Reformed FCFS Disk Scheduling Algorithm

Saman Rasool
Department of Computer Science & Engineering
Faculty of Engineering & Technology
Jamia Hamdard, New Delhi

Ritika Gakher
Department of Computer Science & Engineering
Faculty of Engineering & Technology
Jamia Hamdard, New Delhi

ABSTRACT

In this paper an improved version of the First come first serve (FCFS) disk scheduling algorithm is provided. In the proposed approach we have made use of maximum and minimum service. It provides fast access time and dish bandwidths for disk drives which makes the efficient usage of hardware.

Keywords

Disk Scheduling, FCFS, SSTF, C-LOOK, C-SCAN, Total Head Movement

1. INTRODUCTION

Disks are the secondary storage for bulk data. Present disks are inferred as a logical blocks of large one dimensional array wherein a logical block is considered as the smallest unit of transfer. One of the major responsibilities of the operating system is the efficient use of the hard disk. For meeting this responsibility it requires to have a fast access time and disk bandwidth. In a multiprogramming systems, where there are multiple processes, the disk have many processes pending in a queue, therefore, when each request is processed the operating system picks one of the pending processes from the queue to be processed next.

Access time has two major factors:

Seek time: Time taken by disk arm to move heads to the cylinder containing described sector.

Rotational latency: It is the additional waiting time for disk to rotate on the desired sector to disk head.

There are 5 disk scheduling algorithms which are as follows

FCFS: It is the simplest algorithm and performs operation in order in which requests are received. It does not provide fastest service. Also no rearranging of request is performed.

SSTF: In this request is serviced according to the next shortest distance. It is another form of SJF scheduling. It can cause starvation [6].

SCAN: The working of this approach is same like that of elevator so called elevator algorithm. The disk arm starts from one end of the disk, and moves towards the other end. It services requests till it gets to the other end of disk, where the head movement gets reversed and servicing is continued [6].

C-Scan: It is an improvised version of SCAN aimed at providing a homogeneous wait time wherein it services the requests moving from one end to another. When the head reaches the other end it immediately hops back to the start of the disk without servicing any request. It provides more uniform wait time than SCAN algorithm [1].

C-LOOK: It is a modified version of C SCAN algorithm. The arm services the request only till the last request then it reverses the direction immediately without going till the end of the disk [7].

2. RELATED WORKS

In [1] the researchers suggest a new disk scheduling algorithm that aims at reducing the number of head movements. In [2] the authors intent to improve the existing FCFS algorithm so that the seek time and rotational latency can be reduced by reducing average seek distance. In [3] the investigators looks at the uncertainty associated with the disk scheduling combining two aspects by using Fuzzy logic approach to improve the comprehensive performance. In [4] the author suggests an improved FCFS disk scheduling algorithm in which the disk head is moved in order to serve the first request and if there is any request waiting from the current disk head position, the first request is served. Afterwards the disk head moves towards next request waiting in the queue. The left out request are also further dealt with the same procedure. In [5] the researchers have combined the concept of pipelining with CPU scheduling, thereby improving the performance of CPU scheduling. In [6] the author aims at improving the convergence of genetic algorithm and by introducing the probability of simulated annealing as the principle for recognition of new trial solution.

In [9] the novelists introduce two new algorithms for real time systems i.e. SSEDO and SSEDV. In [10] the paper aims at deriving upper bounds for disk utilization factor. In [11] the researchers introduce two new algorithms which are SRLF and SATF for reduced rational latency.

3. PROPOSED APPROACH

In the first come first serve disk scheduling algorithm the request that arrives first is served first. In this paper, a modification of the FCFS is proposed wherein we proceed as follows.

1. Find k = max – min

   Where max = maximum request

   Min = min request

2. If K< current head position, then serve the request in decreasing order (from current) and those request that are greater than current are arranged in increasing order and served after all the above request are served.

3. If k > current head position, then the serve the request next to the current request in increasing order until the highest request and then serve the current request.

