage devices get connected)

- Client

The CAS architecture is shown in Figure. A client accesses the CAS-Based storage over a LAN through the server that runs the CAS API (application programming interface). The CAS API is responsible for performing functions that enable an application to store and retrieve the data.

CAS architecture is a Redundant Array of Independent Nodes (RAIN). It contains storage nodes and access nodes networked as a cluster by using a private LAN that is internal to it. The internal LAN can be reconfigured automatically to detect the configuration changes such as the addition of storage or access nodes. Clients access the CAS on a separate LAN, which is used for interconnecting clients and servers to the CAS.

The nodes are configured with low-cost, high-capacity ATA HDDs. These nodes run an operating system with special software that implements the features and functionality required in a CAS system.

When the cluster is installed, the nodes are configured with a “role” defining the functionality they provide to the cluster. A node can be configured as a storage node, an access node, or a dual-role node. Storage nodes store and protect data objects. They are sometimes referred to as back-end nodes. Access nodes provide connectivity to application servers through the customer’s LAN. They establish connectivity through a private LAN to the storage nodes in the cluster. The number of access nodes is determined by the amount of user required throughput from the cluster. If a node is configured solely as an “access node,” its disk space cannot be used to store data objects. This configuration is generally found in older installations of CAS. Storage and retrieval requests are sent to the access node via the customer’s LAN. Dual-role nodes provide both storage and access node capabilities. This node configuration is more typical than a pure access node configuration.
The process of storing and retrieving objects in CAS is explained later slides. This process requires an understanding of the following CAS terminologies:

- **Application programming interface (API):** A high-level implementation of an interface that specifies the details of how clients can make service requests. The CAS API resides on the application server and is responsible for storing and retrieving the objects in a CAS system.

- **Binary large object (BLOB):** The actual data without the descriptive information (metadata). The distinct bit sequence of user data represents the actual content of a file and is independent of the name and physical location.
**CAS Terminology (Cont.)**

- **C-Clip**
  - A package containing the user’s data and associated metadata
  - **C-Clip ID** (C-Clip handle or C-Clip reference) is the CA that the system returns to the client application

- **Content Address (CA)**
  - An identifier that uniquely addresses the content of a file and not its location. Unlike location-based addresses, content addresses are inherently stable and, once calculated, they never change and always refer to the same content

- **C-Clip Descriptor File (CDF)**
  - The additional XML file that the system creates when making a C-Clip. This file includes the content addresses for all referenced BLOBs and associated metadata

---

**Content address (CA):** An object’s address, which is created by a hash algorithm run across the binary representation of the object. While generating a CA, the hash algorithm considers all aspects of the content, returning a unique content address to the user’s application. A unique number is calculated from the sequence of bits that constitutes file content. If even a single character changes in the file, the resulting CA is different. A hash output, also called a digest, is a type of fingerprint for a variable-length data file. This output represents the file contents and is used to locate the file in a CAS system. The digest can be used to verify whether the data is authentic or has changed because of equipment failure or human intervention. When a user tries to retrieve or open a file, the server sends the CA to the CAS system with the appropriate function to read the file. The CAS system uses the CA to locate the file and passes it back to the application server.

**C-Clip:** A virtual package that contains data (BLOB) and its associated CDF. The C-Clip ID is the CA that the system returns to the client application. It is also referred as a C-Clip handle or C-Clip reference.

**C-Clip Descriptor File (CDF):** An XML file that the system creates while making a C-Clip. This file includes CAs for all referenced BLOBs and associated metadata. Metadata includes characteristics of CAS objects such as size, format, and expiration date.
Referring to Slide, the data object storage process in a CAS system is as follows:

1. End users present the data to be archived to the CAS API via an application. The application server may also interact directly with the source (e.g., an X-ray machine) that generated this fixed content.

2. The API separates the actual data (BLOB) from the metadata and the CA is calculated from the object’s binary representation.

3. The content address and metadata of the object are then inserted into the C-Clip Descriptor File (CDF). The C-clip is then transferred to and stored on the CAS system.

4. The CAS system recalculates the object’s CA as a validation step and stores the object. This is to ensure that the content of the object has not changed.

5. An acknowledgment is sent to the API after a mirrored copy of the CDF and a protected copy of the BLOB have been safely stored in the CAS system. After a data object is stored in the CAS system, the API is given a C-Clip ID and C-Clip ID is stored local to the application server.

6. Using the C-Clip ID, the application can read the data back from the CAS system.

Once an object is stored successfully, it is made available to end users for retrieval and use. The content address is usually hidden from the user. A user accesses the file stored on CAS by the same file name. It is the application server that references the CA to retrieve the stored content; this process is transparent to the user. No modification is needed to the application’s user interface to accommodate the CAS storage and retrieval process.
How CAS Stores a Data Object

1. Client presents data to API to be archived
2. Unique Content Address is calculated
3. Object is sent to Centera via Centera API over IP
4. Centera validates the Content Address and stores the object
5. Acknowledgement returned to application
6. Clip ID is retained and stored for future use

Referring to Slide, the data object storage process in a CAS system is as follows:

1. End users present the data to be archived to the CAS API via an application. The application server may also interact directly with the source (e.g., an X-ray machine) that generated this fixed content.
2. The API separates the actual data (BLOB) from the metadata and the CA is calculated from the object’s binary representation.
3. The content address and metadata of the object are then inserted into the C-Clip Descriptor File (CDF). The C-clip is then transferred to and stored on the CAS system.
4. The CAS system recalculates the object’s CA as a validation step and stores the object. This is to ensure that the content of the object has not changed.
5. An acknowledgment is sent to the API after a mirrored copy of the CDF and a protected copy of the BLOB have been safely stored in the CAS system. After a data object is stored in the CAS system, the API is given a C-Clip ID and C-Clip ID is stored local to the application server.
6. Using the C-Clip ID, the application can read the data back from the CAS system.

