Indian Journal of Science and Technology, *Vol 9(8), DOI: 10.17485/ijst/2016/v9i8/76368, February 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time

Anju Muraleedharan[1]*, Neenu Antony[2] and R. Nandakumar[3]

[1,2]Department of Computer Science & IT, Amrita School of Arts and Sciences Kochi,India;
[3]Department of Computer Science & IT, Sciences Faculty, Amrita School of Arts and Sciences Kochi,India;
anju.murali36@gmail.com, neenuna92@gmail.com , nandacumar@gmail.com

Abstract

Background: If the processer knows in advance how much time each process takes then the best approach is shortest job first scheduling (SJFS). But knowing the burst time before hand is usual unrealistic. Round Robin scheduling algorithm (RR) is the most commonly used scheduling algorithm in an environment with time sharing among more than one process. RR algorithm is free from starvation, since all processes always get equal time quantum. But the selection of time quantum can influence the performance of the algorithm. If too short, an inordinate fraction of the time is spent in context switches. If too large, it behaves like first come first served (FCFS). We try to improve this aspect of the algorithm. **Methods:**We propose a variant of RR algorithm -*Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time* which could be used if the burst times of the processes are unknown at the beginning. This involves tuning the time quantum at run time. **Findings:**With the proposed method we find that 15% reduction in average waiting time,15% in average turnaround timeand number of context switches can be achieved10%.

Keywords: *CPU, Scheduling criteria, Scheduling algorithm, Round Robin scheduling, Context switch, Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time.*

1. Introduction

In multiprogramming systems, CPU scheduling has to define which job will be activate while many jobs running at the same time. It has a huge influence on resource usage and system functioning Process manager is responsible for process scheduling that controls the transfer of presently executing process from the CPU and choose the next one for execution. Different scheduling algorithms occur and have various benefits and drawbacks. FCFS is very simple in nature in which Jobs are executed on first come, first serve basis.In the case of non preemptive allocation, once a job starts its execution then it will not allow to leave the CPU until its completion. Its limitation is that the waiting time will get larger if small processes are made to wait for the completion of the large process – this can cause poor performance as average wait time is high. In priority scheduling the process will work according to the greatest priority first basis. Selecting the priority for each process is a major concern in this method and starvation is the main problem. Coming toshortest job first scheduling(SJFS), it provides lowest average waiting time. The drawback is the processer needs to know how much time each process will take in advance and the large process will not get chance to work if there are many small processes continuously join.

We focus on RR schedulingalgorithm which is very popular. To overcome the disadvantage of SJFS we can use the Round Robin scheduling algorithm. The RR algorithm is free from starvation, as all jobs will always get equal time quantum. The selection of the time slice is demanding, If it is too short, most part of the time is utilized for context switches. If too large, it behaves like FCFS. This is done by the CPU itself without considering any priority.

*Author for correspondence

1.1 Scheduling Criteria

The properties of the scheduling algorithms are very important. The selection of the algorithm is done on the basis of the following properties.

CPU utilization: It refers to the amount of work handled by the CPU, inorder to keep the CPU occupied with an activity always.

Throughput: Number of tasks finished according to a particular amount of time.

Turnaround time: It means the time interval between starting time and accomplishment time of a process.

Waiting time: The time for which the process stay in the ready queue for its opportunity.

Response time: We can't opt the best scheduling algorithm only by considering the turnaround time. Response time is the first reply time of a process after its submission.

Context Switch: It is the shifting of the CPU from one activity to another. It is pure overhead and adds to the waiting time of all currently active processes.

The goal of scheduling is to diminish most criteria mentioned above.

Many researchers have proposed different methods to improve the performance of RR. For example, Ajith et al[1] suggest a method,it performs by allocating the CPU to every process. Initially they set a specific time slice and when the first cycle completes, the initial time slice is doubled;selects shortest process from the waiting queue and assign the CPU to it and after that, it chooses another process that has the next smallest CPU burst from the remaining set for execution. After completing that cycle, if any process remains after doubling the time slice, it will half the currently used time slice and apply it to remaining jobs.

