

Dynamic Time Slice Calculation for Round Robin Process Scheduling Using NOC

G. Siva Nageswara Rao¹, N. Srinivasu², S.V.N. Srinivasu³, G. Rama Koteswara Rao⁴

^{1,2}Department of Computer Science and Engineering, K L University, India

³Department of Computer Science, HOD, PNC&KR PG College Narasarao PET, India

⁴Department of Computer Science and Engineering, V R Siddhatha Engg. College, Vijayawada, India

Article Info

Article history:

Received Apr 12, 2015

Revised Jul 20, 2015

Accepted Aug 16, 2015

Keyword:

Number of context switches

Number of cycles

Remaining burst time

Time slice

Turnaround time

ABSTRACT

Process scheduling means allocating a certain amount of CPU time to each of the user processes. One of the popular scheduling algorithms is the "Round Robin" algorithm, which allows each and every process to utilize the CPU for short time duration. This paper presents an improvisation to the traditional round robin scheduling algorithm, by the proposed a new method. The new method represents the time slice as a function of the burst time of the waiting process in the ready queue. Fixing the time slice for a process is a crucial factor, because it subsequently influences many performance parameters like turnaround time, waiting time, response time and the frequency of context switches. Though the time slot is fixed for each process, this paper explores the fine-tuning of the time slice for processes which do not complete in the stipulated time allotted to them.

Copyright © 2015 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

G. Siva Nageswara Rao,
Department of Computer Science and Engineering,
K L University,
Greenfields, Vaddeswaram, Guntur District, Andhra Pradesh 522502.
Email: sivanags@kluniversity.in

1. INTRODUCTION

Scheduling is fundamental operating system function and scheduling is needed for each and every resource .CPU is one of the major resource so its scheduling occupies major role in the design of operating system [1]-[3]. CPU scheduling deals with problem of deciding which of the process in the ready queue is to be allotted to the CPU for processing .For doing this FCFS ,SJF ,Priority and Round Robin Algorithms are available [4]-[7].

The following characteristics CPU usages (load the cpu as busy as possible),

1) Throughput (number of processors that complete their execution in unit of time)

2) Turnaround time (amount of time to execute particular process)

3) Waiting time (amount of time the process has been waiting in the ready queue)

4) Response time (amount of time it take from when a request was submitted until the first response is produced, not output) are used compare and determine which algorithm is best.

Round Robin Scheduling is designed for time sharing operating systems, this is very similar to FCFS only difference is preemption is added to it to switch from one process to another process. In round robin algorithm initially time quantum or time slice ,it varies with milliseconds and this is used to preempt i.e. CPU switches from one process to another after each and every time quantum, and sometimes switching occur even before the time completion of time quantum [8]-[9], [17].

Here we keep the ready queue as a circular FIFO queue. The CPU scheduler selects the one of the process from the ready queue and CPU is allotted to it end. When the process schedule one of the two things may happens [13]-[16].

1) The burst time of the process greater than time slice CPU switches to another process which is selected by the CPU Scheduler for the next time quantum, the current process is sent back to the ready queue and it will be observed from the remaining time later [18]-[19], [26].

2) If the burst time of the process is less than the times slice then the current process is removed from the ready queue and the CPU switches another process which is selected by the schedule [20]-[22].

There are numerous researches going around the globe on improving the performance of round robin algorithm. The author of [1] proposes a fuzzy approach to find the suitable time slice for the processes. They have presented the results using different simulations. The novelists of [2] propose a median based approach to find the time slice, combining the conventional shortest job first and Round Robin algorithms. The authors of [3] proposes a new technique using maximum and minimum burst time of the set of processes in the ready queue and calculating a modified time slice. The authors of [4] talks about calculating the time slice using median and highest burst time and then executing the processes as per the new calculated time slice. The researchers of [5] provided a mathematical model for calculating the waiting time and turnaround time. The authors of [6] talks about calculating the mean of the burst times of all the processes and then finds the difference between the mean of the burst time and the burst time of a particular process and allocates the CPU to the process which has the maximum difference.