Once an object is stored successfully, it is made available to end users for retrieval and use. The content address is usually hidden from the user. A user accesses the file stored on CAS by the same file name. It is the application server that references the CA to retrieve the stored content; this process is transparent to the user. No modification is needed to the application’s user interface to accommodate the CAS storage and retrieval process.
The process of data retrieval from CAS follows these steps:

1. The end user or an application requests an object.
2. The application queries the local table of C-Clip IDs stored in the local storage and locates the C-Clip ID for the requested object.
3. Using the API, a retrieval request is sent along with the C-Clip ID to the CAS system.
4. The CAS system delivers the requested information to the application, which in turn delivers it to the end user.
Almost all CAS products have the same features and options. Some may be implemented differently, but the following features are an essential part of any CAS solution:

- **Integrity checking**: It ensures that the content of the file matches the digital signature (hashed output or CA). The integrity checks can be done on every read or by using a background process. If problems are identified in any of the objects the nodes automatically repair or regenerate the object.

- **Data protection**: This ensures that the content stored on the CAS system is available in the event of disk or node failure. Some CAS systems provide local replication or mirrors that copy a data object to another node in the same cluster. This decreases the total available capacity by 50 percent. Parity protection is another way to protect CAS data. It uses less capacity to store data, but takes longer to regenerate the data if corrupted. Remote replication copies data objects to a secondary storage device in a remote location. Remote replication is used as a disaster-recovery solution or for backup. Replication technologies are detailed in Chapters 13 and 14.

- **Load balancing**: Distributes data objects on multiple nodes to provide maximum throughput, availability, and capacity utilization.

- **Scalability**: Adding more nodes to the cluster without any interruption to data access and with minimum administrative overhead.

- **Self-diagnosis and repair**: Automatically enables a CAS system to automatically detect and repair corrupted objects and alert the administrator of any potential problem. These failures can be at an object level or a node level. They are transparent to the users who access the archive. CAS systems can be configured to alert remote support teams who diagnose and make repairs remotely.

- **Report generation and event notification**: Provides on-demand reporting and event notification. A command-line interface (CLI) or a graphical user interface (GUI) can produce various types of reports. Any event notification can be communicated to the administrator through syslog, SNMP, SMTP, or e-mail.

- **Fault tolerance**: Ensures data availability if a component of the CAS system fails, through the use of redundant components and data protection schemes. If remote replication of CAS is implemented, failover to the remote CAS system occurs when the primary CAS system is unavailable.

- **Audit trails**: Enable documentation of management activity and any access and disposition of data. Audit trails are mandated by compliance requirements.
A large health care center examines hundreds of patients every day and generates large volumes of medical records. Each record may be composed of one or more images that range in size from about 15 MB for standard digital X-ray images to over 1 GB for oncology studies. The patient records are stored online for a period of 60–90 days for immediate use by attending physicians. Even if a patient’s record is no longer needed for any reason, HIPAA requirements stipulate that the records should be kept in the original formats for at least seven years.

Beyond 90 days, hospitals may backup images to tape or send them to an offsite archive service for long-term retention. The cost of restoring or retrieving an image in long-term storage may be five to ten times more than leaving the image online. Long-term storage may also involve extended recovery time, ranging from hours to days. Medical image solution providers offer hospitals the capability to view medical records, such as online X-ray images, with sufficient response times and resolution to enable rapid assessments of patients. The patient records are moved to the CAS system after 60-90 days. This facilitates long-term storage and at the same time when immediate access is needed, the records are available by accessing the CAS system.
Example 2: CAS Financial Solution

- Check image size is about 25KB
- Check imaging service provider may process 50–90 million check images per month
- Checks are stored online for a period of 60 days
- Beyond 60 days data is archived

In a typical banking scenario, images of checks, each about 25 KB in size, are created and sent to archive services over an IP network. A check imaging service provider may process 50–90 million check images per month. Typically, check images are actively processed in transactional systems for about five days. For the next 60 days, check images may be requested by banks or individual consumers for verification purposes at the rate of about 0.5 percent of the total check pool, or about 250,000–450,000 requests. Beyond 60 days, access requirements drop drastically, to as few as one for every 10,000 checks. Figure illustrates the use of CAS in this scenario. The check images are stored on a CAS system, starting at day 60 and can be held there indefinitely. A typical check image archive can approach a size of 100 TB. Check imaging is one example of a financial service application that is best serviced with CAS. Customer transactions initiated by e-mail, contracts, and security transaction records may need to be kept online for 30 years; CAS is the preferred storage solution in such cases.
Lesson Summary

Key points covered in this lesson:

- CAS architecture
- Physical and logical elements of CAS
- CAS storage and retrieval process
- CAS solution examples

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Module Summary

Key points covered in this module:

• Benefits of CAS based storage strategy
• Overview of physical and logical elements of CAS
• Storing and retrieving data from CAS
• CAS application examples

This concludes the module. Key points covered in this module are shown here. Please take a moment to review them.
EMC Centera is a simple, affordable, and secure repository for information archiving. EMC Centera is the first platform designed and optimized specifically to deal with the storage and retrieval of fixed content, meeting performance, compliance, and regulatory requirements. Compared to traditional archive solutions, EMC Centera provides faster record retrieval, SiS, guaranteed content authenticity, self-healing, and support for numerous industry regulatory standards.

