International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-2 Issue-4, March 2013

Dynamic Time Quantum in Round Robin Algorithm (DTQRR) Depending on Burst and Arrival Time of the Processes

Dibyendu Barman

***Abstract:- In multitasking and time sharing operating system the performance of the CPU depends on waiting time, response time, turnaround time, context switches from the process mainly depends on the scheduling algorithm. Round Robin is most widely used scheduling algorithm but this algorithm has some disadvantages. Here Time Quantum play very impartment role. If the time quantum is too large then it works like FCFS (First cum First Serve) scheduling algorithm and if time quantum is too small then more context switches is occur which decrease the performance of the CPU. In this paper based on the experiments and calculations a new scheduling algorithm is introduced. In this algorithm the main idea is to adjust time quantum dynamically depending upon arrival time and burst time of* the processes**

Keywords— Round Robin, Context Switch, DTQRR, CPU Scheduling.

I. INTRODUCTION

Operating System is a platform where user can execute program in well and efficient manner. It is an interface between user and computer hardware. Modern operating system is more complex with the features of multiprocessing and multitasking. In this type of operating system more than one process in the job pool or ready queue waiting its turn to be assigned to the CPU. Process is allocated to CPU requires careful awareness that starvation not occur. Scheduling algorithm are mechanism to allocate resources or processes to CPU and execute different way to decrease turnaround time, waiting time, response time and number of context switches. [1]

II. PRELIMINARIES

Control Block (PCB) contain different information of process such as process number, process state, register, list of open files, CPU scheduling information. When process enters into the main memory is kept in the Ready Queue. The processes which are waiting for I/O request are kept in Device Queue. The Long term scheduler or job scheduler select process from job pool and load them into main memory for execution. Short term scheduler or CPU scheduler select from among the processes that are ready to execute and allocates the CPU to one of them. Medium term scheduler is used in time sharing system. [2]

Manuscript published on 30 March 2013.
*****Correspondence Author(s)
Prof. Dibyendu Barman, Assistant Professor, CSE Department Government College of Engineering and Textile Technology, Berhampore, Dist: Murshidabad, State: West Bengal, India.

There exists different CPU scheduling algorithm like FCFS (First Cum First Serve), Priority Scheduling, SJF (Shortest Job First), Round Robin (RR) scheduling with some advantages and disadvantages. FCFS is simplest scheduling algorithm where processes are allocated to CPU according to their arrival time. Its average waiting time quite long so it is not suitable for real time application. In priority Scheduling process is allocated to CPU according to their priority but here is a problem that high priority process may not allow low priority to access CPU which may cause starvation and solution to this problem is aging. Shortest Job First (SJF) provides shortest waiting time but it require precise knowledge of burst time of the process and some time it is not possible and unpredictable. Round Robin scheduling algorithm is simple and most widely used algorithm especially for time-sharing system. All process are kept in circular queue and CPU scheduler goes around the queue allocating the CPU to each process small unit of time call time slices or time quantum. New arriving processes are placed in tail of the queue. The Scheduler picks up first process of the queue to and sets a timer to interrupt after one time quantum and dispatches the process. If the process still running after completion of one Time Quantum the CPU pre-empted and place the process at tail of the queue and second process of the queue allocate to CPU with one time quantum and so on If the time quantum is too small then large number of context switching may occur and if time quantum is too large then response time will increase causes degrade the performance of CPU. [3]

III. PERFORMANCE CRITERIA

Now what is arrival time, burst time, waiting time, turnaround time, response time, Throughput, Response Time?

Arrival Time: Arrival time is the time at which process arrive at main memory.

Burst Time: Burst time is the time for which process holds the CPU.

Waiting Time: Waiting time is amount of time process waiting in ready queue.

Turnaround Time: Turnaround time means time of arrival minus time of completion.

Response Time: Response time means time of arrival minus first response by CPU.

