Mathematical Model for Real Time Disk Scheduling Problem

S. Y. Amdani
Associate Professor,
Department of CSE, BNCOE,
Pusad (India).

M. S. Ali
Principal,
P.R.M.C.E.& M,
Badnera (India).

S. M. Mundada
Assistant Professor,
Department of IT, BNCOE,
Pusad (India).

ABSTRACT
A real time system is a system that must satisfy explicit bounded response-time constraints. Real time data base systems combine the concepts from real time systems and conventional database systems. Real time systems are mainly characterized by their strict timing constraints. Conventional databases are mainly characterized by their strict data consistency and integrity requirements. Thus, real time database systems should satisfy both the timing constraints with data integrity and consistency constraints.

Real-time systems can be defined as those computing systems that are designed to operate in a timely manner. That is, performing certain actions within specific timing constraints; e.g. producing results while meeting predefined deadlines. Real-time disk I/O scheduling is extremely important to the performance improvement of the whole real-time system since the disk devices are the system's bottleneck.

The design of mathematical model shows the correctness of real-time disk scheduling algorithms, as the effectiveness of any algorithm can only be detected or calculated using the mathematical model. This paper represents the design of our mathematical model which helps in evaluating the result of different scheduling algorithms.

Keywords
Real-Time, Disk, Deadline, Scheduling parameters

1. INTRODUCTION
Like a conventional database system, a real time database functions as a repository of data, provides efficient storage, and performs retrieval and manipulation of information. But conventional database system considers only data consistency and integrity constraints and not the timing constraints. However, as a part of a real-time system, whose tasks are associated with time constraints, a real time database has the additional burden of ensuring some degree of confidence in meeting the system's timing constraints and a timing constraint is expressed in the form of a deadline, a certain time in the future by which a transaction needs to be completed. The Transaction is said to be Hit if it is completed within its Deadline.[1][2] Otherwise it is said to be Miss. Thus in real-time database systems, the correctness of transaction processing depends not only on maintaining consistency constraints and producing correct results but also on the time at which a transaction is completed.
In a disk-based database system, disk I/O occupies a major portion of transaction execution time. The order in which I/O requests are serviced has an immense impact on the response time and throughput of the I/O subsystem.

Scheduling I/O operations consists of two parts:

1.Assigning priorities to the various I/O requests. Such priorities determine the order in which various operations are performed.
2. Scheduling the disk head itself in order to cope with the physical limitations of the device, and thereby minimizing delays such as seek time in order to increase throughput.
As we know, before giving service to transactions, it is necessary to schedule them.[3] There are various transaction scheduling algorithms and the maximum time while servicing the transaction is related to disk. So, in order to know what this time required is and how to minimize it, it is mandatory to study the different parameters required for real time disk scheduling.
In this paper we have presented a mathematical model which shows how scheduling result of any algorithm can be evaluated.

2. DISK SCHEDULING PROBLEMS
In a disk-based database system, disk I/O occupies a major portion of transaction execution time.

Disk scheduling involves a careful examination of the pending disk requests to determine the most efficient way to service the disk requests.

The disk scheduling problem involves reordering the disk requests in the disk queue so that the disk requests will be serviced with the minimum mechanical motion by employing seek optimization and latency optimization.

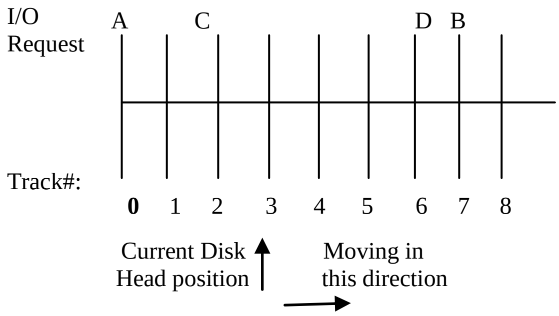

Fig1:Disk Scheduling Problem

3. PARAMETERS
3.1 Transaction Request Parameters

3.1.1 Arrival Time (At) or Release time
The time at which request for the transaction on the disk arrives is the Arrival time for that request.
Inter arrival time: Arrival Time difference of two consecutive request.

$$\text{Inter arrival time} = \left(\frac{1}{\text{Transaction arrival rate}}\right)\left(\log_e \frac{1}{\text{Block accessed}}\right)$$

Arrival Time=Arrival Time + Inter arrival Time

3.1.2 Block Size

It may be called as request size. It is the number of block to be accessed from the disk or you may say number of block residing on disk on which transaction operation (Read, Write, Read-Write or Read-write-compute) has to be performed.

