Volume 5, Issue 5, May 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Comparative Performance Analysis of Modified Even Odd Round Robin Scheduling Algorithm (MEORR) with Round Robin Scheduling Algorithm using Static Time Quantum in Real Time System

Pallab Banerjee[1] ,Sarita Kumari[2],Umara Khatoon[3]

Assistant Professor[1] , B.Tech Scholar[2,3]

[1, 2, 3] Department of Computer Science and Engineering

Cambridge Institute of Technology, Ranchi, Jharkhand, 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. Round Robin is a widely used scheduling algorithm but it has certain limitation because of static time quantum. Time quantum must be large so that context switching becomes reduces and it also effect the response time .So, in this paper we proposed a new algorithm called Modified Even Odd Round Robin (MEORR) in which we focused on dynamic time quantum which gives good result as a very less context switching as well as average waiting time and average turnaround time. It also reduces the overhead of the CPU by adjusting the time quantum according to the highest burst time of the processes in the ready queue.

Keywords-Operating System, Round Robin, Best Performance Round Robin, Turnaround time, Waiting time, Context Switch.

I. INTRODUCTION

An operating system is a system software which makes interface between user and computer hardware so that system perform in an efficient manner. Scheduling is the process of arranging, controlling and optimizing work and workload in a production process. Operating system follows a predefined procedure for selecting processes from memory and allocates resources as per their requirement. 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.CPU scheduling algorithm decides which of the process in the ready queue are to be allocated to the CPU.

II. 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 three 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 The number of context Switch should be less.

III. PROPOSED APPROACH:

The burst time of the processes is taken as unsorted sequence so that it will give better waiting time and turnaround time. In round robin algorithm the performance is based upon the size of static time quantum(TQ). If the Time Quantum is too small the then there will be many context switching between the processes. So our approach solved this problem by taking dynamic time quantum TQ.

TQ_1=Time quantum of even location processes in a Ready Queue.
TQ_2 =Time quqntum of odd location processes in a Ready Queue.
TQ=Largest among TQ_1 and TQ_2.

IV. PROPOSED ALGORITHM

1. Initialize CS=0, AWT=0, ATT=0.
   //ATT=Average Turnaround time.
   //AWT=Average waiting time.
   //CS=Number of context switch.


```
2. while(RQ != NULL)
   {
    // N= Number of Processes in the ready queue.
    // BT= Burst Time of the Processes.
    // RQ= Ready Queue.
    //TQ=Time Quantum.
    //TQ1= Average of even location processes BT in the Ready Queue.
    //TQ2= Average of odd location processes BT in the Ready Queue.

      a.)  //Code for finding Time Quantum.
           Initialize i=0,S1=0,c1=0;
           while(i< N loop)
           {
              S1=S1+P[i];
              i=i+2;
              c1++;
           } //End of while.
            TQ1=S1/c1;
           Initialize j=1,S2=0,c2=0;
           while(j<N loop)
           {
              S2=S2+P[j];
              j=j+2;
              c2++;
           }//End of while.
            TQ2=S2/c2;
           If(TQ1>=TQ2)
           TQ=TQ1;
           Else
           TQ=TQ2;

      b.)  If two processes are there then the TQ is equal to the largest between these two BT.
      c.)  If one process is there then after calculation TQ is equal to BT itself.
      d.)   for( i=1 to N loop)
            // Assign TQ to (1 to N)processes.
           {
             Pi->TQ
            //Assign TQ to all the available processes.
             }
           //End of for.
  } //End of while.
3. 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 4.
4. Calculate ATT,AWT,CS
5. End
```

V. EXPERIMENTAL ANALYSIS

CASE 1:-

(With Zero Arrival Time) We consider five processes with Burst time (P1=7, P2=20, P3=13 ,P4=27,P5=25) and Arrival Time =0 as shown in the Table 1,Table 2 shows the output using RR and MEORR algorithm. Figure 1 and 2 shows Gantt chart of RR and MEORR algorithm respectively.

Table 1:Process with Arrival Time and Burst Time

Process	Arrival Time	Burst Time
P1	0	7
P2	0	20
P3	0	13
P4	0	27
P5	0	25

Table 2: Comparison between RR algorithm and our new proposed MEORR algorithm(Case 1).

