Enhanced Round Robin Technique for Task Scheduling in Cloud Computing Environment

Shubham Mittal
P.G. Scholar,
Graphic Era Hill University,
Dehradun, India

Swati Singh
P.G. Scholar,
Graphic Era Hill University,
Dehradun, India

Ramandeep Kaur
P.G. Scholar,
Graphic Era Hill University,
Dehradun, India

***Abstract*— An operating system is a spine for any processing gadget. Any operating system is said to be a productive working framework, when it works effectively in any condition. The effectiveness of any operating system can likewise be figured from one of its significant working regions, i.e., Process Management. A Round Robin is a sort of procedure planning calculation which is utilized by the CPU with a specific arrangement or requesting in cloud computing. Round Robin is known for its fairness, which is given by its fixed Time Quantum process. In this paper, we have presented another time varying strategy for time quantum with which a framework maximize CPU usage, as well as minimizes the waiting time, turnaround time, context switching and gives better performance with starvation free process planning.**

Keywords—Component; Formatting; Style; Styling; Insert

I. INTRODUCTION

Cloud computing is an Internet-based processing, where shared resources, information and data is provided with other computer devices on demand [12]. This model empowers omnipresent on-demand access to a mutual pool of configurable computing assets furthermore empowers every sort of the application framework to pick up the computation strength, the storage space and a wide range of programming administration as per the need. The pool of assets, in this way called "the cloud". In cloud computing the fundamental focus is on all computing assets and can be overseen automatically through the software without mediating. Cloud computing makes clients think their own particular business without stressing over the monotonous points of interest. Cloud is a model that is likewise known in the business sector for conveying data innovation administrations which is the consequence of constant advancement of information management technology, accordingly is not really the progressive late improvement. In this model, resources are recovered from the web through web based tools and applications rather than a direct connection to a server.

In distributed computing, it is essential to choose good and efficient scheduling algorithm. Keeping this in mind, various researches and experiments are done which results in the advancement of new strategies. Some of those procedures incorporate RR (Round Robin), SJF (Shortest Job First), FCFS (First start things out serve) and many more. One of the most established, critical and well known strategies of process scheduling in timesharing system is Round Robin process scheduling technique. This calculation gives fair allotment of CPU to the processes accordingly understood amongst the general population. In this procedure, allocation of CPU to every process is done for a unit time period in each round. This unit time period is known as Time Quantum.

In Round Robin Technique, the time quantum plays a critical part in planning and scheduling the process [1]. All the processes and time of CPU portion rely on upon time quantum. On the off chance that the estimation of time quantum is too little, it implies the CPU will be occupied for the majority of its time in context switching and this expand the overhead over the CPU. In the event that the time quantum is too huge, this implies process reaction time increments quickly. In the event that the time quantum is steady, then waiting time and response time of the process increments and this debases the system performance.

II. BACKGROUND

Round Robin is one of the most established, critical and prominent procedures of process scheduling in timesharing system. This algorithm is known for its fair assignment of CPU to the processes. In round robin scheduling approach, the selection of time quantum is a critical part. Performance of a system is totally relies on the determination of the time quantum. A percentage of the essential ideas in Round Robin are [2]:

- Time Quantum: Round Robin is known for its fairness as a result of the time quantum procedure. Time Quantum is a period interim for the execution of all procedure in each round. It demonstrates for to what extent a specific procedure will be allotted to the CPU. As time quantum terminates, the CPU acquire to alternate procedure.
- Context Switching: It is a method which is carried out by the CPU to switch with one errand then onto the next and storing the state of an active process. Context switching guarantees that tasks don't get conflict and the execution of preempted procedure can be continued from the same point later.
- CPU Utilization: It quantifies for to what extent CPU was occupied in the execution of the procedure. We will probably make a greatest usage of the CPU.

- Throughput: Total number of processes finished in a unit time is utilized to quantify the throughput of the framework. Since, round robin gives an equivalent measure of time to every procedure, so more often the throughput is entirely low.
- Waiting Time: It is a period measure for which a specific process holds up in a ready queue.
- Turnaround Time: Total time interim from the accommodation of the process to the time of its culmination.
- Response Time: The time interim from the accommodation of the procedure to the first occasion when it gets the CPU for its execution.

