International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Time Efficient Round Robin Job Scheduling (NARR) in Cloud Computing

Afsarjahan, Ritika Mehra, Rakesh Saini

Abstract: ***Cloud computing simply means the advancement of distributed computing which takes data processing computational aspects over networks to centralized high-power data centers. It refers to the use of a centralized pool of resources that are distributed on a pay-per-view model to a large number of customers. This requires scheduling algorithms that allow us to define which task processes, among which resources are first allocated for performance. The Round Robin (RR) scheduling algorithm is the common scheduling algorithm used in multitasking and real-time environments. Its performance is largely determined by the amount of time it takes to carry out a specific task assigned by the CPU. If the less time is chosen the context switch is high and the higher time is chosen, the first-come first-server (FCFS) is selected. System's performance thus totally depends on the optimal quantum time to choose from. In this paper, I'm examining a different way of improving the performance of the round robin scheduling algorithm by means of a dynamic time quantum and comparison of different performance.***

Keywords: ***Round robin algorithm, cloud computing, dynamic time quantum.***

I. INTRODUCTION

Cloud Computing is the popular application processing model. It provides a large scale of resources for high-performance computing. The Cloud Computing is a way of accessing, saving and manipulating data on the web without wasting much of your computer system's memory. This technique reduces the workload at the user's end. Basically, in cloud, we have a large number of resources distributed over the Internet and we can use these resources using the pay- as- you- go model by paying a certain amount. Cloud computing is a new distributed computing paradigm that provides services in infrastructure, platforms and software. Whenever we want to perform a cloud task, we can say that our primary problem is job scheduling and load balance.
CPU scheduling is one of the key features of a multi-tasking operating system (OS). CPU scheduling means that the process to be executed has sufficient resources. The policy for CPU scheduling

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Afsar Jahan*, Assistant Professor, Department of Computer Applications Doon University Dehradun (Uttarakhand) India.
Dr. Ritika, Professor of Computer Science, Head, Department of Computer Applications, DIT University Dehradun (Uttarakhand) India.
Dr. Rakesh Saini, Assistant professor (senior grade) Department of Computer Science and Application, DIT University Dehradun (Uttarakhand) India.

should maximize the CPU performance. This efficiency includes maximizing performance, reducing context interruption, average waiting time (AWT) and average turnaround time (ATT). The number of processes per unit cycle can be defined. The process waiting time of a process is the time it takes before it is carried out in a ready queue. Turnaround time is the sum of the waiting time and time a process takes for its execution.

II. ROUND ROBIN SCHEDULING ALGORITHM

This algorithm allocates the CPU in equal time to all processes. After a constant time slice called time quantum at the end of the ready queue, a process is blocked and placed. When all other processes are performed in their respective times, this process is assigned to CPU again. Round Robin's efficiency completely depends on the quantum time selected. The algorithm becomes FCFS when the quantum time is too large. On the other hand, if the quantum is much small, the waiting and turnaround times will be significantly larger overhead. Many Round Robin algorithms have been developed with technological progress based on dynamic quantum time. In this case, instead of a constant quantum time, a dynamic quantum time is chosen. It can be changed immediately after a cycle or at a ready queue. In the next section we will examine the original round robin algorithms, some of the round robin algorithms problem and a study on various methods for improving the performance of the round robin scheduling algorithm with the use of dynamic time quantum.
This programming algorithm is one of the oldest, simplest, fairest and most widely used, especially for time-share systems. A small unit of time, a time slice or a quantum is defined. Every running process is kept in a circular queue. The CPU scheduler circles this queue and assigns a quantum time interval for each process. New processes have been added to the queue tail. The CPU Scheduler chooses the first process from the queue sets the timer to disrupt the quantity and sends the process. When the process is still running at the end of the quantum, the CPU is pre-empted and added to the queue tail. The process frees CPU voluntarily if it finishes before the end of the quantum. In either case, the planner will assign the CPU to the next process. Whenever a process is given to the CPU, a context switch adds a fee to the run time.

