Unit -2

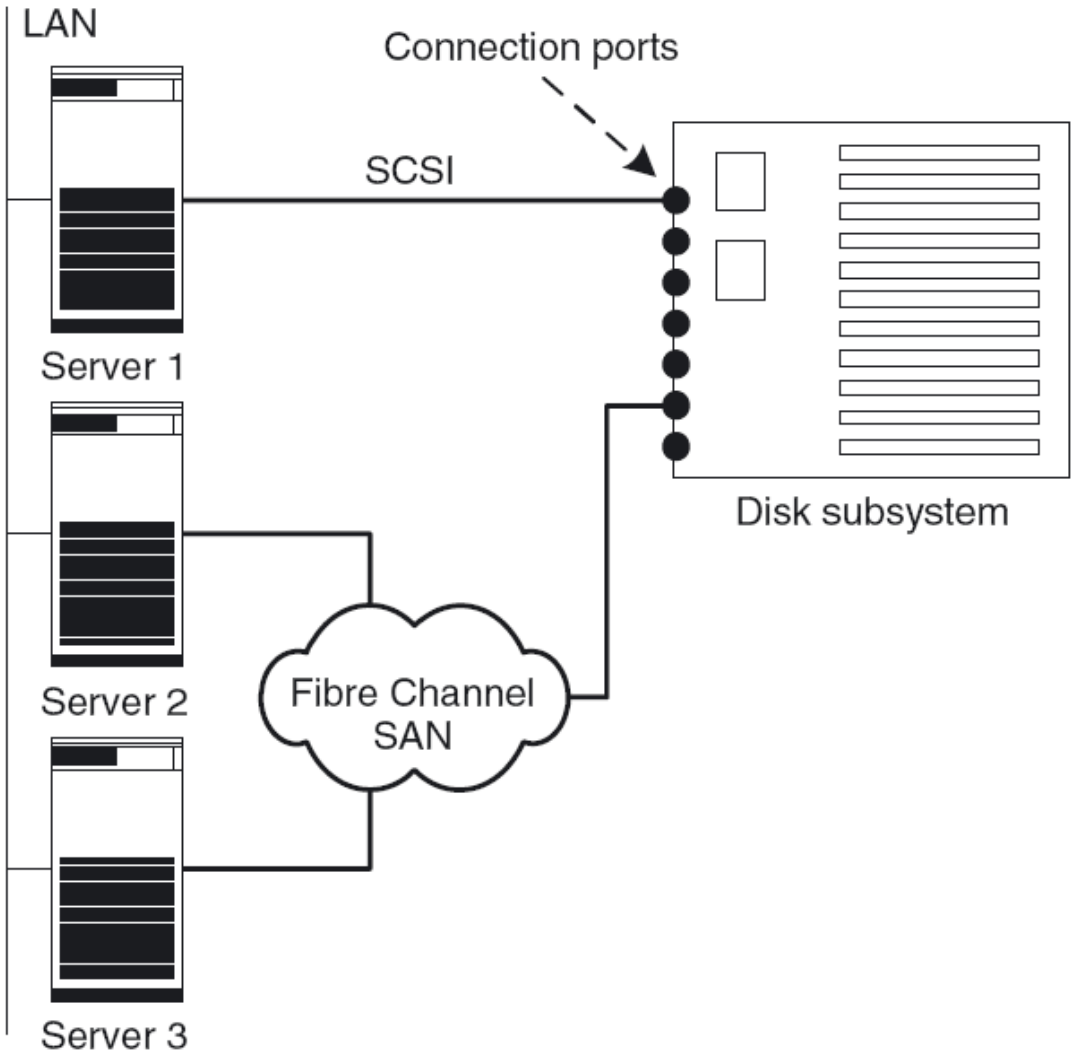
Storage systems architecture

Unit -02/Lecture-01

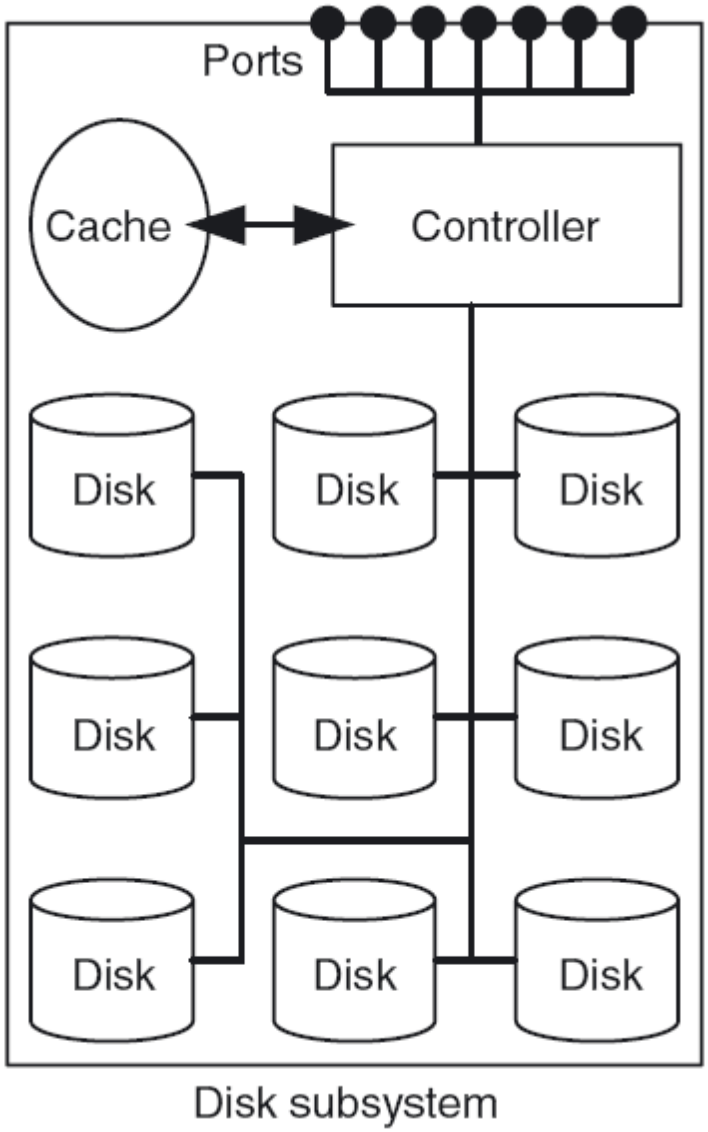
Architecture of Intelligent disk subsystems – [Rgpv/dec 2014(7),Rgpv/dec2013(10)]

In contrast to a file server, a disk subsystem can be visualized as a hard disk server. Servers are connected to the connection port of the disk subsystem using standard I/O techniques such as Small Computer System Interface (SCSI), Fibre Channel or Internet SCSI (iSCSI) and can thus use the storage capacity that the disk subsystem provides. The internal structure of the disk subsystem is completely hidden from the server, which sees only the hard disks that the disk subsystem provides to the server.

The connection ports are extended to the hard disks of the disk subsystem by means of internal I/O channels. In most disk subsystems there is a controller between the connection ports and the hard disks. The controller can significantly increase the data availability and data access performance with the aid of a so-called RAID procedure. Fur- thermore, some controllers realize the copying services instant copy and remote mirroring and further additional services. The controller uses a cache in an attempt to accelerate read and write accesses to the server.



Servers are connected to a disk subsystem using standard I/O techniques. The figure shows a server that is connected by SCSI. Two others are connected by Fibre Channel SAN.