In this paper, we have proposed another procedure over the current round robin process planning approach. We have sorted out this paper in different areas. In Section 3, we talked about the current work done. We have discussed our proposed method in Section 4. In Section 5, we portray our calculation in point of interest with the assistance of an illustration and further comparison is done with the current methodologies. At last, we have concluded our paper in Section 6.

III. RELATED WORK

Operating system follows FCFS (First Come First Serve) procedure to execute the process present in its ready queue. As indicated by the textbooks, [9][10][11] objective of multi-programmed operating system is to augment CPU usage. To accomplish this, the CPU switches between different processes make them keep running in parallel. In this way, we can conclude that Round Robin is an expansion of FCFS notwithstanding preemption and time-slice idea [8]. In cloud computing, selection of a good scheduling algorithm is essential. Remembering this, various researches have experienced numerous experiments and concocted the new methods. In this segment we will examine some of them.

Abbas Noon et al. [6] proposed a technique in which an ideal time quantum can be computed by the mean of burst time for the arrangement of processes in the ready queue. The mean is calculated for the present tasks in ready queue after every cycle. By the utilization of this methodology, they have acquired a decent result. The formula used to acquire time quantum for every interim is:

Time Quant = (Sum of remaining BT) / Number of Tasks

D. Praveen Kumar et al. [3] concocted another technique of integer programming (IP). This method chooses/calculates an integer number that is neither vast nor too little. This choice/calculation gives a reasonable response time to every process and subsequently the throughput of the system is not influenced by unnecessary context switches. This technique is carried on time quantum in each round over the cyclic queue.

Raman et al. [4] presented another calculation which executes the shortest job first rather than FCFS amid the round robin algorithm. They organized the processes in ascending order as indicated by their burst time present in the ready queue. The CPU is allotted to the process utilizing RR scheduling with time quantum esteem equivalent to the burst time of the first process in the ready queue. After first cycle, time quantum is updated by various calculations as:

Time Quant = Minimum BT of the first task.

Updated Time Quant = if{mean>median}

sqrt((Mean * Highest BT) + (Median * Lowest BT))
if {median > mean}
sqrt((median * Highest BT) + (mean * Lowest BT))

Debashree Nayak et al. [5] enhanced the RR algorithm by taking reasonably the dynamic time quantum and the ordering of processes. They arranged the procedures in ascending order as indicated by their burst time present in the ready queue. At that point, the ideal time quantum is computed as:

Time Quant = (Highest BT + Median) / 2

IV. PROPOSED WORK

In time sharing system, the most prominent and broadly utilized process scheduling methodology is round robin. The foundation of this prominent algorithm is its time quantum method, which is not just in charge of the adjustment in the system performance, additionally gives CPU to every process to equivalent and fair time. This methodology made this algorithm known for its fairness procedure. System performance is totally depended over the fair selection of the time quantum. On the off chance that the time quantum is bigger, then waiting time and response time of procedure get increments or if time quantum is little, then greatest time the CPU will be occupied within context switching with one process then onto the next.

So as to increase the system performance, initially, we have to roll out an improvement in the way of the time quantum and second, we have to upgrade time quantum with an enhanced worth which don't expand response time of the process. In our proposed approach, we have made the time quantum dynamic, as well as clubbed the approach with two extra existing procedures, i.e., SJF (Shortest Job First) and RASA (Resource Allocation Scheduling Algorithm). SJF orchestrate all the procedures present in the prepared line so as to their burst time, and RASA helps in making the system starvation free and diminish the response time for the bigger task.