**Centera Architecture**

Centera has a RAIN (redundant array of independent nodes) based architecture designed for high scalability and large content storage. The nodes run a Linux operating system and CentraStar software to implement all the CAS functionality. Each node contains more than 1 TB of usable capacity and can be configured as access and/or storage nodes. EMC Centera has two 24-port 2 gigabit internal switches that provide communications for up to 16 nodes within the private LAN. Several cabinets of these nodes and switches can be connected to form an EMC Centera cluster. The data protection in a Centera star varies depending on whether it uses content protection parity (CPP) or content protection mirrored (CPM). In CPP, the data is fragmented into segments, with an additional parity segment. Each segment is on a different node, similar to a file-type RAID. If a node or a disk fails, the other nodes regenerate the missing segment on a different node. In CPM, each data object is mirrored and each mirror resides on a different node. If a node or a disk fails, the EMC Centera software automatically broadcasts to the node with the mirrored copy to regenerate another copy to a different node so that two copies are always available. Both CPP and CPM provide total protection against failure using EMC Centera’s unique self-healing functions. With the self-healing feature, if any component in the node or the entire node fails, data is regenerated to a different part of the cluster, ensuring that data is always protected.
Centera Video

Centera Video
Check Your Knowledge

• What are the key features of a CAS implementation?
• What are the benefits of a CAS Storage Strategy?
• What are 2 business applications that would benefit from CAS technology?
• What are the logical elements of a CAS system?
• How does data get stored in a CAS environment?
Module 2.6 – Storage Virtualization
Module Objective

Upon completion of this module, you will be able to:

- Identify different virtualization technologies
- Describe block-level virtualization technologies
- Describe file-level virtualization technologies
- Discuss virtual provisioning

The objectives for this module are shown here. Please take a moment to read them.
Lesson: Virtualization Overview

Upon completion of this lesson, you will be able to:

• Identify and discuss virtualization technologies

The objectives for this lesson are shown here. Please take a moment to read them.
What is Virtualization

- Virtualization is a technique of abstracting physical resources into logical view
- Increases utilization and capability of IT resource
- Simplifies resource management by pooling and sharing resources
- Significantly reduce downtime
  - Planned and unplanned
- Improved performance of IT resources

As storage networking technology matures, larger and complex implementations are becoming more common. The heterogeneous nature of storage infrastructures has further added to the complexity of managing and utilizing storage resources effectively. Specialized technologies are required to meet stringent service level agreements and to provide an adaptable infrastructure with reduced cost of management. The virtualization technologies discussed in this module provide enhanced productivity, asset utilization, and better management of the storage infrastructure.

Virtualization is the technique of masking or abstracting physical resources, which simplifies the infrastructure and accommodates the increasing pace of business and technological changes. It increases the utilization and capability of IT resources, such as servers, networks, or storage devices, beyond their physical limits. Virtualization simplifies resource management by pooling and sharing resources for maximum utilization and makes them appear as logical resources with enhanced capabilities.
As previously mentioned, virtualization has been in use for many years. Here are some examples of virtualization:

- Virtual memory
- Virtual networks
- Virtual servers
- Virtual storage
Virtual memory makes an application appear as if it has its own contiguous logical memory independent of the existing physical memory resources.

Since the beginning of the computer industry, memory has been and continues to be an expensive component of a host. It determines both the size and the number of applications that can run on a host.

With technological advancements, memory technology has changed and the cost of memory has decreased. Virtual memory managers (VMMs) have evolved, enabling multiple applications to be hosted and processed simultaneously.

In a virtual memory implementation, a memory address space is divided into contiguous blocks of fixed-size pages. A process known as paging saves inactive memory pages onto the disk and brings them back to physical memory when required. This enables efficient use of available physical memory among different processes. The space used by VMMs on the disk is known as a swap file. A swap file (also known as page file or swap space) is a portion of the hard disk that functions like physical memory (RAM) to the operating system. The operating system typically moves the least used data into the swap file so that RAM will be available for processes that are more active. Because the space allocated to the swap file is on the hard disk (which is slower than the physical memory), access to this file is slower.
Network virtualization creates virtual networks whereby each application sees its own logical network independent of the physical network. A virtual LAN (VLAN) is an example of network virtualization that provides an easy, flexible, and less expensive way to manage networks. VLANs make large networks more manageable by enabling a centralized configuration of devices located in physically diverse locations.

Consider a company in which the users of a department are separated over a metropolitan area with their resources centrally located at one office. In a typical network, each location has its own network connected to the others through routers. When network packets cross routers, latency influences network performance. With VLANs, users with similar access requirements can be grouped together into the same virtual network. This setup eliminates the need for network routing. As a result, although users are physically located at disparate locations, they appear to be at the same location accessing resources locally. In addition to improving network performance, VLANs also provide enhanced security by isolating sensitive data from the other networks and by restricting access to the resources located within the networks.

