Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 4 (2017) pp. 639-644

http://www.ripublication.com

Improved Round Robin Scheduling in Cloud Computing

Priyanka Sangwan[1]
Scholar Mtech, Computer Science, Mody University, Lakshmangarh, Sikar, Rajasthan, India.

Manmohan Sharma[2]
Department of Computer Science, Mody University, Lakshmangarh, Rajasthan, India.

Anil Kumar[3]
Department of Computer Science, Mody University, Lakshmangarh, Rajasthan, India.

Abstract

The fundamental and imperative part of operating system is multiprogramming. It is a strategy for executing various procedures all the while in the memory. RR CPU is ideal algorithm in timeshared frameworks. The implementation of the CPU relies on the mode we decide time quantum in the systems relying on timeshared environments. We have roll out an progress in RR algorithm so that the carrying out of CPU can be moved forward and compared it with original RR scheduling and displayed the consequences of why our model is efficient than the original one. The goal of this term paper is to alter original round robin in a way which will perform superior to anything the straightforward Round Robin as far as limiting average waiting and turnaround_time.

Keywords: Time quantum, turnaround_time, waiting time, round robin

I. INTRODUCTION

Cloud basically can be renamed as a type of parallel and disseminated framework which comprises of an assembly of interrelated and virtualized PCs. The customers may get to information & applications of the Cloud from anyplace whenever [1]. The fundamental advantage of scheduling is to accomplish a superior performance and brilliant system throughput.

RR works using time slice which can be explained as a small entity of time. The job of CPU scheduling process is to circumvent the Ready Queue which distributes the CPU to individual process for a certain period in term of TQ. Performance of any scheduling algorithm can be calculated via following criteria i.e., Turnaround_time, Waiting_time,

Response_time, CPU use, and throughput. TAT can be defined as the time interim from the accommodation time of a procedure to the completion time of a procedure. Waiting_time is the totality number of periods spent holding up in the organized queue. The time from the accommodation of a procedure until the primary reaction is called Response_time. The CPU exploitation is the rate of time CPU stays occupied. The extent of procedures finished in a unit time is called Throughput. Context switch is the measure of times the process changes to get implement. This algorithm can be upgraded by minimizing reaction time, WT and TAT and by amplifying CPU use, throughput. Whatever is left of the paper is sorted out as takes after: In area 2, we describe the related works with an extraordinary emphasis on working. Segment 3 shows the proposed model. In area 4, we evaluate the implementation of our proposed calculation. In area 5 we investigate the results acquired from our examination. Segment 6 gives the closing comments

.

II. RELATED WORKS

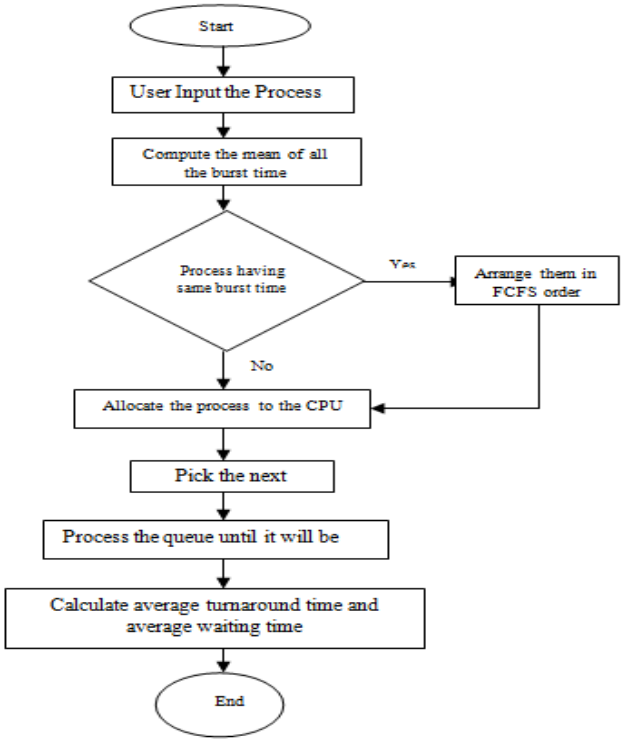
The RR algorithm has demerit that it utilizes static TQ. Studies are done to improve the implementation of the RR scheduling. Portions of the critical works are recorded below.

