USN | | | | | | | | | | |

Internal Assessment Test I – May 2022 scheme

Sub :	Storage Area Network				Sub Code:	18CS822	Branch :	ISE	
Date :	14-05-2022	Duration:	90 min's	Max Marks:	50	Sem/Sec :	VIII A, B & C		OBE
Answer any FIVE FULL Questions							MARKS	CO	RBT

1 Draw a neat diagram of Disk Drive and explain its components. 10 CO1 L2

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.

1. Platter

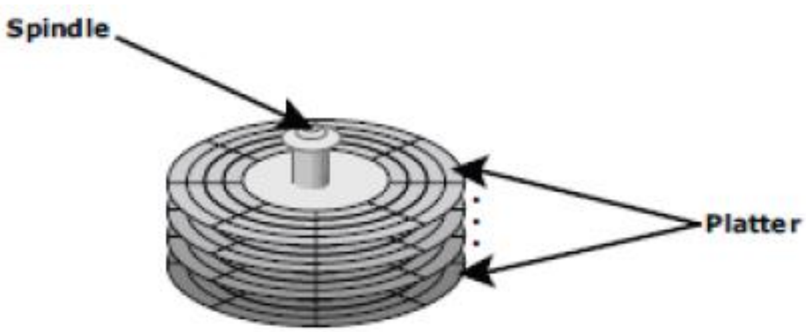

Figure: Spindle and platter

2. Spindle

 A spindle connects all the platters, as shown in above figure and is connected to a motor.

The motor of the spindle rotates with a constant speed.

3. Read/Write Head

 Read/Write (R/W) heads, shown in Figure, read and write data from or to a platter.

 Drives have two R/W heads per platter, one for each surface of the platter.

4. Actuator Arm Assembly

 The R/W heads are mounted on the *actuator arm assembly* which positions the R/W head at the location on the platter where the data needs to be written or read.

5. Controller

 The *controller* is a printed circuit board, mounted at the bottom of a disk drive.

 It consists of a microprocessor, internal memory, circuitry and firmware.

6. Physical Disk Structure

 Data on the disk is recorded on *tracks*, which are concentric rings on the platter around the spindle.

2	Explain the various factors of Disk Drive Performance CO1	10 L2

Disk Drive Performance

A disk drive is an electromechanical device that governs the overall performance of the storage system environment.

1. Disk Service Time

 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.*

2. 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.

3. 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*.

4. Data Transfer Rate

 The *data transfer rate* (also called *transfer rate*) refers to the average amount of data per unit time that the drive can deliver to the HBA.

 It is important to first understand the process of read and write operations in order to calculate data transfer rates.

3	Explain the RAID techniques with suitable diagrams CO1		10 L3

1. Striping

 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*.

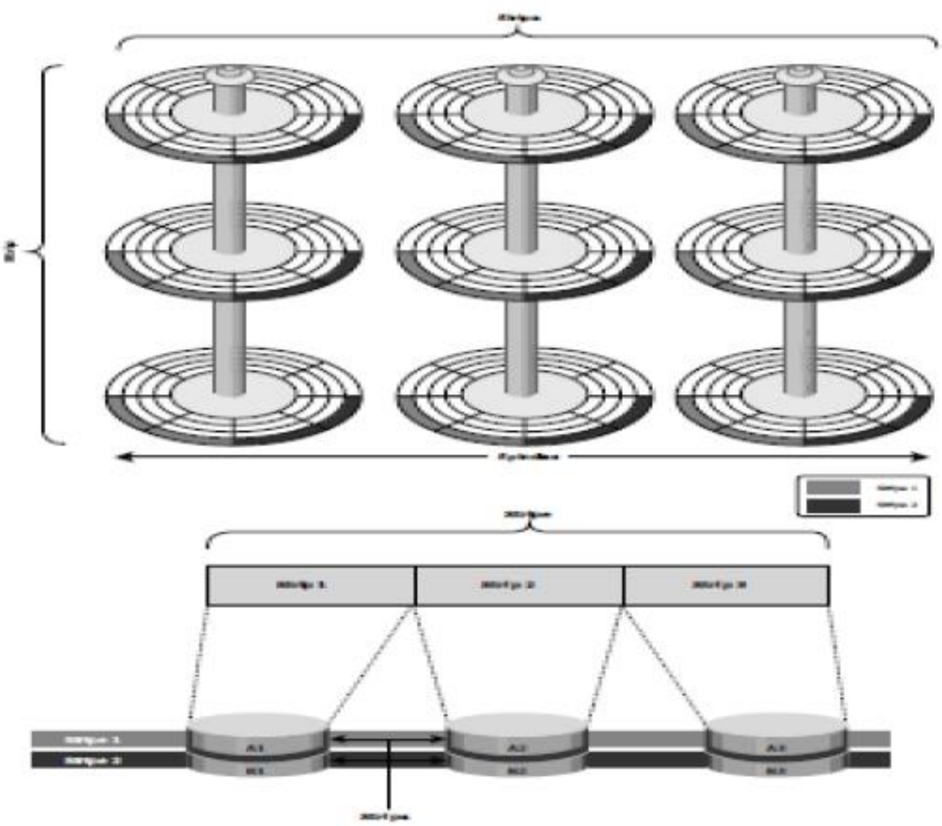

Figure: Striped RAID set

2. **Mirroring**

 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.

