CMR

INSTITUTE OF

TECHNOLOGY

Internal Assessment Test 1 – Sept. 2016 Scheme and solution

Sub:	STORAGE AREA NETWORKS							Code:	10IS765
Date:	8/ 09 / 2016	Duration:	90 mins	Max Marks:	50	Sem:	VII A & B	Branch:	ISE/CSE

Note: ***Answer any five questions.***

1. a.Explain the core elements of the Data Center Infrastructure? (6m)

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 computers, 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.

1.3.1 Core Elements

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

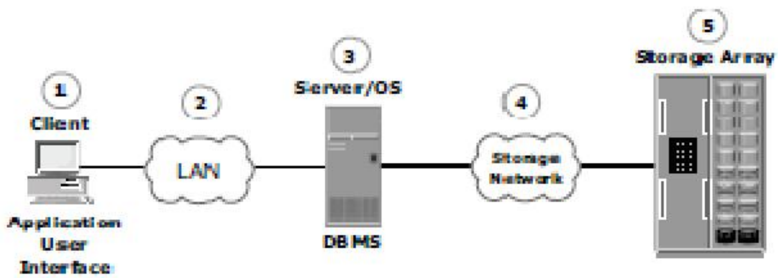
- **Application:** An application is a computer program that provides the logic for computing operations. Applications, such as an order processing system, can be layered on a database, which in turn uses operating system services to perform read/write operations to storage devices.

2.

- **Database:** More commonly, a database management system (DBMS) provides a structured way to store data in logically organized tables that are interrelated. A DBMS optimizes the storage and retrieval of data.
- **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.

Figure 1-5 shows an example of an order processing system that involves the five core elements of a data center and illustrates their functionality in a business process.



(1) A customer places an order through the AUI of the order processing application software located on the client computer.

(2) The client connects to the server over the LAN and accesses the DBMS located on the server to update the relevant information such as the customer name, address, payment method, products ordered, and quantity ordered.

(3) The DBMS uses the server operating system to read and write the data to the database located on physical disks in the storage array.

(4) The Storage Network provides the communication link between the server and the storage array and transports the read or write commands between them.

(5) The storage array, after receiving the read or write commands from the server, performs the necessary operations to store the data on physical disks.

1b Define Seek time and rotational latency(4m)

Seek Time

The *seek time* (also called *access time*) describes the time taken to position the R/W heads across the platter with a radial movement (moving along the radius of the platter). In other words, it is the time taken to 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.

Rotational Latency

To access data, the actuator arm moves the R/W head over the platter to a particular track while the platter spins to position the requested sector under the R/W head. The time taken by the platter to rotate and position the data under the R/W head is called *rotational latency*. This latency depends on the rotation speed of the spindle and is measured in milliseconds. The average rotational latency is one-half of the time taken for a full rotation. Similar to the seek time, rotational latency has more impact on the reading/writing of random sectors on the disk than on the same operations on adjacent sectors.

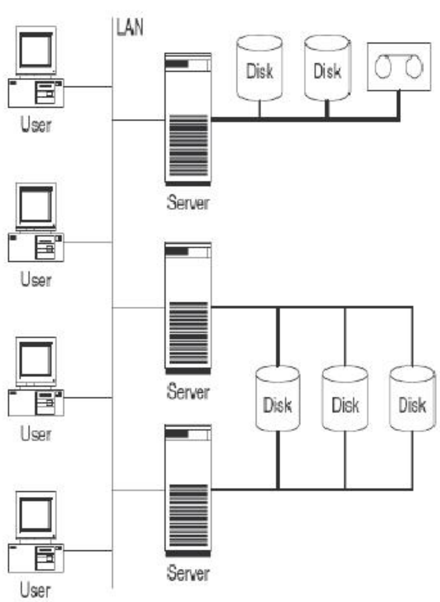
Average rotational latency is around 5.5 ms for a 5,400-rpm drive, and around 2.0 ms for a 15,000-rpm drive.

