Review of Disk Scheduling Algorithms and RAID Levels

Priya Dhir, Poonam
Designation: Assistant Prof, A.S.College, Khanna

Abstract
Disk performance management is an important aspect of operating system research and development. One of the main purposes of the operating system for the disk drives is to use the hardware efficiently. So we can meet this goal using fast access time and large disk bandwidth that depends on the relative positions of the read-write head and the requested data. In this paper we perform an experimental study on disk scheduling algorithms which attempt to optimize head movements and guarantees fairness in response time. Various disk scheduling algorithms are available to service the pending requests. Among these disk scheduling algorithms, the algorithm that yields less number of head movement will remain has an efficient algorithm. An important aspect of disk organization scheme to improve the disk reliability. So the different RAID levels help to maintain data consistency on disk. This paper also describes the summary of different scheduling algorithms and RAID levels.

Keywords: Disk Scheduling, FCFS, SSTF, SCAN, CSCAN, LOOK, CLOOK, RAID

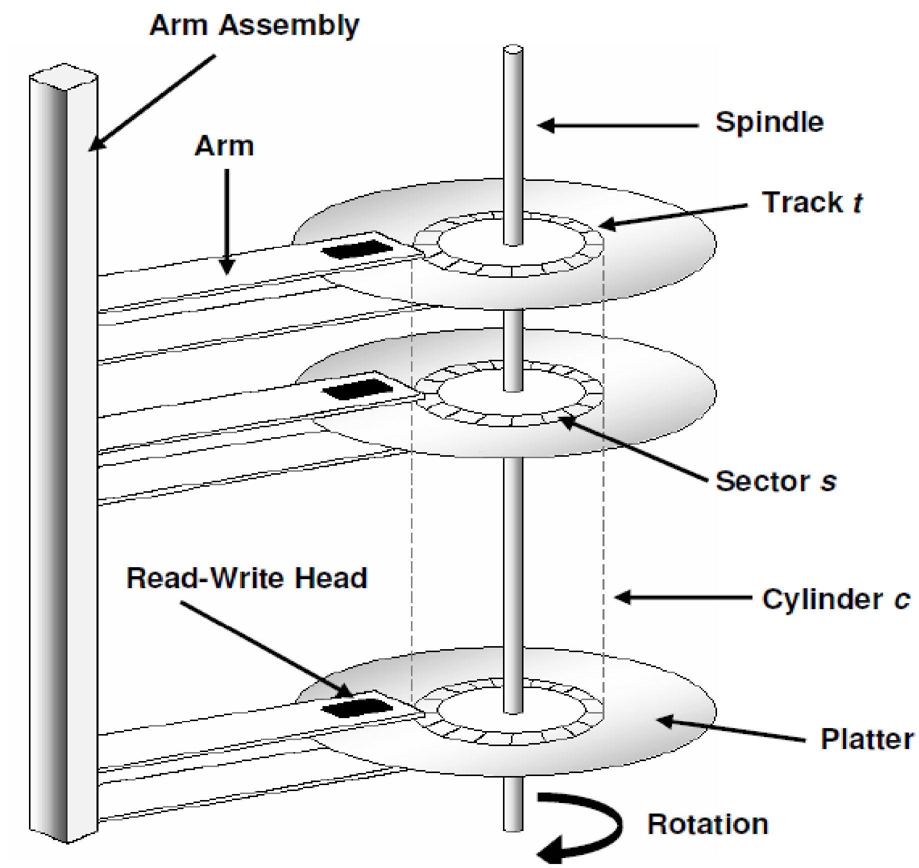
I. Introduction

In multiprogramming operating system there are many processes that may be running simultaneously. Often these processes may requests I/O operation from the hard disk same time. Thus algorithm is used to select which I/O request is going to be satisfied first is called Disk scheduling algorithm. Several algorithms exist to schedule the servicing of disk I/O request. These algorithms are FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK which helps in reducing the average seek time. The main aim of disk scheduling algorithms is to reduce or minimize the seek time for a set of requests. By reducing the average seek time we can improve the performance of disk I/O operation. The two main objectives of disk scheduling algorithm is minimize the response time and maximize the throughput.

