Disk Performance Optimization ch.8

Introduction

In recent years, processors & memory speeds have increased more rapidly than those of hard disk. As a result, processes requesting data from disk tend to experience long service delay. In this chapter, we discuss how to optimize disk performance by recording disk requests to increase throughput, decrease response time & reduce the variance of response times. We also discuss how OSs reorganize data on disk & exploit buffers & caches to boost performance.
Finally, we discuss Redundant Arrays of Independent Disks (RAIDs), which improve disk access times & fault tolerance by servicing requests using multiple disks at once.

Characteristics of Moving-Head Disk Storage

The general structure of hard disk is shown in fig below. In this figure, we notice the followings:

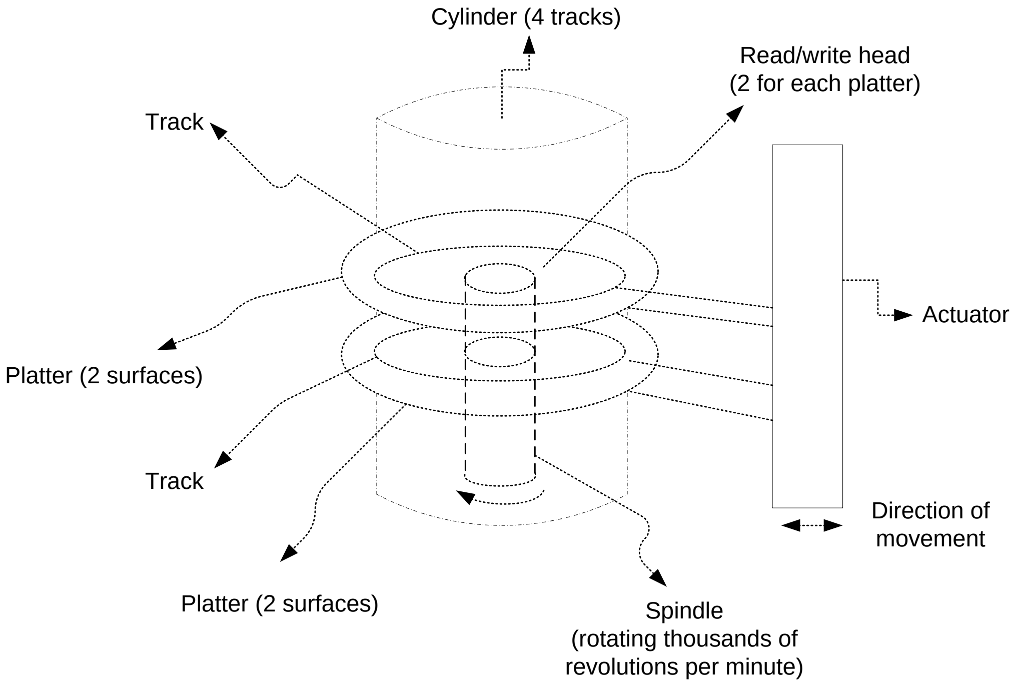

Fig Disk Structure (Schematic Side View)

- The disk storage may consist of several platters & each has a separate read/write moving-head. All heads are fixed to the same actuator & hence move together to select certain cylinder. The cylinder is a set of tracks on all surfaces. Usually, at one time, only one head is active & deals with one track of the whole cylinder. This means that OS has to select the proper head to read/write (r/w) data.
- Each track is divided to several sectors as shown in fig 10.2 each sector is of 512 byte size.

