Improving Round Robin Process Scheduling Algorithm

Barkha Chhugani
Department of Computer Science & Engineering
School of Engineering Sciences & Technology
Jamia Hamdard

Mahima Silvester
Department of Computer Science & Engineering
School of Engineering Sciences & Technology
Jamia Hamdard

ABSTRACT
Round Robin Algorithm is used for process scheduling by assigning a fixed time quantum to every process which has to be executed. In this type of process scheduling, each process waiting in a ready queue is executed for a particular time quantum. If the burst time of the process is finished in one go then the process is removed from the ready queue. Otherwise, it is returned to the ready queue for its next quantum turn. In our proposal, we have calculated a dynamic time quantum for a process which fits to certain dynamically calculated conditions that we have defined later in the module due to which parameters like average turnaround time, average waiting time and the numbers of context switches have been decreased as compared to the standard Round Robin.

Keywords
Turnaround time, Waiting time, context switches, CPU Scheduling, time quantum.

1. INTRODUCTION
CPU scheduling is used to allocate CPU to different processes waiting in the ready queue. While one process is being executed the other processes are in the waiting state as the resources it needs for execution are preoccupied [7]. Some other process scheduling algorithms include First Come First Serve (FCFS) and Shortest Job First (SJF) algorithms [7], [8]. FCFS algorithm executes the first process that comes in the ready queue and that process is executed fully till its burst time is finished [4], [6], [8]. Shortest Job First (SJF) algorithm executes the process with the minimum burst time first and executes it till its burst time is finished. We have divided the paper into three portions. The first portion talks about the previous works in the related subject. The second portion describes the proposed approach and the third portion describes the various results and compares them to the conventional round robin approach.

2. RELATED WORKS
There have been several works done on improvising the round robin process scheduling algorithm. The authors of [1] have proposed a method using changeable time quantum that decides a value that is neither too large nor too small such that this value gives the best scheduling criteria and every process has got reasonable response time and the throughput of the system is not decreased due to unnecessary context switches.

The concept of [2] is to make the time context switches higher average waiting time and higher turnaround time of simple round robin scheduling algorithm which is used for the time sharing system. The objective of [3] is to modify Round Robin algorithm by adjusting time slices of different rounds depending on the remaining CPU bursts of currently running processes and coinciding their waiting time until that round in request of the other processes' waiting time. The [4] proposal calculates different time slices for individual processes coinciding their priorities. The primary objective of [5] is to optimize system performance according to the criteria deemed most important by system designers. The author of [6] proposed a modified round robin approach using the left out burst time. The authors of [10] talked about dynamically calculation the time quantum using some threshold value. The researchers in [11] proceeded by increasing the time quantum of few processes which require fractionally more time to complete their execution than the allocated time quantum.

3. PROPOSED APPROACH
This section describes our proposal. In our proposal, we have first calculated average turnaround time, average waiting time and the number of context switches according to the standard round robin algorithm, then calculated the same parameters using our approach, where we are increasing the time quantum for the processes which satisfy certain conditions described in the pseudo code, in their second last turn.

3.1 Terminologies
- P: Process
- TQ: Time Quantum
- B: Number of Turns
- BT: Burst Time
- AT: Arrival Time
- old_TQ: Original Time quantum
- i: Number of processes in the ready queue
- sqrt: Square root
- B = BT%TQ
- k = | sqrt (max[BT] – min[BT]) |
- NOT = ceil(BT /TQ).

3.2 Pseudo code for the Proposed Approach

1. Initialize: old_TQ = TQ;
2. Initialize: count(i)=1;
3. for (i=0;i<=number of processes in ready queue; i++)
4. {
    a. if (BT[i]%TQ == 0)
    b. {
    c. Execute as per Conventional Round robin
        i. do
        ii. {
        iii. BT[i] = BT[i] – TQ
        iv. if(BT[i]==0)
        v. {
        vi. Remove the process from the ready queue
        vii. }
        viii. else
        ix. {
        x. Send the process back to the ready queue for its execution
        xi. }

xii. }

5. }
6. Else if(BT[i] < TQ)
7. {
   a. Execute process as per Conventional Round Robin and remove from the ready queue
8. }
9. Else if((BT[i]%TQ)>0)
10. {
    a. if (count == (NOT – 1))
    b. {
       i. if (B<=k)
    c. {
       i. TQ = TQ + k
       ii. BT[i] = BT[i] – TQ
       d. }
11. Else
    a. {
    b. BT[i] = BT[i] – TQ
    c. count++
    d. }
12. }
13. }

4. EXAMPLES

Here, we have worked on few examples using standard round robin as well as our approach.

