The Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm
Ahmed Mohamed Mal-Allah Al-Suffar
College of Engineering, AL-IRAQIA UNIVERSITY
ahmed_rnalallah@yahoo.com

ABSTRACT
Multi-programming needs the disk scheduling to specify at which time which process will be executed. The factors of enhancing disk scheduling algorithms by; reduction of the response time and increasing performance have been the most important subjects of efficient researches. Generally, there is an importance of minimize the impact of the slowness of accessing and retrieving data from disk drives proportionally with the accessing of CPU and memory.
The proposal succeeded in achieving the reduction of the total seek time for LOOK algorithm by solid estimation method to determine the suitable track of head movement to serve requests resided at the side of the shortest remote request firstly to achieve its goal.

Key Words: Disk Scheduling Algorithm, Shortest Remote Request, LOOK Algorithm, Total Seek Time, Head Movement.

1. Introduction

Disk Head scheduling algorithms have been studied for many years to increase disk I/O performance. Performance depends on the number and type of request, always considers only the seek distance. Therefore, disk scheduling algorithm objective is to minimize seek time to be the optimal. (Lister, 1984)

SSTF is a common algorithm. It has a natural appeal because it increased performance over FCFS. LOOK algorithm behave almost identically to SSTF in selecting its path by accessing the shortest request to serve it firstly before going back to achieve accessing all requests at other side, (Stallings, 2009).

LOOK improves SSTF in somewhere by avoiding the starvation problem of it because LOOK is biased against the area recently traversed and heavily favors tracks clustered at the outermost or innermost edge of the platter. Therefore, they seem reasonable choice as a default algorithm. Herein, LOOK algorithm has proportionally its
drawback of high total requests' seek time, (Gagne et al., 2013).

The proposal, "The Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm", is focused in how to enhance the features of LOOK algorithm to reduce the total seek time by using a new technique of estimating method to determine the initial head movement to serve requests reside at any side of shortest remote request.

2. Related Work

Many researchers have been done to enhance the disk performance. Martens & Katachabaw (2006) developed a new disk scheduling algorithm focuses on dynamic scheduling algorithm selection and tuning. Dimitrijevic et al. (2005) have presented preemption I/O, which divides disk I/O requests into small temporal units of disk commands to improve the preemptibility of disk access. Alsuffar (2006) Improved Disk Scheduler Shortest Seek-Time First to be Circular (C-SSTF)

3. Disk Drive Background

A hard disk drive is a collection of plates called platters with two surfaces, covered with a magnetic material, divided into circular tracks. A group of tracks are positioned on top of each other from a cylinder. There is a head connected to an arm for each surface, which handles all I/O operations.

The disk drive is the mechanical part including the device motor, the read/write heads, and the associated logic. The other part, called "disk controller", determines the logical interaction in the computer. The controller takes instructions from the CPU and orders the disk drive to carry out the instruction. The scheduler used for divide the total time of the CPU between the numbers or processes so that the processes can execute concurrently at a single time, (Tanenbaum, 2007).

Multi-parts address reference information on the disk that includes the drive number, the surface, and the track. As basic, all tracks of both surfaces of one drive are called cylinder (Fig 1) can be accessed without moving the heads.
Information is written in blocks within each track. The blocks may be a fixed size, specified by the hardware. These are called sectors. Each sector can be separately read or written. Alternatively, the information on a track may be composed of variable length blocks, separated by record gaps. This scheme, even with hardware-variable disk blocks, will pick a block size and fix it in software. In either case, information is read and written in blocks, (Peterson & Silberschatz, 1985). Blocking and unblocking of records in software can easily hide the fixed or variable nature of the physical block size.

Finally, disk speed is comprised of three parts (Choffnes et al., 2004; Gagne et al., 2013):

1- Seek time: To access a block on the disk, the system must first move the head to the appropriate track or cylinder. This head movement and the time needed is called seek time.
2- Latency time: Once the head is at the right track, it must wait until the desired block rotates under the read/write head. This delay is latency time.
3- Transfer time: The actual transfer of data between the disk and main memory can take place. The last part is transfer time.

Therefore the total time to service a disk request is the sum of the seek time, latency time, and transfer time. Studies had been made for many years to increase disk I/O performance. Most of the disks are characterized as a linear seek time and their seek time is responsible for the most time of disk access. Therefore, the existing disk scheduling algorithms have focused on the reduction of the average seek distance. Seek time has improved greatly and today’s disks usually have nonlinear seek time characteristics. The first and third tasks can be reduced, with careful coding, to several hundred instructions. These tasks may take from 1 to 100 microseconds, (Gagne et al., 2013).

