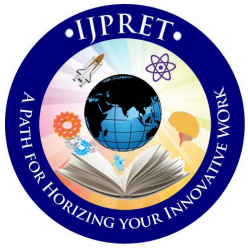
Research Article *ISSN: 2319-507X*
Salim Amdani, IJPRET, 2013; Volume 1(8): 601-609 *IJPRET*



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

HIGH TROUGHPUT ACHIEVEMENT USING A MODIFIED GSR APPROACH

SALIM. Y. AMDANI, M. S. ALI, SUREKHA. ANKITA. A. MAHAMUNE

1. **Associate Professor, Department of CSE, BNCOE, Pusad, India.**
2. **Principal, P.R.M.C.E. & M, Bandera, India.**
3. **M.E. Student, Department of CSE, BNCOE, Pusad, India.**

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Disk,

Real-Time,

Disk scheduling,

Deadline,

Scheduling parameters,

Throughput

Corresponding Author

Mr. Salim. Y. Amdani

Abstract

Disk scheduling algorithm plays an important role in the real-time applications. Real time transactions have the property that they must be completed before their assigned deadlines. If they do not complete within their deadlines, then they are said to be missed. To find the number of missed transactions various real time disk scheduling algorithms are proposed. Such as Earliest-deadline-first (EDF) meets time constraints, but it does not achieve high throughput. By contrast, SCAN can yield significant throughput, but may miss the deadline. So, different hybrid algorithms are proposed, which combines the features of EDF and SCAN. One of these is GSR (Globally Seek Optimizing Rescheduling) algorithm, which reschedule task to anywhere in the input schedule to improve data throughput [9]. However, GSR does not consider initial schedule generation and the track difference between the last task of current scan-group and the first task of the next scan-group. So, in this paper, we have proposed a modified approach based on GSR. First, EDF, GSR and the proposed approach are applied to find the transaction servicing schedule and then using the concept of timing diagram we have found number of hit transactions and throughput for each of these algorithms. Depending upon the obtained results, we have evaluated their performance and shown how our new approach performs better than the previous algorithms.

With the advancement of technology, many applications are requiring large amount of information to be handled and managed in a timely manner. Meanwhile, database management systems are designed around such a concept; that is, with the sole goal of managing data as a resource. In an attempt to achieve the advantages of both systems, an integration of the two technologies resulted in combined systems known as *Real-Time Database Systems* [1]. Typically, a timing

constraint is expressed in the form of a deadline, and 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

T	A set of request **T**= {T0,T1..}
N	No. of input requests
Tid	A real-time request
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
SI	Start Disk Head Location
Ci	Current Head location
Tt	Transfer Time
Ft	Fulfill time
EI	End Disk Head Location
Fj	Fulfil time of last request

[6]. In a disk-based database system, disk I/O occupies a major portion of transaction execution time. 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[3,4].

As we know, before giving service to transactions, it is necessary to schedule

them and find the required time [5]. 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. This task can be achieved with the help of a mathematical model which shows how scheduling result of any algorithm can be evaluated [6].

To meet timing constraints, various real time disk scheduling algorithms are proposed like EDF, SCAN-EDF and GSR. But each has its own drawbacks. So a new

algorithm based on GSR is proposed, which provides a new approach to generate initial schedule. In addition, it considers the track distance between the last task of current group and the first task of the next group that achieves significant data throughput under real-time constraints [7].

USE OF A MATHEMATICAL MODEL

The mathematical model uses **following parameters with their notations:**

Table 1: Table Notations

EXAMPLE: To illustrate the working of EDF, GSR and modified GSR (proposed approach) consider the following set of transactions with their assigned parameter values. Here, we assumed the initial disk

head position (IDHP) = 4.

Tid	Ri	Tp	Bi	Ai (Start block SI)	(End Block EI)	AET =1.5* Bi	Di =2*AET +Ri	Tt=0.6* Bi
T0	1	R	3	12	14	4.5	10	1.8
T1	0	R	2	11	12	03	6	1.2
T2	0	R	5	05	09	7.5	15	3
T3	5	R	6	19	24	09	23	3.6
T4	5	R	4	15	18	06	17	2.4
T5	4	R	5	11	15	7.5	19	3

Table 2: Parameter Calculations

Now, we find the service time Cj,i. We have, Cj,i=Execution time required for ith transaction after servicing jth transaction. And Cj,i= abs[End block of previous(Tj)

Start block of current(Ti)] *0.3 + Transfer time of current(Ti).

