www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 8 Issue 12 December 2019, Page No. 24890-24893
ISSN: 2319-7242 DOI: 10.18535/ijecs/v8i12.4399

Comparative performance analysis of optimized round robin scheduling(ORR) using dynamic time quantum with round robin scheduling using static time quantum in Real Time System

Pallab Banerjee[1], Biresh kumar[2], Amarnath Singh[3] , Rahul Kumar[4], Ritik kumar[5]

[1,2,3]Assistant Professor, [4,5]B.Tech Scholar
Amity School of Engineering and technology, Amity University Jharkhand

Abstract:
One of major component of operating system is task scheduling for the optimum utilization of the resources. Round Robin had been an effective task scheduling method so far, but it has certain limitations. It uses static time quantum which sometimes leads to starvation. The proposed Optimised Round Robin is a modified version of the existing Round Robin scheduling which results in better average time and average turnaround time and overall increase in the performance. The comparative analysis is being done that indicates ORR gives improvement in the system performance.

Keywords: Task scheduling, Round Robin, dynamic time quantum, average waiting time, average turnaround time, context switching.

1. Introduction

An operating system is an interface between computer user and computer hardware. An Operating system is software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers. Modern operating system and time-sharing system are more complex, they have evolved from a single task to multitasking environment in which processes run in synchronized manner. Objective of multiprogramming is to maximize resource utilization, not possible to achieve without proper scheduling. All resources are scheduled before use. In a multiprocessing and multitasking environment if several processes are ready to run at the same time, the system must choose among them and assigned to run on the available CPUs, is called CPU scheduling. Allocating CPU to a process requires careful awareness to assure justice and avoid process starvation for CPU. Scheduling decision try to reduce the following: turnaround time, response time and average waiting time for processes and number of context switches. CPU scheduling algorithm decides which of the processes in the Ready Queue (RQ) are to be allocated to the CPU. There are many different CPU scheduling algorithms used like FCFS, SJF, RR, Priority scheduling algorithm and Short Remaining Time Next (STRN) Remaining Time Next (STRN) algorithm. The processes are scheduled according to the given burst time, arrival time, time quantum and priority. Out of those algorithms, Round Robin (RR) is the oldest, simplest and most widely used proportional share scheduling algorithm. It is like FCFS scheduling, but preemption is added to switch between processes. In Round Robin algorithm a small unit of time slice are required which is called Time Quantum (TQ). The CPU scheduler goes around Ready Queue and allocates the CPU to each process by the help of Dispatcher for a time interval of up to 1 Time Quantum (TQ). If new process arrives then it is added to the tail of Circular Queue. The CPU scheduler picks the first process from the Ready Queue sets a timer to interrupt after one Time Quantum and dispatches the process. After TQ is expired, the CPU preempts the process and the process is added to the tail of the Circular Queue. If the process finishes before the end of the TQ, the process itself preempts the CPU willingly [1]. In this paper, we tried to solve the Time Quantum problem by adjusting the Time Quantum

Dynamically with respect to the existed set of processes in Ready Queue

2. Preliminaries:

In round robin scheduling algorithm, a time quantum is assigned to each process that is static. The performance of RR algorithm depends heavily on the size of the time quantum. For smaller time quantum, the context switching is more and for larger time quantum, response time is more. Overall performance of RR may decrease for weak time quantum selection. Therefore, choice of an appropriate time quantum is necessary. Many researchers had tried to overcome these problems in real by giving their own methodologies. The recent studies made from references have shown that if dynamic time quantum is adapted, waiting time, turnaround time, context switches and throughput will be reduced to some larger extent instead of having fixed time quantum. [2] This algorithm improves the performance better than the priority based round robin scheduling.

3. Proposed Work:

The traditional round robin scheduling is an efficient scheduling method in terms of starvation and execution. In the proposed method, the process is sorted according to their arrival time. Queues are being formed based on the median and Quantum time value is calculated for each ready queue. Each time quantum is valid for each queue. The algorithm will calculate the time quantum value by calculating the mean of burst time of the processes present in ready queue.

$$q = \frac{\sum B_i}{\sum i}$$

where q is the time quantum and B_i is the burst time of task T_i present in Queue.

