New Disk Scheduling Algorithms for Reduced Rotational Latency

Kitae Hwang and Heonshik Shin

Department of Computer Engineering
Seoul National University
Kwanakgu, Seoul 151-742, Korea
E-mail : shinhs@erccw1.snu.ac.kr
Fax : +82-2-886-7589

ABSTRACT

Disk arm scheduling algorithms have been studied for many years to increase disk I/O performance. Most of the disks by the early 1980's are characterized as a linear seek time and their seek time is responsible for the most time of disk access. So, 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, whereas the rotation speed has been steady at the 3600 RPM. So, it is of value to reduce average rotational latency.

In this paper we propose two disk scheduling algorithms, i.e. Shortest Rotational Latency First(SRLF) and Shortest Access Time First(SATF) for reduced rotational latency. SRLF services first a request with the shortest expected rotational latency and SATF services first a request with the shortest expected access time. We analyze the expected access time of the request serviced first by each of SSTF and SRLF under the uniform cylinder access. We evaluated the response time of SSTF, SRLF, and SATF through the simulations under the uniform and localized cylinder access. The results of the analysis and simulations show that the algorithms proposed in this paper are more efficient than SSTF which is known for the fastest response time.

1. INTRODUCTION

Disk arm scheduling algorithms have been studied for many years to increase disk I/O performance[2,3,4,5,7]. Most disks by the early 1980's had been made of simple stepper motors which have a linear seek time characteristic[3]. Since the seek time is responsible for the most time of disk access, most studies on the disk scheduling have focused on the reduction of the average seek distance or the number of cylinders from the current head position to the requested cylinder to improve the response time. SCAN and Shortest Seek Time First(SSTF) are the representative disk scheduling algorithms[4]. SCAN algorithm moves a disk head from the innermost cylinder to the outmost cylinder, or conversely. Then it services first a request closest on the moving direction of the disk head. SSTF services first a request closest to the disk head position. SSTF has been credited with the fastest mean response time among the existing disk scheduling algorithms[4].

Nowadays disk manufacturing technology based on a voice coil actuator is available for disk drives which show the nonlinear seek time characteristics as follows:

$$a + b\sqrt{n}, \qquad n > 0$$

where a is the settling time, b is the acceleration factor, and n is the seek distance[1,5]. In the case of a disk with a voice coil actuator, the reduction of the seek distance makes less contribution to the reduction of the seek time than in the case of a disk with linear seek time characteristics. With the exisiting disk arm scheduling algorithms a request spends rotational latency of a half revolution on the average, which is 8.35ms at 3600 RPM. Actually this time is never small in comparison with the seek time of a disk with a voice coil actuator. When b=0.5ms, the mean rotational time(8.35ms) is approximately equal to the difference between the seek time of 100 cylinders and the seek time of 712 cylinders for a voice coil actuator, i.e., $a+0.5\sqrt{712} - (a+0.5\sqrt{100}) = 8.34$. In addition, the seek time has greatly improved with the nominal seek time of about 15ms, whereas rotation speed has been steady at 3600 RPM for a decade[8]. These facts motivate us to propose new disk scheduling algorithms based on the reduction of rotational latency.

In this paper we present two new disk scheduling algorithms for reduced rotational latency: Shortest Rotational Latency First(SRLF) and Shortest Access Time First(SATF). SRLF first services a request with the shortest rotational latency and SATF services first a request with the smallest sum of seek and rotational latency. It will be shown that the proposed algorithms achieve shorter disk access time than SSTF using analytic and simulation methods.

The remainder of this paper proceeds as follows: In

Section 2 disk access time model is described. New disk scheduling algorithms proposed in this paper are introduced and their expected access time is analyzed in comparison with that of SSTF in Section 3 and estimated through simulation in Section 4. Conclusions are drawn in Section 5.

2. ACCESS TIME MODEL

As shown in Fig. 1, the disk access time is defined as the sum of seek time, rotational latency, and transfer time and the disk access time. The queueing delay increases more rapidly as the mean disk access time becomes larger for a fixed arrival rate.

The transfer time is in proportion to block size, rotational speed, recording density of a track, and speed of the electronics connecting a disk to a computer. Transfer rates in 1990 are typically 1 to 4MB/sec9); so, the transfer time is relatively very small compared with the seek time and rotational latency

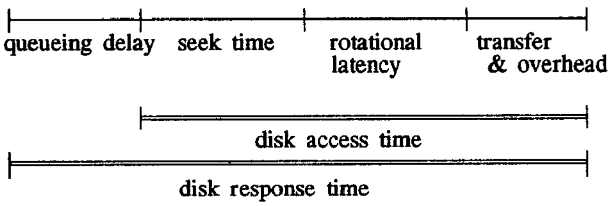

Figure 1 Time components of disk I/O

2.1 Rotational Latency

The rotational latency is defined as the time for a requested sector to rotate under the disk head. The average rotational latency to the desired sector is halfway around the disk. Currently it is approximately 8.3ms at the rotational speed of 3600 RPM. Although some manufacturers plan to go for 5400 RPM in the early 1990's, many difficulties exist in reducing rotational latency by increasing RPM8).

