Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
www.ijarcsse.com
Research Paper
Available online at: www.ijarcsse.com

Comparative analysis of Maximum Performance Round Robin (MPRR) by Dynamic time Quantum with Static time Quantum

Pallab Banerjee[1], Prof. Dr.L.N.Padhy[2]
[1]M.Tech Scholar, [2]Professor
Department of Computer Science and Engg., Konark Institute of Science and Technology,
Biju Patnaik University of Technology, Bhubaneswar, Odisha, India

Abstract- A Round Robin Scheduling Algorithm is a preemptive CPU Scheduling algorithm in which the scheduler processes each job and switches between all the processes when time Quantum expires. Round Robin scheduling algorithm is designed especially for time sharing and real time systems. It is a preemptive CPU scheduling algorithm which switches between the processes when static time quantum expires .Its disadvantages are its Longer Average Waiting Time, Higher Context Switches and Higher Turnaround Time. In this scheduling algorithm the main idea is to adjust the time quantum dynamically so that (MPRR) perform better performance than Round Robin scheduling algorithm.

Keywords-Operating System, Round Robin, Average Mid Max Round Robin, Turnaround time, Waiting time, Context Switch.

I. INTRODUCTION

Operating system is system software which makes an interface between end user and computer hardware, so that the user can handle the system in a convenient manner. Scheduling is the most repetitively used fundamental concept in OS. In multitasking and multiprogramming environment it is necessary to choose the process among the number of process present in the job pool according to their need. Allocation of CPU to the processes is done by scheduler, which operated by some scheduling algorithms. FCFS, SJF, Priority & RR are different type of scheduling algorithms. From these entire algorithm Round Robin algorithm is the most popular non-preemptive scheduling algorithm which works on Real time system also. In non-preemptive scheduling algorithm, CPU is assigned to a process until its execution is completed. But in preemption, running process is forced to release the CPU by the newly arrived process.

II. CPU SCHEDULING ALGORITHMS

There are many scheduling algorithm are designed like FCFS, SJF, PRIORITY, ROUND ROBIN etc. In the First-Come-First-Serve (FCFS) algorithm, process that arrives first is immediately allocated to the CPU based on FIFO policy. In Shortest Job First (SJF) algorithm, process having shortest CPU burst time will execute first. If two processes having same burst time and arrive simultaneously, then FCFS procedure is applied. Priority scheduling algorithm, provides priority (internally or externally) to each process and selects the highest priority process from the ready queue. In case of Round Robin (RR) algorithm, time interval of one time quantum is given
to each process present in the circular queue.

III. PERFORMANCE METRICS

The proposed algorithm is designed to meet all scheduling criteria such as maximum CPU utilization, maximum throughput, minimum turnaround time, minimum waiting time and context switches. Here we are considering the performance criteria in each case of our experiment.
Turnaround Time (TAT)=Finish Time–Arrival Time. Average Turnaround Time should be less.
Waiting Time (WT)= Start Time- Arrival Time. Average Waiting Time should be less.
Context Switch (CS)=Switching between the processes. The number of context Switch should be less.

IV. RELATED WORK

In the last few years different approaches are used to increase the performance of Round Robin scheduling like High performance Round Robin (HPRR), Even-Odd Round Robin(EORR), Average Max Round Robin Algorithm (AMRR),Mid-Average Round Robin Scheduling(MARR),Min-Max Round Robin (MMRR),Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice, Multi-Dynamic time Quantum Round Robin (MDTQRR), Self-Adjustment Time Quantum in Round Robin (SARR),Dynamic Quantum with Re-adjusted Round Robin (DQRRR).

V. PROPOSED APPROACH

Let's assume that the burst time of the processes is taken as sorted increasing order so that it will give better turnaround time and waiting time. Generally in case of Round Robin algorithm the performance depends upon the size of fixed or static Time Quantum (TQ). If TQ is too large then Round Robin algorithm approximate to First Come First Served (FCFS). If the Time Quantum is too small then there will be many context switching between the processes. So, our approach solved this problem by taking a dynamic TQ. Here in the proposed algorithm firstly the mid process is found then Average from 1st process to mid process is found .After getting the average of up to mid process we will calculate **TQ=(AVG+(No of process from Mid to N-1))/(Number of Rest of the process from Mid to N-1)+1.**

VI. PROPOSED ALGORITHM

In our proposed algorithm, processes are already present in the Ready Queue (RQ). By default, Arrival Time (AT) is assigned to zero. The number of processes is 'n' and CPU Burst Time (BT) are accepted as input and Average Turnaround Time (ATT), Average Waiting Time (AWT) and number of Context Switch (CS) are produced as output.

