

An Efficient Approach to Calculate Dynamic Time Quantum in Round Robin Algorithm for Efficient Load Balancing

Debabrata Sarddar
Assistant Professor
Department of CSE
University of Kalyani, India

Shouvik Chakraborty
P.G. Student
Department of CSE
University of Kalyani, India

Mousomi Roy
P.G. Student
Department of CSE
University of Kalyani, India

ABSTRACT

Cloud computing is one of the most emerging technologies in the world. Cloud service providers deal with huge amount of data and processes huge number of requests. It faces various challenges. Load balancing is one of the major challenges. Load balancing is basically the distribution of the tasks among different nodes for perfect utilization of resources and time. Different techniques have been proposed like Shortest Job First, First Come First Serve, Round Robin etc. The main goal of the different load balancing algorithms is to reduce the average waiting time and average turn around time. Round Robin scheduling algorithm is the widely accepted scheduling algorithm in multitasking and real time load balancing environment. It is the most popular algorithm due to its fairness and starvation free nature towards the processes, which is achieved by using the time quantum. As the time quantum is static, it causes less context switching in case of high time quantum and high context switching in case of less time quantum. Increasing context switch leads to high avg. waiting time, high avg. turnaround time which is an overhead and degrades the system performance. So, the performance of the system solely depends upon the choice of optimal time quantum which is dynamic in nature. In this paper, some variants of RR scheduling algorithm have been proposed which can dynamically compute the time quantum. Results of these methods are provided and compared it with other existing methods.

General Terms

Cloud Computing, Load Balancing.

Keywords

Cloud computing, load balancing, round-robin scheduling, dynamic time quantum.

1. INTRODUCTION

Cloud computing can help to develop good business applications that will bring true business value [1]. Cloud computing mainly provides four different services like: virtual server storage (Infrastructure as a service or IaaS) such as Amazon Web Services, software solution provider over the internet (Software as a Service or SaaS), software and product development tools (Platform as a Service or PaaS) such as Google Apps and Communication as a service or CaaS [2][3]. Clouds are deployed on physical infrastructure where Cloud middleware is implemented for delivering service to customers. Such an infrastructure and middleware differ in their services, administrative domain and access to users. Therefore, the Cloud deployments are classified mainly into three types: Public Cloud, Private Cloud and Hybrid Cloud. Due to the exponential growth of cloud computing, it has been

widely adopted by the industry and there is a rapid expansion in data-centers. This expansion has caused the dramatic increase in energy use and its impact on the environment in terms of carbon footprints. Since cloud computing is generally characterized as an IT service (with the vendor providing and maintaining the software and hardware infrastructure), the ability of the client organization to integrate and utilize the vendor's services determines the extent IT benefits are likely to be achieved. Organization-specific capabilities related to implementation, integration, and utilization of cloud services play a key role in deployment performance [3]. Organization-Specific capabilities that can be a source of competitive advantages are [3]:

- Technical. IT resources giving the organization functionality, flexibility, and scalability.
- Managerial. Human IT resources resulting from training, experience, and insight.
- Relational. Ability to develop positive associations with IT providers characterized by trust.

This paper presents some variants of Round-Robin algorithm to achieve reduced average waiting time and average turn around time in load balancing.

2. LOAD BALANCING

The term 'Load Balancing' generally refers to transfer load from the overloaded process to other underloaded processes. The main problem is that how to choose the next node and transfer a load. Based on the implementation method, load balancing algorithms can be classified in two ways [4]:

- Static algorithms does not depends upon the present state.
- Dynamic algorithms checks the present state and then decides.

Existing load balancing techniques in clouds, consider various parameters like:

- 1) Throughput is used to calculate the no. of tasks whose execution has been completed.
- 2) Overhead Associated determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication.[3].
- 3) Fault Tolerance is the ability of an algorithm to perform uni-form load balancing in spite of arbitrary node or link failure.
- 4) Migration time is the time to migrate the jobs or resources from one node to other.

- 5) Response Time is the amount of time taken to respond by a particular load balancing algorithm in a distributed system.
- 6) Resource Utilization is used to check the utilization of resources.
- 7) Scalability is the ability of an algorithm to perform load balancing for a system with any finite number of nodes.
- 8) Performance is used to check the efficiency of the system.
- 9) Energy Consumption- Load balancing helps in avoiding overheating by balancing the workload across all the nodes.
- 10) Carbon Emission - Energy consumption and carbon emission go hand in hand.