2.2 Seek Time

The seek time is a function of seek distance. Nominal seek time is defined as the average time required for the disk head to move from any random cylinder to any other random cylinder. This factor has been the most important subject of research effort for the reduction of the response time. In the early days of disk design, the nominal seek time is one or two orders of magnitude larger than all the other components8, 10): Now the nominal seek time for IBM 3380 disk is reduced to around 16ms.

A disk with a simple stepper motor actuator has a linear seek time characteristic:

$$\mathrm{seek}(n) = \begin{cases} 0 & n = 0 \\ a + bn & n > 0 \end{cases} \tag{1}$$

where n is the seek distance, *a* is the mechanical settling time, and *b* is the acceleration factor. For example, the values of *a* and *b* are 10ms and 0.1ms, respectively. For a disk with the linear seek time, the nominal seek time is simply the time required for a seek of the average seek distance, N/3, where N is the total number of cylinders.

But nominal seek time varies with the various actuator models. Disks with high performance voice coil actuators such as IBM 3380, have nonlinear seek time characteristics1):

$$\mathrm{seek}(n) = \begin{cases} 0 & n = 0 \\ a + b\sqrt{n} & n > 0 \end{cases} \tag{2}$$

Scranton et. al showed that the nominal seek time for a voice coil actuator is the time required for a seek of approximately 0.284N cylinders1). In the case of a disk with a voice coil actuator, therefore, the seek distance makes less contribution to the seek time than in the case of a disk with a simple stepper motor actuator. Meanwhile the rotational latency remains the same regardless of the seek time characteristics. This fact makes it worthwhile to attempt to reduce the rotational latency for shorter response time.

3. DISK SCHEDULING FOR REDUCED ROTATIONAL LATENCY

3.1 SRLF(Shortest Rotational Latency First)

We now describe a new disk scheduling algorithm, SRLF. While the existing disk scheduling algorithms, such as SSTF and variations of SCAN, are based on the seek distances of requests in a disk queue, the SRLF scheduling makes use of the rotational latencies of requests in the queue. SRLF services first a request with the shortest rotational latency.

In our research the disk subsystem is based on a simple model consisting of a single disk, a disk queue, and a scheduling program module as depicted in Fig. 2. For simplicity, it is assumed that the disk has one track per cylinder and that the drive has a single arm and a voice coil actuator; so, the seek time is characterized by the nonlinear seek time model of Eq.(2). The disk is assumed to have the circuitry to sense the rotational position in terms of the sector number that rotates under the disk head. This circuitry can be implemented easily by modifying the circuitry to sense the index point which is the starting point of track. The IBM 3380 disk subsystem also has the circuitry to sense the rotational position9).

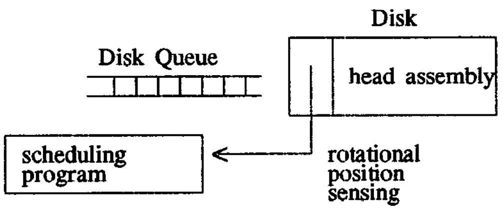

Figure 2 A disk model

We now introduce two basic terms to delineate the disk behavior: the **Expected Rotational Position(ERP)** and **Expected Rotational Distance(ERD)**. ERP is defined as the rotational position located at the end of seek activity resulting from a request in service, and ERD is defined as the rotational distance of a request in service. The expected rotational latency is equal to the multiplication of ERD and the rotational time per sector. The SRLF algorithm calculates ERP and ERD of each request in the disk queue and selects a request with the shortest ERD. The following symbols will be used throughout to describe the SRLF method formally.

r_i = i-th request in the queue
h = the head position(the cylinder number) at the scheduling instant
RP = the rotational position at the scheduling instant
R = one revolution time of a disk
M = the number of sectors in one track
N = the number of total cylinders
SR = R/M, the rotational time per sector
T = transfer time
c_i = the cylinder number of r_i.(ranges from 1 to N)
s_i = the sector number of r_i.(ranges from 0 to M-1)
$TSEEK_i$ = the seek time of r_i,

Let ERP and ERD of $r_i(i=1,..,k$ where k is the queue length) denote ERP_i and ERD_i, respectively. For the seek distance between the disk head and r_i or $|c_i-h|$, $TSEEK_i$, ERP_i, and ERD_i can be calculated to be as follows:

$$TSEEK_i = a + b\sqrt{|c_i-h|}$$

$$ERP_i = (RP + TSEEK_i/SR) \mod M$$

$$ERD_i = \begin{cases} M+s_i-ERP_i, & \text{if } s_i - ERP_i < 0 \\ s_i-ERP_i, & \text{otherwise} \end{cases} \qquad (3)$$

(unit: the number of sectors)

If there is any r_i whose seek distance $|c_i-h| = 0$, that is, whose cylinder number is equal to the head position, SRLF services it first. Otherwise, upon the calculation of $ERD_i(i=1,\ldots,k)$ for all i, SRLF chooses a request with the smallest ERD_i among k requests for immediate service.