Saroj Hiranwal et al[2] proposed that, In the first step allocate the jobs to processer in the increasing orderof their burst time. The next idea is to calculate a smart time slice (STS). When the number of process is odd, STS is the mid one of all burst times. If number of process is even then it is the average CPU burst of all jobs.

Remark: Some other variants on RR that have been proposed are shown in the reference.

2. Proposed Algorithm

In reality, it is often the case that the burst timesof the processes are unknown, i.e. we don't know upfront how much time the process will execute.Our new approach focuses on how round robin will perform if the processes burst times are unknown at the beginning. We propose a refinementto simple RR by altering the time quantum while execution. First and foremost, put a small value to initial time quantum ,and carry through the first cycle with this time quantum.In succeeding cycles,we multiply the time quantum by two if no processes finished its work. We will scrutinize the number of processes completed in each cycle. If at least one of them is completed, then continue the next iteration with an unchanged time quantum. By this method, some of the problems with static time quantum are partially solved.

If the time slice used is larger, then the average waiting time will reduce, in normal cases. But a factor that affects the average waiting time of the RR is the arrival time of the jobs – if several new processes arrived in during the execution, then it may cause an increase in the average waiting time.

```
//TQ -> Time Quantum
//n -> Total number of processes.
//finish -> Counter for number of process
 finished = 0

1. Assign all processes into ready queue when it is empty

2. While (ready queue != NULL)
     Set TQ with an initial value

3. Allocate the CPU to each process according to the FCFS
   manner
   for i= 0 to n loop
   {
        If (process finished == TRUE)
             finish ++
             end of if
   }
   end of for

4. If (finish < 1)
        TQ = TQ * 2
   else if(finish > 2)
        TQ = TQ/2
   else
        Continue with the initial TQ
   end of if

5. If new process arrived, go to step 1
   else go to step 2
   end of if
   end of while
6. Calculate average waiting time, average turnaround time
   and number of context switches
7. End
```

Figure 1. Proposed algorithm

Context switching is an essential parameter in the case of scheduling algorithms. It is pure overhead and adds to the waiting time of all currently active processes. In RR scheduling, the time slice should be chosen as considerably larger than the context switch time. In our approach, we tentatively initialize the time quantum as 10 times the context switch time (the latter is taken as 1 unit). The calculation of the average waiting time includes the overheads from the context switches.

The proposed algorithm will be executed as follows:

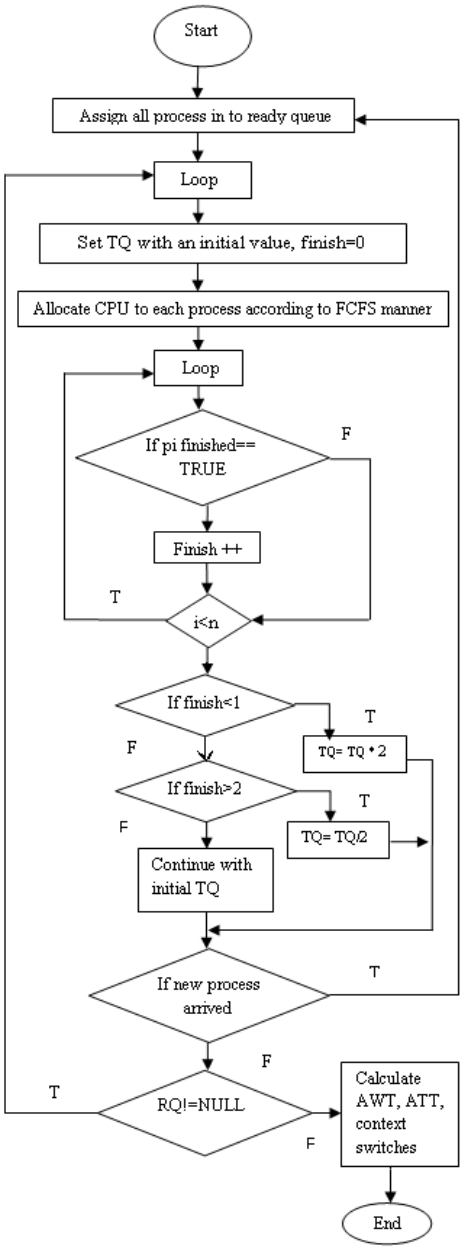

Figure 2. Flow chart for the proposed algorithm

3. Experimental Analysis

For the performance evaluation, our assumptions are,

The arrival time of all the jobs has been known before allocating to the processor. The burst times of the processes are unknown at the beginning. The overhead incurred due to the context switch is added to the waiting time of all currently active processes. The context switching time is equal to one unit.

Experimentally, we compare the scheduling of a set of input processes using basic RR,algorithm[1] and proposed one. We takes the process burst times only for illustration. According to the measures, like, average waiting time, average turnaround time and number of context switches, we calculated and compared the results.

Case 1: Here we consider a case of 5 processes with arrival time zero.

As per Round Robin: Time quantum = 10

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr2	Pr3	Pr4
0	**11**	**22**	**33**	**44**	**55**	**60**	**71**	**82**	**93**

Pr5	Pr 2	Pr3	Pr4	Pr5	Pr2	Pr3	Pr4	Pr5
104	**115**	**126**	**137**	**148**	**153**	**164**	**175**	**186**

Pr3	Pr4	Pr5	Pr4	Pr5	Pr4	Pr5
192	**203**	**214**	**225**	**236**	**239**	**256**

As per Optimized Round Robin Scheduling Algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr2	Pr3	Pr4
0	**11**	**22**	**33**	**44**	**55**	**60**	**81**	**102**

Pr5	Pr2	Pr3	Pr4	Pr5	Pr3	Pr4	Pr5	Pr4	Pr5
123	**144**	**149**	**160**	**171**	**182**	**188**	**209**	**230**	**233**	**250**

Table1. Example1

Process id	Burst Time	Arrival Time
Pr1	14	0
Pr2	34	0
Pr3	45	0
Pr4	62	0
Pr5	77	0

As per Dynamic Time Slice Round Robin Scheduling Algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr2	Pr3
0	**11**	**22**	**33**	**44**	**55**	**60**	**81**	**102**	**123**

Pr4	Pr5	Pr2	Pr3	Pr4	Pr5	Pr4	Pr5
144	**149**	**165**	**186**	**207**	**220**	**248**	

Case 2: Here we consider a case where some of the jobs arrive at times other than 0

As per Round Robin: Time quantum = 10

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr2	Pr3	Pr4
0	**11**	**22**	**33**	**44**	**55**	**66**	**77**	**88**	**99**	**110**

Pr5	Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr3	Pr4
121	**124**	**135**	**146**	**157**	**161**	**172**	**183**	

Pr5	Pr3	Pr4	Pr5	Pr4	Pr5	Pr4	Pr5
194	**203**	**214**	**225**	**236**	**247**	**258**	**272**

As per Optimized Round Robin Scheduling Algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr2	Pr1	Pr3	Pr4	Pr5
0	**11**	**22**	**33**	**44**	**55**	**68**	**89**	**110**	**131**	**152**

Pr1	Pr3	Pr4	Pr5	Pr3	Pr4	Pr5	Pr4	Pr5
156	**167**	**178**	**189**	**198**	**219**	**240**	**251**	**265**

As per Dynamic Time Slice Round Robin Scheduling algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr5	Pr1	Pr2	Pr3
0	**11**	**22**	**33**	**44**	**55**	**76**	**89**	**110**	**131**

Pr4	Pr5	Pr1	Pr3	Pr4	Pr5	Pr4	Pr5
152	**156**	**175**	**196**	**217**	**238**	**263**	

Case 3: Another example where the arrival times are not all zero.