For each I/O request, firstly, the particular head is selected. It is then moved over
the destination track. The disk is then rotated to position the desired sector under the head and finally, the read/write operation is performed. Literature mentioned FCFS, SSTF, SCAN, LOOK algorithms as common algorithms:

It is worthwhile to know that there are several factors to determine which algorithm to choose: (Peterson & Silberschatz, 1985)

- The algorithm must allow for the most efficient use of hardware.
- The file allocation method used on the disk.
- The fairness of the algorithm.
- The size of the queue.

4. Implementation and Experimental Work

In LOOK scheduler, the default movement of disk head is usually to start from the current head position selecting to serving the closest request in any direction it could be. It continues accessing the requests at particular site even last request at that side before returning back to other side, (Tanenbaum, 2007).

In the first instant, it seems reasonable to serve the nearest requests firstly, before moving the head far away to serve other outermost requests but the proposed scheduler "Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm" does not recognize this theory. Herein, some observations (calculations) to be done have been needed before the head being qualified to pick the optimal direction. Therefore, it was never care to start serving from this site or that or even from the current head position toward nearest request or not, but just it serves toward the direction of minimum distance between it and the remote request in any extreme it could be. After the head has finished all requests at that side, it reverses to other end immediately.

Obviously, in dynamic workload, the (SSRRF) scheduler usually gives equitable services to all requests in both directions. Through achieving accessing requests, one by one till the last one in that extreme cylinder, in the direction of the shortest extreme before it enforced to reverse back to serve the requests in the opposite direction until the final request in the second extreme cylinder.

As in LOOK algorithm, if enough requests for the first extreme cylinder keep coming, the head would never mind to them because the strategy does not care to the requests newly got after the head had been traversed. Therefore, in new strategy there are never starving requests in any extreme could be.
Thereby, the new proposal "The Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm" enhanced LOOK algorithm by solving all expected problems surrounded it in a transparent way. Implementation is observed to yield large benefits on a range of decreasing total seek time, in other words, increasing throughput of disk-intensive workload.

The scope of implementation in this research will cover the proposal after traditional LOOK algorithm with some explanation for their features.

To declare the differences between the proposal and traditional LOOK algorithm let suppose the certain set of parameters as conditions needed to implement algorithms in literature:

- Suppose that the disk has 200 tracks (0 to 199).
- Assume that the initial disk head position is at track 53.
- Suppose that the list of (8th) requests consist of: 65, 67, 37, 14, 183, 124, 122, 98.
- Assume that context switching the head from one track to the next takes 0 ms.
- Ignore Latency and Transfer time.

### 4.1 Traditional LOOK algorithm

LOOK scheduler may neither behave identically to any algorithm. That is LOOK avoids the starvation problem of SSTF and biased against the area recently traversed and heavily favors tracks clustered at the outermost and innermost edges of the platter. Dimitrijevic et al. (2005) Also, worthily to say that LOOK has obvious optimization of having the read/write head reversed when the last track in that direction is serviced as in (Fig 2), while SCAN reaches the edge of the disk (or track 0), before reverses direction, so it has additional seek time than LOOK, (Gagne et al., 2013)

**Advantages:** It has obvious optimization of having a good "locality" as compare to SCAN, that is the read/write head reversed when the last track in any direction has been serviced and prevents starvation of some requests in any extreme cylinder.

**Disadvantages:** It does not guarantee the movement of the head to the best and suitable direction because the default head movement is to the request of shortest seek time firstly, (Tanenbaum, 2007).

**Exp:** The current head position is (53) and the Requests are: 65, 67, 37, 14, 183, 124, 122, 98.
(Table 1) demonstrate the calculating of seek time for traditional LOOK disk scheduling algorithm from (Fig 2).

<table>
<thead>
<tr>
<th>Request Position</th>
<th>Tracks Traversed T.</th>
</tr>
</thead>
<tbody>
<tr>
<td>53 - 65</td>
<td>12</td>
</tr>
<tr>
<td>65 - 67</td>
<td>2</td>
</tr>
<tr>
<td>67 - 98</td>
<td>31</td>
</tr>
<tr>
<td>98 - 122</td>
<td>24</td>
</tr>
<tr>
<td>122 - 124</td>
<td>2</td>
</tr>
<tr>
<td>124 - 183</td>
<td>59</td>
</tr>
<tr>
<td>183 - 37</td>
<td>146</td>
</tr>
<tr>
<td>37 - 14</td>
<td>23</td>
</tr>
<tr>
<td>Total Seek Time</td>
<td>299</td>
</tr>
<tr>
<td>Average</td>
<td>37.375</td>
</tr>
</tbody>
</table>

