PERFORMANCE ANALYSIS OF ROUND ROBIN SCHEDULING USING ADAPTIVE APPROACH BASED ON SMART TIME SLICE AND COMPARISON WITH SRR

Vishnu Kumar Dhakad[1], Lokesh Sharma[2]
[1]M.Tech. Student & [2]Asstt. Prof. (CSE),
Sri Balaji College of Engineering & Technology, Jaipur, Rajasthan

ABSTRACT

The Time Sharing System is more multifaceted about the performance and calculating the average waiting time, turnaround time, response time, and context switches from the number of processes mainly depends on the CPU scheduling algorithm where the CPU is one of the most important computer resource and as round robin scheduling is considered most widely used scheduling algorithms. The Round Robin Scheduling Algorithm has disadvantage is longer average waiting time, higher context switches, higher turnaround time and low throughput. In this research a new proposed algorithm called Adaptive Round Robin Scheduling Algorithm is presented which is discussed here. The new proposed algorithm called "Adaptive Round Robin (ARR) Scheduling a Novel Approach Based on Shortest Burst Time Using Smart Time Slice". It is a Priority Driven Scheduling Algorithm based on burst time of processes. It is also uses fixed time quantum same as Simple RR but calculation of time quantum is another factor here. First of all we arrange the processes according to the execution time/burst time in increasing order that is smallest the burst time higher the priority of the running process. The next idea of this approach is to choose the Smart Time Slice (STC) is mainly depends on number of processes. The Smart Time Slice is equal to the mid process burst time of all CPU burst time when number of process given odd. If number of process given even then we choose the time quantum according to the average CPU burst of all running processes. The use of scheduling algorithm increased the performance and stability of the time sharing systems and support building of an self-adaptation operating system, which means that the system is who will adapt itself to the requirements of the user and vice versa.

KEYWORDS: *Round Robin, Time-quantum, CPU scheduling, Average Turnaround Time, Average Waiting Time, Context Switch, SRR, ARR.*

I. INTRODUCTION

A) CPU utilization

We want to keep the CPU as busy as possible that means CPU is not free during the execution of processes. Conceptually the CPU utilization can range from 0 to 100 percent.

(B) Response time

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

(C) Throughput:

One measure work is the number of processes that are completed per time unit that means the number of tasks per second which the scheduler manages to complete the tasks.

(D) Turnaround Time:

The time interval from the time of submission of a process to the time of completion is the turnaround time. Total turnaround time is calculation is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

(E) Waiting Time:

The waiting time is not the measurement of time when a process executes or does I/O completion; it affects only the amount of time of submission of a process spends waiting in the ready queue. We keep average waiting time should be less.

II. AIMS AND OBJECTIVE OF PAPER

Aims and objective of research is to schedule the processes in efficient and convenient way. Scheduling decision try to reduce the following: turnaround time, response time and average waiting time for processes and the number of context switches. So the main objective of research to reduce the load of CPU work, increase the performance of CPU, reduces the overhead and schedule the task in efficient manner [1].

III. METHODOLOGY

In our research work the methodology a new implementation in round robin scheduling to reduce the average waiting time, turnaround time, context switches and achieve high throughput. In round robin scheduling we are using the concept of performance evolution through intelligent time quantum based on execution time [2] [20].

IV. REVIEW OF LITERATURE SURVEY

Burst Round Robin as a Proportional Share in this paper we introduce Burst Round Robin, a proportional-share scheduling algorithm as an attempt to combine the low scheduling overhead of round robin algorithms and favor shortest jobs. [3]. Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm by assigning the processor to processes with shortest remaining burst in round robin manner using the dynamic time quantum [4].Intelligent Time Slice for Round Robin in Real Time Operating System. The main objective of this paper is to develop a new scheduling algorithm for scheduling tasks in real time operating systems. [5]. Self-Adjustment Time Quantum in round robin is the new proposed algorithm called Self-Adjustment-Round-Robin (SARR) based on a new approach called dynamic-time-quantum [6].