3.2 Performance Analysis

For the analysis of the expected access times the following assumptions are made: Each request arrives with a cylinder number randomly distributed within total data band width. The sector number of each request is distributed randomly within the range of [0,M-1]. The cylinder number and the sector number of a request are independent of each other. In reality, however, successive seeks are not independent. They show the localized access pattern such that the cylinder of next request is more likely the same as that of the last request. Nevertheless, it is difficult to analyze the average access time for localized access pattern. The assumption of the uniform disk access provides a good approximation of seek time and most disk scheduling algorithms aim at minimizing the expected access time computed under this assumption6). In this paper, SSTF is selected as the base algorithm for performace comparison with SRLF and SATF because it is known for the fastest mean response time4). The response time for localized access will be evaluated through the simulations in Section 4.

The performance of SSTF and SRLF is evaluated in terms of **Expected Shortest Seek Distance(ESSD)** and **Expected Shortest Rotational Latency(ESRL)**: ESSD and ESRL are defined as the average of the shortest seek distance and that of the shortest rotational latency, respectively, among requests in the disk queue.

3.2.1 Expected Shortest Seek Distance(ESSD)

Assuming that there exist k requests in the disk queue, let $X_i(i=1,2,..,k)$ be the random variables for the seek distance from the disk head position to the cylinder of r_i. These random variables are assumed to have identical distributions. The shortest seek distance for k random requests is the random variable X_s defined to be

$$X_s = \min(X_1,X_2,\ldots X_k)$$

Thus,

$$ESSD = E[X_s]$$

Since the property of uniform access accounts for the fact that the X_i are independent, the seek distances of requests in the disk queue can be modeled as independent random variables. On the other hand, there are N^2 unique seeks: N seeks of length zero and 2(N-i) different seeks of length i, for $i=1,2,\ldots,N-1$. Hence each of the $X_j(j=1,2,\ldots,k)$ has the following distribution:

$$P(X=i) = 2(N-i)/N^2$$

or

$$P(X\geq i) = (2/N^2)\sum_{j=i}^{N-1}(N-j)$$
$$= (N-i)(N-i+1)/N^2$$

Thus

$$P[\min(X_1,X_2,\ldots,X_k)\geq i)]$$
$$= P(X_1\geq i)\cdot P(X_1\geq i)\cdot\cdot\cdot P(X_k\geq i)$$

$$E[X_s] = \sum_{i=1}^{N-1} P[\min(X_1,X_2,\ldots,X_k) \geq i)]$$

According to the result of Bitton6), the equation becomes

$$E[X_s] \fallingdotseq N/(2k + 1) \qquad (4)$$

For a disk with a linear seek time model and k requests in the disk queue, the expected disk access time of the request serviced first by SSTF is presented by

$$ACC_{SSTF(linear)} = a + bN/(2k+1) + R/2 + T$$

since the mean rotational latency of SSTF is R/2. For a disk with a voice coil actuator, however,

$$ACC_{SSTF} \fallingdotseq a + b\sqrt{N/(2k+1)} + R/2 + T$$

Using the simulation method(see Appendix) we have found the expected shortest seek time as a function of the seek distance for a voice coil actuator:

$$ACC_{SSTF(nonlinear)} = a + b\sqrt{SD_kN} + R/2 + T \qquad (5)$$

where SD_k is defined in Appendix.

3.2.2 Expected Shortest Rotational Latency(ESRL)

Let $X_1, X_2, \ldots, X_k$ be the random variables uniformly distributed between 0 and R where subscript k denotes the number of requests in the disk queue. Then, X_i represents the rotational latency of r_i. Thus a random variable X_r or the shortest rotational latency becomes

$$X_r = \min(X_1, X_2, \ldots, X_k)$$

Since X_i is distributed within [0,R], the distribution of X_i is

$$P(X_i < \alpha) = \alpha/R$$

Then the distribution function of X_r is obtained as follows:

$$\begin{aligned} F_{X_r}(x) &= P(X_r < \alpha) \\ &= P(X_1 < \alpha \text{ or } X_2 < \alpha \text{ or } \cdots X_k < \alpha) \\ &= 1 - P(X_1 \geq \alpha \text{ and } X_2 \geq \alpha \cdots X_k \geq \alpha) \\ &= 1 - P(X_1 \geq \alpha)P(X_2 \geq \alpha) \cdots P(X_k \geq \alpha) \\ &= 1 - (1 - \frac{\alpha}{R})^k \end{aligned}$$

Then the density function of X_r is

$$f_{X_r}(\alpha) = F'_{X_r}(\alpha) = \frac{k}{R}(1 - \frac{\alpha}{R})^{k-1}$$

Therefore,

$$\begin{aligned} ESRL = E[X_r] &= \int_0^R \alpha f_{X_r}(\alpha) d\alpha \\ &= \frac{k}{R} \int_0^R \alpha \, (1 - \frac{\alpha}{R})^{k-1} d\alpha \\ &= \frac{R}{k+1} \end{aligned} \qquad (6)$$

Thus, if k=1, then $E[X_r] = R/2$. The distribution of the cylinder numbers of requests serviced by SRLF is random because SRLF services requests considering rotational latency which is related with only sector number but independent of the cylinder number. Then the nominal seek time is equal to $a+b\sqrt{(0.284N)}$ and the expected access time of the request serviced first by SRLF is

$$ACC_{SRLF} = a + b\sqrt{(0.284N)} + R/(k+1) + T \qquad (7)$$

3.2.3 Comparison of the expected access time

For the comparison of access times, ACC_{SSTF} and ACC_{SRLF}, their difference can be calculated as follows:

For the disk queue length of k,

$$\begin{aligned} ACC_{SSTF} - ACC_{SRLF} &= a + b\sqrt{SD_kN} + R/2 + T \\ &\quad - (a + b\sqrt{(0.284N)} + R/(k+1) + T) \\ &= b\sqrt{N}(\sqrt{SD_k} - \sqrt{0.284}) + R(\frac{1}{2} - \frac{1}{k+1}) \end{aligned} \qquad (8)$$

The difference in the expected access time is determined by three constants b, N, and R as shown in Eq.(8) where the first and second multiplication terms are the differences in the seek time and the rotational latency, respectively. For k larger than 1, the first multiplication term becomes negative and the second multiplication term becomes positive. If k=1, Eq.(8) is 0 because the disk access time is the same irrespective of disk scheduling algorithms when there is only one request in the disk queue.

Eq.(8) is plotted for different values of b and R in Fig. 3, where the x-axis represents the number of requests at the scheduling instant. Under the current disk technology, R, N, and b are approximately 16.7ms, 1000, and 0.5ms, respectively. For these values, Eq.(8) is positive for all k as shown in Fig. 3. At present, therefore, we can conclude that SRLF achieves the shorter average access time than SSTF under the current disk technology.

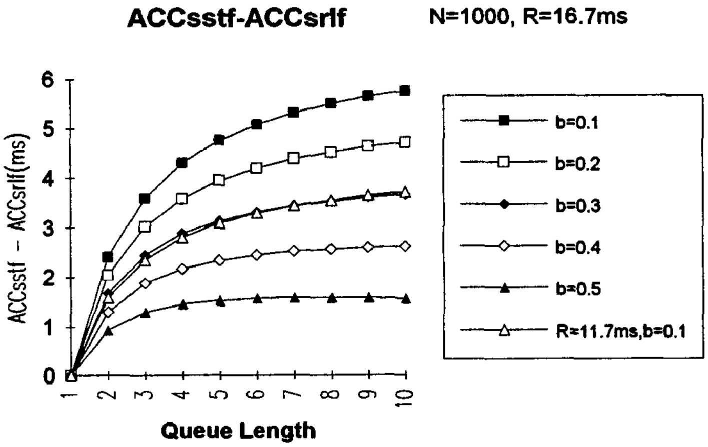

Figure 3(a) ACC_{SSTF} - ACC_{SRLF} with varing b

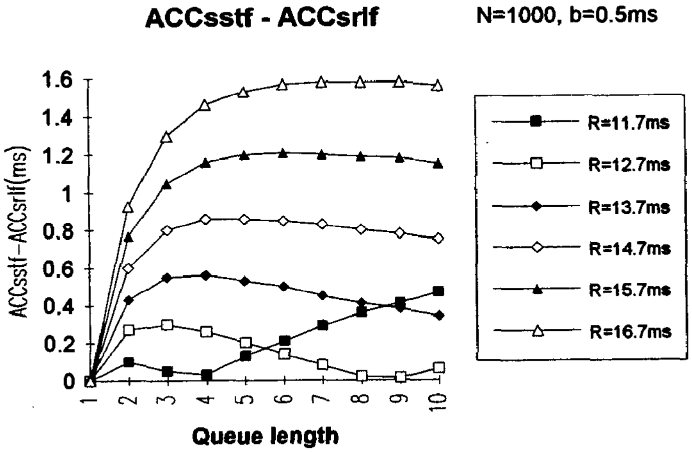

Figure 3(b) ACC_{SSTF} - ACC_{SRLF} with varing R

Which of the two algorithms will achieve the better performance in the future in terms of seek time and rotational speed? Recently, many efforts are being invested in speeding up disk rotation in spite of many difficulties. The seek time of a disk with a voice coil actuator can be improved by making the time factor a and b smaller[1]. Fig. 3a and 3b show some of the results from Eq.(8) for b and R in the range of [0.1ms,0.5ms] and [11.7ms 16.7ms], respectively. (We do not refer to a because a is not related with the Eq.(8)). For all these ranges, Eq. (8) turns out to be positive. As a result, SRLF services requests faster than SSTF. This makes the average response time of SRLF shorter than SSTF. So, we can claim that SRLF is superior to SSTF under the current and future technology.

3.3 SATF(Shortest Access Time First)

In addition to SRLF, we present another efficient disk scheduling algorithm, Shortest Access Time First(SATF). It takes both seek and rotational latency into consideration. For each request in the disk queue, SATF calculates the Expected Access Time(EAT) which is defined as the expected sum of the seek and rotational latency. If there are k requests in the disk queue, EAT of r_i or EAT_i can be derived as follows:

$$TSEEK_i = a + b\sqrt{|c_i - h|}$$

$$ERP_i = (RP + TSEEK_i/SR) \bmod M$$

$$ERD_i = \begin{cases} M + s_i - ERP_i, & \text{if } s_i - ERP_i < 0 \\ s_i - ERP_i, & \text{otherwise} \end{cases}$$

$$EAT_i = TSEEK_i + ERD_i \times SR \qquad (9)$$

SATF first services a request with the smallest EAT. In doing so, SATF achieves the shorter expected access time than SSTF and SRLF. Performance analysis in comparison with other algorithms will be shown in Section 4 through simulations.