After calculations, the values Cji for all the permutations of Tj and Ti are as shown in following table:

Cji	T0	T1	T2	T3	T4	T5
T0	-	2.1	5.7	5.1	2.7	3.9
T1	1.8	-	5.1	5.7	3.3	3.3
T2	2.7	1.8	-	6.6	4.2	3.6
T3	5.4	5.1	8.7	-	5.1	6.9
T4	3.6	3.3	6.9	3.9	-	5.1
T5	2.7	2.4	6	4.8	2.4	-

Table 3 : Service Table

Now, we find Schedule for above example using EDF, GSR and the modified GSR as follows:

EDF-EARLIEST DEADLINE FIRST

EDF serves the transactions in the ascending order of their deadlines. So, the EDF schedule for the given example will be as follows:

EDF SCHEDULE: T1 T0 T2 T4 T5 T3

Timing diagram for the EDF schedule:

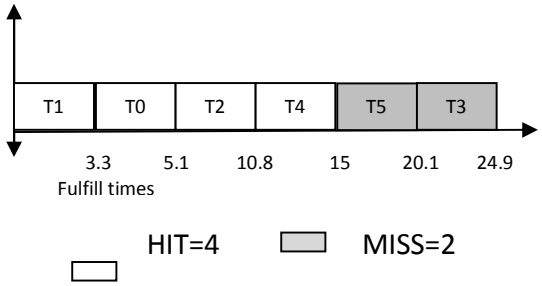


Fig.1. Timing diagram for the EDF

GSR- GLOBALLY SEEK OPTIMIZING RESCHEDULING ALGORITHM

Working of GSR (for above example):

GSR composed of following 3 steps:

1) EDF-to-SCAN Mapping (ESM)
2) Scan Group Identification (SGI)
3) GSR Rescheduling

EDF-to-SCAN Mapping (ESM)

EDF-to-SCAN mapping is a bipartite mapping obtained by connecting each node in EDF schedule to corresponding node in SCAN schedule with an edge. The SCAN and EDF schedules for the above example are shown below:

SCAN Schedule: T2 T1 T0 T5 T4 T3

EDF Schedule: T1 T0 T2 T4 T5 T3

EDF-to-SCAN mapping is done as follows:

SCAN Schedule

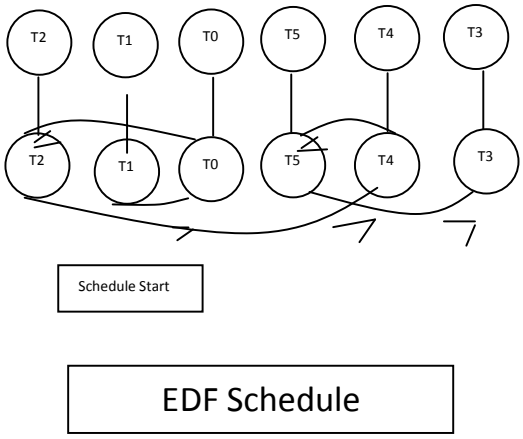
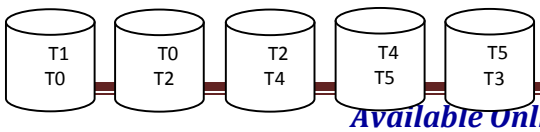


Fig.2. EDF-to-SCAN mapping

Using this mapping, we can investigate the possible transformations from the real-time EDF schedule to the seek-optimized SCAN schedule to find a good real-time disk schedule.

Scan-group identification (SGI):

This algorithm is used to decompose the input schedule into a sequence of scan-groups where, each scan-group contains the maximum number of contiguous transactions with the same SCAN direction. Using this algorithm, for the above mapping, we obtained the following five SCAN groups:



S1 **S2** **S3** **S4** **S5**

Fig.3. SCAN Groups

GSR Rescheduling

After identifying all scan-groups, GSR algorithm tries to reschedule each task into all scan-groups before its own scan-group and compute the improvement of data throughput of each rescheduling result. The result with the largest throughput improvement is selected for rescheduling.