A virtual SAN/virtual fabric is a recent evolution of SAN and conceptually, functions in the same way as a VLAN.
Before Server Virtualization:

- Single operating system image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources

After Server Virtualization:

- Virtual Machines (VMs) break dependencies between operating system and hardware
- Manage operating system and application as single unit by encapsulating them into VMs
- Strong fault and security isolation
- Hardware-independent

Server virtualization enables multiple operating systems and applications to run simultaneously on different virtual machines created on the same physical server (or group of servers). Virtual machines provide a layer of abstraction between the operating system and the underlying hardware. Within a physical server, any number of virtual servers can be established; depending on hardware capabilities. Each virtual server seems like a physical machine to the operating system, although all virtual servers share the same underlying physical hardware in an isolated manner. For example, the physical memory is shared between virtual servers but the address space is not. Individual virtual servers can be restarted, upgraded, or even crashed, without affecting the other virtual servers on the same physical machine.

With changes in computing from a dedicated to a client/server model, the physical server faces resource conflict issues when two or more applications running on these servers have conflicting requirements (e.g., need different values in the same registry entry, different versions of the same DLL). These issues are further compounded with an application’s high-availability requirements. As a result, the servers are limited to serve only one application at a time, as shown in Figure. On the other hand, many applications do not take full advantage of the hardware capabilities available to them. Consequently, resources such as processors, memory, and storage remain underutilized.

Server virtualization addresses the issues that exist in a physical server environment. The virtualization layer, shown in Figure, helps to overcome resource conflicts by isolating applications running on different operating systems on the same machine. In addition, server virtualization can dynamically move the underutilized hardware resources to a location where they are needed most, improving utilization of the underlying hardware resources.
Storage virtualization is the process of presenting a logical view of the physical storage resources to a host. This logical storage appears and behaves as physical storage directly connected to the host. Throughout the evolution of storage technology, some form of storage virtualization has been implemented. Some examples of storage virtualization are host-based volume management, LUN creation, tape storage virtualization, and disk addressing (CHS to LBA).

The key benefits of storage virtualization include increased storage utilization, adding or deleting storage without affecting an application’s availability, and non-disruptive data migration (access to files and storage while migrations are in progress). Figure illustrates a virtualized storage environment. At the top are four servers, each of which has one virtual volume assigned, which is currently in use by an application. These virtual volumes are mapped to the actual storage in the arrays, as shown at the bottom of the figure. When I/O is sent to a virtual volume, it is redirected through the virtualization at the storage network layer to the mapped physical array.
Lesson Summary

Key topics covered in this lesson:

- Various forms of virtualization
  - Memory, network, server and storage virtualization

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
Lesson: Storage Virtualization Implementation

Upon completion of this lesson, you will be able to:

- Discuss SNIA virtualization taxonomy
- Describe Block-Level Virtualization technologies and implementation
- Describe File Level Virtualization technologies and implementation

The objectives for this lesson are listed here. Please take a moment to review them.
The SNIA (Storage Networking Industry Association) storage virtualization taxonomy provides a systematic classification of storage virtualization, with three levels defining what, where, and how storage can be virtualized. The first level of the storage virtualization taxonomy addresses “what” is created. It specifies the types of virtualization: block virtualization, file virtualization, disk virtualization, tape virtualization, or any other device virtualization. Block-level and file-level virtualization are the core focus areas covered later in this module.
The second level describes “where” the virtualization can take place. This requires a multilevel approach that characterizes virtualization at all three levels of the storage environment: server, storage network, and storage, as shown in Figure. An effective virtualization strategy distributes the intelligence across all three levels while centralizing the management and control functions. Data storage functions—such as RAID, caching, checksums, and hardware scanning—should remain on the array. Similarly, the host should control application-focused areas, such as clustering and application failover, and volume management of raw disks. However, path redirection, path failover, data access, and distribution or load-balancing capabilities should be moved to the switch or the network. The third level of the storage virtualization taxonomy specifies the network level virtualization methodology, in-band or out-of-band.
Storage virtualization at the network is implemented using either the in-band or the out-of-band methodology.

In an out-of-band implementation, the virtualized environment configuration is stored external to the data path. The configuration is stored on the virtualization appliance configured external to the storage network that carries the data. This configuration is also called split-path because the control and data paths are split (the control path runs through the appliance, the data path does not). This configuration enables the environment to process data at a network speed with only minimal latency added for translation of the virtual configuration to the physical storage. The data is not cached at the virtualization appliance beyond what would normally occur in a typical SAN configuration. Since the virtualization appliance is hardware-based and optimized for Fibre Channel communication, it can be scaled significantly. In addition, because the data is unaltered in an out-of-band implementation, many of the existing array features and functions can be utilized in addition to the benefits provided by virtualization.

The in-band implementation places the virtualization function in the data path, as shown in Figure. General-purpose servers or appliances handle the virtualization and function as a translation engine for the virtual configuration to the physical storage. While processing, data packets are often cached by the appliance and then forwarded to the appropriate target. An in-band implementation is software-based and data storing and forwarding through the appliance results in additional latency. It introduces a delay in the application response time because the data remains in the network for some time before being committed to disk. In terms of infrastructure, the in-band architecture increases complexity and adds a new layer of virtualization (the appliance), while limiting the ability to scale the storage infrastructure. An in-band implementation is suitable for static environments with predictable workloads.
Block-level storage virtualization provides a translation layer in the SAN, between the hosts and the storage arrays, as shown in Figure. Instead of being directed to the LUNs on the individual storage arrays, the hosts are directed to the virtualized LUNs on the virtualization device. The virtualization device translates between the virtual LUNs and the physical LUNs on the individual arrays. This facilitates the use of arrays from different vendors simultaneously, without any interoperability issues. For a host, all the arrays appear like a single target device and LUNs can be distributed or even split across multiple arrays.