V. THE PROPOSED WORK

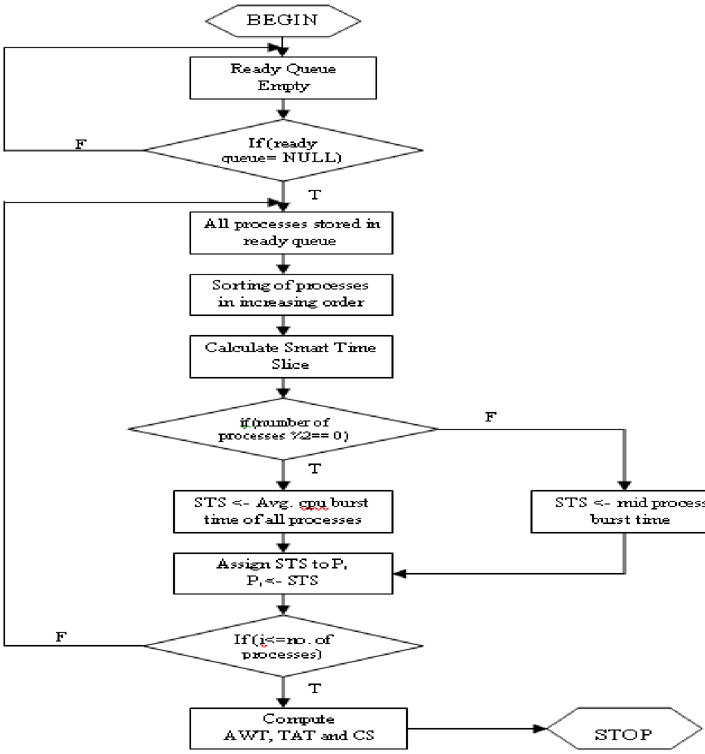
In our proposed algorithm, the number of processes is residing in the ready queue, we assume their arrival time is assigned to zero and burst times are allocated to the CPU. The burst time and the number of processes (n) are accepted as input. Now first of all we arrange all processes in increasing order according to their given burst time and chose smart time slice according some conditions. The smart time slice will be depends on the inputting number of processes burst time. If number of processes are vary then smart time will be vary.

VI. ADAPTIVE RR PSEUDO CODE

1. First of all check ready queue is empty
2. When ready queue is empty then all the processes are assigned into the ready queue.
3. All the processes are rearranged in increasing order that means smaller burst time process get higher priority and larger burst time process get lower priority.
4. While (ready queue! = NULL)
5. Calculate Smart Time Slice:
   If (Number of process%2= = 0)
   STS = average CPU burst time of all processes
   Else
   STS = mid process burst time
6. Assign smart time slice to the i^{th} process:
   Pi ← STS
   i=i+1

7. If (i< Number of process) then go to step 6.
8. If a new process is arrived update the ready queue, go to step 2.
9. End of While
10. Calculate average waiting time, turnaround time, context switches.
11. End

Flowchart for Adaptive RR:



VII. SIGNIFICANCE OF RESEARCH WORK

The significance of our performance metrics for experimental analysis is as follows:

1) Number of Context Switches should be less.

2) Average Waiting time (AWT) should be less.

3) Turnaround Time (TAT): should be less.

4) Throughput: will be high

VIII. FORMULATION OF RESEARCH WORK

The algorithm works effectively even if it used with a very large number of processes. In each case, we have compared the experimental results of our proposed algorithm with the round robin scheduling algorithm with fixed time quantum Q. Here we have assumed a constant time quantum Q equal to 25 in this case.

Case 1: We Assume five processes arriving at time = 0, with increasing burst time (P1 = 14, P2 =45, P3 = 36, P4 = 25, P5= 77) as shown in Table 1. The Table 1 shows the output using SRR algorithm and our ARR algorithm. Figure-1 and Figure-2 shows Gantt chart for both the algorithms respectively.

Table 1 Example of RR and ARR

Process	Arrival Time (ms)	Burst Time (ms)
P1	0	14
P2	0	45
P3	0	36
P4	0	25
P5	0	77

P1	P2	P3	P4	P5	P2	P3	P5	P5	P5

0 14 39 64 89 114 134 145 170 195 197

Figure 7.1 Gantt chart for SRR

AWT = 70.2
TAT = 109.6
CS = 7

According ARR mechanism:

First of all we arrange the processes in ready queue according their given burst time in increasing order that is P1=14, P4=25, P3=36, P2=45 and P5=77 and after that we choosing the time quantum according my algorithm, the time quantum is 36. All the processes have arrival time 0.