TableI. The following is an example of the round robin algorithm:

Table	Scheduling criteria		
	Process ID	***Arrival time (AT)***	***Burst Time (BT)***
	P1	0	105
	P2	0	60
	P3	0	120
	P4	0	48
	P5	0	75

Time quantum=30.

Fig. 1 Gantt chart for round robin

P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P3	P5	P1	P3

0 30 60 90 120 150 180 210 240 258 288 318 348 363 378 408

Table II.Round Robin Scheduling

Table	Scheduling criteria		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	378	273
	P2	210	150
	P3	408	288
	P4	258	210
	P5	363	288
	TOTAL	1617	1209
	AVERAGE	323.5	978.6

A. Complication With Round Robin Scheduling

RR 's performance depends on time quantum. In certain cases, even if only a small amount of BT remains, additional context switches are needed.. The Turn-around time depends on the time quantum as well. Increase in quantum time-causes less context-switch, less turn-around time, but can lead to high response time and waiting time. Reduction in quantum time due to lower waiting time and response time but can lead to high turn-around time and high context-switch. Thus, all factors can be partially reduced by a medium-sized time quantum. The dynamic quantum time method can be used for the medium-sized time quantum. We will now examine various methods based on quantum dynamics.

III. ANALYSIS OF VARIOUS ROUND ROBIN SCHEDULING ALGORITHM

We only considered CPU bound processes during the analysis. In each test case, five independent processes in a uni-processor environment are analysed. Corresponding burst time and process arrival time are known prior to execution. The entire time of arrival and burst time is in M.S. (millisecond). Below are the Arrival time and burst time.

We check the performance of particular algorithm in our analysis for several algorithms. Turn-around time and waiting time can be calculated using the following formula.

Turnaround Time=Completion time-Arrival time

Waiting Time=Turnaround time-Burst Time.

A. Modified Round Robin Algorithm for Resource Allocation based on Average (MRRA).

Procedure

MRRA algorithm is offers by PandabaPradhan, Prafullla Ku. Behera, B N B Ray in account to improve the performance of Round Robin. The MRRA calculates and uses an average burst time of processes for every cycle.

Step1: the process based on A.T comes into ready queue.

Step2: sort the B.T process found on the ready queue.

Step3: find the quantum on the ready queue with the average B.T of the process.

Step4: now set quantum time in the queue for all loaded processes.

Step5: end the process if process B.T-time quantum=0.

Step6: when processB.T-TIME QUANTUM!=0 put the process at the end of the readyqueue. This process has been completed. Continue step:2 to step:6 until all the process is finished.

P4	P2	P5	P1	P3	P1	P3	P3

0 48 108 183 264 345 369 400 408

TQ=81 TQ=31 TQ=8

Fig. 2 Gantt chart for MRRA

Tableiii .Mrra Scheduling Algorithm

Table	Scheduling criteria		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	369	264
	P2	108	48
	P3	408	288
	P4	48	0
	P5	183	108
	TOTAL	1116	708
	AVERAGE	223.2	141.6

B. Self Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes (SRBRR ALGO).

In order to improve the performance of Round Robin, Rami J. Matarneh proposes a SRBRR algorithm. SRBRR calculates the median of the process burst time and it is used as quantum time for each cycle. or we can say TQ= Medium (BT of Ready queue) calculates the quantum time. The median is determined using the following formula.

$$Q=\begin{cases} Y_{(N+1)/2} & \mid \; if\ N\ is\ odd \\ \frac{1}{2}\left(Y_{\left(\frac{N}{2}\right)}+Y_{1+N/2}\right) & \mid\mid if\ N\ is\ even \end{cases}$$

Where, Y is the number in an ascending order, in the middle of a group of numbers.

Procedure

Step1: The process comes at A.T-based ready queue.
Step2: Then, use B.T. process median that is placed on the ready queue to find quantum time.
Step3: Allocate this quantum time to all processes loaded at the ready queue.