After applying GSR, we get the following final GSR schedule.

Final GSR schedule : T1 T0 T4 T2 T5 T3

Timing diagram for the GSR schedule:

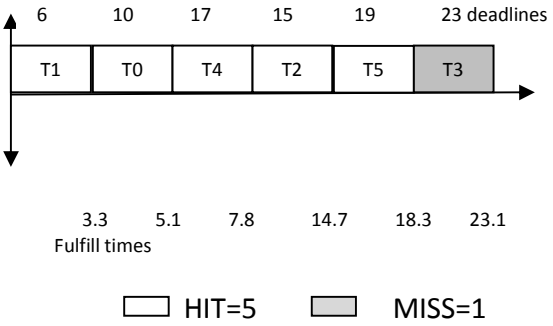


Fig.4. Timing diagram for the GSR

From above figure, it is seen that the fulfill time required for the EDF (=24.9) which is greater than that for the GSR (=23.1). Also GSR has large number of hit transactions =5 than the input EDF schedule =4. Next we illustrate our new algorithm based on GSR for the same set of transactions.

PROPOSED APPROACH

This new approach first generates the initial schedule considering the track locations of the transactions and then this input schedule is decomposed into the SCAN groups and final rescheduled result is found. This new approach consists of three steps which are explained below:

Working of proposed approach:

The new approach differs from GSR only in the first step that is EDF to SCAN mapping. In that, instead of taking EDF as an input schedule we generate the input schedule as follows.

Generating the input schedule

The priority Pi is assigned to each transaction Ti such that **Pi= α*Di + (1- α)* Ai** where, (0< **α** < 1) is the percent of the deadline Di for Ti, Di is the deadline of Ti, and Ai is the track location(Start Block) of Ti. Then initial schedule will contain the transactions in an ascending order of their priority values Pi as follows:

THE INPUT SCHEDULE: T2 T0 T1 T5 T4 T3

Now, mapping is formed considering above schedule as an input schedule unlike GSR where EDF is considered as an input schedule. This mapping is done as follows:

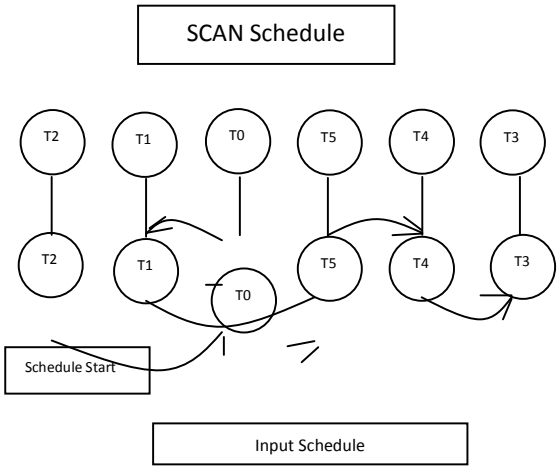
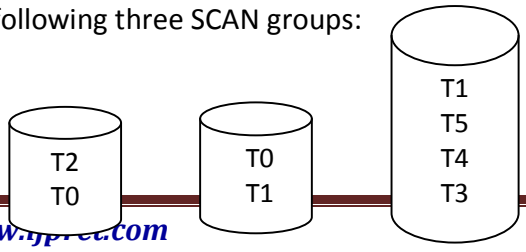


Fig.5. Mapping for proposed approach

Once this mapping is done, we apply second and third steps of GSR as follows:

Generating the SCAN groups

For the above mapping, we obtained the following three SCAN groups:



S1 S2 S3

Fig.6. SCAN Groups

Now the algorithm GSR is applied and we get the following final schedule using our proposed approach :

Final schedule: **T2 T0 T1 T5 T4 T3**

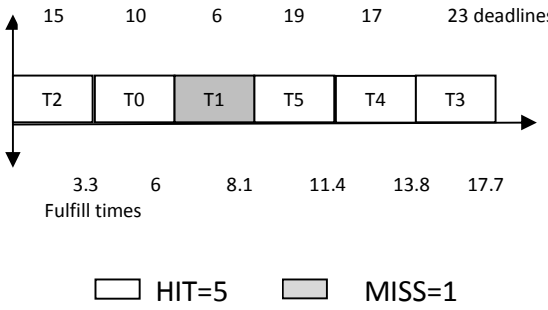


Fig.7. Timing diagram for proposed approach

PERFORMANCE EVALUATION

In this section, we compared our proposed algorithm with the GSR and EDF algorithms. We use data throughput to evaluate the performance of real-time system. The proposed algorithm is efficient to improve disk throughput while guaranteeing the restriction of time. Following graph shows the data throughput improvement of proposed algorithm as compared to EDF and GSR.