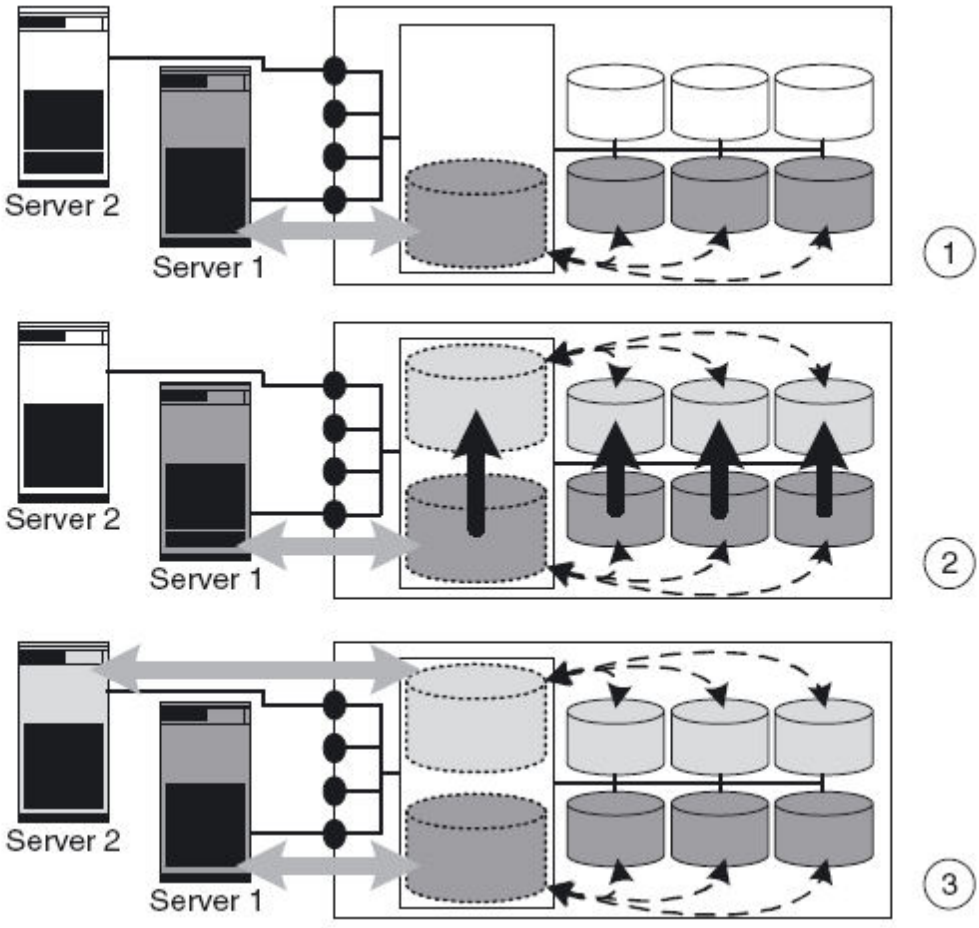


Servers are connected to the disk subsystems via the ports. Internally, the disk subsystem consists of hard disks, a controller, a cache and internal I/O channels.

Disk subsystems are available in all sizes. Small disk subsystems have one to two connection ports for servers or storage networks, six to eight hard disks and, depending on the disk capacity, storage capacity of a few terabytes. Large disk subsystems have multiple ten connection ports for servers and storage networks, redundant controllers and multiple I/O channels. A considerably larger number of servers can access a subsystem through a connection over a storage network. Large disk subsystems can store up to a

petabyte of data and, depending on the supplier, can weigh well over a tonne. The dimensions of a large disk subsystem are comparable to those of a wardrobe. The architecture of real disk subsystems is more complex and varies greatly. Ultimately, however, it will always include the components.

Regardless of storage networks, most disk subsystems have the advantage that free disk space can be flexibly assigned to each server connected to the disk subsystem (storage pooling). All servers are either directly connected to the disk subsystem or indirectly connected via a storage network. In this configuration each server can be assigned free storage. Incidentally, free storage capacity should be understood to mean both hard disks that have already been installed and have not yet been used and also free slots for hard disks that have yet to be installed.



All servers share the storage capacity of a disk subsystem. Each server can be assigned free storage more flexibly as required.

S.no	Rgpv question	Year	Marks
Q.1	Explain component architecture of intelligent disk subsystem?	Dec 2014 Dec 2013	07 10

Unit-02/Lecture -02

Modular vs. Integrated: [Rgpv/dec 2015(7),Rgpv/dec2013(7),Rgpv/dec2011(5)]

Which unified storage architecture to go with when looking at unified systems is probably a decision based on availability more than functionality. Modular solutions may be more appropriate for users that have an existing storage array that offers NAS services with a gateway module. On the other hand, users buying new unified systems will be looking mainly at integrated systems, since most of the new products will have an integrated architecture.

At one time block storage supported the most important applications, typically production databases, and file storage was associated with user home directories and office productivity applications. But now even the most critical databases are being run on NAS devices.

Server virtualization has further raised the profile of file storage, since applications like vmware store and manipulate entire server instances as individual files. Nfs-hosted images are rapidly gaining traction in these environments. "big data" archive systems in industries such as media and entertainment, oil and gas, and remote sensing, to name a few, also do their work at the file level.

S.no	Rgpv question	Year	Marks
Q.1	Differentiate integrated vs modular array?	Dec 2015 Dec 2013 Dec 2011	7 7 5

Unit-02/Lecture – 03

Volume manager vs file system – [Rgpv/dec2013(7), Rgpv/dec2011(5)]]

Current file systems trace their roots from the UFS file system, which was proposed in 1965. By the early 1970s, the UNIX file system was up and running. Since then, not much has changed in file systems and there have only been incremental hardware changes. I think the file system and volume manager are the most critical components in achieving I/O performance from both the OS and underlying hardware. Even the best file system and volume manager can be configured so that the performance is poor. Therefore, my next couple of columns will cover file system and volume management, in addition to file system configuration and tuning.

File System Basics

The purpose of file systems (FS) is to maintain a view of the storage so we can create files. This is done so that users can create, delete, open, close, read, write, and extent files on the device(s). File systems can also be used to maintain security over files.

Volume Manager Basics

The original goal of the UNIX volume management (VM), which was developed in the late 1980s, was to group disk devices together so that file systems larger than a single device could be created, and to achieve high performance by striping devices.

Standard VM Inner Workings (Striping)

Most file systems require a VM to group disk and/or RAID devices together. Striping spreads the data across the devices based on the stripe size set within the volume manager. Note that some volume managers support concatenation, which starts with the first device and then only writes

to the next device when the first device becomes full. The idea behind striping is to spread the data across multiple devices to improve performance and allow multiple I/O disk-head seeks to occur simultaneously. Figure 1 shows what happens with standard striping for allocation of multiple files writing at the same time, and shows what happens when one of those files is removed.

File Systems that Maintain Their Topology

Some file systems maintain and understand the device topology without a volume manager. These file systems support both striping and round-robin allocation. Round-robin allocation means that each device is used individually. In most cases, each file open moves to the next device. In some file systems, it could be that each directory created moves to the next device. Figure 2 shows an example of round-robin allocation, which is very different from striping. I will show how round-robin allocation has some important implications for performance.

File Allocation Comparison

One reason that volume managers do not provide a round-robin allocation method is because of the interaction between the volume manager and the file system. Every file system must allocate space and maintain consistency, which is one of the main purposes of the file system. There are multiple types of file system allocation, but the real issue is that a volume manager presents a single set address range for the block devices in the file system for the file system to allocate from. The volume manager then translates the address to each of the devices. It is difficult, but not impossible, for the volume manager to pass all of the underlying device topology to a file system. Also, most file systems designed with volume managers do not have an interface to understand the underlying volume topology. Other file systems that control their own topology can easily use round-robin allocation, because the file systems understand the underlying topology.

How Volume Managers and File Systems Work

It is important to fully understand how volume managers and file systems work internally to choose the best file system for the application. By understanding the inner workings, you will have a much better idea of what the tunable parameters mean and how to improve performance.