Block-level storage virtualization extends storage volumes online, resolves application growth requirements, consolidates heterogeneous storage arrays, and enables transparent volume access. It also provides the advantage of non-disruptive data migration.

In traditional SAN environments, LUN migration from one array to another was an offline event because the hosts needed to be updated to reflect the new array configuration. In other instances, host CPU cycles were required to migrate data from one array to the other, especially in a multi vendor environment. With a block-level virtualization solution in place, the virtualization engine handles the back-end migration of data, which enables LUNs to remain online and accessible while data is being migrated. No physical changes are required because the host still points to the same virtual targets on the virtualization device. However, the mappings on the virtualization device should be changed. These changes can be executed dynamically and are transparent to the end user.

Deploying heterogeneous arrays in a virtualized environment facilitates an information lifecycle management (ILM) strategy, enabling significant cost and resource optimization. Low-value data can be migrated from high- to low-performance arrays or disks.
File-level virtualization addresses the NAS challenges by eliminating the dependencies between the data accessed at the file level and the location where the files are physically stored. This provides opportunities to optimize storage utilization and server consolidation and to perform non-disruptive file migrations. Figure illustrates a NAS environment before and after the implementation of file-level virtualization.

Before virtualization, each NAS device or file server is physically and logically independent. Each host knows exactly where its file-level resources are located. Underutilized storage resources and capacity problems result because files are bound to a specific file server. It is necessary to move the files from one server to another because of performance reasons or when the file server fills up. Moving files across the environment is not easy and requires downtime for the file servers. Moreover, hosts and applications need to be reconfigured with the new path, making it difficult for storage administrators to improve storage efficiency while maintaining the required service level. File-level virtualization simplifies file mobility. It provides user or application independence from the location where the files are stored. File-level virtualization creates a logical pool of storage, enabling users to use a logical path, rather than a physical path, to access files. File-level virtualization facilitates the movement of file systems across the online file servers. This means that while the files are being moved, clients can access their files non-disruptively. Clients can also read their files from the old location and write them back to the new location without realizing that the physical location has changed. Multiple clients connected to multiple servers can perform online movement of their files to optimize utilization of their resources. A global namespace can be used to map the logical path of a file to the physical path names.
Lesson Summary

Key points covered in this lesson:

• Storage virtualization configuration
• Types of storage virtualization

This concludes the lesson. Key points covered in this lesson are shown here. Please take a moment to review them.
EMC Invista provides block-level storage virtualization in heterogeneous storage environments. It also supports dynamic volume mobility for volume extension and data migration between different storage tiers without any downtime. It integrates with the existing SAN infrastructure and uses the full fabric bandwidth for high-speed I/O processing. It uses intelligent SAN switches with customized hardware to virtualize physical storage in a logical presentation. They use specialized software to examine the port, logical volume, and offset to which the I/O is sent and can control the target path of I/Os to the storage devices. The hosts and storages are connected to the Invista hardware directly or through a SAN switch. Hosts see Invista as a storage device or a virtual target, whereas storage sees Invista as a host or a virtual initiator. Invista uses virtual targets and virtual initiators to map virtual volumes to the physical storage on the arrays. Invista serves as a proxy device, intercepting communications between the host and the storage by providing virtualization.

Control path cluster (CPC), and data path controller (DPC) are the main hardware components of Invista. CPC is a customized storage device running Invista software. The CPC stores Invista configuration parameters, including storage device information, virtual volume information, the clone group, and information about the storage volumes belonging to the storage devices. It also performs all the control and management functions of the virtual storage. The DPC is a special purpose SAN switch/blade. It runs special firmware and layered software that enables the creation and management of virtual initiators and targets. The DPC receives I/O from the host initiator and controls its attributes, such as target, LUN, and offset within the logical unit. The DPC performs I/O-mapping operations and redirection for read and write operations between the hosts (front end) and the storage arrays (back end). The DPC gets its configuration from the CPC.
Rainfinity is a dedicated hardware/software solution for file-level virtualization. The Rainfinity Global File Virtualization (GFV) appliance provides an abstraction of file-based storage transparently to users. Files can be moved from one file server to another even when clients are reading and writing their data. A Rainfinity global namespace transparently maps the logical path names to the physical locations after the files have been moved. Therefore, users and applications are redirected to the new location without reconfiguring the physical path names. The management of the namespace can be accomplished by industry standard protocols and mechanisms, such as a Distributed File System (DFS), NIS, and LDAP. The Rainfinity appliance integrates into the existing IP network and acts like a layer 2 bridge between the client and the file server. This enables Rainfinity to see and process the traffic between clients and file servers with minimal modification to the existing network. Rainfinity is aware of file-sharing protocols (CIFS and NFS). This application-layer intelligence enables Rainfinity to move data from one server to another without interrupting client access.