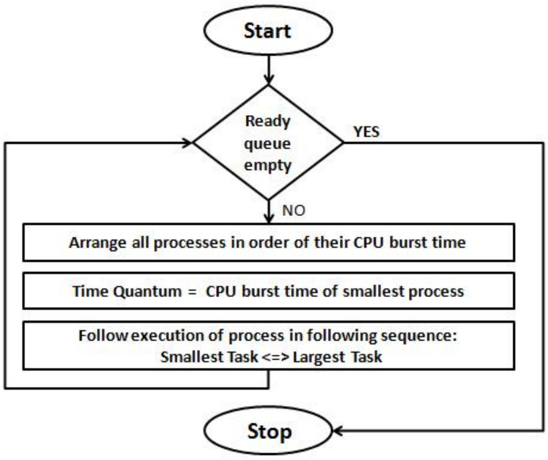
Start
Ready queue empty
YES
NO
Arrange all processes in order of their CPU burst time
Time Quantum = CPU burst time of smallest process
Follow execution of process in following sequence:
Smallest Task <=> Largest Task
Stop

Fig1. Improved Technique for Round Robin (Flowchart)

As per our proposed round robin approach, process present in the ready queue is organized with keeping in mind the end goal to their burst time with the assistance of SJF strategy. Time quantum is redesigned with the estimation of burst time of the smallest task in the queue. Time quantum is updated with the same methodology after each effective round. Since all process are organized so as to their burst time, so it is evident that the largest task will be toward the end of the queue whether it arrives first in the queue. RASA method executed smaller and larger process alternatively and helps in decreasing the response time of the system.

Figure 1 speaks to the flowchart of the proposed round robin process scheduling approach and figure 2 speaks to the pseudo code of the proposed procedure.

```
1. While ready queue is not empty
2.     Arrange all the processes present in the ready
       queue in order of their burst time.
3.     Set the time quantum equals to the CPU burst
       time of the smallest task.
4.     Execute the process up to time quantum
       expires for each round.
5.     Follow the following sequence in order to
       execute the processes:
       Smallest Task <=> Largest Task
6. End while
```

Fig2. Pseudo code for proposed algorithm

The blend of three methods, i.e., RR, SJF and RASA results in fair allocation of CPU to the processes, the decrement in their response time and waiting time makes the system starvation free. This results in the augmentation of the system performance.

V. IMPLEMENTATION AND EXPERIMENTATION

A. *Example and Analysis:*

Let us assume a meta-task containing five different tasks, let say T1, T2, T3, T4 and T5 with their burst time and arrival time as appeared in table 1. Arrival time demonstrates that only one task, i.e., T1 is the only task which is available at first and the rest are arriving at various time interims.

TABLE 1: PROCESS SPECIFICATION

Task	Burst Time	Arrival Time
T1	23	0
T2	75	20
T3	93	22
T4	48	50
T5	2	55

Here, on the off chance that we apply the traditional Round Robin algorithm with fixed time quantum of 20, every procedure will get the CPU for a period quantum of 20 units in each round.

T1	T2	T1	T3	T2	T4	T5	T3	T2	T4	T3	T2	T4	T3
20	40	43	63	83	103	105	125	145	165	185	200	208	241

Fig 3. Gantt chart for the Round Robin approach

Once the time quantum is chosen, it is made fixed until every process present in the ready queue is not completely executed.

In the same situation, in the event that we apply our proposed improved round robin method, the time quantum gets upgraded after each round. Presently time quantum is not any more static in nature. The procedures are executed as:

T1	T2	T3	T5	T4	T3	T3	T4	T4
23	98	173	175	177	179	195	211	241

Fig 4. Gantt chart for the proposed approach

From the above gantt charts, we can compute the turnaround time and waiting time for every procedure with the following formulas:

Average Turnaround Time can be calculated as:

Average Turnaround Time = $\sum$ (Finish time of Process P_i – Arrival time of Process P_i) / Number of Process

Average Waiting Time can be calculated as:

Average Waiting Time = $\sum$ (Turnaround time of process P_i – Burst time of Process P_i) / Number of processes

TABLE 2: AVERAGE TURNAROUND TIME

Task	Round Robin	Proposed Technique
T1	43-0=43	23-0=23
T2	200-20=180	98-20=78
T3	241-22=219	195-22=173
T4	208-50=158	241-50=191
T5	105-55=50	175-55=120
Total	650/5=130	585/5=117