Throughput: Number of process completed per unit time

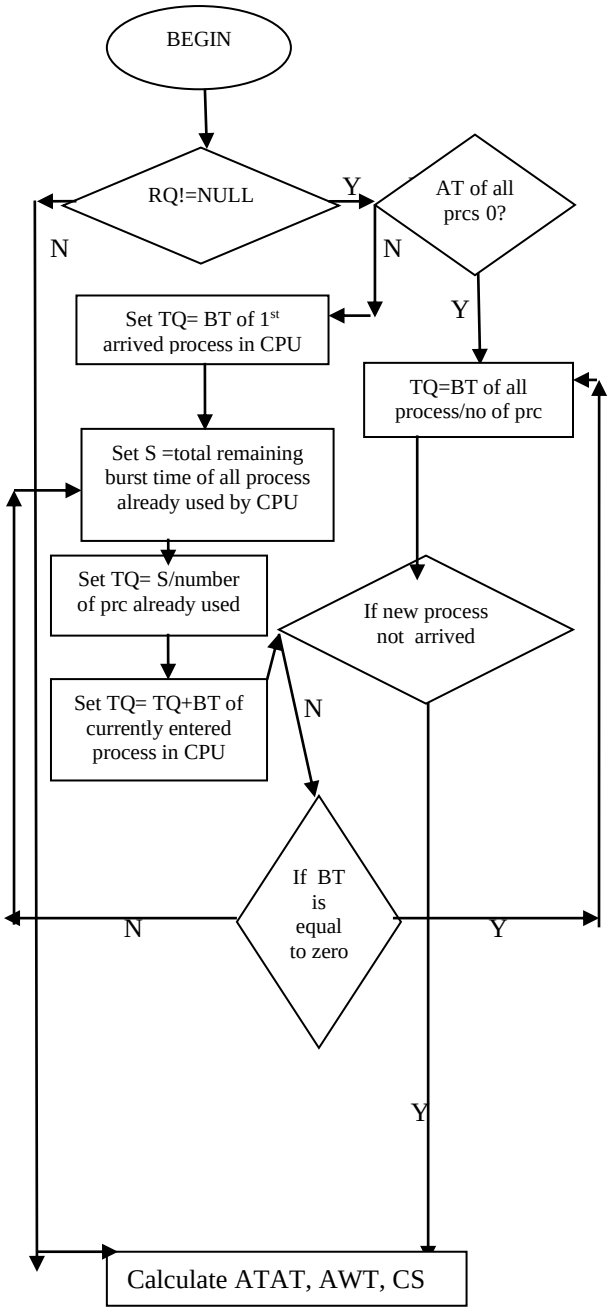
Response Time: Response time is the time from the submission of a request until first response is produced.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

IV. PROPOSED APPROACH

Generally Round Robin algorithm works with fixed Time Quantum (TQ) but the problem with too large or too small Time Quantum already discussed. Here I try to solve the problem by taking Dynamic Time Quantum. Here Time Quantum will automatically generate depending upon the Arrival time & Burst Time of the process.

V. FLOW CHART OF PROPOSED APPROACH



AT: Arrival Time, BT: Burst Time
Prc: Process TQ: Time Quantum
RQ: Ready Queue

Fig1: Flow Chart of DTQRR algorithm

VI. PROPOSED ALGORITHM

A. *If Arrival Time Of All the Processes Are Zero*

If arrival time of all the process is 0 the Time Quantum is set to Average of burst time of all the process i.e.

//TQ is the Time quantum initially set to 0

//Arrival Time[i] is the Burst time of ith process

//N is the number of process

//flag is an integer variable initially set to 0

Pseudo Code 1:

```
While (Ready_Queue! =NULL)
 do
 For i=1 to N
 do
If Arrival_Time[i] ==0 then
Set flag=1;
 else
 Set flag=0
 End If
 done
 If  flag==1 then
For i=1 to  N
 do
Set TQ=TQ+Burst_Time[i];
done
Else
SECONDCASE
End If
 done
```

B. *If Arrival Time Of All the Processes Are Not Zero (Secondcase)*

If arrival time of all the processes is not zero then Time Quantum (TQ) is not fixed throughout the algorithm. It change dynamically depending upon the Arrival Time and Burst Time of the processes i.e. SECONDCASE

//RQ is an integer variable count size of Ready Queue

//Ready_Queue is an integer variable count size of Ready Queue

//NOPA is an integer variable counts number of process already accessed by CPU

// NOPNC is an integer variable counts no of process already accessed by CPU but not completed

// Used_Time[k] amount of time process runs in CPU

//sizeof (X) calculate the size of X

Pseudo Code 2:

```
RQ=sizeof(Ready Queue);
While (Ready_Queue!= NULL)
do
Ready_Queue=Ready_Queue-1;
NOPA=RQ-Ready_Queue;
NOPNC=NOPA;
If (NOPA==1) then
Set TQ=Burst_Time first-arrived
// TQ is set to burst time of 1st
```


Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

// arrived process
End If
If (NOPA>1) then //If number of process accessed by CPU
//more than 1
For k=1 to NOPA
do
If ((Burst_Time[k]-Used_Time[k])==0) then
NOPNC--; //count no of process not fully completed
End If
done
For k=1 to NOPNC
do
TQ=(Burst_Time[k]-Used_Time[k]);
done
TQ=(TQ+Burst_Time currently-arrived) //NOPNC Calculate Time
//Quantum
RQ=sizeof (Ready Queue); //Size of Ready Queue
Calculated
//again
done