Multiple clients (at the top of the diagram) are connected to multiple file servers (at the bottom). The file virtualization appliance (inside the black dotted line) performs this online movement of files. Let us assume the requirement to move blue file system off the yellow file server. File virtualization is triggered, and inserts itself into the I/O stream. This is accomplished by reconfiguring the virtual LANs to make the yellow file server visible only through the file virtualization service, which functions as a network bridge. As a result, the clients retain access to the files, but all that traffic now passes through the file virtualization system. Since file virtualization has control of this traffic, it can move the file system to its new location transparently to the clients. Now the file system is in the new location, but all the I/O is still going through the file virtualization system because, at this stage, it is the only system that knows the new location of the file system. The file virtualization system proceeds to update the Global Namespace. This updates the namespaces on the clients, telling them about the new location of the blue file system. As the clients learn the new location of the blue file system, their I/O goes directly to that new location, removing the file virtualization from the I/O path.
Rainfinity Video

Rainfinity Video
Lesson: Virtual Provisioning

Upon completion of this lesson, you will be able to:

• Explain Virtual Provisioning
• Describe and explain Thin vs. Traditional LUNs
• Explain the benefits of Virtual Provisioning
• Explain how to create, monitor, and manage Thin LUNs

The objectives for this module are shown here. Please take a moment to read them.
What is Virtual Provisioning

• Capacity-on-demand from a shared storage pool
  – Logical units presented to hosts have more capacity than physically allocated
  – Physical storage is allocated only when the host requires it
  – Provisioning decisions not bound by currently available storage

• Above and beyond “Thin Provisioning”
  – Includes management tools that make it easier to configure, use, monitor and manage Thin Pools and Thin LUNs

Storage perceived by the application is larger than physically allocated storage

Virtual Provisioning enables presenting a logical unit, to a host, with more capacity than is physically allocated to it on the storage array. It goes beyond what is conventionally referred to as “Thin Provisioning” in the industry, by including comprehensive management and monitoring capabilities.

Physical storage is allocated to the application “on-demand” as it is needed from a shared pool of physical capacity. This allows for more efficient allocation, and potentially even oversubscription on the storage array - where more capacity can be presented to the hosts than is actually available on the back-end.
Shown is an example comparing Virtual Provisioning (on the right) with traditional storage provisioning (on the left). The example demonstrates the benefit of better capacity utilization.

With traditional provisioning, three LUNs are created and presented to one or more hosts, using traditional provisioning methods. The total usable storage capacity of the storage system is 2 TB.

- The size of LUN 1 is 500 GB, of which 100 GB is actual data and 400 GB is allocated unused capacity.
- The size of LUN 2 is 550 GB, of which 50 GB is actual data and 500 GB is allocated unused capacity.
- The size of LUN 3 is 800 GB, of which 200 GB is actual data and 600 GB is allocated unused capacity.

In total, the storage system stores 350 GB of actual data, 1.5 TB of allocated unused capacity, and only 150 GB of available capacity for other applications.

If we consider the same 2 TB storage system with Virtual Provisioning, the differences are quite dramatic. Although the system administrator creates the same size LUNs for LUNs 1, 2, and 3, there is no allocated unused capacity. In total, the storage system with Virtual Provisioning stores the same 350 GB of actual data as the other storage system, with 1.65 TB of capacity available for other applications, versus only 150 GB in the traditional storage system.
Virtual Provisioning – Benefits

- Reduce administrative costs
  - Simplifies storage provisioning
  - Over-provisioning can eliminate challenges of expansion
  - Reduces time required to repeatedly add storage capacity

- Reduce storage costs
  - Increased space efficiency for primary storage and replicas
  - “Storage on demand” from shared storage pool
  - Deploy assets as needed
  - Reduce levels of unused physical storage
  - Avoid pre-allocating physical storage to applications

- Reduce operating costs
  - Fewer disks consume less power, cooling and floor space

- Reduce downtime
  - Less disruptive to applications

Virtual Provisioning reduces administrative costs by simplifying storage provisioning and eliminating challenges of expansion. It reduces the time required to repeatedly add storage capacity.

Storage costs are reduced through increased space efficiency for primary storage and replicas. “Storage on demand” from a shared storage pool allow assets to be deployed as needed. Levels of unused physical storage are reduced and pre-allocation of physical storage to applications is avoided.

Operating costs are reduced with fewer disks consuming less power, cooling and floor space.

Virtual Provisioning is less disruptive to applications. Administrators do not have to continually take applications off-line to increase the size of file systems.
Virtual Provisioning – Thin Pool Expansion

• Adding drives to the pool non-disruptively increases available shared capacity for all Thin LUNs in pool
  – Drives can be added to a Thin Pool while pool is being used in production

• Allocated capacity is reclaimed by the pool when Thin LUNs are deleted

Adding drives to a Thin Pool non-disruptively increases available shared capacity for all the Thin LUNs in the pool. Drives can be added to a Thin Pool while the pool is being used in production. Allocated capacity is reclaimed by the pool when Thin LUNs are deleted. There is no need to defrag.
Traditional vs. Thin LUNs

<table>
<thead>
<tr>
<th>Use RAID Groups and traditional LUNs</th>
<th>Use Virtual Provisioning with Thin Pools and Thin LUNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>• When microseconds of performance matters</td>
<td>• When the best space efficiency is needed</td>
</tr>
<tr>
<td>• For the best and most predictable performance</td>
<td>• For minimal host impact</td>
</tr>
<tr>
<td>• For precise data placement</td>
<td>• When energy and capital savings are paramount</td>
</tr>
<tr>
<td>• You are not as concerned about space efficiency</td>
<td>• For applications where space consumption is difficult to forecast</td>
</tr>
</tbody>
</table>

The key is to understand your application requirements and select the approach that meets your needs. If conditions change, you have the option of using LUN migration to migrate from a Thin LUN to a traditional LUN. It is important to remember that Thin Pools and Thin LUNs offer many benefits, but they require careful monitoring and are not suited to every application.
Lesson Summary

Key points covered in this module:

- Virtual Provisioning
- Thin vs. Traditional LUNs
- Benefits of Virtual Provisioning

These are the key points covered in this lesson. Please take a moment to review them.
Module Summary

Key points covered in this module:

- Virtualization technologies and forms
- SNIA storage virtualization taxonomy
- Storage virtualization configuration
- Types of storage virtualization
- Virtual provisioning overview

These are the key points covered in this module. Please take a moment to review them.
Check Your Knowledge

• What are the four forms of virtualization?