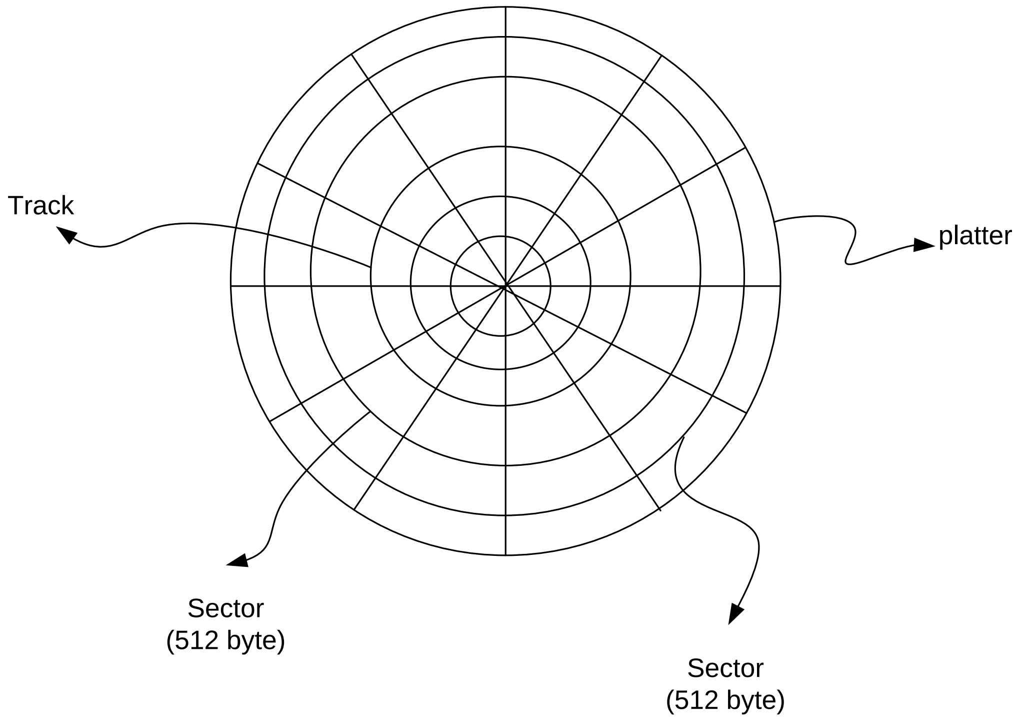

Fig Tracks & Sectors of Disk

From the above, it is clear that for the OS to R/W data from disk, it needs to:

1- Specify the proper surface containing the data & hence the proper moving head.

2- Specify the track & sectors containing the data on that surface.

3- Instruct actuator to move head to the proper track. This movement takes time which is called "Seek Time" & its average value is in the range of few milliseconds (e.g. 7 msec).

4- The platter has to be rotating & the head should wait for the proper sector in the track to get the data.

This time depends on revolution speed & its average value is half of one revolution period & is usually of few milliseconds value (e.g. 4 msec). This time is called "Latency Time".

5- When the head is on the proper sector, it starts reading/writing data & also this process takes time depending on number of sectors to be read.

This time is called "Transmission Time" as shown in Figure.

From the above, It is clear that a few milliseconds are necessary to R/W data from the disk while the CPU can execute millions of instructions in thatv time.

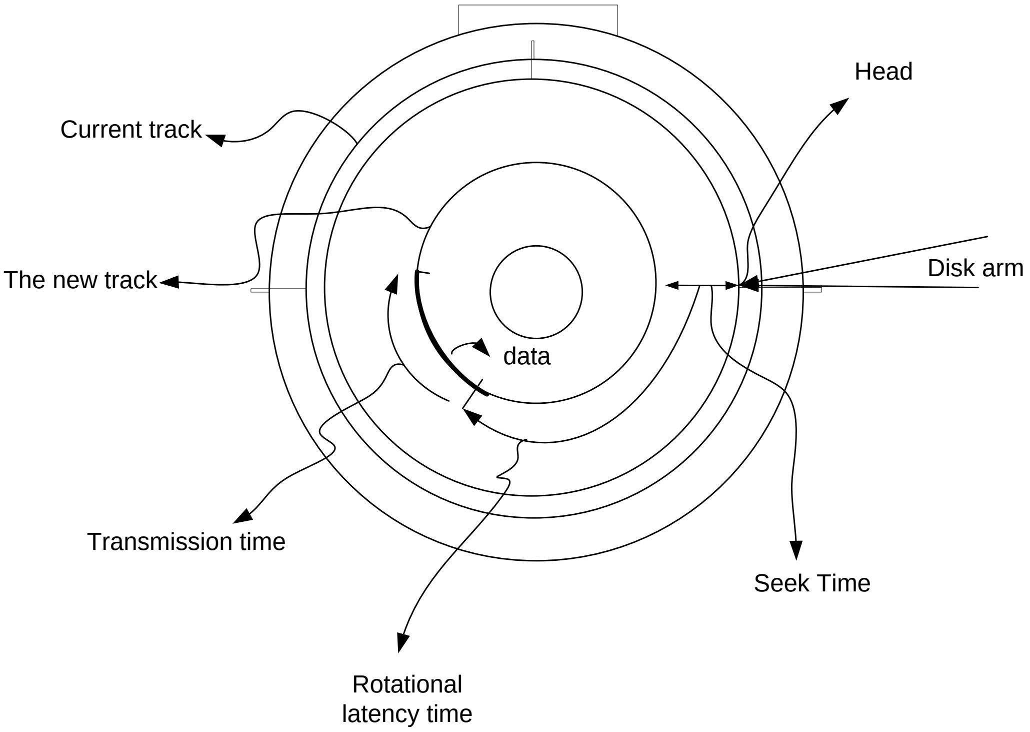

Fig components of disk access (Total time of few milliseconds) e.g. 10 msec

Why Disk Scheduling is Necessary

Many processes can generate requests for reading & writing data on a disk simultaneously. Because these processes sometimes makes requests faster than they can be serviced by the disk, waiting lines or queues build up to hold disk requests. Some early computing systems simply serviced these request on a "First Come First Served FCFS" basis, in which the earliest arriving request is serviced first. FCFS exhibits a random seek pattern in which successive requests can cause time consuming seeks from the innermost to the outermost cylinders (tracks). To reduce the time spent seeking records, it seems reasonable to reorder the request queue in some manners other than FCFS. This process, called disk scheduling, can significantly improve throughput.

The two most common types of scheduling are "Seek optimizing" & "Rotational Optimizing". Because seek times are usually greater than latency times, most scheduling algorithms concentrate on minimizing total seek time for a set of requests.