Performance Comparison

Indirect block allocation and read/write performance can be painfully slow compared to the extent-based allocation method. For example, consider an application doing random reads and writes. To find the block address for the record, a file allocated with indirect blocks must read all of the data areas of the files for the record in question prior to reading the record. With extent-based allocation, the file system can simply read the inodes in question, which will make an enormous difference in performance. I am unaware of any new file systems using indirect blocks for space allocation because of the huge performance penalties for random I/O. Even for sequential I/O, the performance for indirect blocks is generally less than extent-based file systems.

Free Space Allocation and Representation Methods

Each file system uses an algorithm to find and allocate free space within the file system. Most file systems use binary tree (Btree) allocation to represent free space, but some file systems use bitmaps to represent free space. Each method of free space representation has advantages and disadvantages.

Bitmap Representation

The use of bitmap representation is less common. This method is used where each bit in the map represents a single allocation unit such as 1,024 bytes, 512 KB, or even hundreds of megabytes. Therefore, a single bit could represent a great deal of space.

Free Space Allocation

With each allocation type (Btree or bitmap), free space must be found and allocated with the representation method. These allocation algorithms find the free space based on their internal search algorithms. The two most common methods used are first fit and best fit.

First Fit

The first fit method tries to find the first space within the file system that matches the allocation size requested by the file being allocated. In some file systems, the first fit method is used to find the space closest to the last allocation of the file being extended, thereby allowing the allocation to be sequential block addresses allocated for the file within the file system.

Best Fit

The best fit method tries to find the best place in the file system for the allocation of the data. This method is used to try to reduce total file system fragmentation. This method always takes more CPU cycles than first fit, because the whole file system must be searched for the best allocation. (Note that in systems using round-robin allocation only, the device on which the initial allocation was made must be searched.) This method works to reduce fragmentation, especially when files cannot be pre-allocated (for file systems that support this method) or for large allocations, such as multiple megabytes. Most vendors do not support this method, and most allocations in file systems are not large because the overhead would be huge. The old Cray NC1FS supports this method by using hardware vector registers to quickly perform the search.

S.no	Rgpv question	Year	Marks
Q.1	Give difference between Volume manager and file system ?	Rgpv Dec2013(7) Rgpv Dec2011(5)	7 5

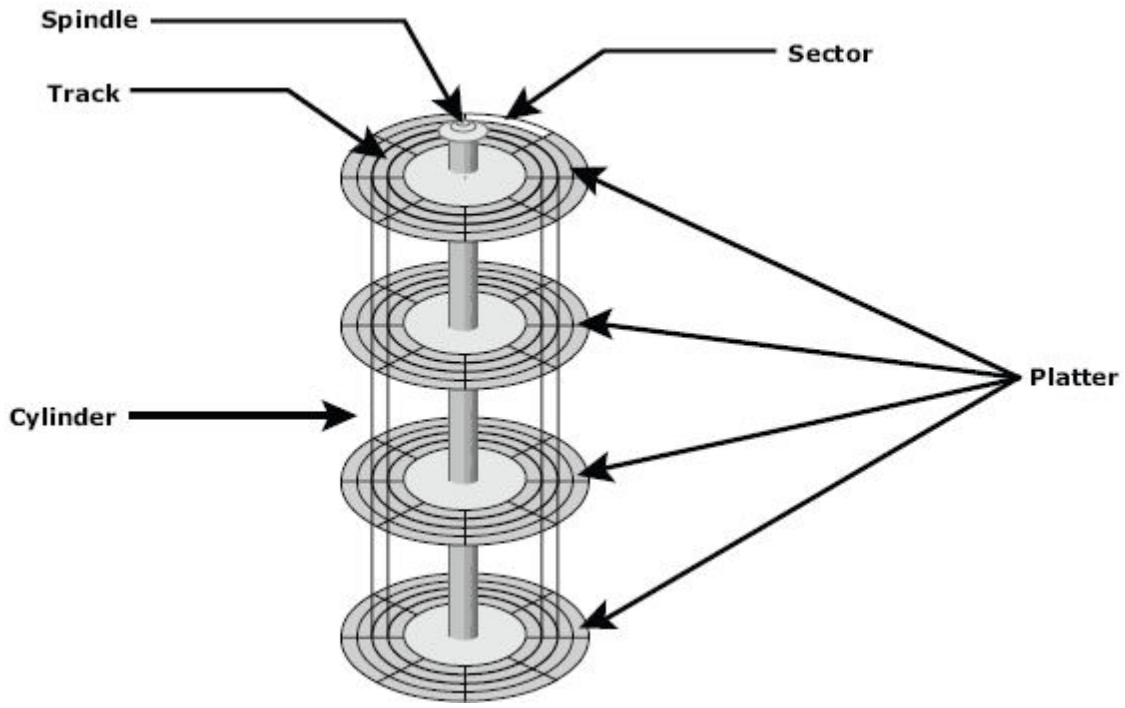
Unit 02/Lecture -04

Physical disk structure[Rgpv/dec2012(10)]

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 small- est, 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 format- ted 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 plat- ter. The location of drive heads is referred to by cylinder number, not by track number.



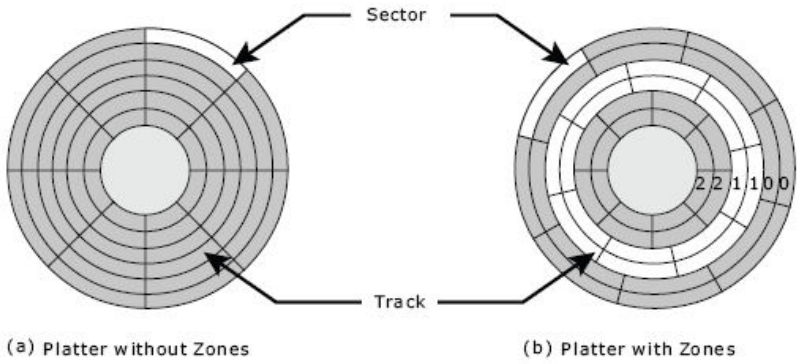
.

Disk structure: sectors, tracks, and cylinders

Zoned bit recording [Rgpv/dec2014 (2), Rgpv/dec2012(10)]

Because the platters are made of concentric tracks, the outer tracks can hold more data than the inner tracks, because the outer tracks are physically longer than the inner tracks, as shown in figure 2-6 (a). On older disk drives, the outer tracks had the same number of sectors as the inner tracks, so data density was low on the outer tracks. This was an inefficient use of available space.

Zone bit recording utilizes the disk efficiently. As shown in figure 2-6 (b), this mechanism groups tracks into zones based on their distance from the center of the disk. The zones are numbered, with the outermost zone being zone 0. An appropriate number of sectors per track are assigned to each zone, so a zone near the center of the platter has fewer sectors per track than a zone on the outer edge. However, tracks within a particular zone have the same number of sectors.



Zoned bit recording

Disk drive performance - [Rgpv/dec2013(10)]

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

disk service time

Disk service time is the time taken by a disk to complete an i/o request. Components that contribute to service time on a disk drive are seek time, rota- tional latency, and data transfer rate.

Seek time

The seek time (also called access time) describes the time taken to position the r/w heads across the platter with a radial movement (moving along the radius of the platter). In other words, it is the time taken to reposition and settle the arm and the head over the correct track. The lower the seek time, the faster the i/o opera- tion. 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 par- ticular 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.

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. 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.