TABLE 3: AVERAGE WAITING TIME

Task	Round Robin	Proposed Technique
T1	43-0=43	23-0=23
T2	200-20=180	98-20=78
T3	241-22=219	195-22=173
T4	208-50=158	241-50=191
T5	105-55=50	175-55=120
Total	650/5=130	585/5=117

Applying a traditional round robin method, it not just expands the waiting time and response time of the processes; additionally degrade system performance by expanding context switching of the process which results in minimizing the CPU utilization. With a specific end goal to accomplish higher throughput, we need to minimize the waiting time and response time of the processes and diminishing the context switching may prompt the better CPU utilization. This can be accomplished by changing static time quantum to alterable as done in our proposed algorithm technique.

B. Order of Execution

Keeping in mind the end goal to show the working of our proposed round robin technique, we will be utilizing the above case. Here, we speak to the regulated working of our proposed method.

TABLE 4: STEP-BY-STEP WORKING OF PROPOSED TECHNIQUE

Time	Ready Queue	Time Quantum	Comments
0	T1	23	T1(23) arrives, run
20	T1, T2		T2(75) arrives and is appended to the queue, T1(3) continuous to run
22	T1, T2, T3		T3(93) arrives and appended to the queue, T1(1) continuous to run
23	T2, T3	75	T1(0) finishes, so T2(75) runs
50	T2, T3, T4		T4(48) arrives and is appended to the queue, T2(48) continuous to run
55	T2, T3, T4, T5		T5(2) arrives and is appended to the queue, T2(43) continuous to run
98	T3, T4, T5		T2(0) finishes, so T3(93) runs
173	T5, T3, T4	2	Time quant expire, T5(2), T3(18), T4(48) arranged in ascending order, smallest task T5(2) runs
175	T4, T3		T5(0) finishes, so largest task T4(48) runs
177	T3, T4		Time quant expires, T4(46) appended to the queue, smallest task T3(18) runs
179	T3, T4	16	Time quant expire, T3(16), T4(46) arranged in ascending order, smallest task T3(16) runs
195	T4		T3(0) finishes, so T4(46) runs
211	T4	30	Time quant expire, T4(30) runs
241	-		T4(0) finishes, all process executed

C. Simulation And Experimental Results

We have performed the recreation with Java 7 Technology. Here, we attempted to assess and contrast our proposed strategy and a portion of the current procedures.

Table 1 demonstrates different tasks present in the ready queue at present. In the event that we apply a percentage of the present procedures over this situation and contrast the same and our proposed approach, procedure will be executed in the accompanying way:

1. Round Robin [9][10][11]:

T1	T2	T1	T3	T2	T4	T5	T3	T2	T4	T3	T2	T4	T3
20	40	43	63	83	103	105	125	145	165	185	200	208	241

Fig 5. Gantt chart for the Round Robin approach

2. Dynamic Quantum Using the Mean Average [6]:

T1	T2	T3	T4	T5	T3	T4
23	98	182	202	204	213	241

Fig 6. Gantt chart for the Dynamic Quantum Using the Mean Average

3. Integer Programming [3]:

T1	T2	T3	T4	T5	T2	T3
23	61	99	147	149	186	241

Fig 7. Gantt chart for the Integer Programming

4. Efficient Dynamic Round Robin [4]:

T1	T2	T3	T5	T4
23	98	191	193	241

Fig 8. Gantt chart for the Efficient Dynamic Round Robin

5. Improved Round Robin [5]:

T1	T2	T3	T5	T3	T4	T4
23	98	187	189	193	219	241

Fig 9. Gantt chart for the Improved Round Robin

6. Enhanced Round Robin (Proposed Technique):

T1	T2	T3	T5	T4	T3	T3	T4	T4
23	98	173	175	177	179	195	211	241

Fig 10. Gantt chart for the Enhanced Round Robin

TABLE 5: AVERAGE TURNAROUND TIME, AVERAGE WAITING TIME & CONTEXT SWITCH FOR VARIOUS TECHNIQUES

Comparison	Technique					
	1	2	3	4	5	6
Average Turnaround Time	130	126.4	119.8	119.8	119.4	117
Average Waiting Time	81.8	78.2	71.6	71.6	71.2	68.8
Context Switching	13	6	6	4	5	6