3.4 Algorithm Overhead

The time complexity of SSTF is O(n), where n is the number of requests in the disk queue. Time complexities of SRLF and SATF are also O(n) because they search the disk queue just once. However, SRLF and SATF have some computational overhead because ERD and EAT must be calculated, respectively. Since these calculations are very simple as shown above, this overhead should not influence their performance too much.

4. PERFORMANCE EVALUATION

4.1 Simulation

We implemented a simulator using SLAM II[11] to evaluate the response time of the three algorithms of SSTF, SRLF, and SATF. The disk parameters used in the simulation is shown in Table 2.

Table 2 Disk parameters

Parameter	meaning	base value
N	Number of total cylinders	1000
a	Mechanical settling time	6ms
b	Acceleration factor	0.5ms
R	Full rotation time	16.7ms
T	Transfer rate	1Mbytes/sec
S	Bytes per sector	512
M	Sector per cylinder	40

The event flow of the simulator and it's description are given in Fig. 4 and Table 3, respectively, in which MAIN and DISKSCHDL are not an event but a program module. MAIN routine initializes SLAM variables and starts simulations by calling event 1 and event 6. The three algorithms are implemented in DISKSCHDL. The results of each experiment are averaged over 20 runs, each of which has executed 4000 disk I/O requests.

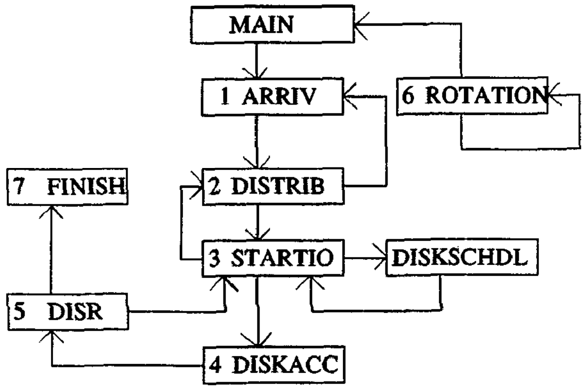

Figure 4 Event flow of simulator

Table 3 Event table

ARRIV	Simulates the request arrival and calls DISTRIB event.
DISTRIB	Inserts the arriving request in a disk queue and calls STARTIO. At the same time, it calls ARRIV for the arrival of next request.
STARTIO	Returns immediately if the disk is busy. If the disk is idle, it calls the scheduler DISK SCHDL which is the disk scheduler to choose a request, changes status of the disk to idle,and calls DISKACC.
DISK-ACC	Simulates physical disk I/O and calls DISR on completion of a disk I/O and calculates disk access time.
DISR	Simulates the disk interrupt service routine. It changes status of the disk to idle and calls FINISH. It also calls STARTIO to service arequest waiting in the queue.
ROTA-TION	Manages the rotational position and keeps up it. DISKACC and DISKSCHDL use this information to know the current rotational position
FINISH	Enumerates the simulation results

4.2 Workload

Requests arrive with arrival time exponentially distributed with mean of MeanArrivalRate. MeanArrivalRate which is defined as the number of disk I/O requests per second varies from 20 to 40. The distribution of requested sectors is assumed to be uniform within the range of 0 to M-1, and the requested size is assumed to be 4 sectors. We evaluate the response time for two cylinder access distributions. One is the uniform cylinder access such that requested cylinder numbers are distributed randomly within the range of 1 to N. The other is the localized cylinder access for more realistic case. Arriving requests are assumed to have the following localized property defined by Hofri3). Let p_{ij} be the probability of a request addressing cylinder j, given that the previous one was destined for cylinder i.

$$p_{ij} = \begin{cases} \alpha + \dfrac{1}{N} & i = j \\ \dfrac{1 - (\alpha + 1/N)}{(N-1)} & i \neq j \quad 0 \leq \alpha < 1 \end{cases}$$

where α reflects the degree of locality. $\alpha = 0$ means uniform cylinder access and high values of α reflect strong locality.

4.3 Results

4.3.1 Experiment 1 : Uniform cylinder access ($\alpha = 0$)

Fig. 5 shows the response time of three algorithms for base values of disk parameters. It turns out that both algorithms, SRLF and SATF considering rotational latency have the shorter response time than SSTF.

Table 4 shows the average seek distance, average seek time, and average rotational latency for base values of disk parameters. As listed in Table 4, among the three algorithms, the average seek distance of SSTF is the smallest while that of SRLF is the largest. Average rotational latency of SRLF is the smallest. SRLF tries to reduce rotational latency at the expense of seek time and achieves the smaller sum of rotational latency and seek time than SSTF. Of course, SATF is the best because it services first a request with the smallest sum of rotational time and seek time.