In [2] an algorithm was proposed which (AAIRR) concentrates on enhancing the RR CPU algorithm. The calculation by lessens the WT and TAT radically contrasted with the basic RR calculation. Its implementation takes into account three phases which gives optimal throughput.

Rami J. Matarneh [3] described a calculation SARR to improve the execution of RR. In SARR for every round the median is computed for all the procedures burst time and utilized as time quantum. H.S.Behera et al. [4] additionally utilize comparable kind of algorithm. But here, they reschedule the procedure at the time of execution. It chooses the procedure possessing least burst time, then process having maximum burst time, then process with second most minimal burst time, et cetera.

III. PROPOSED WORK

The proposed CPU Scheduling calculation relies on implementing change in round-robin scheduling calculation. It diminishes the WT and TAT radically contrasted with the straightforward RR scheduling. Rather than giving static TQ in the CPU scheduling, our algorithm figures the TQ itself. Basically, this is a comparison paper where originally RR scheduling is compared with the one which possesses certain changes. Initially we will keep all the procedures in the arbitrary request of CPU as they arrive. At that point in the subsequent stage the algorithm figures the mean of the burst_time of the considerable number of procedures. After ascertaining the mean, it will define the TQ powerfully i.e. average of mean. In the last stage the algorithm selects the foremost procedure from the queue and allots the CPU for a period interim of the mean TQ.



The steps of the proposed algorithm are as follow.

1. START.
2. Keep the procedures as they turn up in the prepared queue.
3. Calculate the Mean of the CPU burst_time of the considerable number of processes.
4. Set the mean value as the TQ for every procedure.
5. Allocate CPU to the first process waiting in the ready queue for the duration of TQ.
6. If the remaining burst_time of the present procedure is more prominent than the time quantum, expel the present procedure from the organized queue and put it on the end of the queue for further execution.
7. Pick the next procedure which is already waiting in the ready queue and assign the CPU to it up to the duration of the TQ and then again go to the step 6.
8. Process the queue until it will be vacant.
9. Calculate the Average waiting_time and Average Turnaround_time of all process.
10. END.

IV. EXPERIMENTAL ANALYSIS

Here for the sake of evaluating the results of this algorithm it is implicit to use an environment where the entire calculations are performed in a solitary processor. In addition all the procedures possess the identical priority i.e. they are arranged in the queue as they come, no particular strategy is followed in this algorithm. The proposed work is executed in java programming dialect. Various quantities of experiments are likewise completed and their consequences are taken into consideration for releasing the final statements.

CASE 1

Here 5 processes are considered where the arrival time, burst time and the TQ are specified prior to the execution.

Table 1: Processes Specifications

Time Quantum (TQ) = 10 ms		
Process Name	Arrival Time	Burst Time
P1	0	13
P2	0	12
P3	0	22
P4	0	32
P5	0	21

According to original RR Scheduling

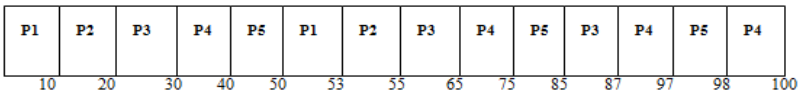


Fig 1: Gantt chart for RR in Table 1

Average Waiting_Time: 58.6
Average Turnaround_Time: 78.6
After implementing the Proposed Model, the average waiting_time and average turnaround_time is:

Time Quantum (TQ) = Average of Burst Time			
Process Name	Burst Time	Waiting Time	Turn Around Time
P1	13	0	13
P2	12	13	25
P3	22	65	87
P4	32	67	99
P5	21	79	100

Fig 2: Results according to Proposed approach

Average Waiting_Time: 44.8
Average Turnaround_Time: 64.8

CASE 2:
Again 5 processes are taken into account for calculation where time quantum is externally provided.