P1	P4	P3	P2	P5	P2	P5	P5

0 14 39 75 111 147 156
192 197

Figure 7.2 Gantt chart for Proposed RR

AWT = 56.8
TAT = 96.2
CS = 6

Table 2 Comparison of SRR and ARR

Algorithm	STS	CS	AWT	TAT	Throughput
Simple RR	25	7	70.2	109.6	low
ARR	36	6	56.8	96.2	high

Case 2: We Assume four processes arriving at time = 0, with random burst time (P1 = 20, P2 =32, P3 = 9, P4 = 19) with time quantum =16 as shown in Table 4.11. The Table 4.12 shows the output using RR algorithm and Adaptive RR algorithm. Figure 4.14 and Figure 4.15 shows Gantt chart for both the algorithms respectively.

Table 3 Example 6 of RR and ARR

Process	Arrival Time (ms)	Burst Time (ms)
P1	0	20
P2	0	32
P3	0	9
P4	0	19

P1	P2	P3	P4	P1	P2	P4

0 16 32 41 57 61
77 80

Figure 7.3 Gantt chart for SRR

AWT = 44.78
TAT = 64.7
CS = 6
According Adaptive RR mechanism:

First of all I arrange the processes in ready queue according their given burst time in increasing order that is P3=9, P4=19, P1=20 and P2=32 and after that we choosing the time quantum according Adaptive RR algorithm, the time quantum is the average processes burst time if the given processes are even, that is 20.The Gantt chart for Adaptive RR

P3	P4	P1	P2	P2

0 9 28 48 68 80

Figure 7.4 Gantt chart for ARR

AWT = 21.2
TAT = 41.2
CS = 3

Table 4 Comparison of simple RR and ARR

Algorithm	Time Quantum	CS	AWT	TAT	Throughput
Simple RR	16	6	44.78	64.7	low
ARR	20	3	21.2	41.2	high

IX. RESULTS

Based on the experiments and calculations that we have made the new modified algorithm which solves the drawbacks of round robin scheduling which is considered a challenge.

Performance Chart:

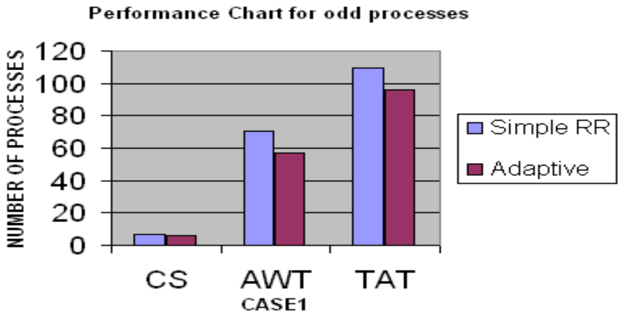


Figure 8.1

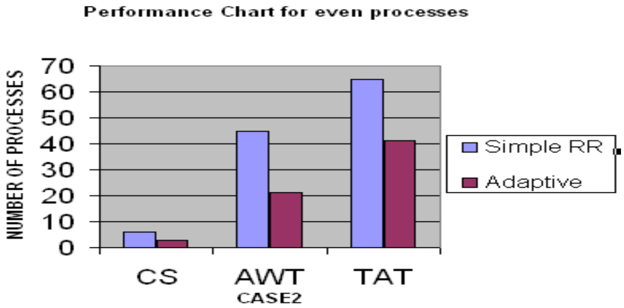


Figure 8.2

X. CONCLUSION

The use of scheduling algorithm increased the performance and stability of the operating system and support building of an self-adaptation operating system, which means that the system is who will adapt itself to the requirements of the user and not vice versa. This paper reduces the drawbacks like higher average waiting time, higher turnaround time and more context switches of Simple Round Robin Scheduling algorithm which is used for the time sharing system. This paper also discovers the high throughput of the system. When throughput is high then the system performance will increase. This Paper is useful in the future with the arrival time of the jobs.

REFERENCES

[1] "Silberschatz, A., P.B. Galvin and G. Gagne, 2004" Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA., ISBN: 13: 978-0471694663, pp: 944.

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

[3]"Tarek Helmy, Abdelkader Dekdouk" *Burst Round Robin: As a Proportional-Share Scheduling Algorithm*, IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations, pp. 424-428, 11-14 November,2007.