3. LOAD BALANCING ALGORITHMS

In the First-Come-First-Serve (FCFS) algorithm, the CPU is assigned immediately to that process which arrives first at the ready queue. In Shortest Job First (SJF) algorithm, process having shortest CPU burst time will execute first. If two processes having same burst time and arrival time, then FCFS procedure is followed. Priority scheduling algorithm, provides the priority to each process and selects the highest priority process from the ready queue. A small unit of time quantum is given to each process present in the ready queue in case of Round Robin (RR) algorithm which maintains the fairness factor.

In RR scheduling, processes get fair share of CPU because of static time quantum assign to each process and the context switch is inversely proposnal to choice of static time quantum which degrades the overall performance of the system (high average waiting time & average turnaround time). This factor motivates us to design an improved algorithm which is able to increase the system performance by reducing the number of context switches, average waiting time & average turnaround time using the concept of dynamic time quantum. The static time quantum which is a limitation of RR was removed by taking dynamic time quantum using median method introduced in SARR algorithm[5]. In DQRRR [6] the median method and job mix concept is reused in a different way. DQRRR gives better result than the classical RR scheduling algorithm. Neural Network based approach has been proposed in [9][12]. In DTQRR [10], the main objective is to adjust the time quantum dynamically depending upon arrival time and burst time of the processes. DTQSJF [11] works for the unpredictable burst time of the processes.

4. PROPOSED APPROACHES

In dynamic time quantum based methods various algorithms have been proposed by many scientists. One of the most popular is the AN algorithm, proposed by Abbas Noon et. al [7]. They present a solution to the time quantum problem by making the operating system adjusts the time quantum according to the burst time of the existed set of processes in the ready queue. In DQRRR [6] the median method and job mix concept is reused in a different way. In this paper some variations of median based methods have been developed and compared them whit fixed time quantum and AN algorithm.

In the median based method [8], at first the task is to calculate median. The median of a sorted list can be calculated using the equation 1.

$$\begin{aligned}
 \text{M dian } (M) &= \text{List } \left(\frac{n}{2}\right) \text{ if } n \text{ is odd} \\
 &= \frac{\text{List } \left(\frac{n}{2}\right) + \text{List } \left(\frac{n}{2} + 1\right)}{2} \text{ if } n \text{ is even} \dots\dots\dots(1)
 \end{aligned}$$

Where n is the number of elements in the list.

Now the problem is how to compute the optimal time quantum using the median method. In this paper, the time quantum has been generated in two ways. Time quantum can be generated using equation 2 and equation 3.

$$TQ = \frac{\text{Highest Burst Time} + \text{Median}}{2} \dots\dots\dots(2)$$

$$TQ = \frac{\text{Lowest Burst Time} + \text{Median} + \text{Highest Burst Time}}{3}$$

In the first case, time quantum is calculated by taking the average of highest burst time and median and in the second approach, time quantum is calculated by taking the average of the lowest burst time, highest burst time and the median. Time quantum is calculated at each iteration. These methods can be applied in three ways. First, the list is sorted in ascending order, Second, the list is sorted in descending order and third, list is not sorted. These methods have been testes in all possible ways. The algorithm is given below.

Algorithm:

Input arameters: Process(Pn), Burst Time(BT), Arrival Time, ready queue.

Output parameters: Avg. Waiting Time(Awt), Avg. Turnarround Time(Att)

Step 1 Initialize: ready queue=0, Awt=0, Att=0

Step 2 While (ready queue!=0)

Step 2.1 Sort the processes in ascending order according to their Burst Time in ready queue

Step 2.2 Find Median

Step 2.3 Calculate TQ

Step 2.4 For each process i=1 to n //assign Oqt to each process

Step 2.4.1 P[i] → TQ

end for

Step 2.5 Check if new process arrives

Step 2.5.1 Update counter and goto step 2

end while

Step 3 Awt, Att are calculated.

Step 4 Stop and exit.

Flow chart of the algorithm is given in figure 1.

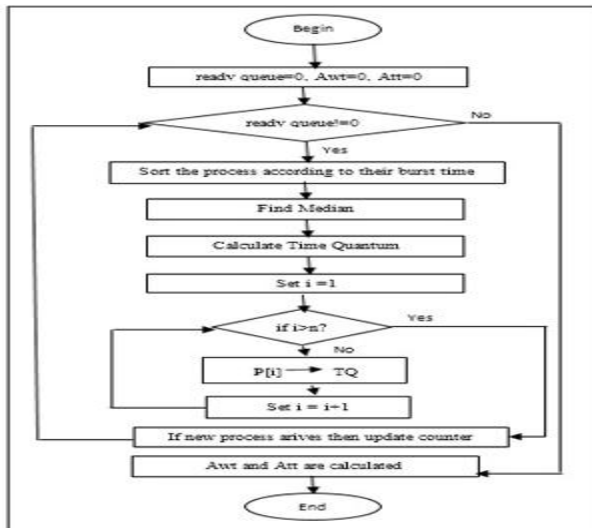


Fig 1: Flowchart of the algorithm

5. EXPERIMENTAL RESULTS

In this paper, two different methods have been implemented i.e. method 1 based on the equation 2 and method 2 based on equation 3 in 3 different orders i.e. ascending, descending and not sorted. All the processes are summarized in table 1.