1. Initialize CS=0, AWT=0, ATT=0.
2. Sort the process in the ready Queue in ascending order .
3. while(RQ != NULL)
{
// N= Number of Processes in the ready queue.
// BT= Burst Time of the Processes.
// RQ= Ready Queue.
//TQn=New Time Quantum.
//Lo=BT of 1st process in ready queue.
//Hi=BT of N process in the ready queue.
Lo=0,Hi=N-1
4. if(Lo<=Hi)
{
Mid=(Lo+Hi)/2;
AVG=((No of processes up to Mid)/(Mid+1));
TQ=(AVG+(No of process from Mid to N-1))/(Number of Rest of the process from Mid to N-1)+1;
}
5. for i=1 to N loop // Assign TQn to (1 to N)processes.
{
Pi->TQ //Assign TQn to all the available processes.
}}
//End of for.
//End of while
//If one process is there then after calculation TQn is equal to BT itself.
6. If (new process arrived and BT!=0 Or new process is arrived and BT==0 Or new process is not arrived and BT!=0)
then go to step 2.
else
go to step 7.
7. Calculate ATT,AWT,CS
//ATT=Average Turnaround time.
//AWT=Average waiting time.
//CS=Number of context switch.
8. End

VII. EXPERIMAENTAL ANALYSIS

In every case we will compare the result of the proposed Mid Average Round Robin (MPRR) method with Round Robin (RR) scheduling algorithm. Here we have taken 20 as the static time quantum (TQ) for RR algorithm.
CASE 1:-Let's consider five processes with Burst time (P1=25, P2=35, P3=55,P4=70,P5=75) and Arrival Time =0 as shown in the Table 1.Table 2 shows the output using RR algorithm and MARR algorithm. Figure 1 and Figure 2 shows Gantt chart of both RR and MPRR algorithm respectively.

Table1: Process with Burst Time

Process	Arrival Time	Burst Time
P1	0	25
P2	0	35
P3	0	55
P4	0	70
P5	0	75

Table 2: Comparison between RR algorithm and our new proposed MPRR algorithm (CASE 1).

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	185	133	14
MPRR	60,13,2	141	89	6

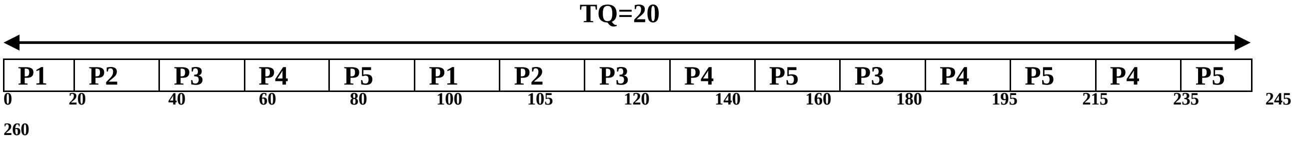

Fig.1: Gantt chart of RR from Table 1 of CASE 1.

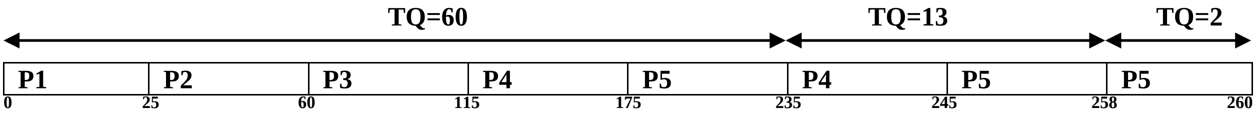

Fig.2: Gantt chart of MPRR from Table 1 of CASE 1.

CASE 2:-Let's consider five processes with Burst time (P1=41, P2=42, P3=43,P4=44,P5=55) and Arrival Time =0 as shown in the Table 3. Table 4 shows the output using RR algorithm and MPRR algorithm. Figure 3 and Figure 4 shows Gantt chart of both RR and MPRR algorithm respectively.

Table3: Process with Burst Time

Process	Arrival Time	Burst Time
P1	0	41
P2	0	42
P3	0	43
P4	0	44
P5	0	45

Table 4: Comparison between RR algorithm and our new proposed MPRR algorithm (CASE 2).