As per Round Robin: Time quantum = 10

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr1	Pr2	Pr3	Pr4	Pr2	Pr3	Pr4
0	**11**	**22**	**33**	**44**	**50**	**61**	**72**	**83**	**94**	**105**

Pr2	Pr4	Pr2	Pr4	Pr2	Pr4	Pr2	Pr4	Pr2	Pr4
116	**127**	**138**	**149**	**160**	**171**	**182**	**193**	**204**	**212**	**227**

Table 2. Comparison of simple RR, Optimized RR and proposed RR

Algorithm	Time Quantum	Context Switches	Average WT	Average TAT
Simple RR	10	24	133	179
Optimized Round Robin	10, 20	18	128.8	175
Dynamic Time Slice Round Robin (proposed)	10, 20	15	121	167

Table 3. Example2

Process id	Burst Time	Arrival Time
Pr1	33	0
Pr2	22	2
Pr3	48	5
Pr4	70	7
Pr5	74	9

Table 4. Comparison of simple RR, Optimized RR and proposed RR

Algorithm	Time Quantum	Context Switches	Average WT	Average TAT
Simple RR	10	25	148	203
Optimized Round Robin	10, 20	18	132	186
Dynamic Time Slice Round Robin (proposed)	10, 20	15	129	183

Table 5. Example3

Process id	Burst Time	Arrival Time
Pr1	15	0
Pr2	77	4
Pr3	30	15
Pr4	85	20

As per Optimized Round Robin Scheduling Algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr1	Pr2	Pr3	Pr4
0	**11**	**22**	**33**	**44**	**50**	**71**	**92**	**113**	**124**

Pr2	Pr4	Pr2	Pr4	Pr2	Pr4	Pr2	Pr4
135	**156**	**177**	**188**	**199**	**207**	**222**	

As per Dynamic Time Slice Round Robin Scheduling algorithm:

Gantt chart

Pr1	Pr2	Pr3	Pr4	Pr1	Pr2	Pr3
0	**11**	**22**	**33**	**44**	**50**	**71**	**92**	**113**

Pr4	Pr2	Pr4	Pr2	Pr4	Pr2	Pr4
134	**155**	**176**	**197**	**205**	**220**	

A possible difficulty: While a set of long processes are being run and the quantum has been increased to several times the initial value, if a fresh short process were to come in, the waiting time for this new process would be much more than in the basic round robin case. This can cause a setback in the performance.

Table 6. Comparison of simple RR, Optimized RR and proposed RR

Algorithm	Time Quantum	Context Switches	Average WT	Average TAT
Simple RR	10	20	86	147
Optimized Round Robin	10, 20	15	80.5	142
Dynamic Time Slice Round Robin (proposed)	10, 20	13	79	141