Step4: Then terminate this process when process B.T-time quantum=0.
Step5: IF process B.T-TIME QUANTUM!=0 , then at the end of the ready queue, put that process. Continue with step:2 until step:6 is completed.

P1	P2	P3	P4	P5

0 105 165 285 333 408

TQ=120

Fig. 3 Gantt chart for SRBRR

TableIV. SRBRR scheduling algorithm

Table	Scheduling criteria		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	105	0
	P2	165	105
	P3	285	165
	P4	333	285
	P5	408	333
	TOTAL	1035	888
	AVERAGE	259.2	177.6

C. Implementation of Alternating Median Based Round Robin Scheduling Algorithm(AMBRR)

A set of two time quanta for scheduling a ready queue is employed in the proposed algorithm. These time quanta are altered in scheduling cycles. Prior to the initial scheduling cycle, one time quanta is set equivalent to the median of burst time of processes in the ready queue while the other one is calculated as the difference between the maximum burst time value and the median of burst in the ready queue of processes. The two time quanta will be fixed over the entire scheduling period and will not be recalculated until the ready queue architecture changes when a new process is arrived. The greater time quantum is used when even the numbers of

scheduling cycles are finalized while the smaller time quantity is used when the impartial numbers of time cycles are completed. The twice quanta therefore is alternatively used in the programming cycles.

Procedure

Step1: The process comes at A.T-based ready queue.
Step2: Sort the process on the basis of Burst time 's ready queue.
Step3: with the above formula, find the Burst time process time quantum placed on the queue.
Step4: All process that is loaded at ready queue are now assigned this time quantum.
Step5: if the process Burst time-time quantum=0 then terminate that process.
Step6: IF process Burst time-TIME QUANTUM!=0 then set this process at end of ready queue. Follow step2 to step5 until all processes have been completed.

TableIV. ***AMBRR scheduling algorithm***

Table	**Scheduling criteria**		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	366	261
	P2	108	48
	P3	408	288
	P4	48	0
	P5	183	108
	TOTAL	1113	705
	AVERAGE	222.6	141

P4	P2	P5	P1	P3	P1	P3

0 48 108 183 258 333 363 408

TQ1=75 TQ2=45

Fig. 4 Gantt chart for AMRBRR

D. Improved Round Robin Scheduling using Dynamic Time Quantum.(IRR)

Procedure

DebashreeNayak offers a Round Robin performance-enhancing IRR algorithm. In IRR, the median of the processes burst time is calculated for each cycle. Median is derived from the formula below.

$$Q=\begin{cases} Y_{(N+1)/2} & \text{if } N \text{ is odd} \\ \frac{1}{2}\left(Y_{\left(\frac{N}{2}\right)}+Y_{1+N/2}\right) & \text{if } N \text{ is even} \end{cases}$$

Where, Y is the number in the center of the ascending group of numbers. Time is calculated by means of below formula:

T.Q= (MEDIAN + HIGH B.T) / 2
Step1: Process arrives at the A.T-based ready queue.
Step2:sort the process in the queue.
Step3:Find the quantum time of the process B.T placed on ready queue using the above formula.
Step4: Now set this time quantum to every process loaded in the queue.
Step5: If process B.T-time quantum=0 then terminate that process.
Step6: If process B.T-TIME QUANTUM!=0 then place that process to the end of ready queue.Continue with step:2 to step:7 until all process is completed.

P4	P2	P5	P1	P3	P1	P3	P3

0 48 108 183 280 377 385 404 408

TQ1=94 TQ2=19

Fig.5 Gantt chart for IRR

TableVI. IRR scheduling algorithm

Table	**Scheduling criteria**		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	385	280
	P2	108	48
	P3	408	288
	P4	48	0

Table	Scheduling criteria		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P5	183	108
	TOTAL	1132	724
	AVERAGE	226.4	144.8

E. New Alternative of Round Robin (NARR)

This algorithm is the new version of round robin. To improve the performance of round robin in cloud computing, NARR calculates the time quantum dynamically by calculating the median based on the processes added in the ready queue.