Abbreviations: TAT: Turnaround Time, WT: Waiting Time, Average TAT: Average Turnaround Time, Average WT: Average Waiting Time.

4.1 Example 1

Table 1. For Conventional Round Robin Approach

Time Quantum(TQ)=1O		
Process	**AT**	**BT**
P1	0	42
P2	0	54
P3	0	35
P4	0	26

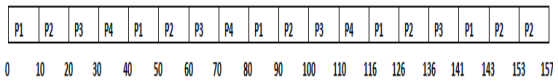

Fig 1. Gantt chart for Example 1 using conventional Round Robin approach

4.1.1. Calculations using Classic Round Robin Algorithm:

TAT(P1)=143 WT(P1)=101
TAT(P2)=157 WT(P2)=103
TAT(P3)=141 WT(P3)=106
TAT(P4)=116 WT(P4)=90
Average TAT= 139.25
Average WT= 100
Context Switches= 18

According to Proposed Approach,

Table 2: Example 1 using Proposed Approach

P	AT	BT	B=BT%TQ	C=INT(BT/TQ)
P1	0	42	2	4
P2	0	54	4	5
P3	0	35	5	3
P4	0	26	6	3

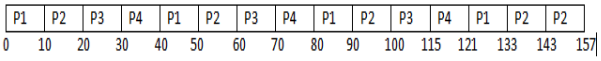

Fig 2: Gantt chart for proposed approach

Calculation of TQ:
Max[BT] = 54
Min[BT] = 26

$$k = \left| sqrt(Max\ [BT] - Min\,[BT]) \right|$$

= |sqrt(54 – 26)|
= |sqrt(28)|
= 5
As, B(for P1,P2,P3) <= k
Then TQ = TQ + k
i.e., TQ = 10 + 5 = 15

Now, as per the proposed approach all the processes which have 'B' less than or equal to the value 'k' then one but last turn of the above mentioned processes will be executed using the new TQ. Other remaining processes that were not able to fulfill the condition will be executed using the original time quantum.

Calculations:
TAT(P1)=133 WT(P1)=91
TAT(P2)=157 WT(P2)=103
TAT(P3)=115 WT(P3)=80
TAT(P4)=121 WT(P4)=85
Average TAT=131.5
Average WT=89.75
Context Switches=15

4.2 Example 2

Table 3: For conventional round robin algorithm

Time Quantum(TQ)=05		
Process	**AT**	**BT**
P1	0	19
P2	0	9
P3	0	23
P4	0	15
P5	0	16

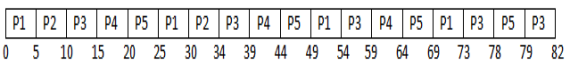

Fig 3: Gantt chart for standard round robin algorithm

4.2.1. Calculations using Classic Round Robin Algorithm:

TAT(P1)=73
TAT(P2)=34
TAT(P3)=82
TAT(P4)=64

TAT(P5)=79

WT(P1)=54
WT(P2)=25
WT(P3)=59
WT(P4)=49
WT(P5)=63

Average TAT=66.4
Average WT=50
Context Switches=18

According to the Proposed Approach

Table 4: For proposed approach

P	AT	BT	B=BT%TQ	C=INT(BT/TQ)
P1	0	19	4	4
P2	0	9	4	2
P3	0	23	3	5
P4	0	15	0	3
P5	0	16	1	3

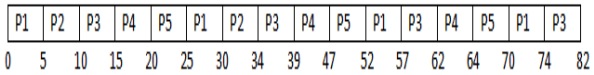

Fig 4: Gantt chart for proposed approach

Calculation of TQ
Max [BT] = 23
Min [BT] = 09

$k = |sqrt(Max\ [BT] - Min\,[BT])|$

= |sqrt(23 – 09)|
= |sqrt(14)|
= 3

As, B(for P3,P4,P5) <= k
Then TQ = TQ + k
i.e., TQ = 5 + 3 = 8
Calculations:

TAT(P1)=74 WT(P1)=55
TAT(P2)=34 WT(P2)=25
TAT(P3)=82 WT(P3)=59
TAT(P4)=64 WT(P4)=49
TAT(P5)=70 WT(P5)=54

Average TAT=64.8
Average WT=48.4
Context Switches=16

4.3 Example 3

Table 5: For Conventional Round Robin

Time Quantum(TQ)=05		
Process	**AT**	**BT**
P1	0	18
P2	0	6
P3	0	27
P4	0	31
P5	0	16

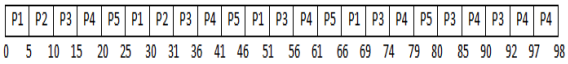

Fig 5: Gantt chart for conventional round robin algorithm

4.3.1. Calculations using Classic Round Robin Algorithm:

TAT(P1)=69 WT(P1)=57
TAT(P2)=31 WT(P2)=25
TAT(P3)=92 WT(P3)=65
TAT(P4)=98 WT(P4)=67
TAT(P5)=80 WT(P5)=64

Average TAT=74
Average WT=55.6
Context Switches=22
According to Proposed Approach,

Table 6: For proposed approach

Process	AT	BT	B=BT%TQ	C=INT(BT/TQ)
P1	0	18	3	4
P2	0	6	1	2
P3	0	27	2	6
P4	0	31	1	7
P5	0	16	1	4

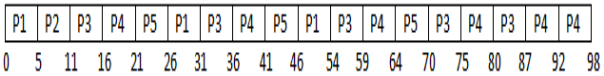

Fig 6: Gantt chart for the proposed approach

Calculation of Time Quantum
Max [BT] = 31
Min [BT] = 06

$k = |sqrt(Max\ [BT] - Min\,[BT])|$

= |sqrt(31 – 06)|
= |sqrt(25)|
= 5

As, B(for P1,P2, P3,P4,P5) <= k , this becomes the ideal case since all the processes fulfil the condition. So, for the processes time quantum will be increased by 5 for one but last turn.
TQ = TQ + k
i.e., TQ = 5 + 5 = 10

Calculations:
TAT(P1)=54 WT(P1)=36
TAT(P2)=11 WT(P2)=05
TAT(P3)=92 WT(P3)=65
TAT(P4)=102 WT(P4)=71
TAT(P5)=70 WT(P5)=54

Average TAT=65.8
Average WT=46.2
Context Switches=17

5. RESULTS AND CONCLUSION

The approach that we have used increases the time quantum for few processes on the basis of some threshold value. This further decreases the average turnaround time, average waiting time and the number of context switches as compared to the conventional round robin algorithm, thereby reducing the overhead of the CPU to some extent. To justify our proposal we have presented the results and comparisons in the form of following graphs and tables The results shows that we have decreased the average turnaround time, average waiting time and number of context switches by 5.5% ,10.25% and 16.6 % respectively for example 1, 2.4%, 3.3% and 11.1% respectively for example 2, 11.08%,16.9% and 22.7 % respectively for example 3.The present work is limited to uni-processor system, the work can be implemented for multiprocessor systems as a future scope.

Table 7: Comparison table for Example 1

Parameter	Standard Round Robin	Proposed Approach	Remarks
Average Turnaround Time(TAT)	139.25	131.5	7.75 units of time saved
Average Waiting Time(WT)	100	89.75	10.25 units of time saved
Number of Context Switches(CS)	18	15	3 context switches saved

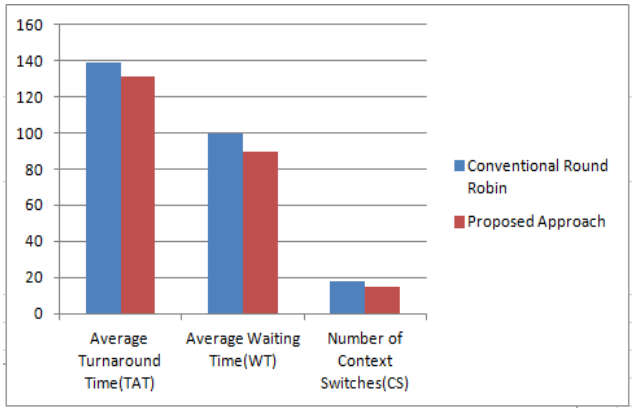

Fig 7: Parameter Comparison Graph for Example 1

Table 8: Comparison table for Example 2

Parameter	Standard Round Robin	Proposed Approach	Remarks
Average Turnaround Time(TAT)	66.4	64.8	1.6 units of time saved
Average Waiting Time(WT)	50	48.4	1.6 units of time saved
Number of Context Switches(CS)	18	16	2 context switches saved