Several improvements in disk use technique have been proposed. These techniques involve the use of multiple disks working cooperatively. To improve speed disk stripping uses a group of disks as one storage unit. RAID scheme improve performance and improve reliability of the storage system by storing redundant data. The simplest RAID organization is called mirroring which keeps a duplicate copy of each disk. Another RAID organization which is called block interleaved parity that uses much less redundancy. A parity RAID system has the combined speed of multiple disks and controllers. If a system has many hundred of disks failures may cause the loss of data several times. Thus the redundant data in RAID are used for recovery from disk failure.

II. Disk Structure

Disks are very important for storage of huge data forever. Disks are secondary storage devices of Choices. Data is stored and retrieved in units called disk blocks or pages. Disk is a platter of metal or plastic with a magnet sable coating on it. It consists of platters that spin around the spindle. The surfaces of platters are covered with a magnetic material. Information is stored magnetically on the platters. The surface of platter is logically divided into circular tracks, which are sub-divided into sectors. The set of tracks that are at one arm position form a cylinder. There may be thousands of concentric cylinders in a disk drive and each track may contain hundred of sectors. The components of the disk have been shown in the following figure.



III. Disk Scheduling Criteria

A set of criteria is established in which various scheduling policies are evaluated.

i. Seek Time: The time taken to move disk arm on the heads to the cylinder containing the desired sector.

ii. Rotational latency: The time spent in waiting for desired block to rotate under the head.

iii. Disk bandwidth: Total number of bytes transferred divided by the total time between the first request for service and the completion of last transfer.

iv. Transfer time: The time required for actually moving the data to/from the disk surface.

IV. Classical Disk Scheduling Algorithms and Their Variations

i. First Come First Serve (FCFS)

This is the simplest form of disk scheduling algorithm. It processes the I/O requests in the same order as they arrive. In this scheduling operating system creates a queue which contains the sequence order in which the CPU will execute the processes. In this the job which had requested first will be executed first by the CPU and the Jobs those are entered later will be executed according to the order in which they are entered. Thus this technique improves the response time as a request gets response in fair amount of time. .Consider an example a disk queue with requests for I/O to blocks on cylinders:

Queue (10-199): 23, 89, 132, 42, 187 and head current position is: 100.

Figure 1.1 shows an example of FCFS algorithm.

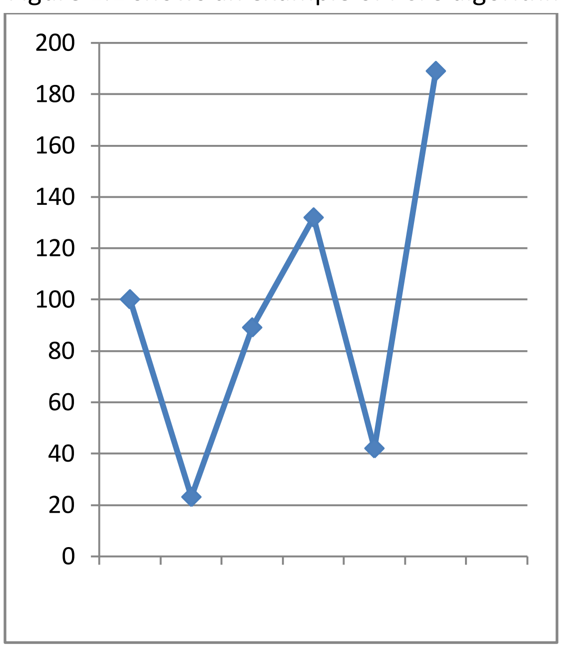


Figure 1.1

Total head movement = (100-23) + (23-89) + (89-132) + (132-42) + (42-187) =421

In FIFO total head movements: 421

ii. Shortest Seek Time First (SSTF)

The SSTF algorithm select the request with the minimum seek time from the current head position. The Shortest seek first algorithm determines which request is closest to the current position of the head, and then services that request next. SSTF picks the request from the queue closest to the current read/write head location. The throughput in this case is much better than in FIFO. However if new requests with shorter seek time arriving then some old processes may have to wait for a long time. Thus it may cause starvation of some request. Consider an example a disk queue with requests for I/O to blocks on cylinders:

Queue (10-199): 23, 89, 132, 42, 187 and head current position is: 100.

Figure 1.2 shows an example of SSTF algorithm.