A. ALGORITHM

Considering that this Optimised round robin scheduling (TARR) considers that processes are arriving at different instances. The steps of algorithm are showing below—

Step 1: Initialization
P_i // Process number
A_i // Arrival time of processes
B_i // Burst time of processed
Step 2: Sort submitted tasks, T_i, i =1, 2, . . . , according to their
Burst time B_i.
Step 3: Compute the median by taking the Burst time Bi of all the process P_i.
Step 4: If a burst time B_i of a Process P_i, is less than or equal to
the median, insert P_i into a Q1 otherwise insert P_i into Q2.
Step 5: The quantum of (q_i) is calculated by calculating the average of all burst times in the queue. (whether it is from Q1 or Q2)
Step 6: In case of the of a new task arrival or a task is finished q ~ will be updated dynamically.
Step 7: If Q1 AND Q2 empty, Terminate.

3 Analysis

To evaluate the efficiency of the suggested algorithm, let us consider some cases. The performance of ORR has been stimulated along with the traditional RR in the below cases considering the arrival time and burst time.

Case 1: Taking five processes P, Q, R, S and T of varying burst times, arriving at the same time, as shown in table1.1. below. The order of execution for both the algorithms is shown, and the outcome is collated in table 1.2.

Processes	Burst time	Arrival time
P	11	0
Q	24	0
R	37	0
S	52	0
T	71	0

Table 1.1. Process with Burst Time and Arrival time

Model	Context Switches	Average Waiting time	Average Turnaround time
RR	12	76.400000	115.400000
ORR	4	48.400000	87.400000

Table 1.2. Comparison between RR & ORR algorithm

TQ=20

P	Q	R	S	T	Q	R	S	T	Q	R	S

0 11 31 51 71 91 95 112 132 152 164 184 195

Fig.1.1. Gantt chart of RR from Table 1.1. of Case 1.

TQ=24 **TQ=61.5**

P	Q	R	T	S

0 11 35 72 124 195

Fig.1.2: Gantt chart of ORR from Table 1.1 of Case 1.

Case 2: Now let us consider another set of processes as given below and compare the test results.

Processes	Burst time	Arrival time
P	21	0
Q	34	0
R	47	0
S	62	0
T	81	0

Table 2.1. Process with Burst Time and Arrival time

Model	Context Switches	Average Waiting time	Average Turnaround time
RR	14	128.4	173.4
ORR	4	68.4	117.4

Table 2.2. Comparison between RR & ORR algorithm

TQ = 20

P	Q	R	S	T	P	Q	R	S	T	R	S

0 20 40 60 80 100 101 115 135 155 175 182 202

T	S	T	T

202 222 224 244 245

Fig.2.1. Gantt chart of RR from Table 2.1. of Case 2.

TQ = 34 **TQ = 71.5**

P	Q	R	R	S	T	T

0 21 55 89 102 164 235.5 245

Fig.2.2. Gantt chart of ORR from Table 2.1. of Case 2.

Case 3: Now let us consider another set of processes as given below and compare the test results.

Processes	Burst time	Arrival time
P	6	0

Q	19	0
R	32	0
S	47	0
T	66	0

Table 3.1. Process with Burst Time and Arrival time

Model	Context Switches	Average Waiting time	Average Turnaround time
RR	9	58.400000	88.400000
ORR	4	38.400000	72.400000

Table 3.2. Comparison between RR & ORR algorithm

TQ = 20

P	Q	R	S	T	R	S	T	S	T	T

0 6 25 45 65 85 97 117 137 144 164 170

Fig.3.1. Gantt chart of RR from Table3.1. of Case 3.

TQ = 19 **TQ = 56.5**

P	Q	R	R	S	T	T

0 6 25 44 57 104 160.5 170

Fig.3.2. Gantt chart of ORR from Table 3.1. of Case 3.

Case 4: Now let us consider another set of processes as given below and compare the test results.