S.no	Rgpv question	Year	Marks
Q.1	What do you understand by Zoned bit recording and logical block addressing?	Dec 2014 Dec 2012	2 10
Q.2	Explain various performance criteria of disk? Write specification of disk?	Dec 2013	10

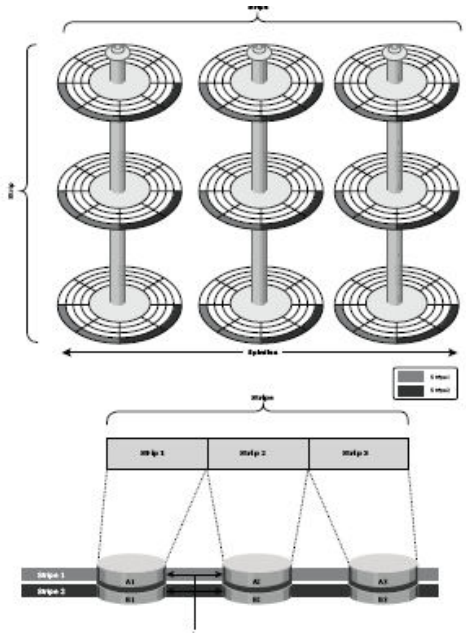
Unit 02/Lecture-05

Raid levels – [Rgpv/dec 2015(7),Rgpv/dec2012(10),Rgpv/dec2012(10)]

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.

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 3-2 shows physical and logical representations of a striped raid set.



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.

Mirroring

Mirroring is a technique whereby data is stored on two different HDDs, yield- ing two copies of data. In the event of one HDD failure, the data is intact on the surviving HDD (see figure 3-3) and the controller continues to service the host's data requests from the surviving disk of a mirrored pair.

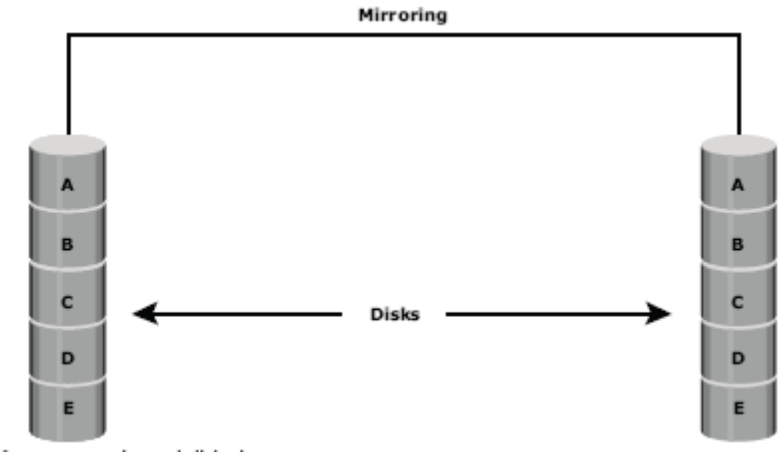
When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair. This activity is transparent to the host.

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.

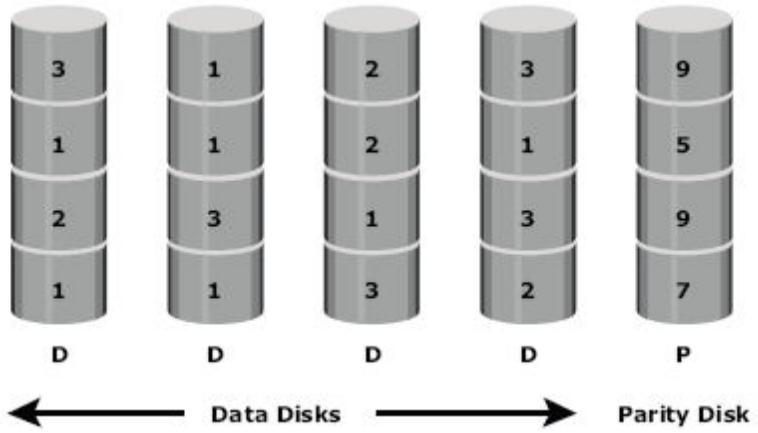


Mirrored disks in an array

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. Parity is a redundancy check that ensures full protection of data without maintaining a full set of duplicate data.

Parity information can be stored on separate, dedicated HDDs or distributed across all the drives in a raid set. Figure 3-4 shows a 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 ds fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value.



Parity raid

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.

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. 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.

S.no	Rgpv question	Year	Marks
Q.1	Write short note on stripping and mirroring?	Dec 2012	10

Unit-02/Lecture-06

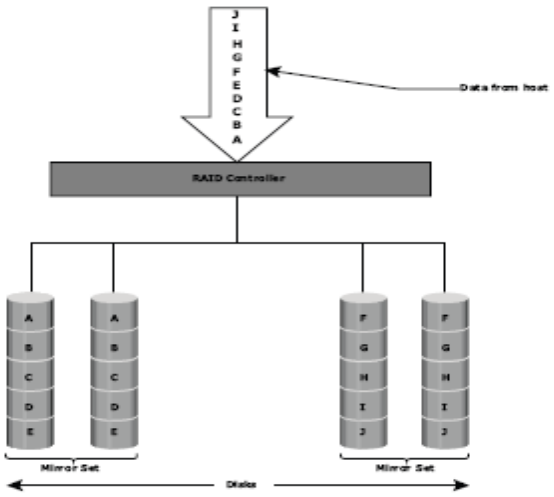
Raid levels – [Rgpv/dec 2014(3),Rgpv/dec2012(10)]

Raid 0

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. Figure 3-5 shows raid 0 on a storage array in which data is striped across 5 disks. 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.

Raid 1

In a raid 1 configuration, data is mirrored to improve fault tolerance . A raid 1 group consists of at least two HDDs. As explained in mirroring, every write is written to both disks, which is transparent to the host in a hardware raid implementation. In the event of disk failure, the impact on data recovery is the least among all raid implementations. This is because the raid controller uses the mirror drive for data recovery and continuous operation. Raid is suitable for applications that require high availability.



Raid 1

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 (see figure 3-7).

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:

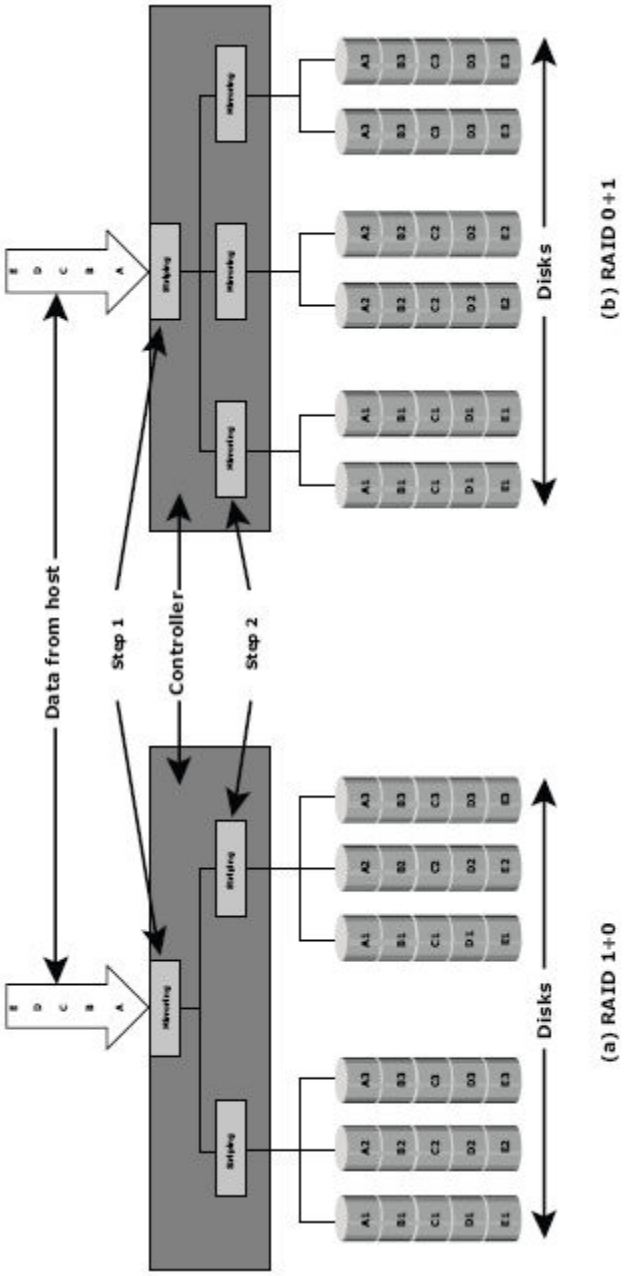
- High transaction rate online transaction processing (oltp)
- Large messaging installations
- Database applications that require high i/o rate, random access, and high availability

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.