Procedure

Step1: Process based on arrival time added in the ready queue.

Step2: Set 3 variables i.e. process counter =0 , Time quantum =0 and median =0.

Step3: Sort the processes according to their burst time in ascending order.

Step4: Find the time quantum by calculating median of the burst time of the processes.

If(PC= =1)

TQ= burst time of that process.

Else

TQ= median of the burst time of the processes.

Step5: If any of the process complete its execution then there is no change in time quantum and process counter become program counter-1. In case any other process arrived then repeat step2, step3 and step4 to calculate new time quantum.

Step6: If the time quantum is over and process is still in running state, the CPU is preempted and the process is added to the ready queue.

Step7: if process counter = = 0 means that all the processes complete their execution time then compute turnaround time and waiting time.

Fig. 6 Gantt chart for NARR

P4	P2	P5	P1	P3	P1	P3

0 48 108 183 258 333 363 408

TQ=75

Table VII.NARR scheduling algorithm

Table	Scheduling criteria		
	Process ID	***TURN AROUND TIME***	***WAITING TIME***
	P1	363	258
	P2	108	48
	P3	333	213
	P4	48	0
	P5	183	108
	TOTAL	1035	627
	AVERAGE	207	125.4

IV. COMPARABILITY OF DIFFERENT VARIANTS IN ROUND ROBIN ALGORITHMS

After analyzing all these 6 algorithms we found that NARR gives better turnaround time and waiting time for given data values as shown in fig7.All algorithms described above primarily include ascending order processes of burst times. In addition, it is an issue for itself to sort out the problem in ascending order. The quantum time of the MRR algorithm is importantly to note that it is significantly different due to small or large differences in execution time of task.This means that the average burst times may vary significantly from the median time of all tasks with smaller T.A.T and less context but greater time of reaction and waiting time. Thus, if process sorting is applied in SRBRR, SRBRR algorithm is more stable than the MRR Algorithm. However, the process is not used in the original SRBRR so that in case of the process time is identical, it leads to F.C.F.S.If the arrival time of the procedure is different, SRBRR works very well. The time quantum calculation of AMBRR algorithm is an extremely computational task that takes up a large amount of time to calculate median and average burst time in each cycle.In this section we will view the comparison between average turnaround time and average waiting time.