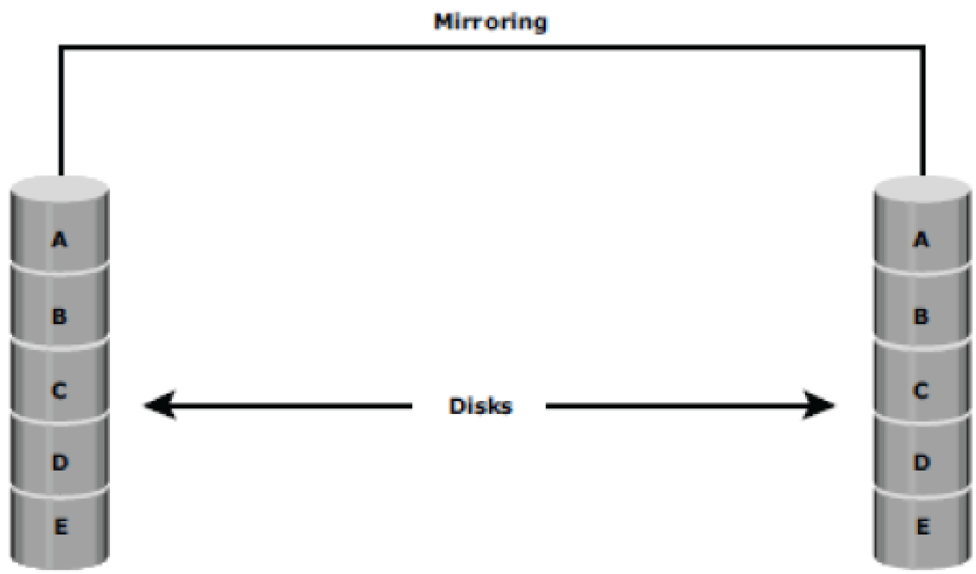

Figure: Mirrored disks in an array

3. Parity

 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.

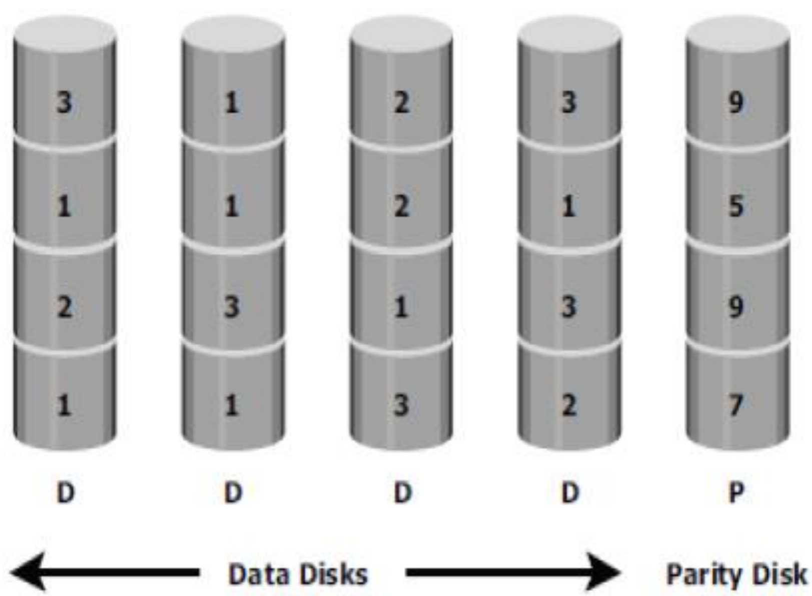

Figure: Parity RAID

 The first four disks, labeled *D*, contain the data.

 The fifth disk, labeled *P*, stores the parity information, which in this case is the sum of the elements in each row.

 Now, if one of the *D*s fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value.

 Parity calculation is a *bitwise XOR* operation.