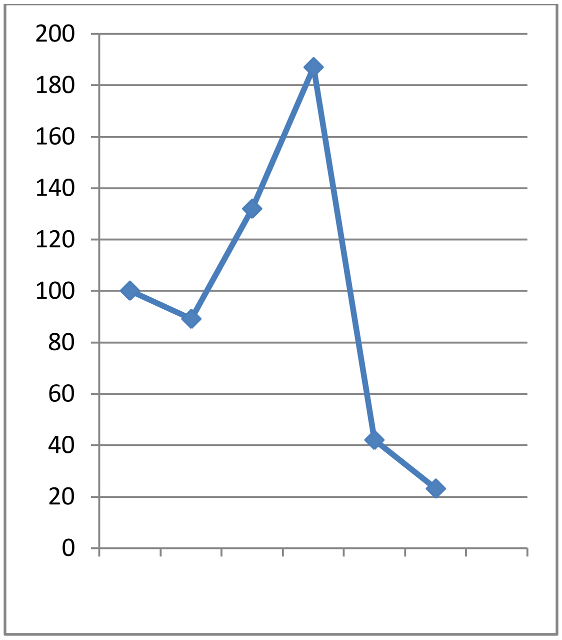


Figure 1.2

Total Head Movement = (100-89) + (89-132) + (132-187) + (187-42) + (42-23) =273

In SSTF total head movements:273

iii. SCAN

This method works like an elevator. The disk starts at one end of the disk and moves toward the other end, servicing the requests until it gets to other end of the disk. Sometimes It is also called elevator algorithm. Its throughput is better than FCFS.The scan algorithm is more fair than SSTF as far as starvation of request is considered. Consider an example a disk queue with requests for I/O to blocks on cylinders:

Queue (10-199): 23, 89, 132, 42, 187. And head current position is: 100.

Figure 1.3 shows an example of SCAN algorithm

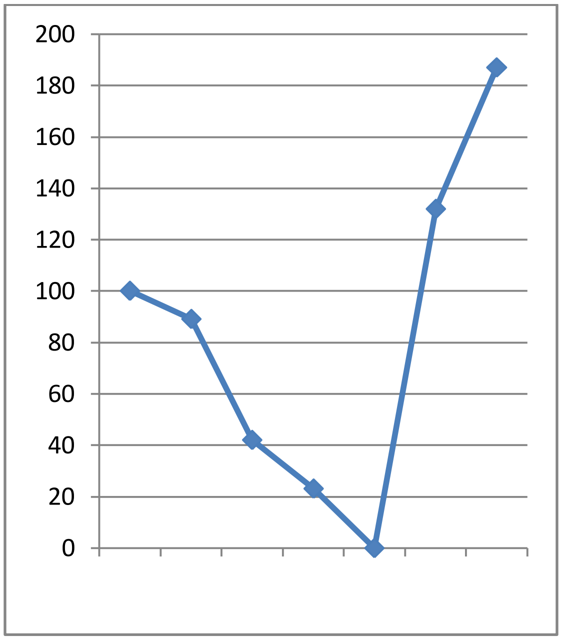


Figure 1.3

Total Head Movement = (100-89) + (89-42) + (42-23) + (23-0) + (0-132) + (132-187) =287

In SCAN total head movements: 287

iv**. Circular Scan(C-SCAN)**

Circular scan (C-SCAN) scheduling is a variant of scan designed to provide a more uniform wait time. In SCAN, the most outer and most inner cylinder has less opportunity to be accessed than the middle. C-SCAN eliminates this by satisfying requests only when the head moves in one direction and not satisfying any requests when it moves back. Consider, for example a disk queue with requests for I/O to blocks on cylinders:

Queue 23,89, 132, 42, 187. And head current position is: 100

Figure 1.4 shows an example of C-Scan algorithm.