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. If one drive fails, then the entire stripe is faulted. A rebuild operation copies the entire stripe, copying data from each disk in the healthy stripe to an equivalent disk in the failed stripe. This causes increased and unnecessary i/o load on the surviving disks and makes the raid set more vulnerable to a second disk failure.

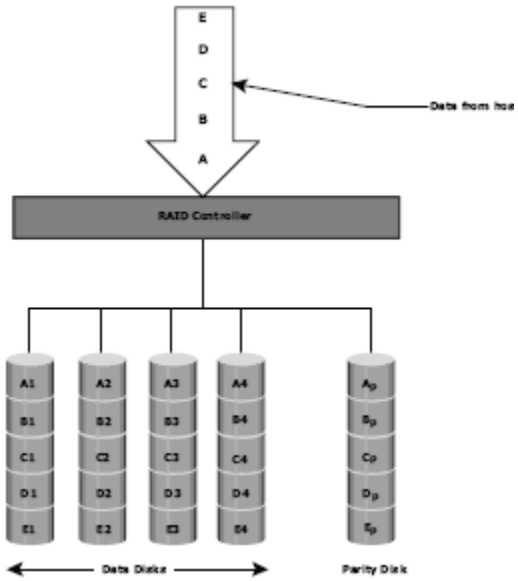


Nested

Raid 3

Raid 3 stripes data for high performance and uses parity for improved fault tolerance. Parity information is stored on a dedicated drive so that data can be reconstructed if a drive fails.

For example, of five disks, four are used for data and one is used for parity. 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.



raid 3

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.

Raid 4

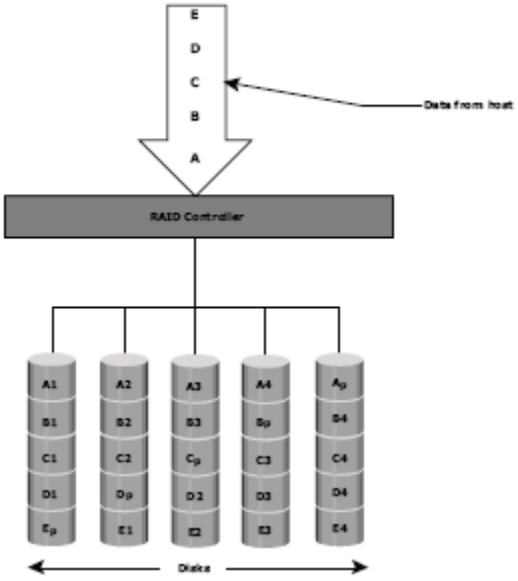
Similar to raid 3, raid 4 stripes data for high performance and uses parity for improved fault tolerance . Data is striped across all disks except the parity disk in the array. Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails. Striping is done at the block level.

Unlike raid 3, data disks in raid 4 can be accessed independently so that specific data elements can be read or written on single disk without read or write of an entire stripe. Raid

4 provides good read throughput and reason- able write throughput.

Raid 5

Raid 5 is a very versatile raid implementation. 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. The raid 5 implementation. 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.

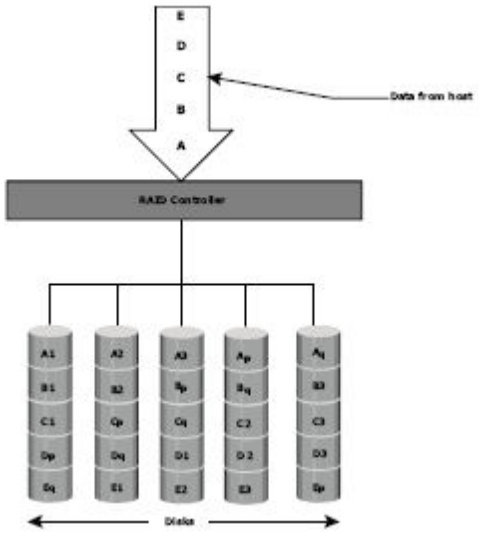


raid 5

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 6

Comparison of different raid types

RAID level	Fault-tolerance	Read performance	Write performance	Space requirement
RAID 0	None	Good	Very good	Minimal
RAID 1	High	Poor	Poor	High
RAID 10	Very high	Very good	Good	High
RAID 4	High	Good	Very very poor	Low
RAID 5	High	Good	Very poor	Low
RAID 6	Very high	Good	Very very poor	Low

S.no	Rgpv question	Year	Marks
Q.1	Write down different levels of RAID and compare them?	Dec 2015 Dec 2012 Dec 2011	7 10 10
Q.2	How RAID4 is different from RAID3?	Dec 2014	3

Unit-02/Lecture-07

Disk service time - [Rgpv/dec2012(10)]

Disk Service time (Ts) can be calculated using its Rotational Latency (L), Average Seek Time (T) and Internal Data Transfer Time (X).

i.e Ts = L + T + X

Hence the components of Disk Service Time are Disk Rotational Latency, Average Seek Time and Internal Data Transfer Time.

In random I/O operation seek time will be high, as the R/W head to seek different sectors on different tracks on the platter to read/write an I/O to/from disk.

So Seek Time contributes largest percentage of the disk service time in a random I/O operation.

S.no	Rgpv question	Year	Marks
Q.1	Which components constitute the disk service time? Which component contributes the largest percentage of the disk service time in a random I/O operation?	Dec 2012	10

Additional Topic/Lecture-08

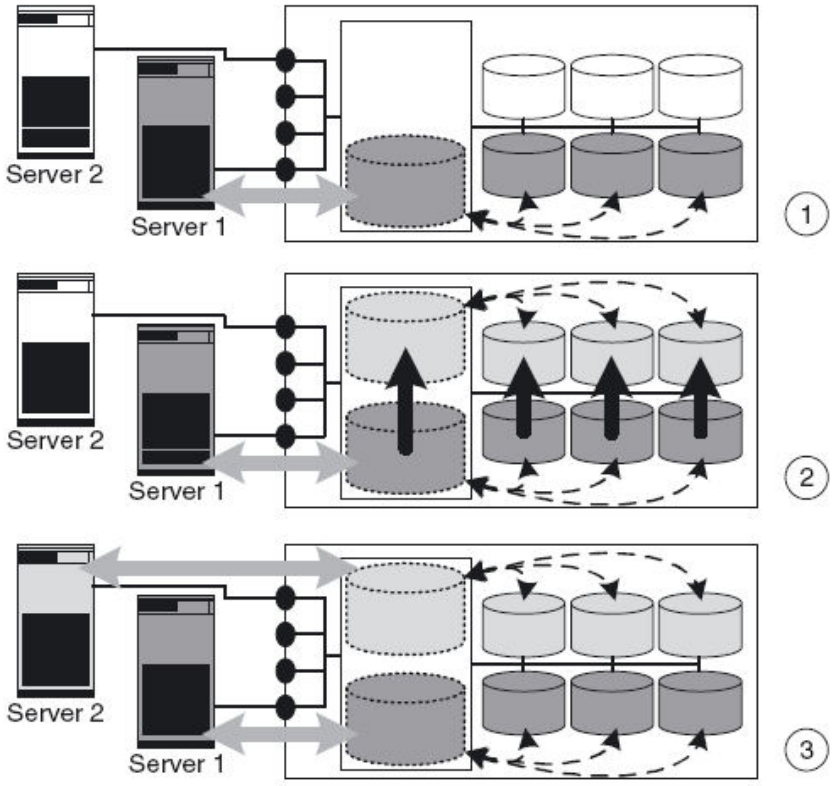
Intelligent disk subsystems overview

Intelligent disk subsystems represent the third level of complexity for controllers after jbods and raid arrays. The controllers of intelligent disk subsystems offer additional functions over and above those offered by raid. In the disk subsystems that are currently available on the market these functions are usually instant copies, remote mirroring and lun masking.