3.1.3 Block Access or Access block or Block location to be Accessed

It may be called as block location on disk. It is analogous to the track location or the sector location on disk.

3.2 Scheduling Parameters

3.2.1 Transaction ID

Whenever any request for the transaction on the disk arrives, it is assigned an ID by which it can be identified by the scheduler. It is unique for every transaction that is arriving on the disk.

3.2.2 Average Execution Time(AET)

Average Execution Time=1.5*Block Size

3.2.3 Slack Factor

The Slack factor is a constant that provides control over the tightness or slackness of the transaction deadlines..

Slack Factor ≥ 1.

3.2.4 Deadline

The latest time at which a request must be completed is the deadline.

Deadline = Arrival Time + (Slack factor * Average Execution Time)

Or Deadline =Arrival Time + (Slack Factor * (1.5 * Block Size)

3.3 Disk Parameters

3.3.1 Disk Size

Disk consists of number of tracks and number of tracks forms a sector and number of sectors forms a cylinder.

3.3.2Seek Time:

Time required to move the disk head from current position to the target track.

Seek Time = Seek factor * [abs (Block Access – Current Head Position)]

3.3.3 Rotational Latency

Once the desired track has been reached, then time required to rotate the disk until desired sector is under the disk head.

3.3.4 Transfer Time

The time required for actual transfer of data between the disk and main memory.

Transfer Time = Block size * Transmission Factor

4. MATHEMATICAL MODE

4.1 Some Notations used in the paper

Following table shows the Notations that we are going to used. [4]

Table 1:Table Notations

T	A set of request **T**= {T0,T1..}
N	No. of input requests
Ti	A real-time request Ti=(ri,tp,bi,ai)
Ri	Release Time of *Ti*
Ai	Block access of *Ti*
Bi	Block Size of *Ti*
Tp	Transmission type
Di	Deadline of *Ti*
AET	Average Execution time of *Ti*
Ei	Response time of *Ti*
Ci	Current Head location
Si	Desired Disk Head Location
Ts	Seek Time of *Ti*
Tt	Transfer Time
Ttt	Total transaction Time
Ft	Fulfill time
Li	End Disk Head Location
Fj	Fulfil time of last request

4.1.1 Transmission Type

Different operations perform on disk are:-

1) Read(R)
2) Write (W)
3) Read-Write (RW)
4) Read-Write-Compute (RWC)

Transmission Factor considered according to the transmission type:

Table 2: Transmission Factors

Transmission Type	Transmission Factor
Read	0.6
Write	0.6
Read-Write	1.2
Read-Write-Compute	1.8

4.1.2 Assumptions made while Calculations

Table 3: Assumptions

Assumption	**Value**
Slack factor	2
Seek Factor	0.3

4.1.3 Example

Suppose a real-time transaction request are as Ti(Release Time,Transaction Type,Block Size, Block Location)

T0(0,R,4,30); T1(2,R,2,24)

T2(1,R,6,26) T3(3,R,4,14)

T4(2,R,8,16) T5(4,R,6,21)

T6(5,R,3,13) T7(1,R,2,20)

T8(5,R,8,15) T9(6,R,4,32).