4. EXAMPLES USING AND COMPARING THE CONVENTIONAL FCFS, SSTF AND C-LOOK ALGORITHMS AND THE PROPOSED APPROACH

4.1.1 Example 1: Using FCFS disk scheduling algorithm

Current head movement=10

Sequence of requests: 3, 7, 26, 17, 21

Rearranged Sequence of request: 10, 3, 7, 26, 17, 21

Calculating Total head movement = | (3-10) + (7-3) + (26-7) + (21-17) |

Total head movement: 7+4+19+9+4= 43

4.1.2 Example 1: Using C-SCAN disk scheduling algorithm:

Current head movement=10

Sequence of requests: 3, 7, 26, 17, 21

Rearranged Sequence of request: 10, 17, 21, 26, 0, 3, 7

Calculating Total head movement =| (17-10) + (21-17) + (26-21) + (0-26) + (3-0) + (7-3) |

Total head movement: 7+4+5+26+3+4= 49

4.1.3 Example 1: Using SSTF disk scheduling algorithm

Current head movement=10

Sequence of requests: 3, 7, 26, 17, 21

Rearranged Sequence of request: 10, 7, 3, 17, 21, 26

Calculating Total head movement = | (7-10) + (3-7) + (17-3) + (21-17) + (26-21) |

Total head movement: 3+4+14+4+5=30

4.1.4 Example 1: Using proposed approach:

Current head movement=10

Sequence of requests: 3, 7, 26, 17, 21

K= max-min

= 26-3 =23

If K> current

As 23< 10

Rearranged Sequence of request: 17, 21, 26, 10

Total head movement= | (21-17) + (26-21) + (10-26) |

= 4+5+16= 25

Hence the proposed approach is improved.

Table1: Comparison Table for Example 1

FCFS	SSTF	C-SCAN	PROPOSED
10	10	10	17
3	7	17	21
7	3	21	26
26	17	26	10
17	21	0	
21	26	3	
		7	
THM=43	THM=30	THM=49	THM=25

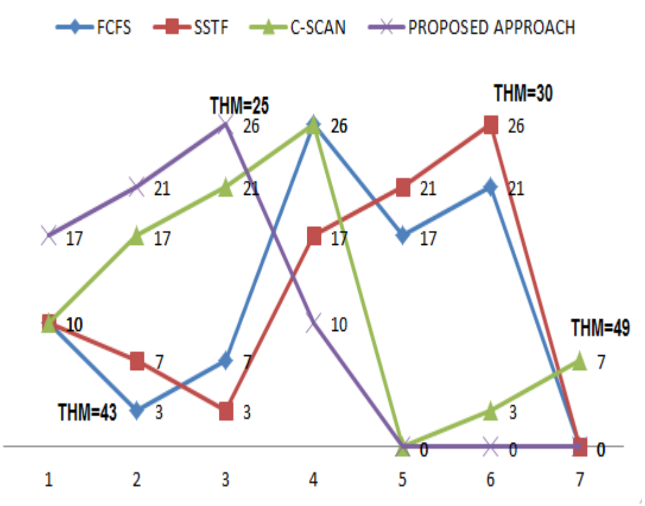

Fig 1: Graph defining the total head movement for FCFS, C-SCAN, SSTF and proposed approach of example 1

4.2.1 Example 2: Using FCFS disk scheduling algorithm

Current head movement=60

Sequence of requests: 90, 40, 45, 67, 53

Rearranged Sequence of request: 60, 90, 40, 45, 67, 53

Calculating Total head movement =| (90-60) + (40-90) + (45-40) + (67-45) + (53-67) |

Total head movement: 30+50+5+22+14=121

4.2.2 Example 2: Using C-SCAN disk scheduling algorithm

Current head movement=60

Sequence of requests: 90, 40, 45, 67, 53

Rearranged Sequence of request: 60, 67, 90, 0, 40, 45, 53

Calculating Total head movement =| (67-60) + (90-67) + (0-90) + (40-0) + (45-40) + (53-45) |

