JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

An Optimal Approach to Selecting the Time Quantum for Dynamic Round Robin

Sumit Mohan[1*], Rajnesh Singh[2]

[1] IEC College of Engineering & Technology, AKTU, Greater Noida, India,
[2]EC College of Engineering & Technology, AKTU, Greater Noida, India,

Available online at: www.ijcseonline.org

Accepted: 20/Jun/2018, Published: 30/Jun/2018

Abstract— in the modern era of technology, Multiprogramming (Multi-Tasking) became a major issue for a system because every user wants to run many applications at a time. In time shared-environment, system resources distributed among the available application in main memory. Improper uses of system resources may degrade the system performance, so we need proper utilization of resources. Hence scheduling is a mechanism by which we can utilize the system resources efficiently. We have many CPU scheduling algorithms but in contrast to time shared environment, round robin is the best choice. In this algorithm, processes get equal opportunity to execute its task. System performance solely depends on scheduling algorithm and round robin performance only depends on the choice of time quantum. Thus selecting the optimal time quantum is the main problem in this algorithm because if it very small then CPU will spend lots of time context switches which will degrade the system performance and if it is too large then response time processes will be increased which cannot be tolerated in time shared environment. Therefore this shows an optimal approach for selecting time quantum and comparison shows that the OASRR algorithm performance better than all variants of round robin algorithms and this algorithm having minimum avg. turnaround time, waiting time and less no. of context switches.

Keywords— CPU Scheduling, Round Robin, Time Quantum, Waiting Time, Turnaround Time, Context Switches

I. INTRODUCTION

Improper use of CPU can affect the efficiency of a multi-programmed system so we need to schedule task properly in which we can utilize system resources efficiently. Hence Scheduling algorithms are used for proper utilization of system resources, we have many algorithms and each algorithm has a specific environment. We can select the algorithm as per our requirements. In contrast of time-sharing environment, we have round robin scheduling algorithm here every process gets equal opportunity to execute its task [3]. Hence selecting the optimal time is the main problem because time quantum is the key element of round robin performance which leads to also system performance.

Every process has following parameter CPU time, Arrival time and priority. CPU time is the of the process which needs CPU resources or I/O operations, the priority of a process tells CPU to its importance and Arrival time of any process when a process submit a request to the ready queue, Turnaround Time is the difference of submission of a request to completion[10]. The Waiting Time of a process is the amount of time in which process spends for its execution. Context Switching when the processor switches from current process to other processes, Effectiveness of CPU scheduling algorithm always depends on waiting, turnaround time and context switches [11].

CPU scheduling algorithm performance criteria are following**:**

Processor Utilization: Sum of work done by processor We can estimate the performance of a system by processor utilization. Always we want processor utilization as much as possible [8].

Throughput: Measure the work of a processor done in one unit of time.

Throughput = (Completed Processes) / (One Unit Time)

Turnaround Time: The time which is spent by a process in the waiting queue for the ready queue.

TAT = Completion Time – Request Submission Time

Waiting Time (WT): When process wait for its turn in the ready queue is called Waiting Time.

WT=Turnaround Time –Burst Time

Response Time (RT): when processor starts processing the request of the process

RT = First Response – Arrival Time

II. Related Work

There are many problems with static round robin because in this algorithm we choose the time quantum randomly; sometimes it gives a good performance. But we cannot say that always [4] it will give a good performance because time quantum is the main key of the performance of round robin. We cannot choose time quantum very small and too large it Must be optimal because in the small time quantum more no. of context switches occur and response time of the process will be increased if we choose time quantum too much large[3].

Recently many approaches proposed for selecting Time Quantum. Some of the approaches are very optimal and these approaches are adaptive Scheduling using Shortest Burst Approach Based on Smart Time Slice [1] in this algorithm all processes are arranged in increasing (burst time) order, shortest burst time process will get first priority to run its task and time quantum calculated by average of all process if no. of processes are even otherwise mid process burst time will be selected as a time quantum. Min-Max Round Robin [2] in this algorithm time calculated by maximum burst time of process – minimum burst time of the process.

MDTRR [5] this algorithm gave another approach for time quantum here time quantum selected median of all processes and this time quantum assigned till the median after that time quantum will be assigned by the first process of the first quarter.