4.2Calculations for AET and Deadline

4.2.1 For Transaction T0

Arrival Time ri = 0
Transmission operation tp = R mean read operation
Transmission Factor =0.6
Block size bi=4
Access Block ai = 30
Average Execution Time AET = 1.5 * bi =1.5 *4= 6
Deadline = ri + Slack factor* AET
= 0 + 2 * 6 = 12

4.2.1 For Transaction T1

Arrival Time ri = 2
Transmission operation tp = R mean read operation
Transmission Factor =0.6
Block size bi=2
Access Block ai = 24
Average Execution Time AET = 1.5 * 2 =3 Deadline = ri + Slack factor * AET = 2 + 2 * 3 = 8.
Similarly we get the values for other transactions. Following table shows AET and Deadline for all the transactions:

Table 4: AET and Deadline Calculations

Tid	Ri	Tp	bi	ai	AET	Di
T0	0	R	4	30	6	12
T1	2	R	2	24	3	8
T2	1	R	6	26	9	19
T3	3	R	4	14	6	15
T4	2	R	8	16	12	26
T5	4	R	6	21	9	22
T6	5	R	3	13	4.5	14
T7	1	R	2	20	3	7
T8	5	R	8	15	12	29
T9	6	R	4	32	6	18

4.3 Calculations for other parameters

Now for calculating other parameters consider following schedule according to their deadline order and arrival time.

{T7 T1 T0 T6 T3 T9 T2 T5 T4 T8}

4.3.1For Transition T0

Response time= max (arrival time of current record, Fulfill Time of last record)=max(*ri*,*fj*)=max(0,0)=0
Current Disk Head Location =Initial Disk Head Location=12
Desired Disk head Location=Block to access=30
Seek Time = seek factor * [abs(Desired Disk head Location – Current disk Head Location)]=0.3 * abs(12-30) = 0.3 * 18 = 5.4
Transfer Time = Transmission factor * Block size = 0.6 * 4=2.4
Total transmission Time = seek Time + Transfer Time = 5.4 + 2.4 = 7.8
Fulfill time = Total Transmission Time + Response Time = 7.8+ 0 =7.8

4.3.2For Transition T1

Response time= max (arrival time of current record, Fulfill Time of last record)=max(*ri*,*fj*)=max(2,7.8)=7.8
Current Disk Head Location =Initial Disk Head Location=24
Desired Disk head Location=Block to access=34
Seek Time = seek factor * [abs(Desired Disk head Location – Current disk Head Location)]=0.3 * abs(34-24) = 0.3 * 10 = 3
Transfer Time = Transmission factor * Block size = 0.6 * 2 =1.2
Total transmission Time = seek Time + Transfer Time = 3 + 1.2 = 4.2
Fulfill time = Total Transmission Time + Response Time = 4.2+ 7.8=12
Similarly we get the values for other transactions. Following table shows calculated parameters for all the transactions:

Table 5: Parameter Calculations

Tid	Rt	Ht	Si	Li	ST	TT	TTT	Ct
T0	0	12	30	34	5.4	2.4	7.8	7.8
T1	7.8	34	24	26	3	1.2	4.2	12
T2	12	26	26	32	0	3.6	3.6	15.6
T3	15.6	32	14	18	5.4	2.4	7.8	23.4
T4	23.4	18	16	24	0.6	4.8	5.4	28.8
T5	28.8	24	21	27	0.9	3.6	4.5	33.3
T6	33.3	27	13	16	4.2	1.8	6	39.3
T7	39.3	16	20	22	1.2	1.2	2.4	41.7
T8	41.7	22	15	23	2.1	4.8	6.9	48.6
T9	48.6	23	32	36	2.7	2.4	5.1	53.7

4.4 Calculations for Service Table

4.4.1Service Table:

The service table is used to draw the timing diagram for particular algorithm that schedules the transactions.[5] From the timing diagram we can decide the number of hits and missed that occurs in schedule.

Cj,i=Execution time required for ith transaction after servicing jth transaction.

Cji= abs[End block of previous(Tj) – Start block of current(Ti)] *0.3 + Transfer time of current(Ti).