2. Explain server centric IT architecture and storage centric IT architecture with advantages and limitations.(10m)

In conventional IT architectures, storage devices are normally only connected to a single server. To increase faulttolerance, storage devices are sometimes connected to two servers, with only one server actually able to use the storage device at any one time. In both cases, the storage device exists only in relation to the server to which it is connected. Other servers cannot directly access the data; they always have to go through the server that is connected to the storage device. This conventional IT architecture is therefore called server-centric IT architecture. In this approach, servers and storage devices are generally connected together by SCSI cables. In conventional server-centric IT architecture storage devices exist only in relation to the one or two servers to which they are connected. The failure of both of these computers would make it impossible to access this data. Most companies find this unacceptable. At least some of the company data (for example, patient files, and websites) must be available around the clock.
Figure 1.1 In a server-centric IT architecture storage devices exist only in relation to servers
Figure 1.2 The storage capacity on server 2 is full. It cannot make use of storage space free on server 1 and server 3

Although the storage density of hard disks and tapes is increasing all the time due to ongoing technical development, the need for installed storage is increasing even faster. It is necessary to connect ever more storage devices to a computer. But each computer can accommodate only a limited number of I/O cards (for example, SCSI cards). Furthermore, the length of SCSI cables is limited to a maximum of 25 m. This means that the storage capacity that can be connected to a computer using conventional technologies is limited. In server-centric IT environments the storage device is statically assigned to the computer to which it is connected. In general, a computer cannot access storage devices that are connected to a different computer. This means that if a computer requires more storage space than is connected to it, it cannot still use the free space present in another computer. Storage devices are often scattered throughout an entire building or branch. Computers may be consciously set up where the user accesses the data in order to reduce LAN data traffic. The result is that the storage devices are distributed throughout many rooms, which are neither protected against unauthorized access nor sufficiently air conditioned

Storage networks can solve the problems of server-centric IT architecture. Storage networks open up new possibilities for data management. The idea behind storage networks is that the SCSI cable is replaced by a network that is installed in addition to the existing LAN and is primarily used for data exchange between computers and storage devices. In contrast to server-centric IT architecture, in storage networks storage devices exist completely independently of any computer. Several servers can access the same storage device directly over the storage network without another server having to be involved. Storage devices are thus placed at the centre of the IT architecture. IT architectures with storage networks are therefore known as storage-centric IT architectures. When a storage network is introduced, the storage devices are usually also consolidated. This involves replacing the many small hard disks attached to the computers with a large disk subsystem. Disk subsystems currently have a maximum storage capacity of several ten terabytes. The storage network permits all computers to access the disk subsystem and share it. Free storage capacity can thus be flexibly assigned to the computer that needs it at the time. In the same manner, many small tape libraries can be replaced by one big one.



ure 1.1 In a server-centric IT architecture storage devices exist only in relation to servers

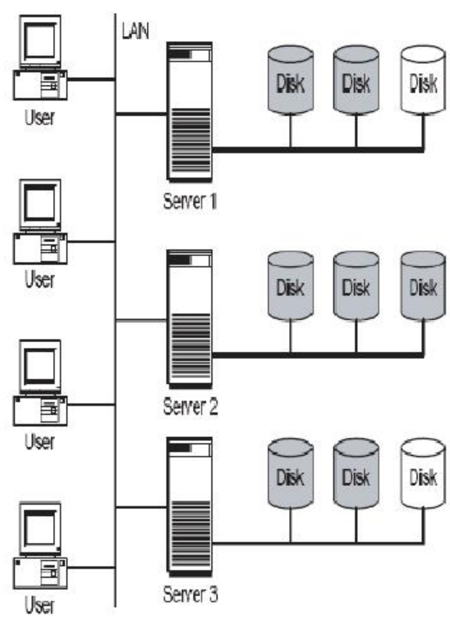
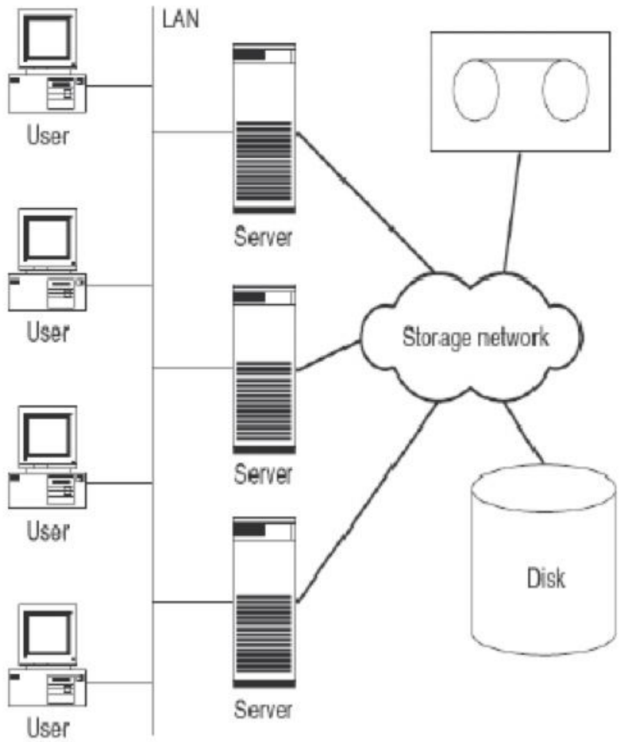