Table 1. Summarization of the Methods

| | Ascending Order | Descending Order | Not Sorted |
|---------------------------------------|-----------------|------------------|------------|
| Method 1 (based on equation 2) | M1ASC | M1DSC | M1NS |
| Method 2 (based on equation 3) | M2ASC | M2DSC | M2NS |

So, six different methods have been implemented according to the table 1 along with other methods in MATLAB. Some processes have been taken with their burst time compared these methods. Graphical demonstration also given with the result. Experimental results have been shown in table 2 and in table 3. Table 2 shows the result of the average waiting time and table 3 shows the result of average turnaround time.

Figure 2 provides the graphical representation of average waiting time and average turnaround time in figure 3.

Table 2. Average waiting times of 10 schedules with different load balancing algorithms

| Serial No. | Burst Time | RR (TQ=20) | AN | M1ASC | M1DSC | M1NS | M2ASC | M2DSC | M2NS |
|------------|------------------|------------|-------|-------|--------|-------|-------|-------|-------|
| 1 | [20,40,60,80] | 70.0 | 62.5 | 50.0 | 118.75 | 50.0 | 62.5 | 122.5 | 62.5 |
| 2 | [80,45,62,34,78] | 190.6 | 174.4 | 108.8 | 206.8 | 160.8 | 132.4 | 211.4 | 173.6 |
| 3 | [54,19,45,14,77] | 90.2 | 103.4 | 51.4 | 129.4 | 75.4 | 60.4 | 130.8 | 94.6 |
| 4 | [34,30,62,88,74] | 152.8 | 122.4 | 84.0 | 176.0 | 99.8 | 120.0 | 200.8 | 123.6 |
| 5 | [28,17,29,35,41] | 77.0 | 57.2 | 49.0 | 88.0 | 51.2 | 54.8 | 96.2 | 57.0 |
| 6 | [44,75,55,43,55] | 188.4 | 150.8 | 93.8 | 155.2 | 135.4 | 93.8 | 148.8 | 130.6 |
| 7 | [32,56,87,32,18] | 109.6 | 108.2 | 57.6 | 127.6 | 99.2 | 66.6 | 131.4 | 108.2 |
| 8 | [79,27,66,51,44] | 173.6 | 150.4 | 94.6 | 184.2 | 163.0 | 92.0 | 168.0 | 149.4 |
| 9 | [24,56,40,64,32] | 123.2 | 101.6 | 76.0 | 146.8 | 108.8 | 74.0 | 134.4 | 100.8 |
| 10 | [25,5,28,3,33] | 45.4 | 43.0 | 21.0 | 63.2 | 34.8 | 33.0 | 61.2 | 45.4 |

Table 3. Average turn around times of 10 schedules with different load balancing algorithms

| Serial No. | Burst Time | RR (TQ=20) | AN | M1ASC | M1DSC | M1NS | M2ASC | M2DSC | M2NS |
|------------|------------------|------------|-------|-------|--------|-------|-------|-------|-------|
| 1 | [20,40,60,80] | 120.0 | 112.5 | 100.0 | 168.75 | 100.0 | 112.5 | 172.5 | 112.5 |
| 2 | [80,45,62,34,78] | 250.4 | 234.2 | 168.6 | 266.6 | 220.6 | 192.2 | 271.2 | 233.4 |
| 3 | [54,19,45,14,77] | 132.0 | 145.2 | 93.2 | 171.2 | 117.2 | 102.2 | 172.6 | 136.4 |
| 4 | [34,30,62,88,74] | 210.4 | 180.0 | 141.6 | 233.6 | 157.4 | 177.6 | 258.4 | 181.2 |
| 5 | [28,17,29,35,41] | 107.0 | 87.2 | 79.0 | 118.0 | 81.2 | 84.8 | 126.2 | 87.0 |
| 6 | [44,75,55,43,55] | 242.8 | 205.2 | 148.2 | 209.6 | 189.8 | 148.2 | 203.2 | 185.0 |
| 7 | [32,56,87,32,18] | 154.6 | 153.2 | 102.6 | 172.6 | 144.2 | 111.6 | 176.4 | 153.2 |
| 8 | [79,27,66,51,44] | 227.0 | 203.8 | 148.0 | 237.6 | 216.4 | 145.4 | 221.4 | 202.8 |
| 9 | [24,56,40,64,32] | 166.4 | 144.8 | 119.2 | 190.0 | 152.0 | 117.2 | 177.6 | 144.0 |
| 10 | [25,5,28,3,33] | 64.2 | 61.8 | 39.8 | 82.0 | 53.6 | 51.8 | 80.0 | 64.2 |