Performance Graph:

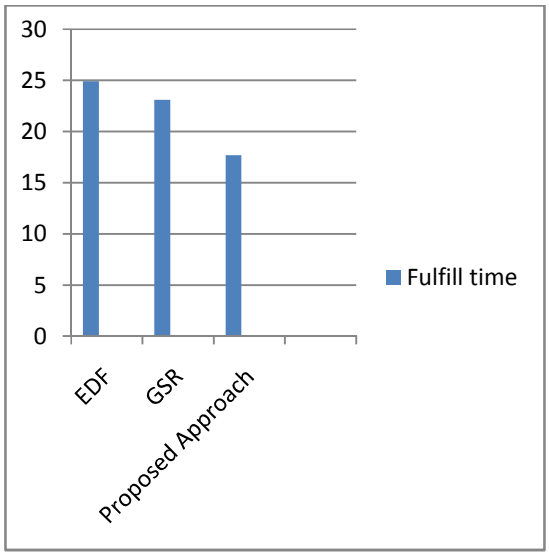


Fig.8. Performance Graph

Above performance graph shows that the fulfill times for EDF and GSR are greater than that of proposed approach. This means number of transactions completing per unit time are more in case of our proposed approach and thus achieves higher disk throughput while guaranteeing the real time constraints.

CONCLUSION

In order to improve data throughput, the seek-optimizing SCAN scheme should be employed to reschedule the input tasks as much as possible. In this paper, we have proposed a new real-time disk scheduling algorithm based on GSR. This new algorithm is superior to EDF and GSR in that it gives greater number of hit transactions than EDF and also fulfill time required is less than the EDF and GSR. This new algorithm can get a rescheduled result where GSR may fail and obtains the shorter response time (fulfill time) than that of the GSR. Performance graph shows that our new algorithm obtain significant data throughput in real-time disk scheduling.

REFERENCES

1. Saud A. Aldarmi: “Real-Time Database Systems: Concepts and Design”. Tech. Report. Department of Computer Science ,The University of York (April 1998)]

2. S.Y. Amdani and M.S. Ali: “An Overview of Real-Time Disk Scheduling Algorithms”. International Journal on Emerging Technologies 2(1): 126-130(2011)].

3. [3] Sameh Mohamed Ibrahim Elnikety "A Real-Time Disk Scheduling Algorithm for Multimedia Storage Servers" Thesis, Alexandria University, 1999.

4. Arezou Mohammadi and Selim G. Akl "Scheduling Algorithms for Real-Time Systems" Technical Report No. 2005-499,, This work was supported by the Natural Sciences and Engineering Research Council of Canada, 2005.

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

6. S.Y.Amdani, M.S.Ali, S.M.Mundada: “Mathematical Model for Real Time Disk Scheduling Problem”, Emerging Trends in Computer Science and Information Technology -2012(ETCSIT2012) Proceedings published in International Journal of Computer Applications® (IJCA)

7. Nianmin YAO, Jinzhong CHEN, Ang LI: “An Inter-group Seek-optimizing Disk Scheduling Algorithm for Real-time System”,Journal of

Computational Information Systems 8: 18 (2012) 7579-7586].

8. Chang, R. I., Shin, W. K., Chang, R. C. "Deadline-modification-SCAN with maximum-scannable -groups for multimedia real-time disk scheduling". In: proceedings of the 19th IEEE Real-Time System Symposium, pp. 40 -49, 1998.

9. Chang, H. P., Chang, R. I., Shih, W. K., Chang, R. C. "GSR: A global seek-optimizingreal-time disk-scheduling algorithm". Journal of Systems and Software, 198- 215,2007.