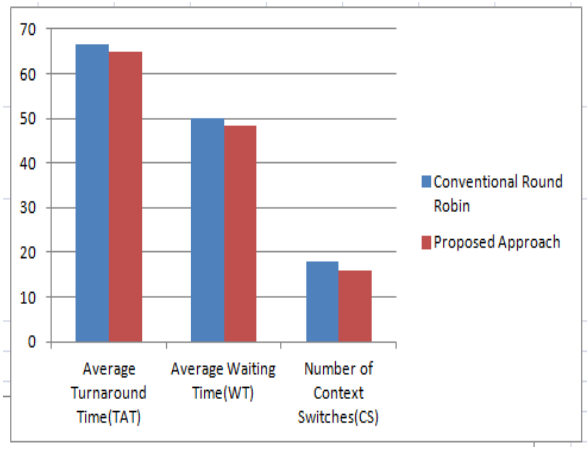

Fig 8: Parameter Comparison Graph for Example 2

Table 9: Comparison table for Example 3

Parameter	Standard Round Robin	Proposed Approach	Remarks
Average Turnaround Time(TAT)	74	65.8	8.2 units of time saved
Average Waiting Time(WT)	55.6	46.2	9.4 units of time saved
Number of Context Switches(CS)	22	17	5 context switches saved

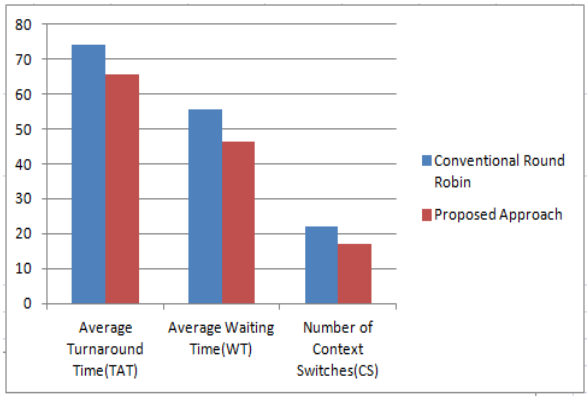

Fig 9: Parameter Comparison Graph for Example 3

6. REFERENCES

[1] Mostafa SM, Hamad SH, Rida SZ (2011) Improving Scheduling Criteria of Preemptive Tasks Scheduled under Round Robin Algorithm using Changeable Time Quantum. J Comput Sci Syst Biol 4:071-076. doi:10.4172/jcsb.1000078

[2] Neha Mittal , Khushbu Garg , Ashish America, "A Paper on Modified Round Robin Algorithm, IJLTEMAS, Volume IV, Issue XI, November 2015

[3] Lipika Datta, Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice, I.J. Education and Management Engineering, 2015, 2, 10-19 Published Online June 2015 in MECS (http://www.mecs-press.net) DOI: 10.5815/ijeme.2015.02.02

[4] Lipika Datta, Modified RR Algorithm with Dynamic Time Quantum for Externally Prioritized Tasks, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 1,Pp 217 - 221

[5] Amit Kumar Sain, Dynamical Modified R.R. CPU Scheduling Algorithm, International Journal of Computer Trends and Technology- volume4, Issue2- 2013

[6] Mohd Abdul Ahad. Article: Modifying Round Robin Algorithm for Process Scheduling using Dynamic Quantum Precision. IJCA Special Issue on Issues and Challenges in Networking, Intelligence and Computing Technologies ICNICT(3):5-10, November 2012., Pp 5-10

[7] Silberschatz, A. P.B. Galvin and G. Gagne (2012), Operating System Concepts, 8th edition, Wiley India

[8] Operating Systems Sibsankar Haldar 2009, Pearson

[9] D.M. Dhamdhere operating Systems A Concept Based Approach, Second edition, Tata McGraw-Hill, 2006.

[10] Aashna Bisht, Mohd Abdul Ahad, Sielvie Sharma, Enhanced Round Robin Algorithm for Process Scheduling using varying quantum precision, Proceedings of ICRIEST AICEEMCS, 29th December,2013.pp 11-15

[11] Aashna Bisht, Mohd Abdul Ahad and Sielvie Sharma. Article: Calculating Dynamic Time Quantum for Round Robin Process Scheduling Algorithm. International Journal of Computer Applications 98(21):20-27, July 2014.