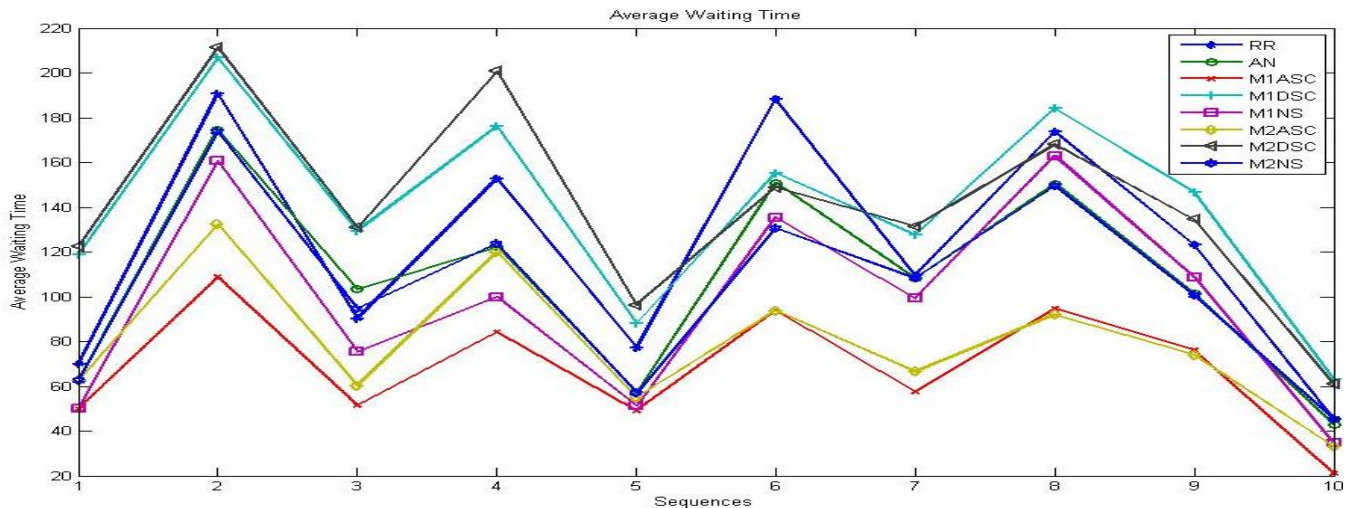


Fig 2: Comparison of Average Waiting Time

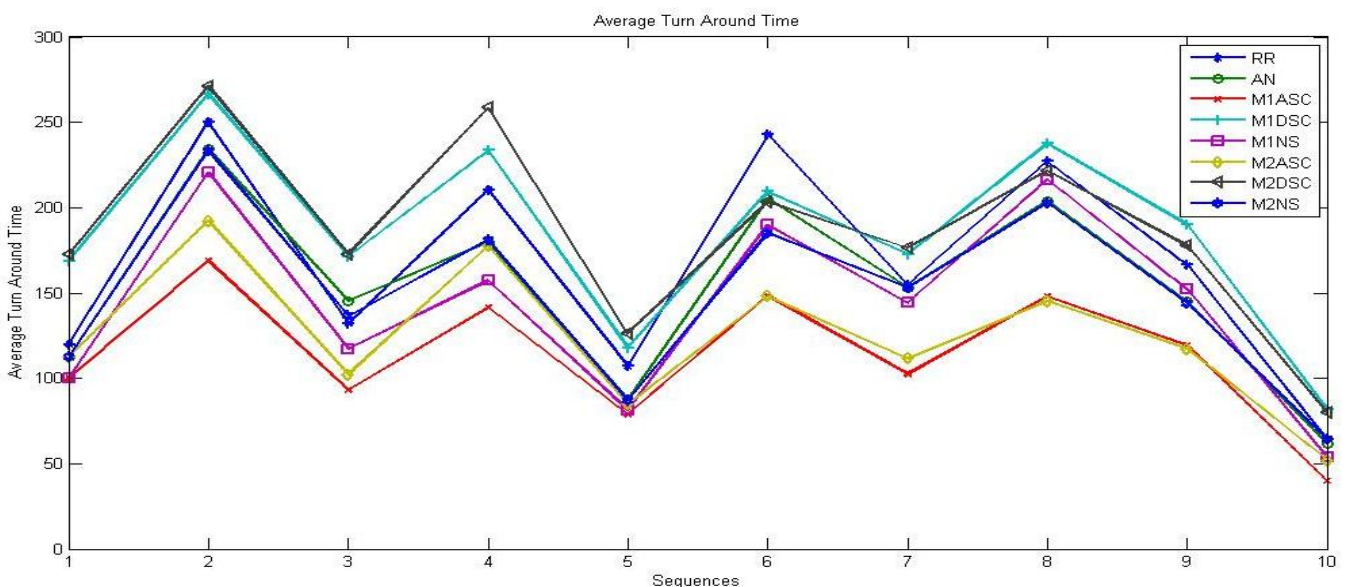


Fig 3: Comparison of Average Turn Around Time

6. CONCLUSION

The Round robin algorithm faces the problem to find the optimal time quantum to be used in the algorithm. This paper provides a solution to this problem by using dynamic time quantum instead of fixed time quantum.

In this paper, different round robin algorithm have been discussed that could be a simple step for a huge aim in obtaining an optimal scheduling algorithm. It will need much more efforts and researches to score a goal. From the simulation study, an important conclusion can be obtained; that the performance of M1ASC and M2ASC algorithm is higher than that of RR in any system. The use of dynamic scheduling algorithm increased the performance and stability of the system. The proposed algorithm can be modified using soft and hard real time systems. This work can be extended in future in many ways because no methods have been proposed to be optimal. Soft and evolutionary computing methods can be applied to determine the time quantum. Evolutionary and

heuristic approaches can be applied to generate dynamic time quantum.

7. ACKNOWLEDGMENTS

We would like to thank Dr. Kalyani Mali, Head of the department, department of computer science & engineering, University of Kalyani for purpose of laboratory and other university infrastructure.

8. REFERENCES

- [1] Abdulaziz Aljabre, 2012 " Cloud Computing for Increased Business Value", International Journal of Business and Social Science, Vol. 3 No. 1; January 2012.
- [2] K. Dinesh, G. Poornima, K.Kiruthika, 2012 " Efficient Resources Allocation for Different Jobs in Cloud", International Journal of Computer Applications (0975 – 8887) Volume 56– No.10, October 2012.