Intelligent Time Slice for Round Robin [7] here time quantum depends on various factor OTS, Priority Component and context switches component.

[6]SRBRR in this paper time quantum calculated by the median of the processes multiplied by highest burst time process then applied sqrt function and then finally take the ceil value of calculated s result.

III. Methodology

The proposed algorithm (OASRR) follows simple approach to calculate the optimal time quantum as static round robin follows. OASRR algorithm has a special parameter which adjustment factor alpha, it's a constant and its value is α=0.9. This algorithm is taking advantages of shortest job first but here less possibility of the starvation problem because it has optimal time quantum value which is calculated by as follows:

$$\boxed{\mathbf{TQ = Ceil\ [\{(\textstyle\sum (Pi\ /n)^*\alpha\}]}}$$

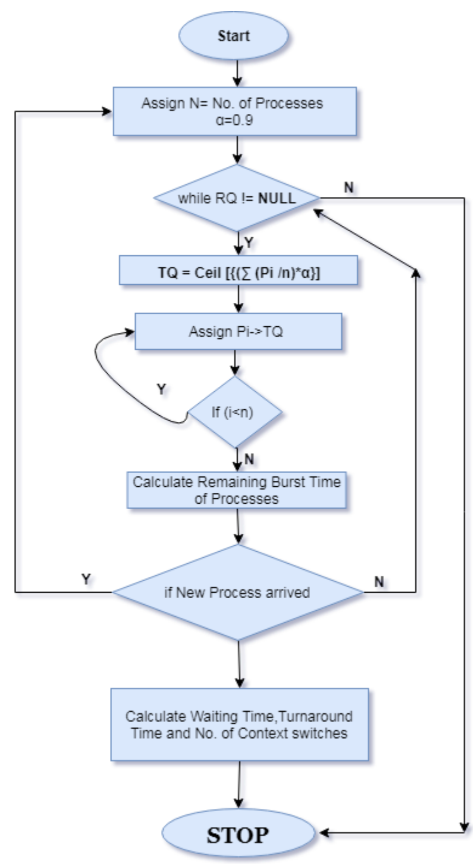
The proposed (OASRR) algorithm follows:

a. Initialize n= No. of Processes and α=0.9.
b. While (ready queue! = **NULL**)

 $\mathbf{TQ = Ceil\ [\{(\sum (P_i\ /n)^*\ \alpha\ \}]}$
 For i=1 to n…// i loop variable
 Where α is the Adjustment Factor.
c. Assign TQ to
 I_{th}Process Pi→TQ

d. If (i<n) then go to step c.
e. Calculate the remaining burst time of processes
f. If other processes (new) come then
 Update n and goto: a

 End of while
f. Calculate Turnaround Time, Waiting Time and No.of Context Switches.
g. End



Flow Chart of The Proposed Algorithm

IV. Results and Discussion

Assumptions' assumed that Attributes of a process (CPU Time, Priority) must be known before submission of processes, and CPU bound. The Arrival Time of all processes must be same and this experiment performed in a single processor system, the processes are independent of each other.

Case: 1

Let's take 5 Processes in increasing order at arriving time=0
As shown below.

Table 1: Burst Time of Processes

Process	CPU Time
P_1	37
P_2	58
P_3	76
P_4	115
P_5	137

TQ calculated by the proposed formula

TQ = Ceil [{(∑ (Pi /n)*α}]

TQ=Ceil [{(37+58+76+115+137)/5}* α]
= (423) ÷5=84.6
= 84.6*0.9= 75.14
= ceil (75.14) =76
TQ= 76

Gantt chart Case: 1

P_1	P_2	P_3	P_4	P_5	P_4	P_5

0 37 95 171 247 323 362 423

No. of Context Switches = 6

Avg. Waiting Time = (0+37+95+247+286) / 5 = 665/5=133
Avg. Turnaround Time = (37+95+171+362+423) / 5 =1088/5=217.6

Table 2: Comparative Analysis RR, OASRR algorithm (case – 1)

Algorithm	**TQ**	**Avg. TAT**	**Avg. W T**	**C S**
RR	50	272.4	187.8	10
OASRR	76	217.6	133	6

Case: 2

Let's Take Same Processes burst time in Decreasing order and processes at t=0 Arriving Time
As shown below