Figure 1.2 The storage capacity on server 2 is full. It cannot make use of storage space free on server 1 and server 3



In storage-centric IT architecture the SCSI cables are replaced by a work. Storage devices now exist independently of a server

3. Define ILM.Explain in detail. (10m)

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 (see Figure 1-7). The value of the information is highest when a company receives a new sales order and processes it to deliver the product. After order fulfillment, the customer or order data need not be available for real-time access. The company can transfer this data to less expensive secondary storage with lower accessibility and availability requirements unless or until a warranty claim or another event triggers its need. After the warranty becomes void, the company can archive or dispose of data to create space for other high-value information.

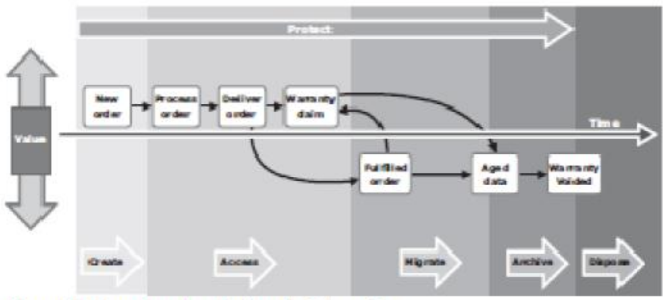


Figure 1-7: Changing value of sales order information

1.5.1 Information Lifecycle Management

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.

1.5.2 ILM Implementation

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. Figure 1-8 illustrates a three-step road map to enterprise-wide ILM.

Steps 1 and 2 are aimed at implementing ILM in a limited way across a few enterprise-critical applications. In Step 1, the goal is to implement a storage networking environment. Storage architectures offer varying levels of protection and performance and this acts as a foundation for future policy-based information management in Steps 2 and 3. 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.

Step 2 takes ILM to the next level, with detailed application or 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.

Step 3 of the implementation is to automate more of the applications or data classification and policy management activities in order to scale to a wider set of enterprise applications.

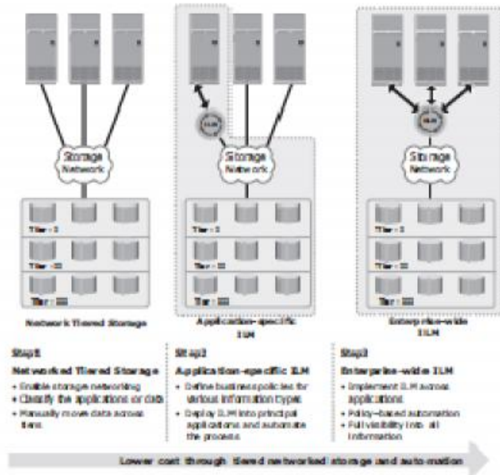


Figure 1-8: Implementation of ILM

1.5.3 ILM Benefits

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.

4.a.Write a note on volume manager(4m)

2.5.3 Volume Manager

In the early days, an HDD appeared to the operating system as a number of continuous disk blocks. The entire HDD would be allocated for the file system or other data entity used by the operating system or application. The disadvantage was lack of flexibility: As an HDD ran out of space, there was no easy way to extend the file system's size. As the storage capacity of the HDD increased, allocating the entire HDD for the file system often resulted in underutilization of storage capacity.

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)* (see Figure 2-11). 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 (see Figure 2-11).

The evolution of *Logical Volume Managers (LVMs)* enabled the dynamic extension of file system capacity and efficient storage management. LVM is software that runs on the host computer and manages the logical and physical storage. LVM is an optional, intermediate layer between the file system and the physical disk. It can aggregate several smaller disks to form a larger virtual disk or to partition a larger-capacity disk into virtual, smaller-capacity disks, which are then presented to applications. The LVM provides optimized storage access and simplifies storage resource management. It hides details about the physical disk and the location of data on the disk; and it enables administrators to change the storage allocation without changing the hardware, even when the application is running.