Instant copies

Instant copies can virtually copy data sets of several terabytes within a disk subsystem in a few seconds. Virtual copying means that disk subsystems fool the attached servers into believing that they are capable of copying such large data quantities in such a short space of time. The actual copying process takes significantly longer. However, the same server, or a second server, can access the virtually copied data after a few seconds.

There are numerous alternative implementations for instant copies. One thing that all implementations have in common is that the pretence of being able to copy data in a matter of seconds costs resources. All realizations of instant copies require controller computing time and cache and place a load on internal i/o channels and hard disks. The different implementations of instant copy force the performance down at different times. However, it is not possible to choose the most favorable implementation alternative depending upon the application used because real disk subsystems only ever realize one implementation alternative of instant copy.



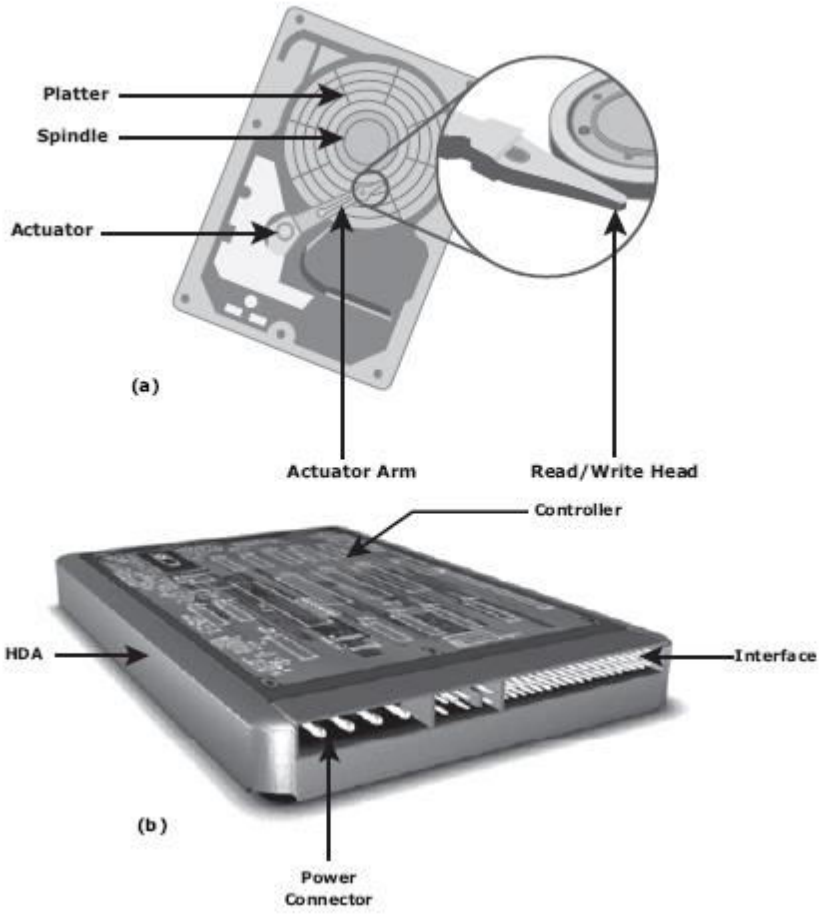
Instant copies can virtually copy several terabytes of data within a disk sub- system in a few seconds: server 1 works on the original data (1). The original data is virtually copied in a few seconds . Then server 2 can work with the data copy, whilst server 1 continues to operate with the original data .

Additional Topic Unit - 02/Lecture-09

Disk drive components

A disk drive uses a rapidly moving arm to read and write data across a flat plat- ter 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.

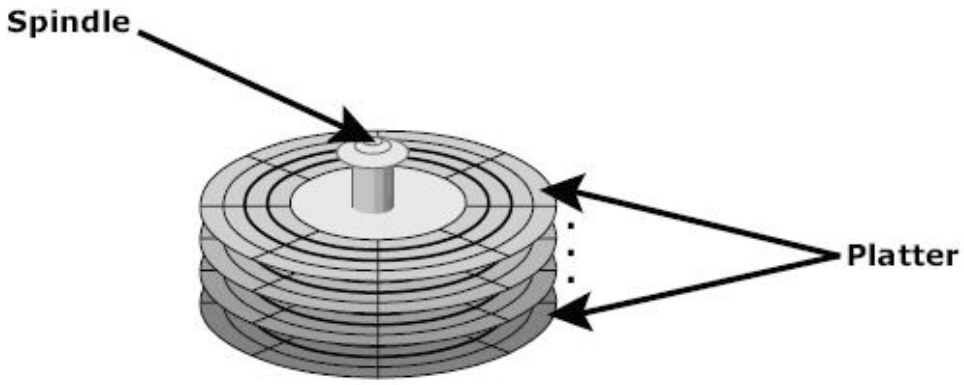
Key components of a disk drive are platter, spindle, read/write head, actuator arm assembly, and controller



Disk drive components

Platter

A typical HDD consists of one or more flat circular disks called platters. The data is recorded on these platters in binary codes . 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 bothSurfaces (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.



Spindle

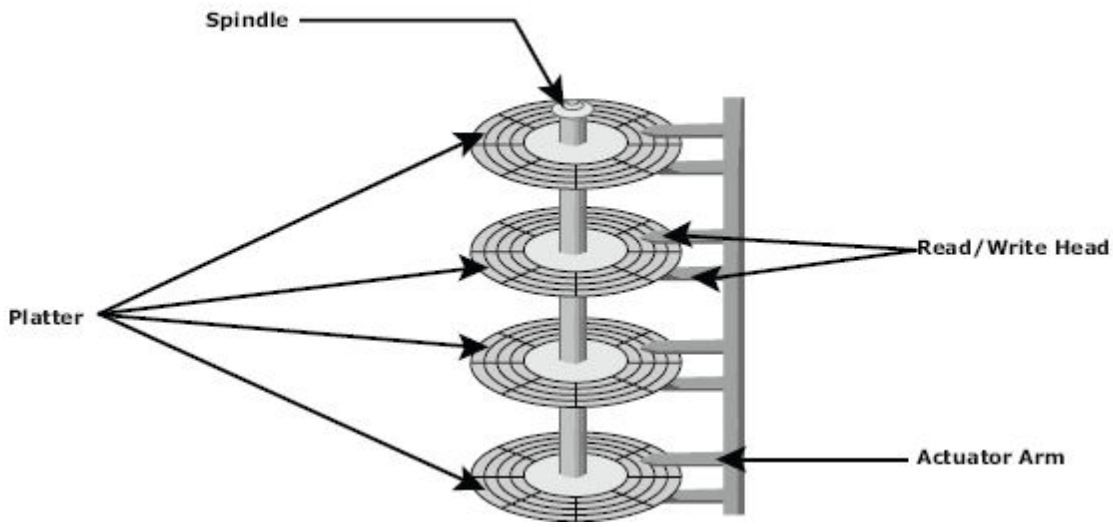
A spindle connects all the platters, as shown in figure 2-3, and is connected to a motor. The motor of the spindle rotates with a constant speed.The disk platter spins at a speed of several thousands of revolutions per minute (rpm). Disk drives have spindle speeds of 7,200 rpm, 10,000 rpm, or 15,000 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.

read/write head

Read/write (r/w) heads, shown in figure 2-4, read and write data from or to a platter. Drives have two r/w heads per platter, one for each surface of the platter. The r/w head changes the magnetic polarization on the surface of the platter when writing data. While reading data, this head detects magnetic polarization on the surface of the platter. During reads and writes, the r/w head senses the magnetic polarization and never touches the surface of the platter. When the spindle is rotating, there is a microscopic air gap between the r/w heads and the platters, known as the head flying height. This air gap is removed when the spindle

stops rotating and the r/w head rests on a special area on the platter near the spindle. This area is called the landing zone. The landing zone is coated with a lubricant to reduce friction between the head and the platter.