Algorithm	Time quantum	Turnaround time	Waiting time	Context switch

RR	10	60.6	42.2	11
MEORR	24,3	51.4	33	7

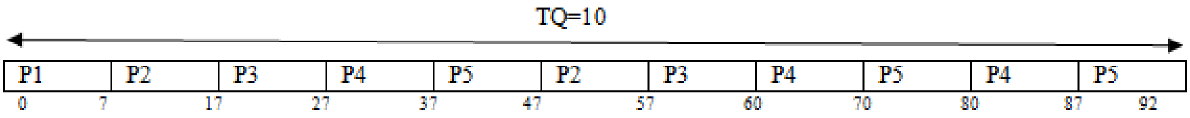

Fig1: Gantt chart of RR form Table1 (CASE 1)

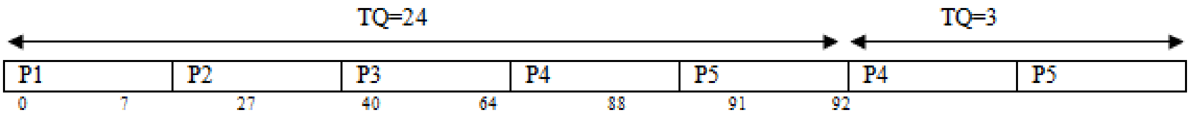

Fig1 : Gantt chart of MEORR form Table1(CASE 1)

CASE 2:-

(With Zero Arrival Time) We consider five processes with Burst time (P1=30, P2=5, P3=15 ,P4=18,P5=25) and Arrival Time =0 as shown in the Table 3. Table 4 shows the output using RR and MEORR algorithm. Figure 3and 4 shows Gantt chart of RR and MEORR algorithm respectively.

Table 3:Process with Arrival Time and Burst Time

Process	Arrival Time	Burst Time
P1	0	30
P2	0	5
P3	0	15
P4	0	18
P5	0	25

Table 4: Comparison between RR algorithm and our new proposed MEORR algorithm(Case 2).

Algorithm	Time quantum	Turnaround time	Waiting time	Context switch
RR	10	64.8	46.2	11
MEORR	23,7	63.2	44.6	7

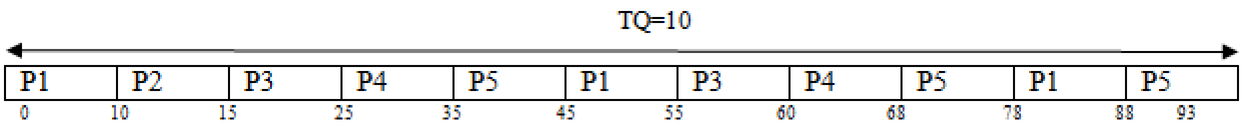

Fig3: Gantt chart of RR form Table3 (CASE 2)

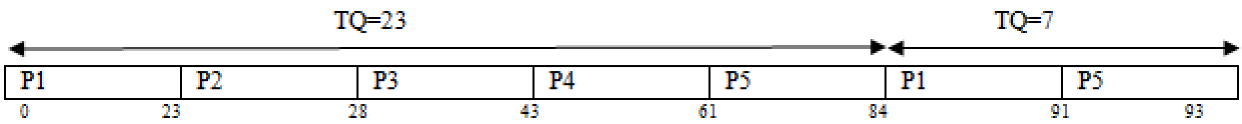

Fig4: Gantt chart of MEORR form Table3 (CASE 2)

CASE 3:-

(With Zero Arrival Time) We consider five processes with Burst time (P1=55, P2=53, P3=51 ,P4=54,P5=52) and Arrival Time =0 as shown in the Table 5. Table 6 shows the output using RR and MEORR algorithm. Figure 5 and 6 shows Gantt chart of RR and MEORR algorithm respectively.

Table 5:Process with Arrival Time and Burst Time

Process	Arrival Time	Burst Time
P1	0	55
P2	0	53
P3	0	51
P4	0	54
P5	0	52

Table 6: Comparison between RR algorithm and our new proposed MEORR algorithm(Case 3).