4.3.2 Experiment 2 : Localized cylinder access

Fig. 6 shows the response time of the three algorithms for α(α = 0.3, 0.5, 0.7, 0.9) using the base values of disk parameters. For all ranges of arrival rate, SATF shows the best performance among the three algorithms and the second is SRLF. As the locality becomes larger, it is natural that the average seek distance of SSTF become smaller. It may be asserted that SSTF should be better than SRLF at a high locality. As the locality increases, however, the average seek distance of SRLF is no longer the nominal seek distance(=N/3). Rather, it decreases. This is because SRLF services it first if there is any request whose cylinder is equal to the head position,.

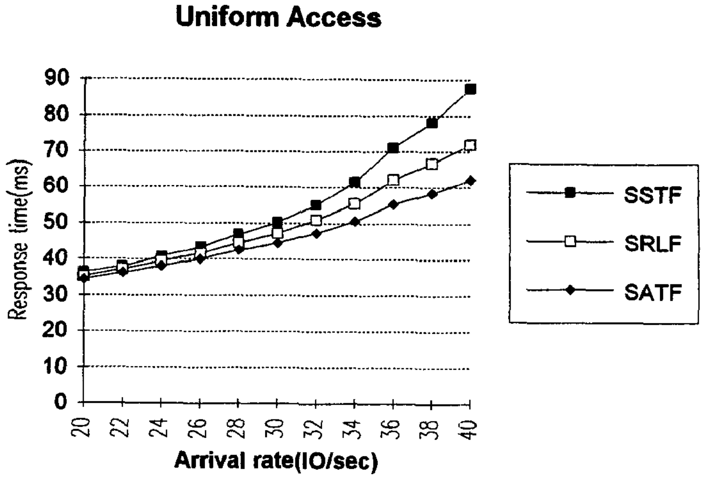

Figure 5 Response time under the uniform access

Table 4 The results of simulations for the uniform cylinder access

(a) Average seek distance

Arrival rate	20	24	28	32	36	40
SSTF	309	297	280	260	225	196
SRLF	333	333	332	333	333	333
SATF	320	313	304	296	280	268

(b) Average seek time(ms)

Arrival rate	20	24	28	32	36	40
SSTF	14.0	13.8	13.5	13.2	12.6	12.1
SRLF	14.4	14.4	14.4	14.4	14.4	14.4
SATF	14.2	14.1	14.0	13.8	13.6	13.4

(c) Average rotational latency(ms)

Arrival rate	20	24	28	32	36	40
SSTF	8.33	8.34	8.32	8.33	8.34	8.35
SRLF	7.86	7.56	7.16	6.98	6.05	5.52
SATF	7.89	7.62	7.29	7.15	6.47	6.11

At a high locality, SATF is more efficient than the other two algorithms as shown in Fig. 6. Fig 7 shows the average seek distances and the average rotational latencies for arrival rate of 26 IOs/sec. At the locality greater than 0.5, the seek distances of the three algorithms are almost the same as shown in Fig. 7a. At the locality of 0, SRLF has the smallest average rotational latency among the three algorithms. However, at the locality greater than about 0.2, SATF has the smallest average rotational latency. This result can be explained as follows: As the locality increases, more requests addressing the same cylinder arrive at the disk device and there may exist more requests whose seek distances are 0 in the disk queue. Thus a request with the shortest expected rotational latency may be serviced first by SATF because the expected access time of the request may be the smallest. This possibility becomes higher as the locality increases. If all requests address the same cylinder, the requests may be serviced in the order of rotation by SATF. The results of these simulations allow us to claim that the two proposed algorithms which consider the rotational latency are superior to SSTF which considers only seek time for a disk with a voice coil actuator.

5. CONCLUSION

So far, we have presented two disk scheduling algorithms of Shortest Rotational Latency First(SRLF) and Shortest Access Time First(SATF). SRLF services first a request whose cylinder is equal to the disk head position; if there is no such a request, it services first a request with the shorest expected rotational latency. SATF services first a request with the shortest expected disk access time.

We found out the seek distance which produced the average shotest seek time for a disk with a voice coil actuator for the disk queue length of k using the simulation. We analyzed the expected access time of the request serviced first by each of SSTF and SRLF, for a disk queue length of k, under the asssumption of the uniform cylinder access. The result of the analysis shows that SRLF has the shorter expected access time than SSTF under the current disk technology, i.e. N=1000, R=16.7ms, and b=0.5ms, and under the future disk technology supporting faster R and shorter b.

Recent studies show that the locality is an important property in accessing disk cylinders. In this study experiments have been conducted to find out the response time of SSTF, SRLF, and SATF through simulations with varing the degree of locality. For all ranges of workload settings, the results show that SATF has the best of the three algorithms and the second is SRLF. For the localities greater than 0.5, seek distances of the three algorithms are almost the same. However, the two algorithms proposed in this paper have the shorter average rotational latency. Especially, SATF is more efficient than the other two algorithms at the higher locality because it first services a request with the shorter rotational latency among the requests addressing the same cylinder. Consequently, the two disk schduling algorithms of SRLF and SATF for reduced rotational latency are more efficient than the existing algorithms.

< Appendix : Simulation results for $ACC_{SSTF(nonlinear)}$ >

Simulations were performed as follows:

a, b, and N are assumed to be 6ms, 0.5ms, and 1000, respectively. The following Step 1 to Step 5 are repeated for 10 values of k (k=1,2,...,10).