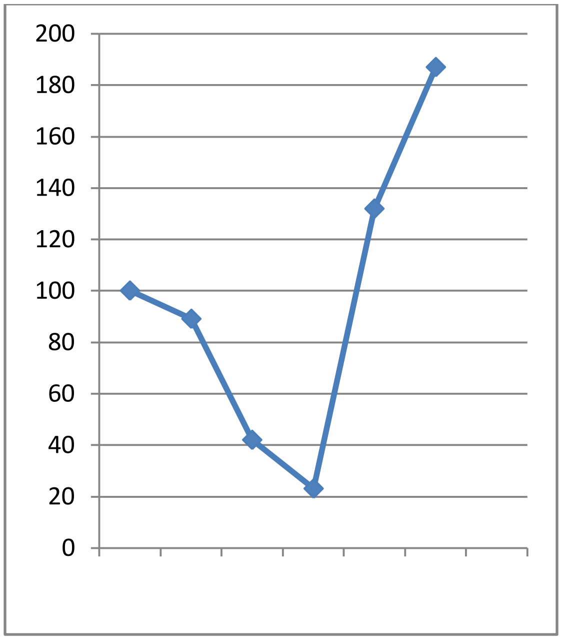


Figure 1.4

Total Head movement: (100-89) + (89-42) +(42-23)+(23-132) +(132-187)= 241

In C-SCAN total head movements: 241

v. LOOK

As in SCAN, requests are served when the head moves in both directions. The difference is that LOOK uses information about the requests to change the direction of the head. When it knows that there are no requests beyond the current point, it changes the direction of the head. This improves the throughput and response time. . Consider, for example a disk queue with requests for I/O to blocks on cylinders:

Queue (10-199):23, 89, 132, 42, 187. And head current position is: 100.

Figure 1.5 shows an example of C-Scan algorithm.

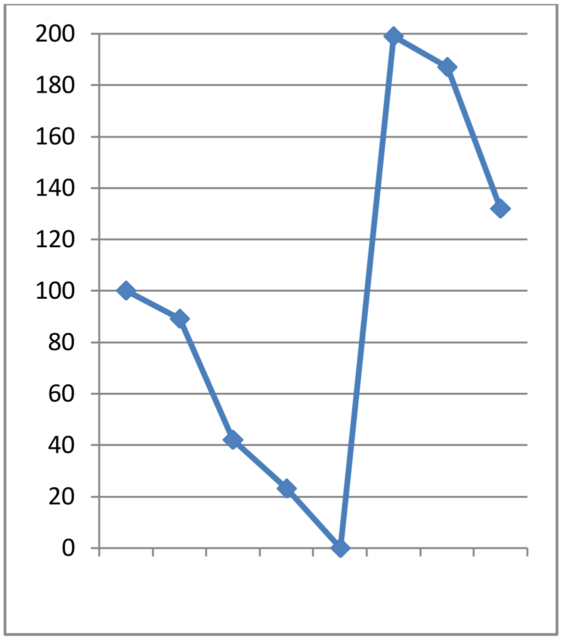


Figure 1.5

Total Head movement: (100-89) +(89-42) +(42-23)+(23-0) +(0-199)+(199-187)+(187-32)=366.

In LOOK total head movements: 366

vi. C-LOOK

This is the variation of LOOK where requests are satisfied only when the head moves outwards, as in C-SCAN. Thus, no request is satisfied when the head moves inwards after determining that no requests are there beyond the current point. . Consider, for example a disk queue with requests for I/O to blocks on cylinders: Queue (10-199): 23, 89, 132, 42, 187. And head current position is: 100.Figure 1.6 shows an example of C-Look algorithm.

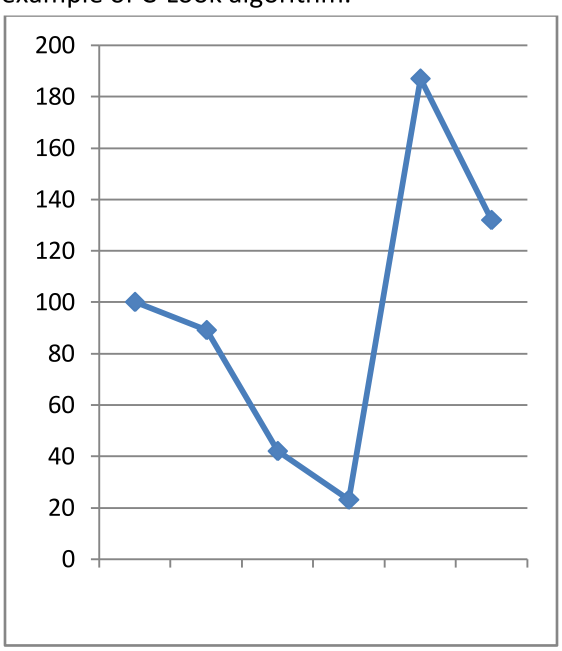


Figure 1.6

Total Head movement: (100-89) + (89-42) + (42-23) + (187-23) + (187-132)=258
In C-Look total head movements are: 258