Algorithm	Time quantum	Turnaround time	Waiting time	Context switch
RR	10	260	207	30
MEORR	54,1	201.2	148.2	6

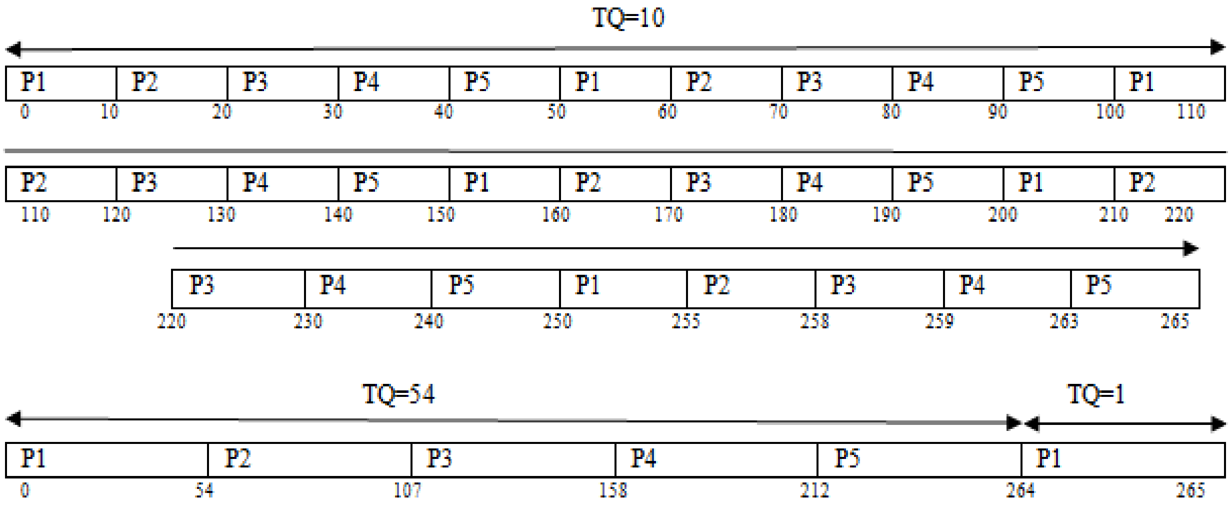

Fig6: Gantt chart of MEORR form Table5 (CASE 3)

CASE 4:-

(With Arrival Time) We consider five processes with Burst time (P1=15, P2=7, P3=13 ,P4=16,P5=18) and Arrival Time (P1=0,P2=5,P3=7,P4=15,P5=20) as shown in the Table 7. Table 8 shows the output using RR and MEORR algorithm. Figure 7and 8 shows Gantt chart of RR and MEORR algorithm respectively

Table 7:Process with Arrival Time and Burst Time

Process	Arrival Time	Burst Time
P1	0	15
P2	5	7
P3	7	13
P4	15	16
P5	20	18

Table 8: Comparison between RR algorithm and our new proposed MEORR algorithm(Case 4)

Algorithm	Time quantum	Turnaround time	Waiting time	Context switch
RR	10	41.4	27.6	9
MEORR	15,3	32	18.2	7

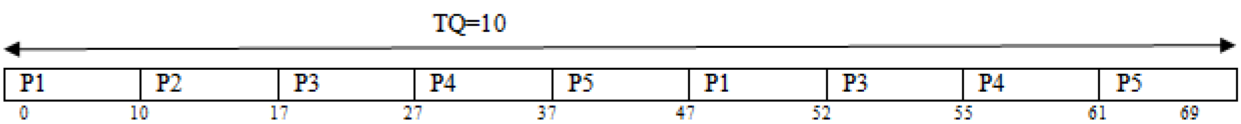

Fig7: Gantt chart of RR form Table7 (CASE 4)

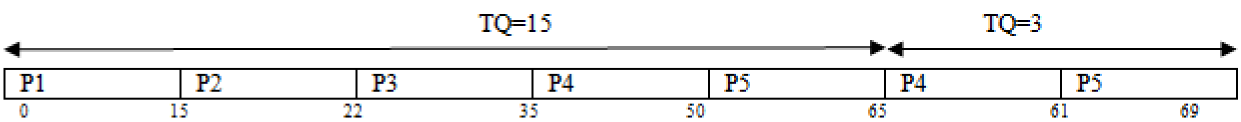

Fig8: Gantt chart of MEORR form Table7 (CASE 4)