4-2 The Proposal; "Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm"

According to the default strategy of SSTF, SCAN, and LOOK schedulers the first head movement is to the side of either increasing or decreasing track numbers contains proportionally the closest request to the current head position to serve it firstly. While the proposed scheduler (SSRRF) implements a new intelligent method to control the first
head movement after observing (controlling) the remote outermost requests at both edges to determine which side has shorter remote request's distance than other. So, herein, once the disk head starts moving to achieve the set of requests in queue, it does not like other schedulers that it doesn't care to select site of nearest request or not as in (Fig 3).

*Exp:*
1- If the current head position is (53) and the Requests are: 65, 67, 37, 14, 183, 124, 122, 98.
2- Implements a new intelligent method to control the first head movement to select the suitable path after observing (controlling) the remote outermost requests at both edges to determine which side has shorter distance of remote request than other (Fig 3).

**First path is:** 53 - 14 = | 39 | Total seek time to Left side . *(which is shorter path)*

**Second path is:** 53 - 183 = | 130 | Total seek time to Right side..*(which is longer path)*

(Table 2) demonstrate the calculating of seek time for traditional LOOK disk scheduling algorithm from (Fig 3).
Table (2): Calculating Seek Time for Proposal LOOK Disk Scheduling Algorithm

<table>
<thead>
<tr>
<th>Request Position</th>
<th>Tracks Traversed T.</th>
</tr>
</thead>
<tbody>
<tr>
<td>53 - 37</td>
<td>16</td>
</tr>
<tr>
<td>37 - 14</td>
<td>23</td>
</tr>
<tr>
<td>14 - 65</td>
<td>51</td>
</tr>
<tr>
<td>65 - 67</td>
<td>2</td>
</tr>
<tr>
<td>67 - 98</td>
<td>31</td>
</tr>
<tr>
<td>98 - 122</td>
<td>24</td>
</tr>
<tr>
<td>122 - 124</td>
<td>2</td>
</tr>
<tr>
<td>124 - 183</td>
<td>59</td>
</tr>
<tr>
<td>Total Seek Time</td>
<td>208</td>
</tr>
<tr>
<td>Average</td>
<td>26</td>
</tr>
</tbody>
</table>

5. Evaluation

Proposal Advantages: According to (Table 2), experimentally and deterministically had been proven that proposed scheduler (SSRRF) is the optimal, moreover it has as many as advantages extracted from just the best features of SSJF, SCAN, and LOOK schedulers.

1- Maybe it seems fair and reasonable to serve requests close to the current head position like SSTF, but moving the head to opposite side to serve remote requests that have shortest total seek time is fairer through assessment of which selected side has the minimum need of seek time. Thereby, the proposed algorithm (SSRRF) of 26 total seek time is the best.

2- This policy has an economical goal to maximize throughput.

3- The proposed algorithm produces more uniform response time.

4- The proposed algorithm prevents starvation of some requests.

5- Proportionally to other schedulers, the proposed algorithm (SSRRF) consumes minimum Seek-Time (26) to access all request. Therefore, it is the best algorithm to reduce CPU idle time to wait I/O fetching data from the disk.

6- The proposal usually gives more equitable service to innermost and outermost requests.

7- In dynamic scope, because of the property of never serve traversed tracks, proposed algorithm extremely limiting how long the heads to stay more at any one extreme
Proposal Disadvantages: The head should not move randomly across the requests in both sides of particular surface of the disk unless has the suitable calculations to determine which side has the remote request with shorter distance than others. Since the procedures of calculating differences of distances among the requests takes place, so it must be time consuming in contrast with other algorithms.

6. Conclusion

Obviously, this research fulfilled the basic idea behind the project's goal to obtain the improvement of performance. That is the proposal "The Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm" had been evaluated through the simulation under the uniform and localized cylinder access depending on well-known criteria. Moreover, the implementing of new intelligent controlled scheduler at the beginning to achieve optimized head-movement implies guarantee high throughput and fairness in response time to treat all I/O requests equally.

Reference

Martens, D. L., & Katachabaw, M. J. (2006, April). Disk access analysis for system performance optimization. In the Proceedings of the 5th WSEAS International Conference on APPLIED COMPUTER SCIENCE (ACOS'06), Hangzhou, China (pp. 531-316).