So, a good scheduling algorithm should possess the following characteristics [2]:

- 1) Minimize the context switches.
- 2) Maximize the CPU utilization.
- 3) Maximize the throughput.
- 4) Minimize the turnaround time.
- 5) Minimize the waiting time.
- 6) Minimize response time.

2. PROPOSED APPROACH

Our proposed approach does not aim to change the behavior of the conventional round robin algorithm but to improve it further. In our proposed approach, we will be modifying the time slice of only those processes which require a slightly greater time than the allotted time slice cycle(s) [1]-[3]. The remaining processes will be executed in the basic Round Robin manner. Hence we calculate the remaining burst time and no. cycles for each process [23]-[25]. Based on the remaining burst time, we sort the process, if the remaining burst time is less than or equal to the one time slice then execute the same process otherwise go for next process. If more than one process having the same remaining burst time then use the Shortest Job First Scheduling Algorithm [10]-[12].

TS: Time Slice

BT : Burst Time

RBT : Remaining Burst time

$RBT [P_i] = BT [P_i] \% TS$

NOC : Number of Cycles

$NOC [P_i] = \text{ceil} (BT [P_i] / TS)$, where ceil function gives the largest integer greater than or equal to the number.

2.1. Proposed Algorithm

Step1: START

Step2: Make a ready queue of the Processes say Request.

Step3: Calculate the Time Slice $(TS = \text{floor} ((\sum(BT [P_i]) / N))$

Step4: Calculate the Remaining burst time and Number of Cycles for all processes
($RBT = BT[P_i]\%TS$, $NOC = BT[P_i]/TS$)

Step5: Sort the all processes based on remaining burst time & NOC.

Step6: Pick the process from the ready queue and allocate the CPU to it for a Time interval of up to 1 time quantum.

Step7: If the remaining CPU burst time of the currently running process is less than or Equal to the one time quantum then allocate CPU again to the currently running process, otherwise to the next process.

Step8: Repeat Step6 & Step 7 until all process are scheduled.

Step9: END.

Example1: Consider the Five Process and their Time slice is given in the following Table.

(a) Static Time Slice:

Consider the Static Time Slice = 10

Table 1. Process table

Process Name	Burst Time
PS1	12
PS2	15
PS3	23
PS4	37
PS5	21

Table 2. Comparison table for various scheduling algorithms using static time slice

Type of Algorithm	Avg. TAT.	Avg. TWT.	NCS
Basic Round Robin	81.6	60	13
Aashna Bisht Method	71.2	49.6	10
My Proposal1	65.2	40.2	7
My Proposal2	59.2	37.6	7

In the above Table 1 My Proposal2 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high.

(b) Dynamic Time Slice:

TQ = Avg. Burst time of the all the process (i.e. TS= 22)

Table 3. Grant Chat

0	PS1	12	12	PS2	27	27	PS5	48	48	PS3	71	71	PS4	108
---	-----	----	----	-----	----	----	-----	----	----	-----	----	----	-----	-----

Table 4. Comparison table for various scheduling algorithms using Dynamic time slice

Type of Algorithm	Avg. TAT.	Avg. TWT.	NCS
Basic Round Robin	59	40.4	6
Aashna Bisht Method	58	36.4	5
My Proposal1	53.2	31.8	4

In the above Table 4 My Proposal1 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high.

Example2: Consider the Five Process and their Time slice is given in the following Table.