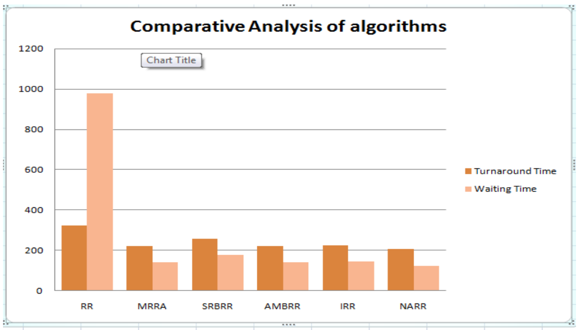


Fig. 7 Comparative analysis

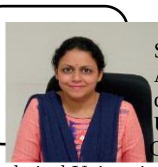
REFERENCES

1. Sindhu, M., Rajkamal, R., Vigneshwaran, P.: An Optimum Multilevel CPU Scheduling Algo-rithm. 978-0-7695- 4058-0/10 $26.00 © 2010 IEEE
2. Silberschatz, A., Galvin, P.B., Gangne, G.: Operating System Concepts, 6 edn. Wiley, INC (2002)
3. Saleem, U., YounusJaved, M.: Simulation of CPU Scheduling Algorithm. 0-7803-6355-8/00/ $10.00 @ 2000 IEEE
4. Huajin, S., Deyuan, G., Shengbing, Z., Danghui, W.: Design Fast Round Robin Scheduler in FPGA. 0-7803-7547- 5/021/$17.00 @ 2002 IEEE
5. Mamunur Rashid, Md., NasimAdhtar, Md.: A new multilevel CPU scheduling algorithm. J. Appl. Sci. **6**(9), 2036–2039 (2009)
6. Suranauwarat, S.: A CPU Scheduling Algorithm Simulator. In: WI 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee (2007)
7. Black, D.L.: Scheduling support for concurrency and parallelism in the mach operating system. Computer **23**, 123–126 (1985)
8. Bhardwas, A., Rachhpal Singh, Gaurav: Comparative study of scheduling algorithm in operating system. Int. J. Comput. Distrib. Syst. **3**(I) (2013)
9. A. Alsheikhy, R. Ammar, and R. Elfouly, "An improved dynamic Round Robin scheduling algorithm based on a variant quantum time",IEEE, 2015 11th International Computer Engineering Conference (ICENCO), 2015, pp. 98-104.
10. Rami J. Matarneh,"Self-Adjustment Time Quantum in Round Robin Algorithm Dependingon Burst Time of Now Running Processes", American J. of Applied Sciences 6(10): 1831 -1837,2009.
11. Pandaba Pradhan, Prafulla Ku. Behera and B N B Ray, "Modified Round Robin Algorithm for Resource Allocation in Cloud Computing ", ScienceDirect, Procedia Computer Science 85 (2016) 878 – 890,2016.

12. Salman Arif, Saad Rehman and Farhan Riaz "Design of A Modulus Based Round Robin Scheduling Algorithm",IEEE, 9th Malaysian Software Engineering Conference, Dec. 2015
13. Salman Arif,Naveed Ghaffar,Ali Javed "Implementation of Alternating Median Based Round Robin Scheduling .

AUTHORS PROFILE

Afsar Jahan, Afsarjahan is working as a Assistant Professor, Department of Computer Applications at Doon University Dehradun. Currently she is persuing her Ph.D. degree in Computer Science from DIT University, Dehradun(2018-2021). She received her MCA degree from Graphic Era University, Dehradun and BCA from Graphic Era University Dehradun. She specializes in core areas of computer science and holds experience of more than 2.5 years She is a Ph.D. scholar. Her Ph.D. topic is “Scheduling In Cloud Computing”. She is NET Qualified in 2019. She specializes in core areas of computer science including computer networks, Network security, Mobile computing, concurrent and distributing computing, cloud computing and J2EE based applications.



Ritika,Dr. Ritika is working as a Professor of Computer Science and Head, Department of Computer Applications at DIT University. She received her Ph.D. degree in Computer Science from Gurukul Kangri University, Haridwar in the year 2010, M.Tech degree in Computer Science and Engineering from Uttarakhand Technical University, Dehradun and MCA from Gurukul Kangri University Dehradun. She specializes in core areas of

computer science and holds experience of more than 17 years. She is an innovative person with deep knowledge of Bigdata Analytics, Data Mining, Mobile Computing, Computer Networks, J2EE based applications etc. She has also published many research papers in various National and International Journals. She is life time membership of the Indian Science Congress Association, Computer Society of India, ACM, Institute of Electrical and Electronics Engineers and The Institution of Electronics & Telecommunication .Engineers. She is SPOC of NPTEL Local Chapter DIT University Dehradun. Received certificate of appreciation for instrumental role as SPOC NPTEL DIT Chapter (from Jan-May 2016) from Coordinator NPTEL, IIT Madras.

Rakesh Saini, Dr. Rakesh Saini is currently working as assistant professor(senior grade) in Department of Computer science and Application At DIT University Dehradun. He has completed his MCA from UPTU, Lucknow2005 and M Tech (Computer science nd engineering) degree from UTU, Dehradun in 2012and Ph.D. degree from DIT University, Dehradun in 2017. He is having 15years of teaching experience. He has published variousresearch papers in reputed journal like Scopus, Elsevier indexed journals. He is author of around 8 books. His research interests include Wireless Sensor Network, Energy –Efficient in Wireless sensor Network, Cloud Computing.