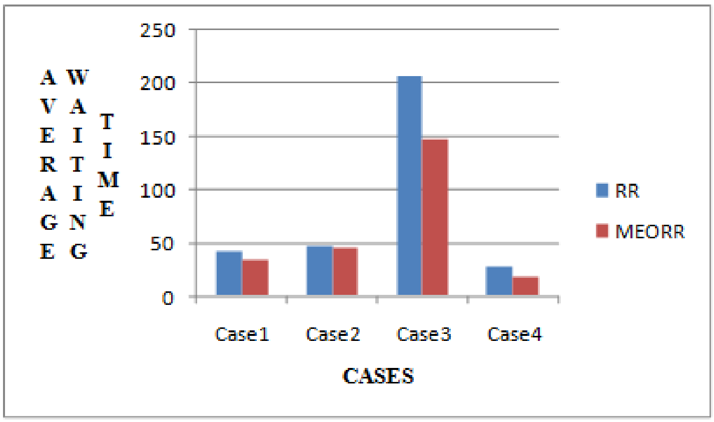

Fig.9: Comparison of average Waiting Time of RR and MEORR taking arrival time into consideration.

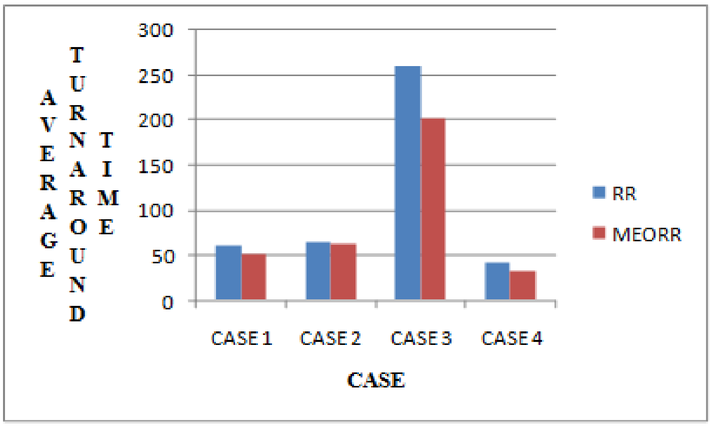

Fig.10: Comparison of average Turnaround Time of RR and MEORR taking arrival time into consideration.

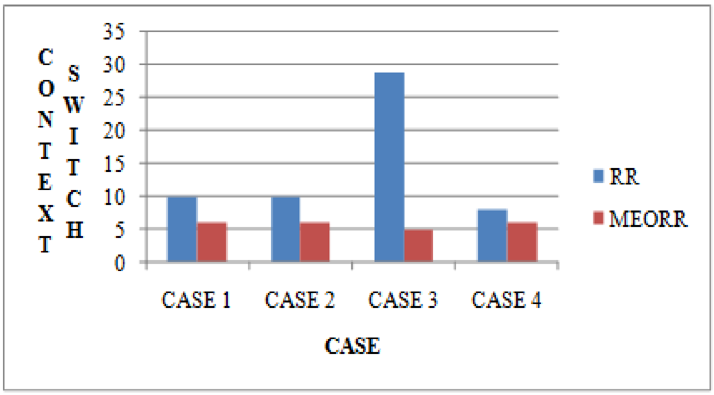

Fig.11: Comparison of Context Switch of RR and MEORR taking arrival time into consideration.

VI. CONCLUSION

CPU is one of the most important component of the computer resources.CPU scheduling involves careful examination of waiting processes to determine the most efficient way to service the request. In this paper an optimized performance round robin scheduling algorithm is proposed. This proposed MEORR CPU scheduling algorithm always gives better performance than RR. This is achieved by increasing Time Quantum Dynamically with decreasing the total Turnaround Time, Average Waiting Time and Number of Context Switching.

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.

[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