Total head movement: 7+23+90+40+5+8= 173

4.2.3 Example 2: Using SSTF disk scheduling algorithm

Current head movement=60

Sequence of requests: 90, 40, 45, 67, 53

Rearranged Sequence of request: 60, 53, 45, 40, 67, 90

Calculating Total head movement = | (53-60) + (45-53) + (40-45) + (67-40) + (90-67) |

Total head movement: 7+8+5+27+23= 70

4.2.4 Example 2: Using proposed approach:

Current head movement=60

Sequence of requests: 90, 40, 45, 67, 53

K= max-min

= 90-40 =50

As, K< current

As 50< 60

Rearranged Sequence of request: 60, 53, 45, 40, 67, 90

Total head movement= | (53-60) + (45-53) + (40-45) + (67-40) + (90-67) | = 7+8+5+27+23= 70

Hence the proposed approach is improved.

Table2: Comparison table for Example 2

FCFS	CSAN	SSTF	PROPOSED
60	60	60	90
90	67	53	40
40	90	45	45
45	40	40	67
67	45	67	53
53	53	90	
THM=121	THM=173	THM=70	THM=70

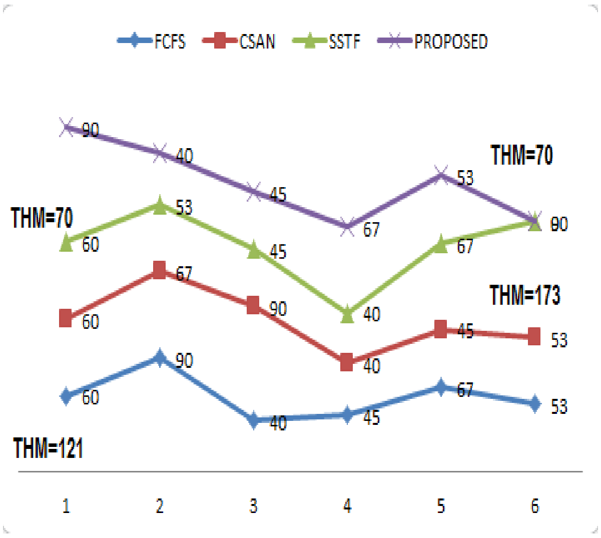

Fig 2: Graph defining the total head movement for FCFS, C-SCAN, SSTF and proposed approach of example 2

4.3.1 Example 3: Using FCFS disk scheduling algorithm

Current head movement=35

Sequence of requests: 10, 30, 40, 20, 15, 25

Rearranged Sequence of request: 35, 10, 30, 40, 20, 15, 25

Calculating Total head movement = | (10-35) + (30-10) + (40-30) + (20-40) + (15-20) + (25-15) |

Total head movement: 25+20+10+20+5+10=90

4.3.2 Example 3: Using C-SCAN disk scheduling algorithm

Current head movement = 35

Sequence of requests: 10, 30, 40, 20, 15, 25

Rearranged Sequence of request:

35, 40, 0, 10, 15, 20, 25

Calculating Total head movement = | (40-35) + (0-40) + (10-0) + (15-10) + (20-15) + (25-20) |

Total head movement: 5+40+10+5+5+5= 70

4.3.3 Example 3: Using SSTF disk scheduling algorithm

Current head movement=35

Sequence of requests: 10, 30, 40, 20, 15, 25

Rearranged Sequence of request: 35, 30, 25, 20, 15, 10, 40

Calculating Total head movement = | (30-35) + (25-30) + (20-25) + (15-20) + (10-15) + (40-10) |

Total head movement: 5+5+5+5+5+30= 55

4.3.4 Example 3: Using proposed approach

Current head movement=35

Sequence of requests: 10, 30, 40, 20, 15, 25

K= max-min

= 40-10 =30

As, K< current

As 30<35

Rearranged Sequence of request: 35, 30, 25, 20, 15, 10, 40

Total head movement=| (30-35) + (25-30) + (20-25) + (15-20) + (10-15) + (40-10) |

=5+5+5+5+5+30= 55

Hence the proposed approach is improved.