1. Generate the random head position and k independent random requests in a disk queue for 1000 cylinders.
2. Calculate the shortest seek time.
3. Repeat Step 1 and 2 100,000 times and average the shortest seek time.
4. Calculate the seek distance, SeekDist, which produces the average shortest seek time, AveShort which is the result of Step 3.

$$\text{SeekDist} = \left[\frac{(\text{AveShort} - a)}{b} \right]^2 = \left[\frac{(\text{AveShort} - 6)}{0.5} \right]^2$$

5. Calculate SD_k defined as SeekDist divided by 1000.

$$SD_k = \text{SeekDist}/1000$$

Table 1 shows SD_k: For k=1, the result(0.284) is in accord with the seek distance for the nominal access time derived by Scranton and Thompson1). Also for k=2, the result is approximately consistent with the seek distance (0.16N) obtained by Bitton and Gray to produce the expected seek time for mirrored disk6). Then, if there exist k random requests in a disk queue, the expected shortest seek time is expressed as

$$a + b\sqrt{SD_k N}$$

Hence, ACC_{SSTF} for a voice coil actuators can be expressed as

$$ACC_{SSTF(nonlinear)} = a + b\sqrt{SD_k N} + R/2 + T$$

Table 1 Seek distance for the expected shortest seek time for a voice coil actuator.

queue length k	1	2	3	4	5
SD_k	0.284	0.170	0.121	0.093	0.075
queue length k	6	7	8	9	10
SD_k	0.063	0.0544	0.0477	0.0427	0.0835

REFERENCE

[1] R. A. Scranton and D. A. Thompson, "The access time myth," IBM Res. Rep., Nov. 1983.

[2] Neil C. Wilhelm, "An abnomaly scheduling: A comparison of FCFS and SSTF seek scheduling using an empirical model for disk accesses," Comm. of ACM, Vol. 19, No. 1, Jan. 1976.

[3] Micha Hofri, "Disk scheduling : FCFS vs. SSTF revisited," Comm. of ACM, Vol. 23, No. 11, Nov. 1980.

[4] R. Geist and S. Daniel, "A continuum of disk scheduling algorithms," ACM Trans. on Computer Systems, Vol. 5, No. 1, Feb. 1987.

[5] Dina Bitton, "Arm scheduling in shadowed disks," Spring, COMPCON89, 1989, pp. 132-136.

[6] Dina Bitton and Jim Gray, "Disk shadowing," Proc. of the 14th VLDB Conf., 1988.

[7] Richard P. King, "Disk arm movement in anticipation of future requests," ACM Trans. on Compter Systems, Vol. 8, No. 3, 1990, pp. 214-229,

[8] S. W. Ng, "Improving disk performance via latency reduction," IEEE Trans. on Computers, Vol. 40, No. 1, Jan. 1991, pp. 22-30

[9] J. Hennessy and D. A. Patterson, Computer Architecture : A Quantitative Approach, Morgan Kaufmann Pub. INC., 1990.

[10] R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk system architectures for high performance computing," Proceedings of the IEEE, Vol. 77, No. 12, Dec. 1989.

[11] A. Alan B. Pritsker, Introduction to Simulation and SLAM II, John Wiley & Sons, New York, 1986.