Processes	Burst time	Arrival time
P	14	0
Q	25	9
R	35	11
S	47	14
T	62	18

Table 4.1. Process with Burst Time and Arrival time

Model	Context Switches	Average Waiting time	Average Turnaround time
RR	9	49.200000	83.800000
ORR	5	39.200000	75.800000

Table 4.2. Comparison between RR & ORR algorithm

TQ = 20

P	Q	R	S	T	Q	R	S	T	S	T	T

0 14 34 54 74 94 99 114 134 154 161 181 183

Fig.4.1. Gantt chart of RR from Table 4.1. of Case 4.

TQ=14 **TQ = 30** **TQ = 47** **TQ = 62**

P	Q	R	R	S	T

0 14 39 69 74 121 183

Fig.4.2. Gantt chart of ORR from Table 4.1. of Case 4.

Comparative analysis of all the cases:

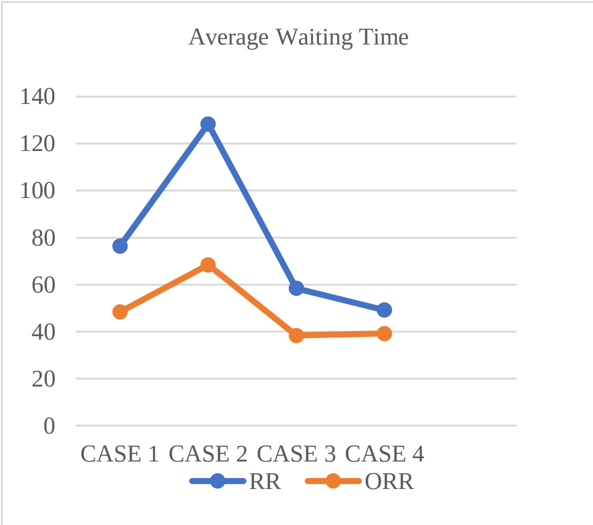

Fig. 5.1. Comparison of average Waiting Time of RR and ORR.

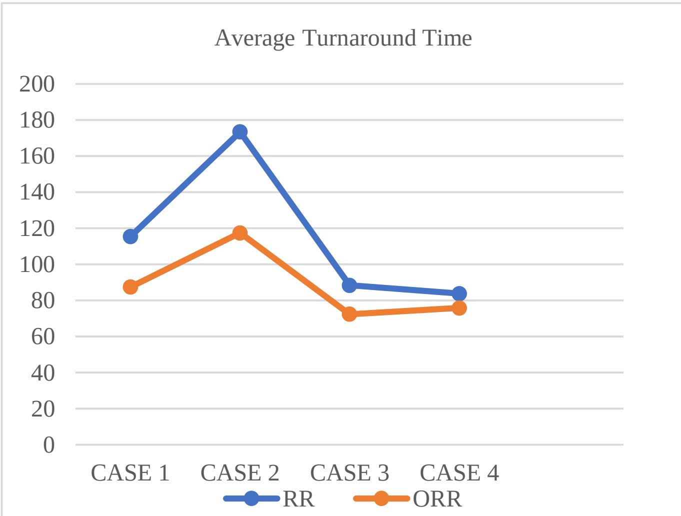

Fig.5.2. Comparison of average Turnaround Time of RR and ORR.

4. Conclusion

We have successfully compared the Round Robin (RR) algorithm and the optimized RR algorithm and derived a conclusion that the proposed algorithm is effective in terms of context switches, throughput, average turnaround time and waiting time which in turn increase the overall performance. Using this algorithm, the performance of time-sharing systems can be enhanced, and further modifications can be done to amplify the performance of a multiprogramming operating system and real-time systems. For the future perspective, this paper might help in enhancing the algorithm for much better results.

5. References

[1] "Tanebaun, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13:9780136006633, pp: 1104.

[2] Raja Ram J., Geetha K., Mohan R. (2013),"An Intelligent Adaptive Round Robin (IARR) Scheduling algorithm for performance in real time system", International Conference on Advances in Engineering and Technology Pages 813-819.