(a) Static Time Slice:

Consider the Static Time Slice = 4

Table 5. Process Table

Name of the Process	Burst Time
PS1	19
PS2	9
PS3	23
PS4	13
PS5	17

Table 6. Grant Chat

0	PS2	9	9	PS4	22	22	PS5	39	39	PS1	58	58	PS3	81
---	-----	---	---	-----	----	----	-----	----	----	-----	----	----	-----	----

Table 7. Comparison table for various scheduling algorithms using static time slice

Type of Algorithm	Avg. TAT.	Avg. TWT.	NCS
Basic Round Robin	68.6	52.4	22
Aashna Bisht Method	60.6	44.4	18
My Proposal1	57	40.8	16
My Proposal2	57	40.8	16

(b) Dynamic Time Slice:

TS = Avg. Burst time of the all the process (i.e. TS= 16)

Table 8. Comparison table for various scheduling algorithms using Dynamic time slice

Type of Algorithm	Avg. TAT.	Avg. TWT.	NCS
Basic Round Robin	62.6	46.4	7
Aashna Bisht Method	51.2	35.6	5
My Proposal1	41.8	25.6	4

3. RESULTS & GRAPH

Example 1:

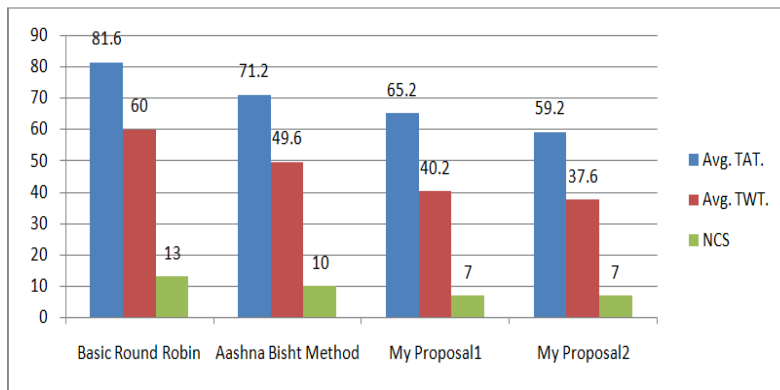


Figure 1. Comparison table for various scheduling algorithms using static time slice

In the above Figure 1 My Proposal2 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high.

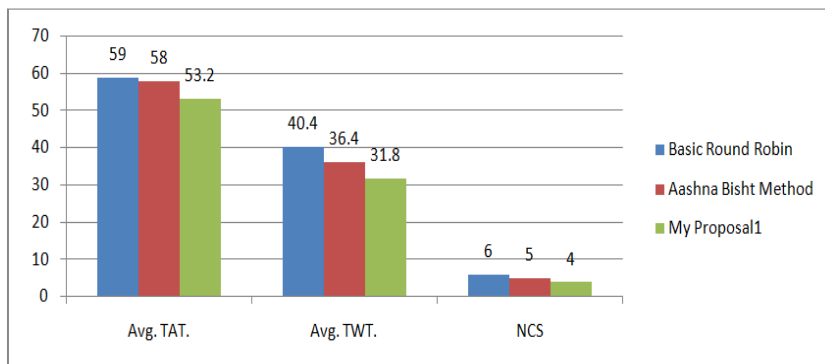


Figure 2. Comparison table for various scheduling algorithms using Dynamic time slice

In the above Figure 2, My Proposal1 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high.

Example 2:

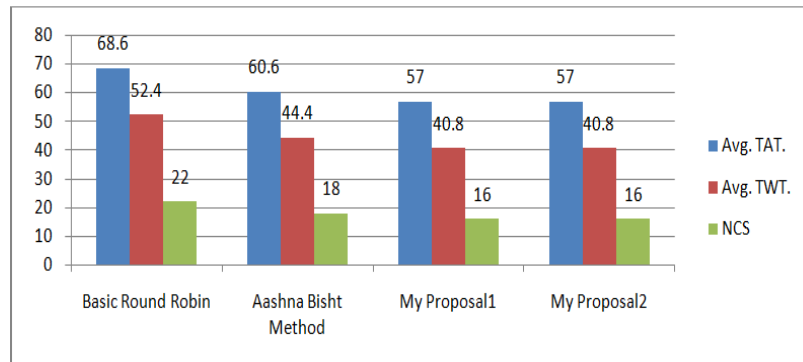


Figure 3. Comparison table for various scheduling algorithms using static time slice