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	207	164	14
MPRR	44,1	127	105	4

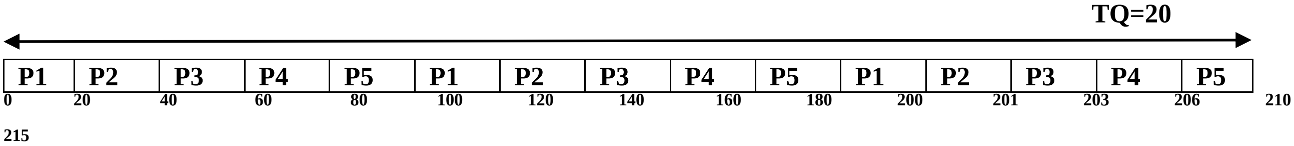

Fig.3: Gantt chart of RR from Table 3 of CASE 2

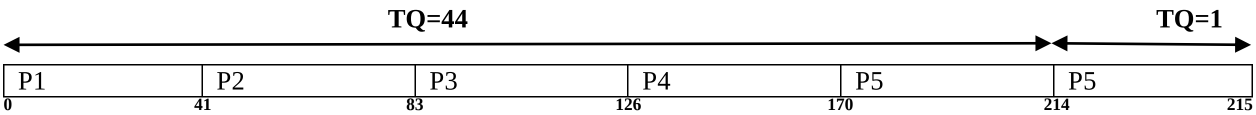

Fig.4: Gantt chart of MPRR from Table 3 of CASE 2.

CASE 3:-Let's consider five processes with Burst time (P1=9, P2=10, P3=36, P4=72, P5=144) and Arrival Time =0 as shown in the Table 5. Table 6 shows the output using RR algorithm and MPRR algorithm. Figure 5 and Figure 6 shows Gantt chart of both RR and MPRR algorithm respectively.

Table5:Process with Burst Time

Process	Arrival Time	Burst Time
P1	0	9
P2	0	10
P3	0	36
P4	0	72
P5	0	144

Table 6: Comparison between RR algorithm and our new proposed MPRR algorithm (CASE 3).

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	185	133	14
MPRR	60,13,2	141	89	6

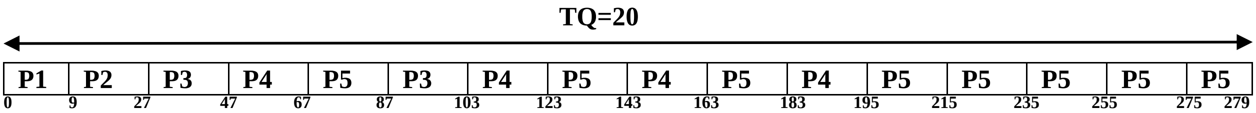

Fig.5: Gantt chart of RR from Table 5 of CASE 3

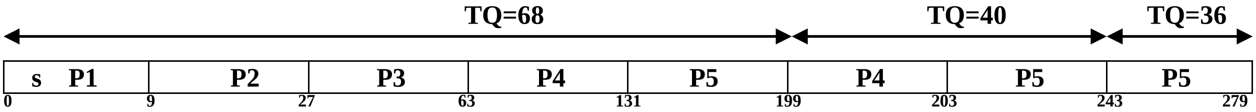

Fig.6: Gantt chart of MPRR from Table 5 of CASE 3

CASE 4:-Let's consider five processes with Burst time (P1=25, P2=30, P3=45, P4=55, P5=75) and Arrival Time as (P1=0, P2=10, P3=15, P4=25, P5=30) shown in the Table 7. Table 8 shows the output using RR algorithm and MPRR algorithm. Figure 7 and Figure 8 shows Gantt chart of both RR and MPRR algorithm respectively.

Table7:Process with Burst Time

Process	Arrival Time	Burst Time
P1	0	25
P2	10	30
P3	15	45
P4	25	55
P5	30	75

Table 8: Comparison between RR algorithm and our new proposed MPRR algorithm (CASE 4).

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	132	103	12
MPRR	52,13,10	102.4	61	6