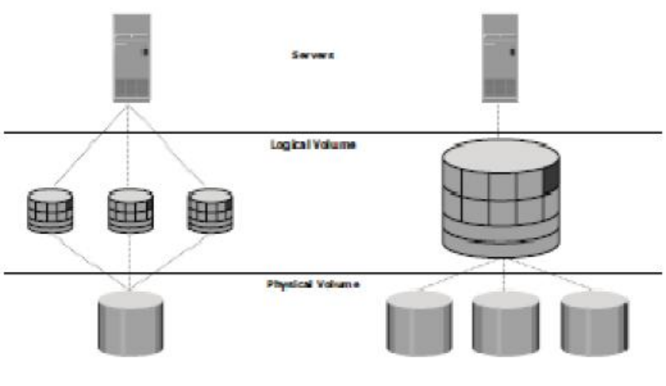


Figure 2-11 : Disk partitioning and concatenation

The basic LVM components are the *physical volumes*, *volume groups*, and *logical volumes*. In LVM terminology, each physical disk connected to the host system is a *physical volume (PV)*. LVM converts the physical storage provided by the physical volumes to a logical view of storage, which is then used by the operating system and applications. A *volume group* is created by grouping together one or more physical volumes. A unique *physical volume identifier (PVID)* is assigned to each physical volume when it is initialized for use by the LVM. Physical volumes can be added or removed from a volume group dynamically. They cannot be shared between volume groups; the entire physical volume becomes part of a volume group. Each physical volume is partitioned into equal-sized data blocks called *physical extents* when the volume group is created.

Logical volumes are created within a given volume group. A logical volume can be thought of as a virtual disk partition, while the volume group itself can be thought of as a disk. A volume group can have a number of logical volumes. The size of a logical volume is based on a multiple of the physical extents.

The logical volume appears as a physical device to the operating system. A logical volume can be made up of noncontiguous physical partitions and can span multiple physical volumes. A file system can be created on a logical volume; and logical volumes can be configured for optimal performance to the application and can be mirrored to provide enhanced data availability.

4 b.An application specifies a requirement of 500 GB to host a database & other files. It also specifies that storage environment should support 2000 IOPS during its peak overloads. The disks available for configuration provides 60GB of usable capacity & manufacturer specifies that they can support a maximum of 110 IOPS .The application is response time sensitive & disk utilization beyond 60% does not meet the response time requirement. Compute & Explain(6m)

User specification of storage capacity = 500 GB

Vendor specification of storage capacity = 60 GB

user specification of disk load = 2000 IOPS

Vendor specification of diskload = 110 IOPS

disk utilization = 60%

$D_R = Max(D_C, D_I)$

D_C [Disks needed for capacity]

$= \frac{\text{user specification}}{\text{vendor specification}} = \frac{500}{60} = 8.3$

D_C = 9 disks

D_I [Disks needed for IOPS]

$= \frac{\text{user specification}}{\text{Vendor spec} \times \text{utilization}} = \frac{2000}{110 \times 0.6}$

$= 30.2 = 31$ disks

$D_R = Max(D_C, D_I) = 31$ disks

5.a.Compare virtual Storage provisioning with traditional storage provisioning.(6m)

Feature	**Thick LUN**	**Thin LUN**	**RAID Group LUN**
Pool RAID types	RAID 6, RAID 5, or RAID 1/O.	RAID 6, RAID 5, or RAID 1/O.	RAID 6, RAID 5, RAID 3, RAID 1/0, or RAID 1 RAID groups, individual disk, or hot spare.