• Difference between in-band and out-of-band implementation.

• What is virtual provisioning?
Cloud Storage Infrastructure
Cloud Computing Video

Cloud Computing Video
Traditional storage technologies were not designed to scale in the multi-petabyte Internet Era. With traditional storage architectures, new arrays need to be added as capacity requirements increase. As the numbers of arrays under management grows, the storage environment becomes increasingly complex, harder to manage, and more costly to operate. This brings negative consequences to the business: increased time to market, loss of productivity, and decreased flexibility. Traditional storage technologies continue to excel in the areas they were designed to address—namely, transactional and distributed computing—but these solutions fall short for Internet Era requirements.
To deal with Internet Era data growth, a massively scalable infrastructure is required—one that offers global data distribution, self-healing, self-management, and multi-tenancy features. The storage infrastructure needs to store files or objects and be accessible via multiple protocols. It must have rich metadata to tag content and use those data tags to apply policies, improve searches, or build custom queries. It must have policies that can leverage metadata to determine where and how the system stores data so it can meet information performance and availability requirements. For example it must provide information such as how data is protected, for how long, with how many copies, and where it should be stored geographically to best meet requirements. It should also include data services to handle policy, replication, versioning, compression, de-duplication, and spin-down.

**Infinite Scale** – Essentially, a cloud should remove the limitations of individual devices, thereby removing the boundaries of the boxes and enabling efficient single-level management of multiple locations. A unified namespace enables cloud storage to scale massively while still operating as a single entity.

**No Boundaries** – The global distribution capabilities allow for policies to match content popularity, geographic location, and desired retention periods. In other words, the system needs to be able to respond to the demands of Internet Era information—by scaling quickly and having the high bandwidth required to handle large objects with high levels of concurrent data access. **Operational Efficiency** – Cloud storage systems must leverage commodity components, policy-based automation, and secure multi-tenancy to realize economies of scale. Multi-tenancy allows enterprises or service providers to support multiple applications from within the same infrastructure and, in essence, fence them off from one another.

**Self-Management** – With a unified namespace, policy-based data services, spin-down, compression, replication and other services should be native to the cloud—rather than purchased, licensed, managed, and maintained separately—to produce a solution that is fully integrated rather than bolted together.

**Self-Healing** – Cloud storage needs to protect data, not disks. Data needs to be protected based on policy.
Use of Cloud Computing Resources

- “Cloud computing” takes hold as 69% of all internet users have either stored data online or used a web-based software application

### Cloud Computing Activities

<table>
<thead>
<tr>
<th>Why people use “cloud” applications</th>
<th>% of those who use online applications and services to store data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Major reason</td>
</tr>
<tr>
<td>It is just easy and convenient.</td>
<td>51%</td>
</tr>
<tr>
<td>I can access this information from whatever computer I am using.</td>
<td>41%</td>
</tr>
<tr>
<td>I can easily share information with others.</td>
<td>39%</td>
</tr>
<tr>
<td>I won’t lose this information if my computer fails.</td>
<td>34%</td>
</tr>
</tbody>
</table>

Defining Cloud Computing

“Cloud Computing is an emerging IT development, deployment and delivery model, enabling real-time delivery of products, services and solutions over the Internet (i.e. enabling cloud services)”

- Services include
  - Software-as-a-Service (SaaS)
  - Platform-as-a-Service (Paas)
  - Infrastructure-as-a-Service (IaaS)

Examples:
  - Amazon: Elastic Compute Cloud (EC2), Simple Storage Services (S3)
  - Google Apps
  - Storage Cloud - Decho (Mozy Online Backup), EMC Atmos
  - Salesforce.com…..

Cloud computing refers to the architecture that is scalable in real-time, offering resources as a service to its users who need not have knowledge of, expertise in, or control over the supporting infrastructure, technology. In cloud computing offerings applications reside in data centers, where compute resources can be dynamically provisioned and shared to achieve significant economies of scale.

Customers are not explicitly buying “cloud computing”, but the “cloud services” that are enabled by cloud computing environments. This abstraction alleviates IT consumers from the operations of applications, allowing end users to focus instead on the execution and high-value activities.

In short, a cloud service is virtually any business or consumer service that is delivered and consumed over the Internet in real-time.
Followings are the key attributes of Cloud Services.

**Offsite, provided by third-party provider** - “In the cloud” execution, which for most practical purposes means offsite (really, location-agnostic). Specifying “third-party provider” simply highlights that the services we’re focused on in our analysis are *commercial* cloud services.

**Accessed via the Internet** - standards-based, universal network access. This does not preclude service providers offering security or quality-of-service value-added options.

**Minimal/no IT skills to “implement”** - online, simplified specification of services requirements; need is eliminated for lengthy implementation services for on-premise systems that support the service (the service provider offloads this).