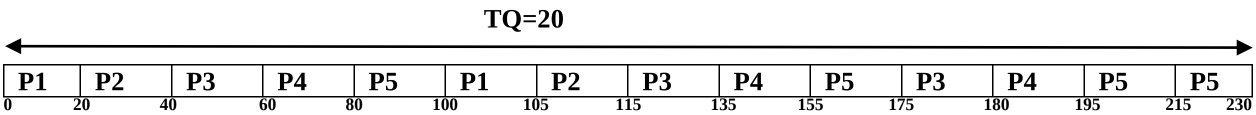

Fig.7: Gantt chart of RR from Table 7 of CASE4

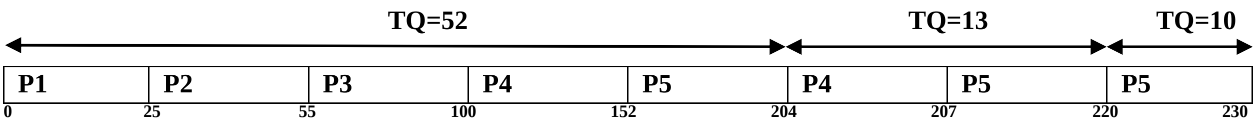

Fig.8: Gantt chart of MPRR from Table 8 of CASE 4

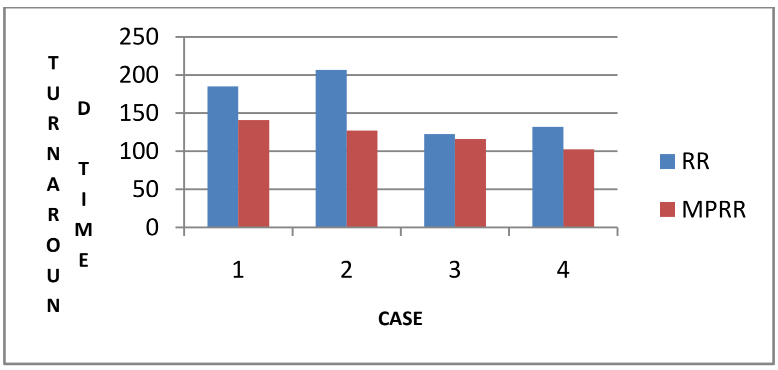

Fig.9: Comparison of average turnaround time of RR and MPRR taking arrival time into consideration.

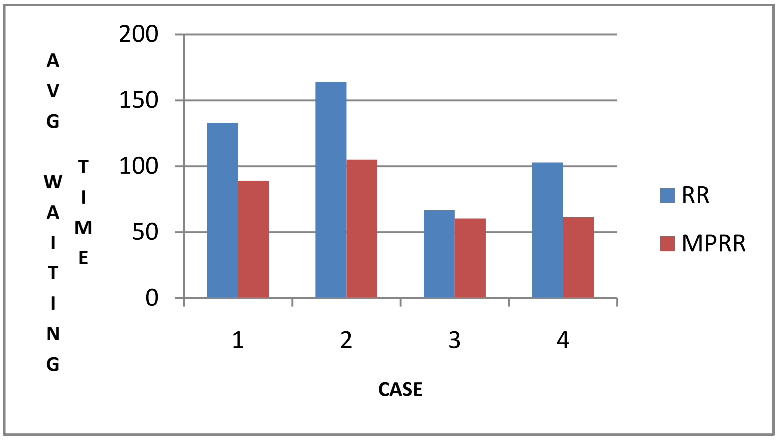

Fig.10. Comparison of average waiting time of RR and MPRR taking arrival time into consideration.

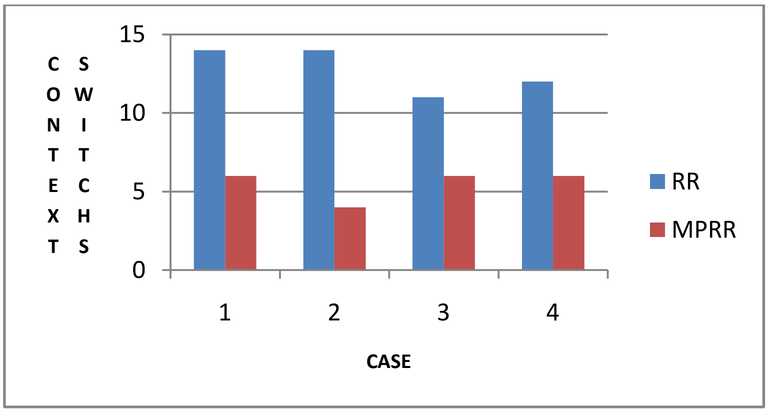

Fig.11. Comparison of Context Switches of RR and MPRR taking arrival time into consideration.

VIII. CONCLUSION

From the above study we conclude that the proposed Maximum Performance Round Robin Scheduling algorithm shows better performance as compared to Round Robin Scheduling algorithm by decreasing the Total Turnaround Time, Average Waiting Time and Number of context switching .This is achieved by increasing the Time Quantum Dynamically.

REFERENCE

[1] "Silberschatz, A., P.B. Galvin and G. Gagne, 2008"Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA, ISBN: 13: 978-0471694663 , pp: 944.

[2] Pallab banerjee, probal banerjee, shweta sonali dhal, " Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum ",IJITEE, ISSN: 2278-3075, Volume-1, Issue-3, August 2012.

[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 Dispersion Measure" 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 for round 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 multi-dynamic time quantum round robin (mdtqrr) algorithm with arrival time", ISSN : 0976-5166, Vol. 2, No. 2,Apr-May2011,pp.262-271.