4 Explain Components of an Intelligent Storage System 10
CO1 L2

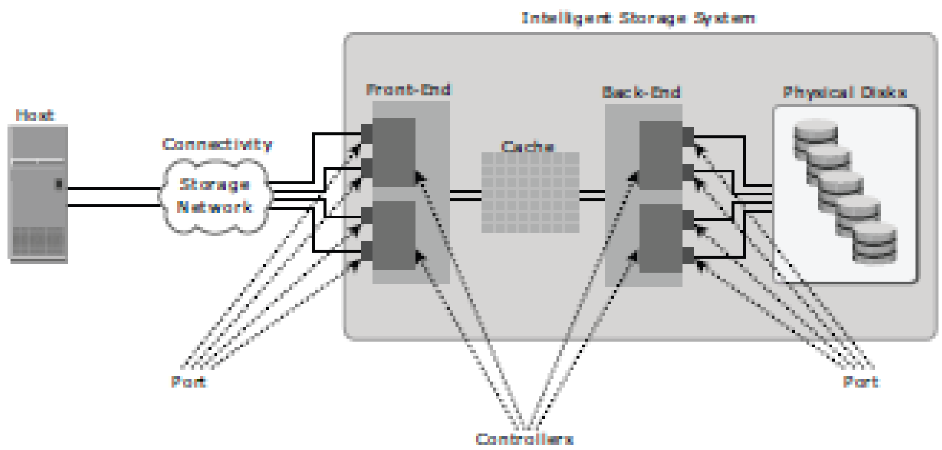

Figure 4-1: Components of an intelligent storage system

Front End

The front end provides the interface between the storage system and the host. It consists of two components: 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 pro- tocol, such as SCSI, Fibre Channel, or iSCSI, for storage connections. Redundant ports are provided on the front end for high availability.

Front-End Command Queuing

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 the default algorithm where commands are executed in the order in which they are received (Figure 4-2 [a]).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 4-2(a), 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

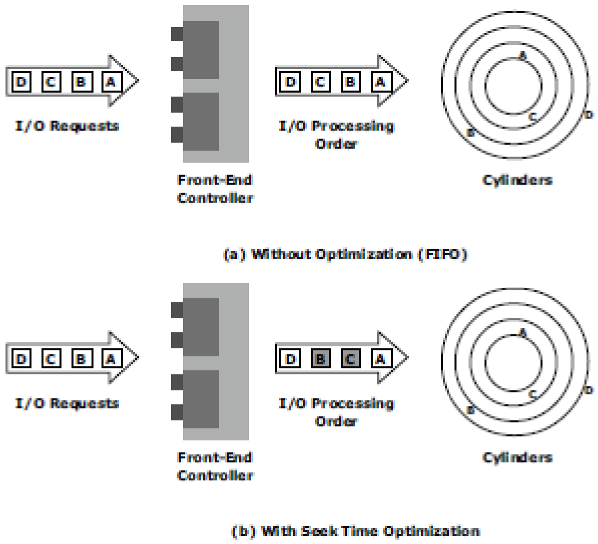

Figure 4-2: Front-end command queuing

Back End

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

Physical Disk

A physical disk stores data persistently. Disks are connected to the back-end with either SCSI or a Fibre Channel interface (discussed in subsequent chapters). An intelligent storage system enables the use of a mixture of SCSI or Fibre Channel drives and IDE/ATA drives

5 The average I/O size of an application is 64 KB. The following specifications are available from the disk manufacturer: average seek time = 5 ms, 7,200 RPM, and transfer rate = 40 MB/s. Determine the maximum IOPS that could be performed with this disk for the application. Using this case as an example, explain the relationship between disk utilization and IOPS. 10

CO1 L4

Seek time is 5ms.
Disk rotation speed is 7,200 rpm or 120 rps, from which rotational latency is determined:
L = 0.5/120 = 4.2ms (one half of the time taken for a full rotation).
The data transfer rate is derived from the block size:
X = 64Kb / 40 Mb = 1.6ms
The time taken by the I/O controller to serve an I/O block size of 64Kb (disk service time) is:
Ts = T + L + X = 5ms + 4.2ms + 1.6ms = 10.8ms
Therefore, the maximum number of IOPS is:
1 / Ts = 92 IOPS

This represents the IOPS that can be achieved at potentially high levels of disk utilization, close to 100%. At lower disk utilization, the number of IOPS a disk can perform is reduced, for example, at 70-percent utilization the IOPS is 64 (92 x 0.7). In the same way, an increase in disk controller utilization causes an increase in the application response time. For example, the response time for an I/O at 96 percent disk controller utilization is:

R = Ts / (1-U) = 10.8 / (1-0.96) = 270ms

6 a) An application generates 7650 IOPS with 50% being READ operation with disk handling capacity of 180 IOPS. Determine the disk workload and number of disks required in RAID5 10
CO1 L3

Disk workload=(7650*0.5)+4*(7650*0.5)=19125
No of disks required = 19125/180 = 106 disk

b) List out Comparison between different RAID levels
CO1 L2

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
5	3	(n-1)*100/n where n= number of disks	Moderate	Very good for random reads. Good for sequential reads	Fair for random writes. Slower due to parity overhead. Fair to good for sequential writes.	High
6	4	(n-2)*100/n where n= number of disks	Moderate but more than RAID 5	Very good for random reads. Good for sequential reads.	Good for small, random writes (has write penalty).	Very High
1+0 and 0+1	4	50	High	Very good	Good	Moderate