The logic on the disk drive ensures that heads are moved to the landing zone before they touch the surface. If the drive malfunctions and the r/w head accidentally touches the surface of the platter outside the landing zone, a head crash occurs. In a head crash, the magnetic coating on the platter is scratched and may cause damage to the r/w head. A head crash generally results in data loss.



Actuator arm assembly

Controller

The controller (see figure 2-2 [b]) is a printed circuit board, mounted at the bottom of a disk drive. It consists of a microprocessor, internal memory, circuitry,

And firmware. The firmware controls power to the spindle motor and the speed of the motor. It also manages communication between the drive and the host. In addition, it controls the r/w operations by moving the actuator arm and switching between different r/w heads, and performs the optimization of data access.

Additional Topic Unit - 02/Lecture -10

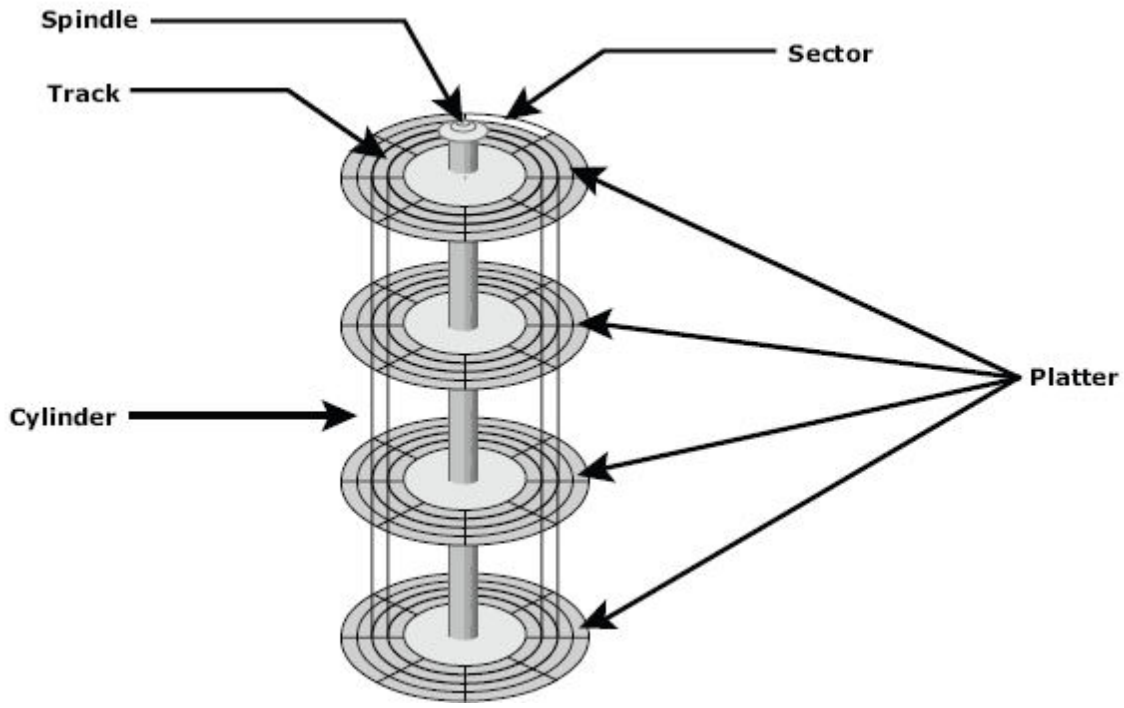
Physical disk structure

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 small- est, 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 format- ted 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 plat- ter. The location of drive heads is referred to by cylinder number, not by track number.



.

Disk structure: sectors, tracks, and cylinders

Additional Topic Unit -0 2/Lecture-11

Hot spares[Rgpv/dec2015(2)]

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:

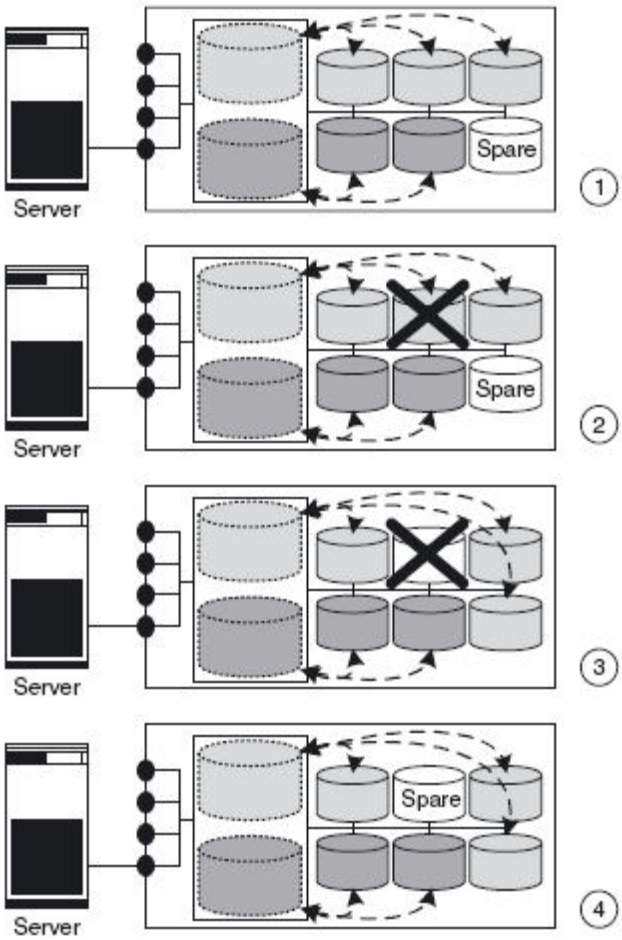
- the hot spare replaces the new HDD permanently. This means that it is no longer a hot spare, and a new hot spare must be configured on the array.
- when a new HDD is added to the system, data from the hot spare is copied to it. The hot spare returns to its idle state, ready to replace the next failed 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.

Modern raid controllers can manage a common pool of hot spare disks for several virtual raid disks. Hot spare disks can be defined for all raid levels that offer redundancy.
The recreation of the data from a defective hard disk takes place at the same time as Write and read operations of the server to the virtual hard disk, so that from the point of view of the server, performance reductions at least can be observed. Modern hard disks come with self-diagnosis programs that report an increase in write and read errors to the system administrator in plenty of time: 'caution! I am about to depart this life. Please replace me with a new disk. Thank you!' to this end, the individual hard disks store the data with a redundant code such as the hamming code. The hamming code permits the correct recreation of the data, even if individual bits are changed on the hard disk. If the system is looked after properly you can assume that the installed physical hard disks will hold out for a while. Therefore, for the benefit of higher performance, it is generally an acceptable risk to give access by the server a higher priority than the recreation of the data of an exchanged physical hard disk.

A further side-effect of the bringing together of several physical hard disks to form a virtual hard disk is the higher capacity of the virtual hard disks. As a result, less device addresses are used up in the i/o channel and thus the administration of the server is also simplified, because less hard disks (drive letters or volumes) need to be used.