Summary of Disk Scheduling algorithms:

Algorithm	**Main Idea**
FIFO	Jobs are taken into service in their coming order.
SSTF	Seek reducing algorithm. The next job to be processed is chosen as the nearest job in seek distance to the job under service.
SCAN	A modification of SSTF. It applies SSTF in one direction of disk, when head comes to the edge it reverses the direction and makes a new SSTF application on reverse direction until the edge.
C-SCAN	Cyclic SCAN makes the head run only in single direction and when it comes to the edge it returns to the starting point and redo the run in same direction.
LOOK	A modification of SCAN. While the head goes in a direction, if there is no job in that direction, it instantaneously reverses the direction instead of going to the disk surface's edge and apply SSTF in new direction.
C-LOOK	A modification of LOOK to apply it in single direction only.

V. RAID (Redundant Array of Independent Disks)

One of the important disk organization schemes to improve the disk reliability is RAID. It is a short form for Redundant Array of Independent Disks. It is a collection of multiple disks that work cooperatively. The reliability of storage system is improved by storing redundant data. This enables the system to continue operation even one of the disks crashes. RAID will be used as the basis for examination of the performance, cost and reliability of disk arrays. There are a number of RAID levels.

RAID Level 0 – It provides data striping but no redundancy. Disk stripping is a technique for spreading data over multiple disks. It can speed up operation that retrieves data from disk storage. The computer system breaks a body of data into units and spread these units across the multiple disks. This improves performance as the time required to transfer a block to memory is reduced.

RAID Level 1- It provides disk mirroring. Disk mirroring is a technique in which data is written to multiple disks simultaneously. This way if one of the disk drives fails the system can switch to the other disk without any loss of data.

RAID Level 3- It is same as level 0 but also reserves one dedicated disk for error correction data. It provides good performance and some level of fault tolerance also.

RAID Level 5- It provides disk stripping in which data as well as error correction information is spread across number of disks used in the array. This results in excellent performance and good fault tolerance.

RAID Level 6- This level is very much similar to RAID level 5 but it has more one advantage added to it. The added advantage is that it can sustain 2 drive failures instead of 1. This is achieved with the help of parity.

RAID Level 10- Raid 10 is a good solution that will give you both the performance advantage of raid 0 and also the redundancy of raid 1 mirroring. Raid 10 was made by a combination of raid 0 and raid 1. And hence you get qualities of both the raid levels.

Summary of different RAID levels:

- RAID 0 uses striping for high performance. Raid 0 cannot be considered as RAID as it does not provide fault tolerance.
- RAID 1 uses mirroring for redundancy.
- RAID 5 uses striping as well as parity for redundancy. It is well suited for heavy read and low writes operations.
- RAID 6 uses striping and double parity for redundancy.
- RAID 10 is a combination of raid 1 and raid 0. It also provides heavy redundancy because of mirroring and also provides performance as the data is striped across multiple raid 1 groups.

Conclusion

Disk scheduling algorithms are used to allocate the services to the I/O requests on the disk and improve its performance. When selecting a Disk Scheduling algorithm, performance depends on the number and types of requests. SSTF is common and has a natural appeal. SCAN gives better performance than FCFS and SSTF. Then we examine different RAID levels which are used to improve the performance of the disk arrays.

Refernces:

[1] Operating System Principles (6th edition) Abraham Silberschatz, Peter Bare Galvin, Greg Gagne

[2] H. M. Deitel, “Operating Systems”, 2nd Edn., Pearson Education Pte. Ltd., 2002, ISBN 81-7808-035-4.

[3] Sandipon Saha,Md. Nasim Akhter, Mohammod Abul Kashem, A New Heuristic Disk Scheduling Algorithm

[4] Tarsem Singh A Comparative Study of Disk Scheduling Algorithms , *International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 1, Jan - Feb 2016*

[5] Sourav Kumar Bhoi, Sanjaya Kumar Panda, Imran Hossain Faruk, Design and Performance Evaluation of an Optimized Disk Scheduling Algorithm (ODSA), *International Journal of Computer Applications (0975 – 8887) Volume 40– No.11, February 2012*