Table 2: Process Specifications

Time Quantum (TQ) = 20 ms		
Process Name	Arrival Time	Burst Time
P1	0	40
P2	0	35
P3	0	80
P4	0	45
P5	0	25

According to original RR Scheduling

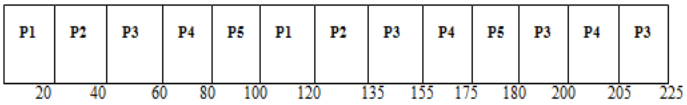


Fig 3: Gantt chart for RR in Table 2

Average Waiting_Time: 128.0
Average Turnaround_Time: 173.0

According to the Proposed Model, the average waiting_time and average turnaround_time is:

Time Quantum (TQ) = Average of Burst Time			
Process Name	Burst Time	Waiting Time	Turn Around Time
P1	40	0	40
P2	35	40	75
P3	80	145	225
P4	45	120	165
P5	25	165	190

Fig 4: Results according to Proposed approach

Average Waiting_Time: 94.0
Average Turnaround_Time: 139.0

V. COMPARISON OF RESULTS

Performance Attribute	Original RR Scheduling	Proposed Model	Remark
Average Waiting Time	58.6	44.8	13.8 Units of Time Saved
Average Turnaround Time	78.6	64.8	13.8 Units of time Saved

Table 4: Results for Case 1

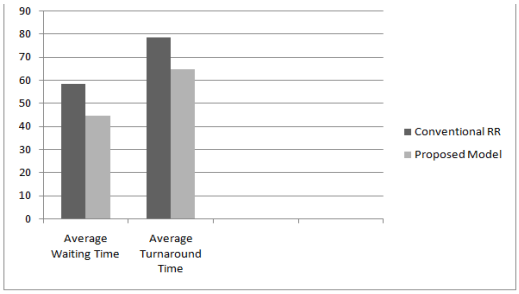


Fig 7: Performance Comparison for Case 1

Performance Attribute	Original RR Scheduling	Proposed Model	Remark
Average Waiting Time	128.0	94.0	34 Units of Time Saved
Average Turnaround Time	173.0	139.0	34 Units of time Saved

Table 5: Results for Case 2

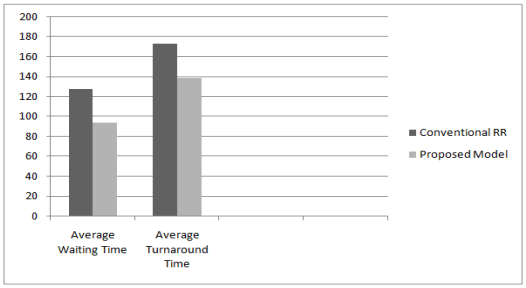


Fig 8: Performance Comparison for Case 2

VI. CONCLUSION AND FUTURE SCOPE

We have compared the original round robin with our proposed model and displayed the consequences. Here an enhanced version of RR is proposed. From the graphs and calculations it was proved that the best/worst case of original algorithm is equal to the worst_case of proposed algorithm. The graphs and consequences speaks that the projected algorithm is much better than conventional RR when contrasted with the cases of average WT and TAT. RR scheduling can be used in cloud computing in balancing the load at the earliest as response time gets reduced effectively.

In this paper, only the average waiting_time and average turnaround_time is compared. No spot light has been given on number of switch_cases used in executing the algorithm. In future, we will enhance the work using this attribute too and will fetch the consequences as they will appear.

REFERENCES

[1] Hitoshi Matsumoto, Yutaka Ezaki," Dynamic Resource Management in Cloud Environment", July 2011, FUJITSU science & Tech journal, Volume 47, No: 3, page no: 270-276.

[2] Abdulraza, Abdulrahim,Salisu Aliyu, Ahmad M Mustapha, Saleh E Abdullahi, " An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 2, February 2014 ISSN: 2277 128X

[3] Rami J. Matarneh (2009) "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes", American Journal of Applied Sciences, pp 1831-1837.

[4] H.S.Behera, R. Mohanty, Debashree Nayak, (2010) "A New Proposed Dynamic Quantum with Readjusted Round Robin Scheduling Algorithm and Its Performance", International Journal of Computer Applications, pp 10-15.

[5] Sandeep Negi, "An Improved Round Robin Approcah using Dynamic Time Quantum for Improving Average Waiting Time", International Journal of Computer Applications (0975-8887) Volume 69- No.14, May 2013.

[6] Liang Luo, Wenjun Wu, Dichen Di, Fei Zhang, Yizhou Yan, Yaokuan Mao, "A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods", 2012, 978-1-4673-2154-9/12, IEEE.

[7] Saroj Hiranwal and K. C. Roy "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice" International Journal of Computer Science and Communication Vol. 2, No. 2, July- December 2011, pp. 319-323