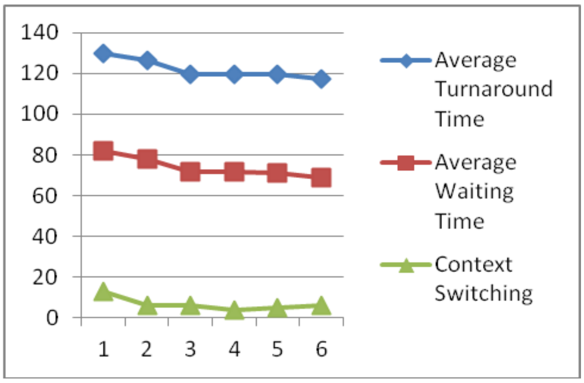

Fig 11. Comparison of Average Turnaround Time, Average Waiting Time and Context Switching amongst various techniques

From the above example, experiments and result correlations, we can conclude that our proposed technique not just minimizes the waiting time and turnaround time additionally diminishes the superfluous context switching which amplify the CPU utilization, as well as expansion the system performance.

VI. CONCLUSION AND FUTURE ENHANCEMENT

Round robin process scheduling methodology is not just broadly known for its fairness approach in computing environment, additionally known for its starvation free performance. Time quantum not just plays a vital part in giving fairness amongst processes additionally influences the system performance. As per our methodology, we have not just makes the time quantum dynamic in nature, additionally attempted to diminish the waiting time and response time of the processes. Each round gives assurances for the fulfillment of one process and additionally decrement in the remaining time of the largest process.

Further, this work can be extended by executing this strategy in a real distributed computing environment under various parameters and conditions and measure the system performance in genuine.

ACKNOWLEDGMENT

We might want to introduce our earnest on account of Ms. Avita Katal, Assistant Professor, University of Petroleum and Energy Studies, for her empower and consistent support. Other than her, we might want to display our respects to our loved ones who have upheld us all through the exploration work.

REFERENCES

[1] Sandeep Negi and Priyanka Kalra, "A Comparative Performance Analysis of Various Variants of Round Robin Scheduling Algorithm," International Journal of Information & Computation Technology, ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 765-772.

[2] Abdulrazaq Abdulrahim, Saleh E Abdullahi and Junaidu B. Sahalu, "A New Improved Round Robin (NIRR) CPU Scheduling Algorithm," International Journal of Computer Applications (0975–8887), Volume 90–No 4, March 2014.

[3] D Praveen Kumar, T. Sreenivasula Reddy and A. Yugandhar Reddy, "Finding Best Time Quantum for Round Robin Scheduling Algorithm to avoid Frequent Context Switch," International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6750-6754, ISSN: 0975-9646.

[4] Raman and Dr. Pardeep Kumar Mittal, "An Efficient Dynamic Round Robin CPU Scheduling Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014, ISSN: 2277-128X.

[5] Debashree Nayak, Sanjeev Kumar Malla and Debashree Debadarshini, "Improved Round Robin Scheduling using Dynamic Time Quantum," International Journal of Computer Applications (0975–8887), Volume 38–No.5, January 2012.

[6] Abbas Noon, Ali Kalakech and Seifedine Kadry, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average," International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011, ISSN: 1694-0814

[7] Samih M. Mostafa, S. Z. Rida & Safwat H. Hamad, “Finding Time Quantum of Round Robin CPU Scheduling Algorithm In General Computing Systems Using Integer Programming,” International Journal of Research and Reviews in Applied Sciences, IJRRAS 5 (1), October 2010.

[8] Rakesh Kumar Yadav, Abhishek K Mishra , Navin Prakash and Himanshu Sharma, "An Improved Round Robin Scheduling Algorithm for CPU scheduling," International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 1064-1066, ISSN: 0975-3397.

[9] Operating Systems Sibsankar Haldar, 2009, Pearson Education, India.

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

[11] Silberchatz, Galvin and Gagne, 2003 operating systems concepts, (6th edition, John Wiley and Sons).

[12] http://luitinfotech.com