Table 3: Burst Time of Processes

Process	**CPU Time**
P_1	137
P_2	115
P_3	76
P_4	58
P_5	37

TQ calculated by the proposed formula
TQ = Ceil [{(∑ (Pi /n)*α}]

TQ=Ceil [{(137+115+76+58+37)/5}* α]
= (423) ÷5=84.6
= 84.6*0.9= 75.14
= ceil (75.14) =76
TQ= 76

Gantt chart Case: 2

P_1	P_2	P_3	P_4	P_5	P_4	P_5

0 37 95 171 247 323 362 423

No. of Context Switches = 6
Avg. Waiting Time = (247+308+152+228+286) / 5 = 1221/5=244.2
Avg. Turnaround Time = (384+423+228+286+323) / 5 =1664/5=328.8

Table 4: Comparative Analysis RR, OASRR algorithm (case – 2)

Algorithm	**TQ**	**Avg. TAT**	**Avg. W T**	**C S**
RR	50	360.4	275.8	10
OASRR	76	328.8	244.2	6

Case: 3

Let's Take Same Processes burst time in Random order and processes at t=0 Arriving Time

As shown below

Table 5: Burst Time of Processes

Process	CPU Time
P_2	58
P_1	37
P_4	115
P_3	76
P_5	137

TQ calculated by the proposed formula
TQ = Ceil [{(∑ (Pi /n)*α}]

TQ=Ceil [{(58+37+115+76+137)/5}* α]
= (423) ÷5=84.6
= 84.6*0.9= 75.14
= ceil (75.14) =76
TQ= 76

Gantt chart Case: 3

P1	P2	P3	P4	P5	P3	P5	
0	58	95	171	247	323	362	423

Table 6: Comparative Analysis RR, OASRR algorithm (case – 3)

Algorithm	TQ	Avg. TAT	Avg. W T	C S
RR	50	292.4	207.8	10
OASRR	76	239.4	154.8	6

V. CONCLUSION AND FUTURE SCOPE

My above-proposed algorithm comparison shows that an Optimal Approach of Selecting the Time Quantum for Dynamic Round Robin has better performance than existing Round Robin in case of waiting, turnaround time and no. of context switches. This proposed algorithm gave the best optimal approach for time quantum which leads to better performance in term of context switches and avg. waiting time and turnaround time, In future work, there may another way of selecting time quantum value, which will give better performance than mine and it can be proposed for multicore processor system and also for the distributed environment.

REFERENCES

[1] Saroj hiranwal and D.r. K.C.Roy"Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice".volume 2,issue 3.

[2] 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.

[3] "Tannenbaum, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13:9780136006633, pp: 1104.

[4] "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.

[5] 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-May 2011.

[6] "Rakesh Mohanty, H.S. Behera and et. al, Design and Performance Evaluation of a new proposed Shortest Remaining Burst Round Robin(SRBRR) scheduling algorithm, Proceedings of the International Symposium on Computer Engineering and Technology(ISCET), March 2010.

[7] "Yaashuwanth .C & R. Ramesh" Intelligent time slice for round robin in a real-time operating system, IJRRAS 2 (2), February 2010

[8] Milan Milinkovic, "Operating Systems Concepts and Design", McGraw-Computer Science Series, second edition.

[9] A. Dhore "Operating Systems", Technical Publications.

[10] M. Dietel, "Operating Systems", Pearson Education, Second Edition.

[11] "Real Time Task Scheduling ", NPTEL, IIT Kharagpur
P. Balakrishna Prasad, "Operating Systems" Second Edition

Authors Profile

"Sumit Mohan" Graduated in 2016 from GCET Galgotia's College of Engineering & Technology and now pursuing his Master degree in Computer Science from Dr. APJ Abdul Kalam Technical University Lucknow (U.P.). He has also done three- year Polytechnic diploma in Computer Science. He has written in several international journals and conference. He is now active in writing papers and joining conferences.

"Prof. Rajnesh Singh" Presently working as a **Professor** and **HOD (CS)** in IEC College of Engineering and Technology, Greater Noida since 2009. He has done M.Tech from CDAC Noida GGSIPU New Delhi in 2006. He has published many research papers in the field of Networking in IEEE and 10 years of teaching experience.