[3] "Tanebaun, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13:9780136006633, pp: 1104.

[4] Pallab banerjee, probal banerjee, shweta sonali dhal, "Performance Evaluation of a New Proposed Average Mid Max Round Robin (AMMRR) Scheduling Algorithm with Round Robin Scheduling Algorithm",IJARCSSE,ISSN:2277-128X, Volume-2, Issue- 8, August 2012.

[5] Pallab banerjee, probal banerjee, shweta sonali dhal,"Comparative Performance Analysis of Even Odd Round Robin Scheduling Algorithm (EORR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum" IJARCSSE,ISSN: 2277-128X,Volume-2, Issue-9, August 2012.

[6] Pallab banerjee, probal banerjee, shweta sonali dhal,"Improved High Performance Round Robin Scheduling Algorithm(HPRR)using Dynamic Time Quantum" International Journal of Computer Information System,ISSN: 2277-128X,Volume-5, No-3, 2012.

[7] Sarojhiranwal and D.r. K.C.Roy"Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice".volume 2,No. 2,July-Dec 2011,pp. 319-32.

[8] Pallab banerjee, probal banerjee, shweta sonali dhal, "Comparative Performance Analysis of Mid Average Round Robin Scheduling Algorithm (MARR) using Dynamic TimeQuantum with Round Robin Scheduling Algorithm having static Time Quantum",IJECSE,ISSN: 2277- 1956, Volume-1,Issue-4, August 2012.

[9] H. S. Behera, R. Mohanty, and D. Nayak, "A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis," vol. 5,no. 5, pp. 10-15, August 2010.

[10] Sanjay Kumar Panda and Saurav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max DispersionMeasure" ISSN : 0975-3397 ,Vol. 4 No. 01, January 2012.

[11] Tarek Helmy, Abdelkader Dekdouk " Burst Round Robin: As a Proportional-Share Scheduling Algorithm", IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations, pp. 424-428, 11-14 November,2007.

[12] Yaashuwanth .C & R. Ramesh "Inteligent time slice forround robin in real time operating system", IJRRAS 2 (2), February 2010.

[13] R. J. Matarneh, " Seif-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Proceses ", American Journal of Applied Sciences 6 (10),pp. 1831-1837, 2009.

[14] H. S. Behera, Rakesh Mohanty, Sabyasachi Sahu and Sourav Kumar Bhoi. "Comparative performance analysis of multidynamic time quantum round robin (mdtqrr) algorithm with arrival time", ISSN : 0976-5166, Vol. 2, No. 2,Apr- May2011,pp.262-271.

[15] Pallab banerjee ,Prof Dr L.N.Padhy. "Comparative analysis of Maximum performance round robin(MPRR) by Dynamic Time Quantum with static time quantum",ISSN:2277- 128X,pg.372-377Vol-4,Issue-11,Nov-2014

[16] Pallab banerjee,Talat Zabin,Shweta Kumari,Pushpa Kumari "Comparative performance analysis of best performance of round robin scheduling algorithm(BPRR) using Dynamic Time quantum with priority based round robin(PBRR) CPU Scheduling algorithm in Real Time System".ISSN:2277- 1956,Vol-4,Number-2, pg. 151-159,May 2015.

[17] R.Nallakumar ,Dr.N.Sengottaiyan ,S.Nithya "A Servey of Task Scheduling Methods in Cloud Computing "IJCSE ,ISSN :2347-2693 ,Volume-2,Issue-1,2014

[18] Abbas Noon,Ali Kalakech,Seifedine Kadry"A New Round Robin Scheduling Algorithm for Operating Systms: Dynamic Quantum Using Mean Average"IJCSI,ISSN:1694- 0814,Vol-8,Issue-3,No-1,May-2011

[19] J. Geralds, "Sega Ends Production of Dreamcast," vnunet.com, para. 2, Jan. 31, 2001. [Online]. Available: http://nl1.vnunet.com/news/1116995. [Accessed: Sept. 12, 2004]. (General Internet site)