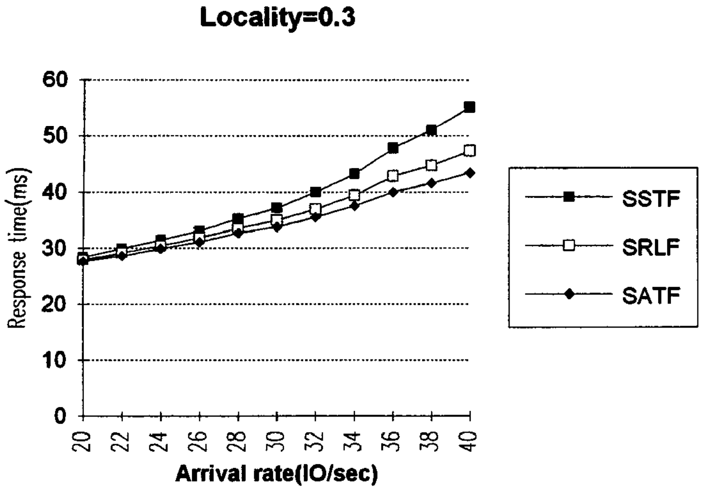

Figure 6(a) Response time for the locality of 0.3

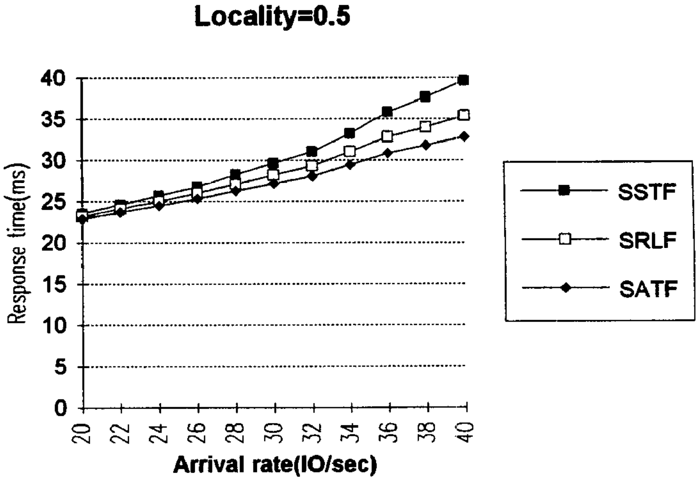

Figure 6(b) Response time for the locality of 0.5

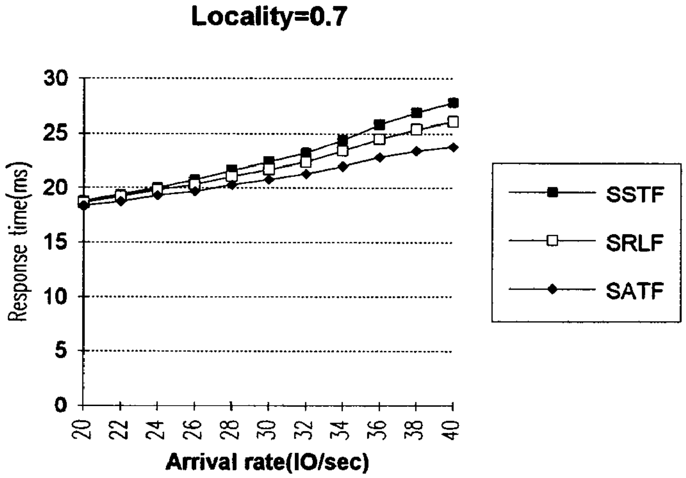

Figure 6(c) Response time for the locality of 0.7

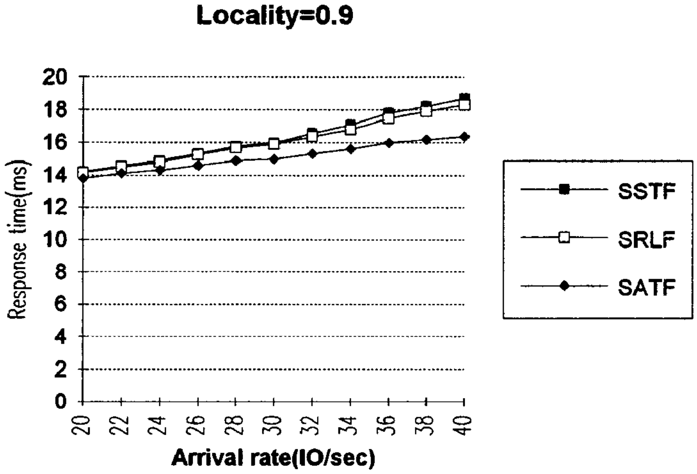

Figure 6(d) Response time for the locality of 0.9

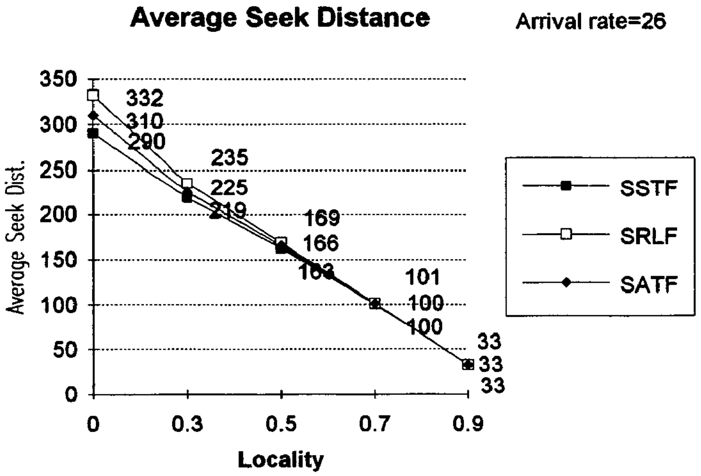

Figure 7(a) Average seek distance according to locality

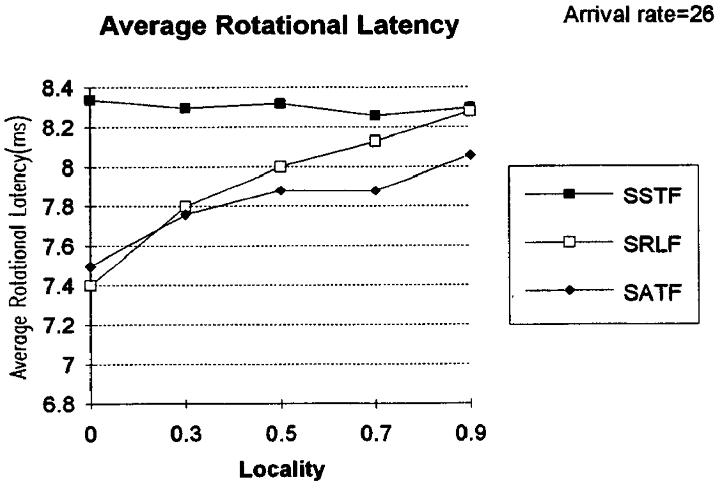

Figure 7(b) Average rotational latency according to locality