- [3] Gary Garrison, Sanghyun Kim, and Robin L. Wakefield, 2012" Success Factors for Deploying Cloud Computing",

- communications of the ACM, September 2012, vol. 55, no. 9, doi:10.1145/2330667.2330685.
- [4] R. L. Grossman, “The case for cloud computing”, 2009 IT professional, vol. 11, no. 2, (2009), pp. 23-27.
- [5] Rami J. Matarneh. 2009 “Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes”, American J. of Applied Sciences 6(10): 1831-1837,2009.
- [6] H.S.Behera, R.Mohanty, Debashree Nayak. 2010 “A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis “International Journal of Computer Applications (0975 – 8887) Volume 5– No.5, August 2010.
- [7] Abbas Noon, Ali Kalakech, Seifedine Kadry “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011 ISSN (Online): 1694-0814.
- [8] Debashree Nayak, Sanjeev Kumar Malla, Debashree Debadarshini “Improved Round Robin Scheduling using Dynamic Time Quantum” International Journal of Computer Applications (0975 – 8887), Volume 38, No.5, January 2012.
- [9] Deepali Maste, Leena Ragha, Nilesh Marathe “Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling” International Journal of Information and Computation Technology ISSN 0974-2239 Volume 3, Number 4 (2013), pp. 311-322.
- [10] Dibyendu Barman “Dynamic Time Quantum in Round Robin Algorithm (DTQRR) Depending on Burst and Arrival Time of the Processes” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-4, March 2013.
- [11] Dibyendu Barman, Iti Sarkar “Dynamic Time Quantum in Shortest Job First Scheduling Algorithm (DTQSJF) for Unpredictable Burst Time of Processes” International Journal of Computer Science & Engineering Technology (IJCSET), ISSN : 2229-3345 Vol. 4 No. 03 Mar 2013 pp: 208-212.
- [12] Nada M. Al Sallami, Ali Al daoud, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 10, 2013 pp: 138-145.