LUN expansion	Fully supported.	Fully supported.	Only supported using metaLUNs.
LUN shrinking	Fully supported for Windows Server 2008 hosts.	Fully supported for Windows Server 2008 hosts.	Fully supported for Windows Server 2008 hosts connected to a storage system running R30 or later for pool LUNs (thin and thick) and R29 for RAID group LUNs.
LUN compression	Fully supported if the Compression enabler is installed.	Fully supported if the Compression enabler is installed.	Supported if the Compression enabler is installed, but migrates the RAID group LUN to a pool LUN.
LUN migration	Fully supported.	Fully supported.	Fully supported.
Disk usages	Any type of disk, including Flash (SSD) disks, can be in the pool with the thick LUN. The disks in the pool with the thick LUN cannot be the vault disks 000–004.	Any type of disk, including Flash (SSD) disks, can be in the pool with the thin LUN. The disks in the pool with the thin LUN cannot be the vault disks 000–004.	All disks in the RAID group with the LUN must be of the same type.
Space efficiency	When you create a thick LUN, the LUN is assigned physical space on the pool equal to the LUN's size. This space is always available to the LUN even if it does not actually use the space.	When you create a thin LUN, a minimum of 2 GB of space on the pool is reserved for the thin LUN. Space is assigned to the thin LUN on an as-needed basis. Since the thin LUNs compete for the pool's space, a pool can run out of space for its thin LUNs. Such an event is an unrecoverable write error and data from the last write operation will not be available. Some applications, such as VERITAS Storage Foundation, may return space no longer needed to the pool by an appropriate signaling to the storage system.	When you create a LUN, the LUN is assigned physical space on the RAID group equal to the LUN's size. This space is always available to the LUN even if it does not actually use the space.
Performance	Thick LUN performance is comparable to the performance of a RAID group LUN and is typically faster than the performance of a thin LUN.	Thin LUN performance is typically slower than thick LUN performance.	RAID group LUN performance is comparable to the performance of a thick LUN and is typically faster than thin LUN performance.
Manual administration	Pools require less manual administration than RAID groups.	Pools require less manual administration than RAID groups.	RAID groups require more manual administration than pools.
Use with SnapView	Fully supported for thick LUNs. A thick LUN cannot be a clone private LUN.	A thin LUN can be a snapshot source LUN, a clone LUN, a clone source LUN, but not a clone private LUN, and not in the reserved LUN pool. A thin LUN cannot be a clone private LUN.	Fully supported for RAID group LUNs.
Use with MirrorView/A	Fully supported for thick LUNs. A thick LUN cannot	Mirroring with thin LUNs as primary or secondary images is supported only between storage systems	Fully supported for RAID group LUNs.

or MirrorView/S	be used in the write intent log.	running FLARE 04.30 or later. For mirroring between storage systems running FLARE 04.30, the primary image, secondary image, or both images can be thin LUNs. A thin LUN cannot be used in the write intent log.	
Use with SAN Copy	Fully supported for thick LUNs in all configurations.	Thin LUNs are supported only for SAN Copy sessions in the following configurations:1)Within a storage system running FLARE 04.30 or later, 2)Between systems running FLARE 04.30 or later The source LUN must be on the storage system that owns the SAN Copy session.	Fully supported for RAID group LUNs in all configurations.

b. Consider an application that generates 6000 IOPS with 70% of them being reads calculate the disk load in RAID 1 & RAID 5(4m)

RAID 1 – Write penalty = 2

Disk load RAID 1 $= 0.70 \times 6000 + 2 \times 0.30 \times 6000$

$= 7800$ IOPS

RAID 5 – Write penalty = 4

Disk load RAID 5 $= 0.70 \times 6000 + 4 \times 0.30 \times 6000$

$= 11400$ IOPS //

6. a.DAS benefits and Limitations-Write in detail.(6m)

5.1.1 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.

5.1.2 External DAS

In *external DAS* architectures, the server connects directly to the external storage device (see Figure 5-1). 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.

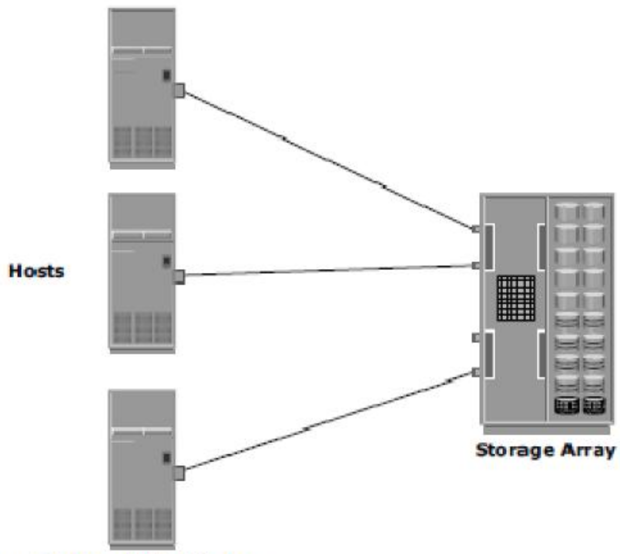


Figure 5-1: External DAS architecture

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.

However, 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. The distance limitations associated with implementing DAS because of direct connectivity requirements can be addressed by using Fibre Channel connectivity. 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.

b.Give a Brief note on Hotspares (4m)

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.