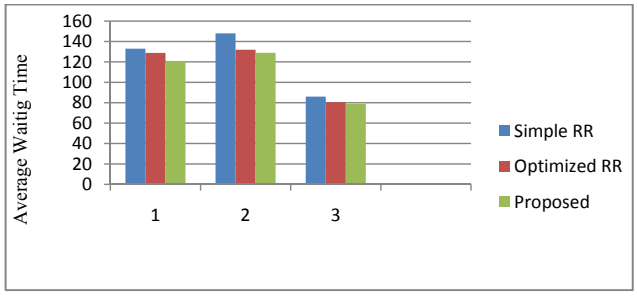

Figure 3. Waiting time cases

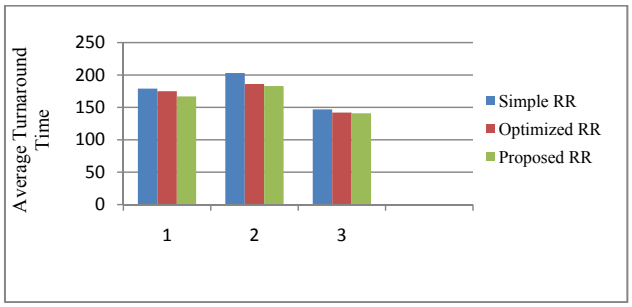

Figure 4. Turnaround time cases

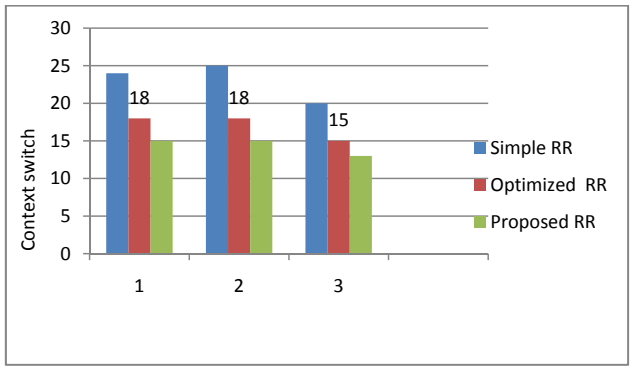

Figure 5. Context switches cases

4. Conclusion

With the proposed method we can reduce average waiting time, average turnaround time and number of context switches, provides better performance than simple RR.When the quantum has been increased and a process with small burst time were to arrive in the middle of execution then the algorithm could suffer- the new one has to wait more time than the basic rr. But even in this case, if the new incoming process is one that takes long CPU time, the setback in performance will be offset by the increase in quantum.

5. References

1. Hemamalini M, SrinathM. V.Memory Constrained Load Shared Minimum Execution Time Grid Task Scheduling Algorithm in a Heterogeneous Environment. Indian Journal of Science and Technology. 2015 July; 8(15).
2. AjithS, Priyanka G, Sahil B. An Optimized Round Robin Scheduling Algorithm for CPU Scheduling. International Journal on Computer Science and Engineering (IJCSE). 2010; 2(7): 2382-85.

3. Hiranwal S, Dr. RoyK C. Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice. International Journal of Data Engineering (IJDE).2011; 2(3): 319-323.
4. Banerjee P,Banerjee P, Dhal S S.Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum.International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2012 Aug; 1(3): 56-62.
5. Matarneh R J.Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes.American Journal of Applied Sciences ISSN 1546-9239. 2009; 6 (10): 1831-37.
6. Behera H S, Mohanty R, Sahu S, Bhoi S K.Comparative Performance Analysis Of Multi-Dynamic Time Quantum Round Robin (Mdtqrr) Algorithm With Arrival Time. Indian Journal of Computer Science and Engineering (IJCSE). ISSN 0976-5166. 2011; 2(2): 1-10.
7. Rajput I S, Gupta D. A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems. International Journal of Innovations in Engineering and Technology (IJIET). ISSN 2319 – 1058.2012 Oct; 1(3): 1-11.
8. Abdulrahim A,Aliyu S, Mustapha A M, Abdullahi S E. An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm.International Journal of Advanced Research in Computer Science and Software Engineering. ISSN 2277 128X.2014 Feb; 4(2):1-8.
9. Mishra M K, Khan A K. An Improved Round Robin CPU Scheduling Algorithm. Journal of Global Research in Computer Science.ISSN 2229-371X .2012Jun; 3(6): 64-69.
10. Noon A, Kalakech A, Kadry S.A New Round Robin Based Scheduling Algorithm for OperatingSystems: Dynamic Quantum Using the Mean Average. IJCSI International Journal of Computer Science Issues.2011 May;8(3): 224-29.
11. Varma P S. A Best possible Time quantumfor Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm. International Journal of Advanced Research inComputer Science and Software Engineering.ISSN 2277 128X. 2012 Nov; 2(11): 1-6.
12. BeheraH S,Mohanty R, Nayak D. A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis. International Journal of Computer Applications.2010 Aug; 5(5): 10-15.
13. Goel N, Dr. Garg R B.An Optimum Multilevel Dynamic Round Robin Scheduling Algorithm.National Conference on Information Communication & Networks. Organized by Tecnia Institute of Advanced Studies.Delhi. 2013 Apr.
14. SilberschatzA, GalvinP B,Gagne G. Operating System Concepts , 8th Edition. John Wiley and Sons