In the Figure 3, My Proposal2 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high. The authors of [6] talks about calculating the mean of the burst times of all the processes and then finds the difference between the mean of the burst time and the burst time of a particular process and allocates the CPU to the process which has the maximum difference

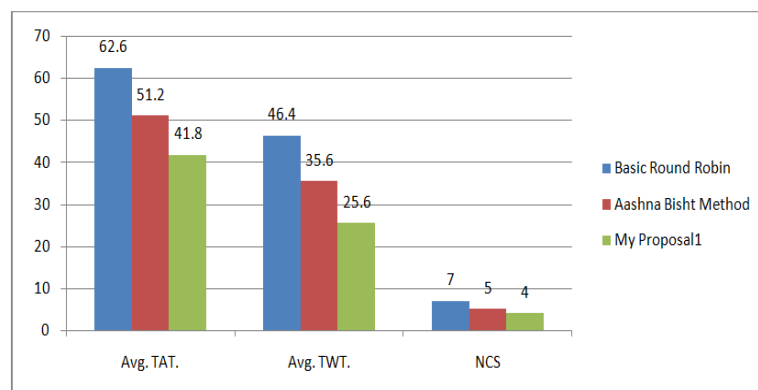


Figure 4. Comparison table for various scheduling algorithms using Dynamic time slice

With the reference of the [6], we modified the algorithm with better results than earlier scheme scheduling process. We scheme takes lower time than the basic round robin method, which is shown in the Figure 4. In the above Figure 4 My Proposal1 Algorithm has minimum Avg. TAT, minimum TWT and minimum no. of Context switches. Due to less context switching, the processor idle time is low and resource utilization is very high.

4. CONCLUSION

In this paper an improvement for the conventional round robin algorithm is proposed which is being supported by a set of hypothetical examples and a better amount of improvement is observed. The approach can be further refined using the concept of arrival time.