[4] "Prof. Rakesh Mohanty, Prof. H. S. Behera, Khusbu Patwari, Manas Ranjan Das, Monisha Dash, Sudhashree" Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm, Am. J. Applied Sci., 6 (10): 1831-1837, 2009.

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

[6] "Rami J. Matarneh" *Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the now Running Processes*, Department of Management Information Systems, American Journal of Applied Sciences 6 (10):1831- 1837, 2009, ISSN 1546-923

[7] "Shreedhar, M.; Varghese,G. (October 1995)" Efficient fair queueing using deficit round robin .ACM SIGCOMM Computer Communication Review 25 (4): 231. Doi:10.1145/21739.

[8]"Tong, W. and J. Zhao, 2007" Quantum varying deficit round robin scheduling over priority queues. Proceedings of the International Conference on Computational Intelligence and Security, Dec. 15- 19, Computer Society, Harbin, China, pp: 252-256.

[9] "Back, D.S., K. Pyun, S.M. Lee, J. Cho and N. Kim, 2007" A hierarchical deficit round-robin scheduling algorithm for a high level of fair service. Proceedings of the International Symposium on Information Technology Convergence, Nov. 23-24, IEEE Computer Society, Washington DC., USA., pp: 115-119.

[10] "Zhao, W. and J.A. Stankovic, 1989" Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems. Proceeding of the IEEE Real-Time Systems Symposium, Dec. 1989, pp: 156-165.
[11] "Lupetti, S. and D. Zagorodnov, 2006" Data popularity and shortest-job-first scheduling of network transfers. Proceeding of the International Conference on Digital telecommunications, Aug. 29 EEE Computer Society, USA, pp: 26-26. DOI: 10.1109/ICDT.2006.28.
[12] "Sandmann, W., 2006' Benefits of alternating FCFS/SJF service order. Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Aug. 18-20,World Scientific and Engineering Academy and Society, Stevens Point, Wisconsin, USA., pp: 194-199.
[13] "Caprita, B., W.C. Chan, J.N.C. Stein and H. Zheng, 2004" Group ratio round-robin: O(1) Proportional Share scheduling for unprocessed and multiprocessor systems. Columbia University, Technical Report CUCS-028-04.
[14] "Place, J., 1989" FCFS": A novel scheduling policy for tightly-coupled parallel computer systems. Proceeding of the ACM Annual Computer Science Conference, Proceedings of the 17th ACM Annual Conference on Computer Science, Feb. 21-23, ACM Press, Louisville, Kentucky, pp: 188-194. DOI: 10.1145/75427.75450
[15]"H. M. Deitel" An introduction to operating systems. Published by Addison-Wesley,Inc. ISBN 0-201-18038-3, 1990.
[16]"T. A. Funkhouser, C.H. Sequin" Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. Proceedings of SIGGRAPH'93, pages 247-254, 1993.
[17] "P. S. Heckbert, M. Garland" Surface Simplification Using Quadric Error Metrics. Proceedings of SIGGRAPH'97, pages 43-50, 1997.
[18] "H. Hoppe" Efficient Implementation of Progressive Meshes. Computers & Graphics,1998.
[19] "M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, S. Zeswitz" NPSNET: A Network Software Architecture for Large-Scale Virtual Environments. Presence, 3(4): 265-287, 1994.
[20] "W. Stallings" Operating systems Published by Prentice-Hall, Inc. ISBN 0-02-415493-8, 1995.

Authors:

Vishnu Kumar Dhakad was born on 21 July 1982. He is a M. Tech. (CSE) student in Sri Balaji college of Engineering and Technology, Jaipur, Rajasthan, He has completed B.E.(CSE) in 2006 from University of Rajasthan, Jaipur. He has experienced in education. His research area is in Real Time System, Operating System, Distributed System and Expert System**.**

Lokesh Sharma is working as assistant professor in Sri Balaji College of Engineering and Technology Jaipur. He has done his B.E.(IT) in 2005 from UOR, Jaipur and M.Tech.(CSE) in 2010 from RTU, Kota and pursing his Ph. D in the field of Wireless Ad-hoc Network. He has more than 6-years of teaching experience. He has also work done in Amity University. His research areas are operating system, wireless network and sensor network.