Table3: Comparison table for Example 3

FCFS	CSCAN	SSTF	PROPOSED
35	35	35	35
10	40	30	30
30	0	25	25
40	10	20	20
20	15	15	15
15	20	10	10
25	25	40	40
THM=90	THM=70	THM=55	THM=55

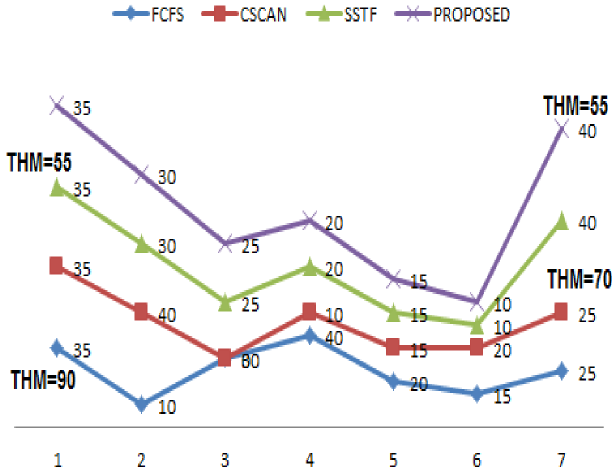

Fig 3: Graph defining the total head movement for FCFS, C-SCAN, SSTF and proposed approach of example 3

5. CONCLUSION

The proposed approach is observed, and it shows an upper hand over the conventional First Come First Serve Disk scheduling algorithm. A similar approach can also be used in the real time operating systems with minor alterations.

6. REFERENCES

[1] Sandipon Saha, Md. Nasim Akhter, Mohammod AbulKashem, "A New Heuristic Disk Scheduling Algorithm", International Journal of Scientific & Technology Research Volume2, Issue 1, January 2013.

[2] Priya Hooda, Supriya Raheja, "A New Approach to Disk Scheduling Using Fuzzy Logic", Journal of Computer and Communications, 2014.

[3] Manish Kumar Mishra , An Improved FFS(IFCFS) Disk Scheduling Algorithm, International Journal of Computer Applications (0975-888), Volume 47- No. 13, June 2012.

[4] Hamid Husain, Karnav Gupta, Shradha Taneja, "Modified First Come First Serve (MFCFS)"

[5] Himanshi Arora, Deepanshu Arora, Bagish Goel, Parita Jain, "An Improved CPU Scheduling Algorithm", International Journal of Applied Information Systems (IJAIS), Volume 6– No. 6, December 2013

[6] R Muthu Selvi, R Rajaram, A genetic based approach for multi objective optimization of Disk Scheduling to reduce the completion time and missed task. International Journal of Information Technology Convergence and Services, Volume.1, No. 4 , August 2011.

[7] Shenze Chen, John A. Stankovnic, James F. kurose, Don Towsley, "Performance Evaluation of two New Disk Scheduling Algorithm for Real Time System". Office of Naval under contract N00014-87-K-796 by NSF under contract IRI-8908693, and by NSF equipment grants CERDCR 8500332.

[8] Kitae Hwang , Heonshik Shin, New Disk Scheduling Algorithm for Reduced Rotational Latency.pp 395 – 402.

[9] Giorgio Gallo, Federico Malucelli, Martina Marre, Hamiltonian Path Algorithms for Disk Scheduling, pp 1-8.

[10] Silberschatz ,Galvin and Gagne, Operating systems concepts, 8th edition, Wiley, 2009.

[11] D.M. Dhamdhare, Operating Systems, Tata McGraw Hill, Second Edition,2006.

[12] S Yashvir, Om Prakash, Selection of Scheduling Algorithm, IJASCSE Volume 1, Issue 2 , 2012.