REFERENCES

- [1] G Siva, Nageswara Rao, *et al.*, "An Enhanced Dynamic Round Robin CPU Scheduling Algorithm", *International Journal of Applied Engineering Research*, Vol. 9, No. 15, pp. 3085-3098, 2014.
- [2] G Siva, Nageswara Rao, *et al.*, "Comparison of Round Robin CPU Scheduling Algorithm with Various Dynamic Time Quantum", *International Journal of Applied Engineering Research*, Vol. 9, No. 18, pp. 4905-4916, 2014.
- [3] G Siva, Nageswara Rao, *et al.*, "A NEW PROPOSED DYNAMIC DUAL PROCESSOR BASED CPU SCHEDULING ALGORITHM", *Journal of Theoretical and Applied Information Technology 20th March 2015*, Vol.73, No. 2, 1992.
- [4] Bashir Alam, "Fuzzy Round Robin CPU Scheduling Algorithm", *Journal of Computer Science*, Vol. 9, No. 8, pp. 1079-1085, 2013.
- [5] Lalit Kishor, Dinesh Goyal, "Time Quantum Based Improved Scheduling Algorithm", *Issue International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3, 2013.
- [6] P. Surendra Varma, "A Best possible time quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm", *International Journal of Advanced Research in Computer Science and software Engineering*, Vol. 2, No. 11, 2012.
- [7] K. N. Rout, G. Das, B. M. Sahoo, and A. K. Agrawalla, "Improving Average Waiting Time Using Dyanamic Time Quantum".
- [8] Shahram Saiedi, Hakimeh Alemi Baktash, Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method, *International Journal of Information Technology and Computer Science*, pp: 67-73 2012.
- [9] H. S. Behera, Simpi Patel, and Bijayalakshmi Panda, "A New Dynamic Round Robin and SRTN Algorithm with Variable Original Time Slice and Intelligent Time Slice for Soft Real Time Systems".
- [10] Supriya Raheja, Reena Dhadich, and Smita Rajpal, "An Optimum Time Quantum Using Linguistic Synthesis for Round Robin CPU Scheduling Algorithm", *International Journal on Soft Computing (IJSC)*, Vol. 3, No. 1, 2012.
- [11] Nayana Kundargi, Sheetal Bandekar, "CPU Scheduling Algorithm Using Time Quantum For Batch System," *International Journal of Latest Trends In Engeneering And Technology (IJLTET)*, 2013.
- [12] Himanshi Arora, Deepanshu Arora, Bagish Goel, and Parita Jain, "An Improved Scheduling Algorithm", *International Journal of applied Information Systems(IJAIS), Foundation of Computer Science FCS*, New York, USA, Vol. 6, No. 6, pp 7- 9, 2013.
- [13] Sandeep Negi, "An Improved Round Robin Approach Using Dyanamic Time Quantum For Improving Average Waiting Time", *International Of Computer Applications*, Vol. 69, No. 14, pp. 12-16, 2013.
- [14] Rakesh Patel, Mrs. Mili Patel, "SJRR CPU Scheduling Algorithm", *International journal of Engineering And Computer Science*, Vol. 2, No. 12, pp. 3396-3399, 2013.
- [15] Adeeba Jamal, Aiman Jubair, "A Varied Round Robin Approach Using Harmonic Mean Of The Remaining Burst Time Of Processes", *Special Issue of International Journal of Computer Applications, 3rd International IT Summit Confluence*, pp. 11- 17, 2012.
- [16] Mohd Abdul Ahad, "Modyfying Round Robin Algorithm For Process Scheduling Using Dynamic Quantum Precision", *Special Issue Of International Journal of Computer Applications On Issues AND Challenges In Networking ,Intelligence And Computing Technologies, ICNICT 2012*, pp. 5-10, 2012.
- [17] C. Yaashuwanth, Dr. R. Ramesh, "A New Scheduling Algorithms for Real Time Tasks", *(IJCSIS) International Journal of Computer Science and Information Security*, Vol. 6, No. 2, pp. 61-66, 2009.
- [18] Sanjaya Kumar Panda, and Sourav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure", *IJCSE*, Vol. 4, No. 01, pp. 45-53, 2012.
- [19] Aashna Bisht, Mohd Abdul Ahad, Sielvie Sharma, "Enhanced Round Robin Algorithm for Process Scheduling using varying quantum precision", *Proceedings of ICRIEST AICEEMCS*, pp. 11-15, 2013.
- [20] R. N. D. S. S Kiran, Polinati Vinod Babu, and B. B. Murali Krishna, "Optimizing CPU Scheduling for Real Time Applications Using Mean-Difference Round Robin (MDRR) Algorithm", *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*, Vol. 1, Advances in Intelligent Systems and Computing, Vol. 248, pp. 713-721, 2014.
- [21] M. H. Zahedi, M. Ghazizadeh, and M. Naghibzadeh, "Fuzzy Round Robin CPU Scheduling (FRRCS) Algorithm", *Advances in Computer and Information Sciences and Engineering*, pp. 348-353, 2008.
- [22] Bin Nie, Jianqiang Du, Guoliang Xu, Hongning Liu, Riyue Yu, and Quan Wen, "A New Operating System Scheduling Algorithm", *Advanced Research on Electronic Commerce, Web Application, and Communication, Communications in Computer and Information Science*, Vol. 143, pp. 92-96, 2011.
- [23] T. Gunasekhar, and K. Thirupathi Rao, *et al.*, "A Survey on Denial of Service Attacks", *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 2, 2373-2376, 2014.
- [24] T. Gunasekhar, K. Thirupathi Rao, *et al.*, "Mitigation of Insider Attacks through Multi-Cloud", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 1, pp. 136-141, 2018.
- [25] T.Gunasekhar, K. Thirupathi Rao, "EBCM: Single Encryption, Multiple Decryptions", *International journal of Applied engineering Research*, Vol. 9, No. 19, pp. 5885-5893, 2014.
- [26] Adam Hendra Brata, Deron Liang, and Sholeh Hadi Pramono, *et al.*, "Software Development of Automatic Data Collector for Bus Route Planning System", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 1, pp. 150-157, 2015.