7. a. Differentiate Write back and write through cache with neat diagrams(6m)

Write operations with cache provide performance advantages over writing directly to disks. When an I/O is written to cache and acknowledged, it is completed in far less time (from the host's perspective) than it would take to write directly to disk. Sequential writes also offer opportunities for optimization because many smaller writes can be coalesced for larger transfers to disk drives with the use of cache.

A write operation with cache is implemented in the following ways:

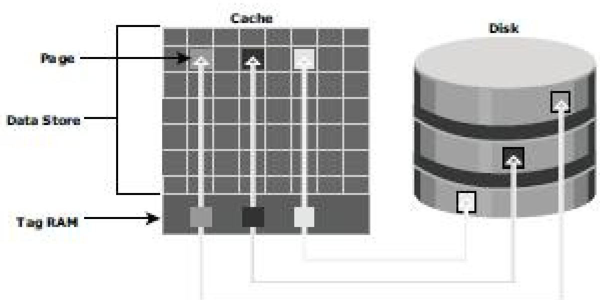
- **Write-back cache:** Data is placed in cache and an acknowledgment is sent to the host immediately. Later, data from several writes are committed (de-staged) to the disk. Write response times are much faster, as the write operations are isolated from the mechanical delays of the disk. However, uncommitted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

Cache can be bypassed under certain conditions, such as 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.

b.Mention the role of tag RAM in cache(4m)

Cache is organized into pages or slots, which is the smallest unit of cache allocation. The size of a cache page is configured according to the application I/O size. Cache consists of the *data store* and *tag RAM*. The data store holds the data while tag RAM tracks the location of the data in the data store (see Figure 4-3) and in disk.

Entries in tag RAM indicate where data is found in cache and where the data belongs on the disk. Tag RAM includes a *dirty bit* flag, which indicates whether the data in cache has been committed to the disk or not. It also contains time-based information, such as the time of last access, which is used to identify cached information that has not been accessed for a long period and may be freed up.



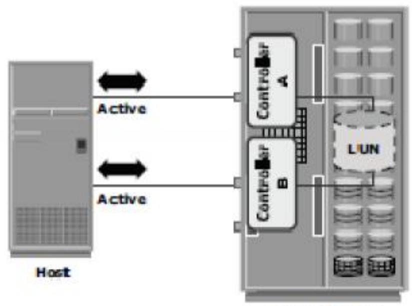
8. a. Compare the RAID levels(5m)

Table 3-2: Comparison of Different RAID Types

RAID	MIN. DISKS	STORAGE EFFICIENCY %	COST	READ PERFORMANCE	WRITE PERFORMANCE	WRITE PENALTY
0	2	100	Low	Very good for both random and sequential read	Very good	No
1	2	50	High	Good. Better than a single disk.	Good. Slower than a single disk, as every write must be committed to all disks.	Moderate
3	3	(n-1)*100/n where n= number of disks	Moderate	Good for random reads and very good for sequential reads.	Poor to fair for small random writes. Good for large, sequential writes.	High
4	3	(n-1)*100/n where n= number of disks	Moderate	Very good for random reads. Good to very good for sequential writes.	Poor to fair for random writes. Fair to good for sequential writes.	High

b. High-end and midrange storage array-Write in detail.(5m)

High-end storage systems, referred to as *active-active arrays*, are generally aimed at large enterprises for centralizing corporate data. These arrays are designed with a large number of controllers and cache memory. An active-active array implies that the host can perform I/Os to its LUNs across any of the available paths (see Figure 4-7).



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 functionals that are required for mission-critical applications in large enterprises.

4.2.2 Midrange Storage System

Midrange storage systems are also referred to as *active-passive arrays* and they are best suited for small- and medium-sized enterprises. 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 4-8, 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.

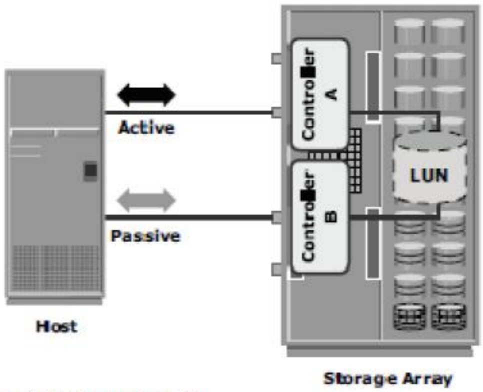


Figure 4-8: Active-passive configuration

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.