[T0,T1]=[34-24]*0.3+1.2=4.2

[T0,T2]=[34-26]*0.3+3.6=6

[T0,T3]=[34-14]*0.3+2.4=22.7

[T0,T4]=[34-16]*0.3+4.8=10.2

[T0,T5]=[34-21]*0.3+3.6=31.3

[T0,T6]=[34-13]*0.3+1.8=8.1

[T0,T7]=[34-20]*0.3+1.2=5.4

[T0,T8]=[34-15]*0.3+4.8=10.5

[T0,T9]=[34-32]*0.3+2.4=3

Similarly we get the values for other transactions. Following table shows service time calculations for all the transactions:

Table 6 : Service Table

Cji	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
T0	-	4.2	6	22.7	10.2	31.3	8.1	5.4	10.5	3
T1	3.6	-	3.6	6	7.8	5.1	5.7	3	8.1	0.6
T2	3	3.6	-	7.8	7.2	6.9	7.5	4.8	9.9	2.4
T3	1.2	0.6	1.2	-	5.4	2.7	3.3	0.6	5.7	1.8
T4	0.6	4.2	3	5.4	-	2.7	3.3	0.6	5.7	1.8
T5	1.5	5.1	3.9	6.3	8.1	-	7.8	3.3	8.4	3.9
T6	1.8	1.8	0.6	3	4.8	5.1	-	2.4	5.1	7.2
T7	8.1	3.6	2.4	4.8	6.6	2.7	4.5	-	6.9	0.6
T8	0.3	3.9	2.7	5.1	6.9	5.4	4.8	2.1	-	0.3
T9	4.2	7.8	6.6	9	10.8	8.1	8.7	6	11.1	

4.4.2 Timing Diagram

Following figure shows the timing Diagram for the above mentioned schedule as:

T7 T1 T0 T6 T3 T9 T2 T5 T4 T8

The figure2 shows the number of Hits and number of Miss. The Transaction is said to be Hit if it is completed within its Deadline. Otherwise it is said to be Miss. Here number of Hits are two{ T7 T0} and number of Miss are eight{T1 T2 T3 T4 T5 T6 T8 T9 }

Fig2: Timing diagram

5. CONCLUSION

Thus, in case of Real time database systems along with consistency and integrity constraints it also considers timing constraints. A timing constraint is expressed in the form of a deadline, a certain time in the future by which a transaction needs to be completed. If we have to minimize the execution time of the transaction, then we should minimize the time related to the disk, as most of the time required while serving the transaction is related to disk. Minimizing time related to disk require to minimize the rotational latency, seek time, and transfer time. If we have to minimize the rotational latency, we need to rotate the disk in high speed. But rotating disk in high speed has disadvantages like, disk becomes hotter and it consumes more power. Decreasing Transfer time is not our task as it depends on the transaction size, more is the transaction size more is the time required. The only scope to decrease the time related to disk is by decreasing the seek time. Decreasing seek time means *seek optimization*. Seek optimization can be done by decreasing mechanical motion of the disk head.

To achieve this we have present our mathematical model for real Time Database System and we have made a brief discussion over the real-time disk scheduling parameters. We are familiar with the terms & parameter to be used in the upcoming scheduling algorithm. And also we have learned how to calculate these parameter and how to draw timing diagram with the help of service table.

6. REFERENCES

[1] Ben Kao and Hector Garcia-Molina “An Overview of Real-Time Database Systems”, in proceedings of NATO Advanced Study Institute on Real-Time Computing. St. Maarten, Netherlands Antilles, Springer-Verlag, 1993.

[2] Saud A. Aldarmi, “*Real-time Database Systems: Concepts and Design” ,* Department of Computer science, The University of York, April 1998.

[3] Fengxiang Zhang, Alan Burns, Sanjoy Baruah, “Task Parameter Computations for Constraint Deadline Real-Time Systemswith EDF Scheduling” 201O International Conference On Computer Design And Appliations (ICCDA 2010) Volume 3.

[4] R. 1. Chang, W K. Shih, and R. C. Chang, "Deadline-modification-scan with maximum scannable-groups for multimedia real-time disk scheduling," in Proceedings of the 19th IEEE Real-Time Systems Symposium, pp. 40-49, 1998.

[5] H.-P. Chang, R.-I. Chang, W.-K. Shih, and R.-C. Chang, "GSR: A global seek-optimizing real-time disk-scheduling algorithm," The Journal of Systems and Software, vol. 80, no. 2, pp. 198-215, 2007.