VII. WHY WE USE THE ALGORITHM

In last few years many algorithms are used to increase the performance of Round Robin Scheduling (RR) but in those algorithms most of the cases processes have to stored in the ready queue in ascending order of the arrival time but if number of process is high then extra time require to sort those large number of process this may decrease the performance of the CPU. Another case is that a process may come any time during running condition of the CPU i.e. when a process arrive after CPU start execution to sort the newly arrived process sorting algorithm have to run entire period of CPU execution. This may decrease the performance of CPU. In some algorithm processes are sorted depending upon Burst time but Burst time may vary or change at the time of execution of the processes. But In this DTQRR algorithm process need not have to sort depending on either arrival time nor in Burst time of the processes

VIII. EXPERIMENTAL ANALYSIS

To evaluate the performance of the DTQRR algorithm asset of different cases have taken. Here only 5 processes have taken but the algorithm works effectively & efficiently if large numbers of processes are used.

CASE1:- Let's take five process with Burst Time (P1=13, P2=12, P3=10, P4=20, P5=15) with Time Quantum (TQ) =8 with Arrival Time=0 as shown in the Table1.Table 2 shows the output using RR algorithm and DTQRR algorithm. Figure 2 and Figure 3 shows Gantt chart for both simple RR and Proposed algorithm respectively

Table 1.Process with Arrival time and Burst Time

Processes	Arrival Time	Burst Time
P1	0	13
P2	0	12
P3	0	10
P4	0	20
P5	0	15

Table 2. Comparison of simple RR and DTQRR algorithm (CASE1)

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	8	56.2	42.2	10
DTQRR	14	42.4	28.4	6

P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P4
0	8	16	24	32	40	45	49	51	59	66	70

Fig 2: Gantt chart of RR from Table 1(CASE1)

P1	P2	P3	P4	P5	P4	P5
0	13	25	35	49	63	69

70

Fig 3: Gantt chart of DTQRR from Table 1(CASE1)

CASE2:- Let's take five process with Burst Time (P1=14, P2=34, P3=45, P4=62, P5=77) with Time Quantum (TQ) =25 with Arrival Time=0 as shown in the Table3.Table42 shows the output using RR algorithm and DTQRR algorithm. Figure 4 and Figure 5 shows Gantt chart for both simple RR and Proposed algorithm respectively

Table 3.Process with Arrival time and Burst Time

Processes	Arrival Time	Burst Time
P1	0	14
P2	0	34
P3	0	45
P4	0	62
P5	0	77

Table 4. Comparison of simple RR and DTQRR Algorithm (CASE2)

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	25	143.4	97	11
DTQRR	46	117.6	71.2	6

P1	P2	P3	P4	P5	P2
0	14	39	64	89	114	123

P3	P4	P5	P4	P5	P5
143	168	193	205	230	232

Fig 4: Gantt chart of RR from Table 3(CASE2)

P1	P2	P3	P4	P5	P4	P5
0	14	48	93	139	185	201	232

Fig 5: Gantt chart of DTQRR from Table 3(CASE2)

CASE3:- Let's take five process with Burst Time (P1=20, P2=30, P3=50, P4=60, P5=70) with Time Quantum (TQ) =20 with Arrival Time (P1=0, P2=5, P3=20, P4=25, P5=30) as shown in the Table3.Table42 shows the output using RR algorithm and DTQRR algorithm. Figure 6 and Figure 7 shows Gantt chart for both simple RR and Proposed algorithm respectively

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table 5.Process with Arrival time and Burst Time

Processes	Arrival Time	Burst Time
P1	0	20
P2	5	30
P3	20	50
P4	25	60
P5	30	70

Table 6. Comparison of simple RR and DTQRR algorithm (CASE3)

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	132	86	11
DTQRR	20,22,40,38,50	106	52	6

P1	P2	P3	P4	P5	P2	P3
0	20	40	60	80	100	110	130

P4	P5	P3	P4	P5	P5
150	170	180	200	220	230

Fig 6: Gantt chart of RR from Table 5(CASE3)

P1	P2	P3	P4	P5	P4	P5
0	20	50	100	150	200	210	230

Fig 7: Gantt chart of DTQRR from Table 5(CASE3)

CASE4:- Let's take five process with Burst Time (P1=25, P2=40, P3=55, P4=80, P5=100) with Time Quantum (TQ) =20 with Arrival Time (P1=0, P2=15, P3=30, P4=40, P5=50) as shown in the Table3.Table42 shows the output using RR algorithm and DTQRR algorithm. Figure 8 and Figure 9 shows Gantt chart for both simple RR and Proposed algorithm respectively

Table 7.Process with Arrival time and Burst Time

Processes	Arrival Time	Burst Time
P1	0	25
P2	15	40
P3	30	55
P4	40	80
P5	50	100

Table 8. Comparison of simple RR and DTQRR algorithm (CASE4)

Algorithm	Time Quantum	Turnaround Time	Average Waiting Time	Context Switch
RR	20	171	111	14
DTQRR	25,50,45,62	113	55	4