Hot spare disk

The disk subsystem provides the server with two virtual disks for which a common hot spare disk is available (1). Due to the redundant data storage the server can continue to process data even though a physical disk has failed, at the expense of a reduction in performance (2). The raid controller recreates the data from the defective disk on the hot spare disk (3). After the defective disk has been replaced a hot spare disk is once again available (4).

S.no	Rgpv question	Year	Marks
Q.1	What is hot sparing?	Dec 2014	2

Additional Topic Unit - 02/Lecture-12

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 controllers route data to and from cache via the internal data bus. When cache receives write data, the controller sends an acknowledgment mes- sage back to the host. Controllers optimize i/o processing by using command queuing algorithms.

Front-end command queuing

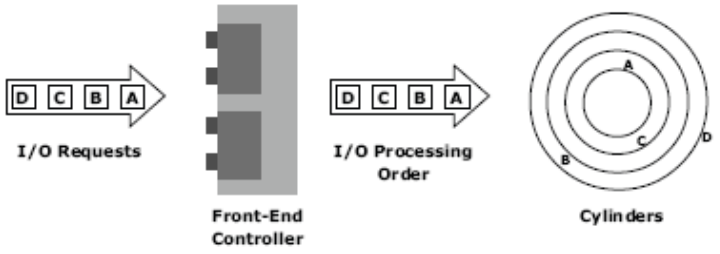
Command queuing is a technique implemented on front-end controllers. It determines the execution order of received commands and can reduce unnec- essary drive head movements and improve disk performance. When a com- mand is received for execution, the command queuing algorithms assigns a tag that defines 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:

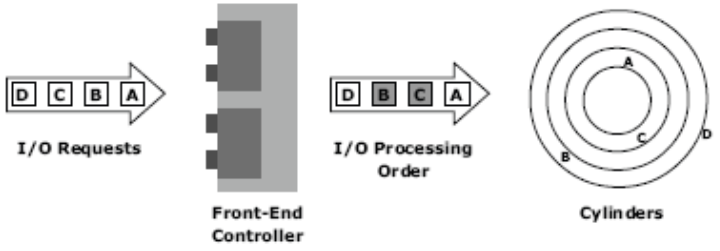
- First in first out (fifo): this is the default algorithm where commands are executed in the order in which they are received. There is no reordering of requests for

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. 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.



(a) Without Optimization (FIFO)



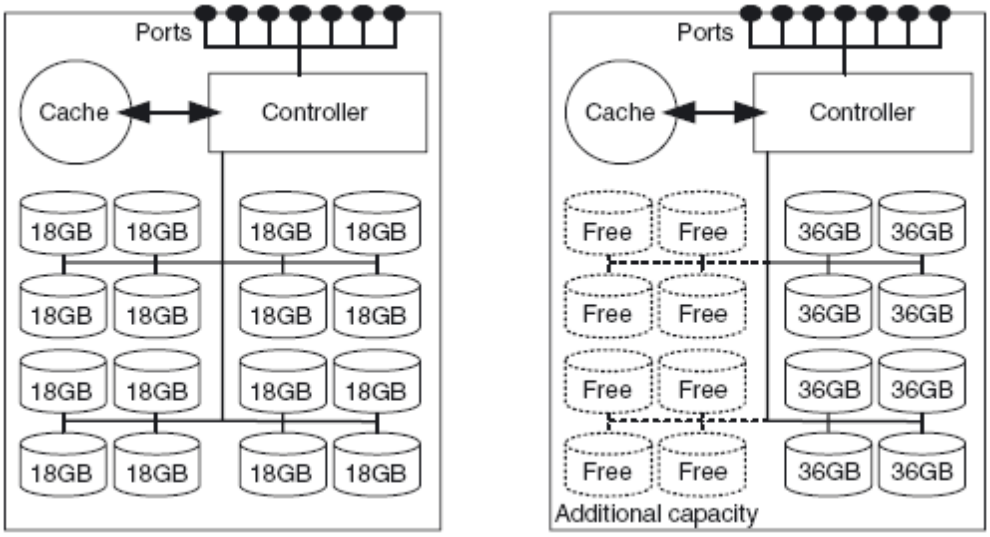
(b) With Seek Time Optimization

Additional Topic Unit - 02/Lecture -13

Hard disks and internal i/o channels

The controller of the disk subsystem must ultimately store all data on physical hard disks. Standard hard disks that range in size from 36 gb to 1 tb are currently used for this purpose. Since the maximum number of hard disks that can be used is often limited, the size of the hard disk used gives an indication of the maximum capacity of the overall disk subsystem.

When selecting the size of the internal physical hard disks it is necessary to weigh the requirements of maximum performance against those of the maximum capacity of the overall system. With regard to performance it is often beneficial to use smaller hard disks at the expense of the maximum capacity: given the same capacity, if more hard disks are available in a disk subsystem, the data is distributed over several hard disks and thus the overall load is spread over more arms and read/write heads and usually over more i/o channels (figure 2.4). For most applications, medium-sized hard disks are sufficient.



Disk subsystem (small disks) Disk subsystem (large disks)

If small internal hard disks are used, the load is distributed over more hard disks and thus over more read and write heads. On the other hand, the maximum storage capacity is reduced, since in both disk subsystems only 16 hard disks can be fitted.

For applications with extremely high performance requirements should smaller hard disks be considered. However, consideration should be given to the fact that more modern, larger hard disks generally have shorter seek times and larger caches, so it is necessary to carefully weigh up which hard disks will offer the highest performance for a certain load profile in each individual case.

Standard i/o techniques such as scsi, fibre channel, increasingly serial ata (sata) and serial attached scsi (sas) and, still to a degree, serial storage architecture (ssa) are being used for internal i/o channels between connection ports and controller as well as between controller and internal hard disks. Sometimes, however, proprietary – i.e., manufacturer-specific – i/o techniques are used. Regardless of the i/o technology used, the i/o channels can be designed with built-in redundancy in order to increase the fault-tolerance of a disk subsystem. The following cases can be differentiated here:

- active

  In active cabling the individual physical hard disks are only connected via one i/o channel (figure 2.5, left). If this access path fails, then it is no longer possible to access the data.

- active/passive

  In active/passive cabling the individual hard disks are connected via two i/o channels (figure 2.5, right). In normal operation the controller communicates with the hard disks via the first i/o channel and the second i/o channel is not used. In the event of the Failure of the first i/o channel, the disk subsystem switches from the first to the second I/o channel.

- active/active (no load sharing)

  In this cabling method the controller uses both i/o channels in normal operation (figure 2.6, left). The hard disks are divided into two groups: in normal operation the first group is addressed via the first i/o channel and the second via the second i/o channel. If one i/o channel fails, both groups are addressed via the other i/o channel.

- active/active (load sharing)

  In this approach all hard disks are addressed via both i/o channels in normal operation (figure 2.6, right). The controller divides the load dynamically between the two i/o channels so that the available hardware can be optimally utilized. If one i/o channel fails, then the communication goes through the other channel only.

Active cabling is the simplest and thus also the cheapest to realise but offers no protection against failure. Active/passive cabling is the minimum needed to protect against failure, whereas active/active cabling with load sharing best utilises the underlying hard- ware.

Implementation of raid

There are two types of raid implementation, hardware and software.

Software Raid

Software raid uses host-based software to provide raid functions. It is implemented at the operating-system level and does not use a dedicated hardware controller to manage the raid array.

Software raid implementations offer cost and simplicity benefits when com- pared 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, as detailed later in this chapter.
- 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.

Hardware raid

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.

Reference

Book	Author	Priority
Information storage management	G. Somasundaram Alok Shrivastava	1
Storage Network explained : Basic and application of fiber channels, SAN, NAS, iSESI	Ulf Troppens, Wolfgang Mueller-Friedt, Rainer Erkens, Rainer Wolafka, Nils Haustein	2
Nick Antonopoulos, Lee Gillam	Cloud Computing : Principles, System & Application	3