**Provisioning** - self-service requesting, near real-time deployment, dynamic & fine-grained scaling.

**Pricing** - fine-grained & usage-based pricing capability. (As a convenience to some customers, providers may *mask* this pricing granularity with long-term, fixed price agreements.)

**User Interface** - browser & successors. Browsers will evolve, for a wider variety of devices, and with richer capabilities. But the basic aspects of a browser - intuitive/easy-to-use, standards-based, application/service-independent, multi-platform - remain the attributes of cloud services UIs.

**System Interface** - Web services APIs, providing a standards-based framework for accessing and integrating with and among cloud services (*and* web services-based/enabled in-house systems). In our view, this is a critically important aspect of cloud services: that they provide well-defined, programmatic access for users, partners and others who want to leverage the cloud service within a broader solution context.

**Shared resources** - the shared asset approach improves supplier and customer economics; there is some ability to customize “around” the shared services, via configuration options within the service, workflow/process management among services, et al.
**Cloud Applications**

- **Enterprise Solutions**
  - Transactional data or high performance file sharing applications
    - Example: Amazon EC2
  - Cloud storage infrastructure
    - Example: EMC Atmos

- **End-user Solutions**
  - Rich Internet applications and online service providers
    - Examples: Social media sites, Online photo sharing
  - Online data backup
    - Example: Mozy online backup
Concept in Practice – Atmos Video

Atmos Video
1. Which logical element of CAS is a set of function calls that enable communication between applications, or between an application and an operating system?
   a. Application Programming Interface (API)
   b. Content Address (CA)
   c. Metadata
   d. Server

2. Which two identifiers represent a member of a hard zoning?
   a. Domain ID and Switch port number
   b. HBA WWN and Switch port number
   c. Switch security ID and Storage port number
   d. HBA WWN and Switch WWN

3. What term is used to describe the virtual space in which all storage nodes communicate with each other over distances?
   a. Fabric
   b. Fibre Channel
   c. Loop
   d. Port

4. Which block storage over IP protocol requires a pair of bridges that uses IP as the transport protocol?
   a. FCoE
   b. FCIP
   c. iSCSI
   d. iFCP

5. Which IP SAN topology allows Initiators to exist in an Ethernet environment while the storage remains in a Fibre Channel SAN?
   a. Extension
   b. Native
   c. Reduction
   d. Bridging

6. Which term is used to describe a device that is connected to the SAN for purposes of requesting or supplying data?
   a. Fiber
   b. Host Bus Adapter
   c. Node
   d. Ports

7. Why is multimode fiber (MMF) transmission typically used for relatively short distances?
   a. Optical fiber cables are only available in short sizes due to manufacturing constraints
   b. Cost rises with the distance
   c. Optical signal strength tends to degrade over greater distances due to modal dispersion
   d. Multiple beams of light does not disperse easily over a long distance
8. What best describe external DAS?
   a. Scalable Connectivity
   b. Less protocol overhead
   c. Ideal for file level access
   d. Low complexity

9. Which protocol is used to connect and manage physical disk storage resources in a NAS device?
   a. DART
   b. NFS and CIFS
   c. Fibre Channel
   d. IP

10. What is an example of virtual storage?
    a. Disk Drive
    b. Cache
    c. Swap files
    d. RAM

11. In the context of virtual memory, what is a swap file?
    a. Reserved address space in memory
    b. High speed memory for critical applications
    c. A portion of a physical disk
    d. RAM

12. What is a function of block-level storage virtualization?
    a. To abstract multiple physical networks as one logical network
    b. To abstract multiple operating systems as one operating system
    c. To abstract multiple physical arrays as a single target device
    d. To abstract multiple file servers as one file server

13. What is a key benefit of virtual provisioning?
    a. Reduced storage costs
    b. Reduced network traffic
    c. Reduced storage traffic
    d. Support ILM strategy

14. A customer wants to migrate their existing DAS to a SAN environment. Where is the most cost reduction achieved with the new implementation?
    a. Storage subsystem
    b. People
    c. Backup media
    d. Software

15. What enables a logical unit to be presented to a host with more capacity than is physically allocated to it on the storage array?
    a. Server Virtualization
    b. Thin Provisioning
    c. Storage Virtualization
    d. File Virtualization
16. What is the size of payload in a Fibre Channel frame?
   a. 1500 bytes
   b. 2000 bytes
   c. 2112 bytes
   d. 9000 bytes

17. What are the features of a content addressable storage (CAS) system?
   a. Automatic zone allocation, faster record retrieval, and self-configuring
   b. Faster record retrieval, self-healing, and better read performance than a NAS
      OLTP application
   c. Faster record retrieval, self-healing, and record-level retention and protection
   d. Self-configuring, better read performance than a NAS OLTP application, and
      record-level retention and protection

18. Which characteristic defines typical archival data?
   a. Final data for system recovery
   b. Intermediate data for system recovery
   c. Primary data that is accessed less frequently
   d. Secondary copy of data that is accessed less frequently

19. Which login process is related to Upper Layer Protocols such as SCSI?
   a. Fabric login (FLOGI)
   b. Node login (NLOGI)
   c. Port login (PLOGI)
   d. Process login (PRLI)

20. What represents a unique iSCSI identifier assigned to iSCSI initiators and targets?
   a. FQDN
   b. IQN
   c. WWPN
   d. WWNN