P1	P2	P3	P4	P5	P1	P2	P3
0	20	40	60	80	100	105	125	145

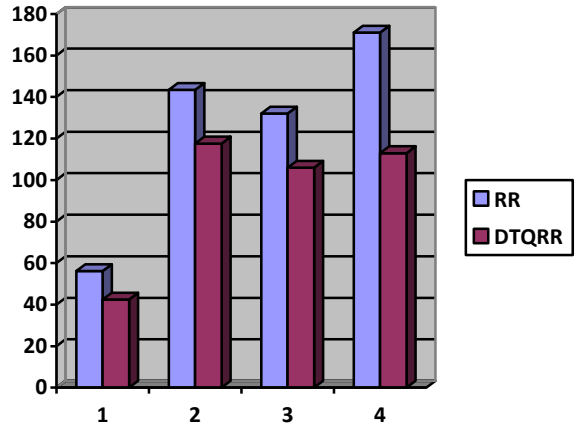
P4	P5	P3	P4	P5	P4	P5	P5
165	185	200	220	240	260	280	300

Fig 8: Gantt chart of RR from Table 7(CASE4)

P1	P2	P3	P4	P5
0	25	65	120	200	300

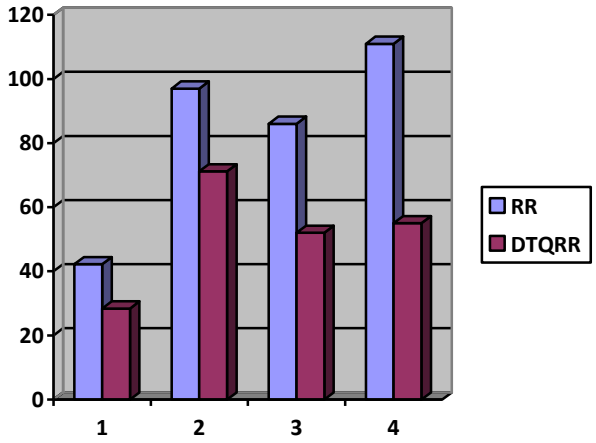
Fig 9: Gantt chart of DTQRR from Table 7(CASE4)

IX. PERFORMANCE CHART



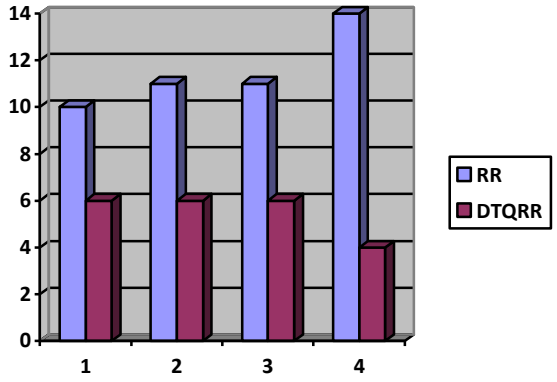
CASE---→

Fig10: Comparison of Average Turnaround Time



CASE---→

Fig11: Comparison of Average Waiting Time



CASE---→

Fig12: Comparison of Number of Context Switches

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

X. CONCLUSION

From the above experimental analysis it's conclude that DTQRR algorithm give better performance than simple Round Robin Algorithm in respect of Average Turnaround Time, Average Waiting Time and number of Context switches by changing Time Quantum Dynamically. For the future perspective the research should be useful with knowing of arrival time, burst time.

REFERENCES

1. "Silberschatz, A., P.B. Galvin and G. Gagne, 2008" Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA., ISBN: 13: 9780471694663, pp.944.
2. "Tanenbaun, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13: 9780136006633, pp. 1104.
3. 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.
4. 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), 2009, pp.1831-1837.
5. Stallings, W.: Operating Systems Internals and Design Principles, 5th edition, Prentice Hall, (2004).
6. Kaladevi, M.C.A.,M.Phil., and Dr.S.Sathiyabama, M.Sc.,M.Phil.,Ph.D, "A Comparative Study of Scheduling Algorithms for Real Time Task", International Journal of Advances in Science and Technology, Vol. 1, No. 4, 2010.
7. Houssine Chetto and Maryline Chetto (1989) " Some Results of Earliest Deadline Scheduling Algorithm" journal on IEEE TRANSACTION ON SOFTWARE ENGINEERING. VOL 15. NO.10, pg 1261-1269.
8. 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",JARCSSE,ISSN:2277-128X,Volume-2,Issue-8,August-2012.
9. S. M. Mostafa, S. Z. Rida and S. H. Hamad, "Finding Time Quantum of Round Robin CPU Scheduling Algorithm in General Computing Systems using Integer Programming", IJRRAS 5 (1), October 2010, pp.64-71.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)