International Journal of Grid Distribution Computing
Vol. 8, No.4, (2015), pp. 171-178
http://dx.doi.org/10.14257/ijgdc.2015.8.4.17

Load Balancing Scheduling with Shortest Load First

Ranjan Kumar Mondal[1], Enakshmi Nandi[2] and Debabrata Sarddar[3]

[1]Department of Computer Science & Engineering,
University of Kalyani, Kalyani, India
[2]Ph.D. Student, Department of Computer Science & Engineering
University of Kalyani, Kalyani, India
[3]Assistant Professor, Department of Computer Science & Engineering,
University of Kalyani, Kalyani, India
[1]ranjan@klyuniv.ac.in, [2]pamelaroychowdhurikalyani@gmail.com,
[3]dsarddar1@gmail.com

Abstract

Cloud computing is an internet based technology. This computing paradigm has increased the utility of network where the potentiality of one node can be used by Other node, cloud provides services on demand to distributive resources such as Database, servers, software, infrastructure etc. in pay per use basis, load balancing is One of the unique and important issues for distributing a larger processing lode to smaller processing nodes for increasing total performance of system, in load balancing method the workload not only distribute across multiple computers but also other resources over the network links to gain optimum resource utilization, minimum average response time and avoid overload condition. Different load balancing algorithms have been launched in order to manage the resources of service provider efficiently and effectively. The objective of this paper is to propose efficient scheduling algorithm that can maintain the load balancing and provide improved strategies through efficient job scheduling that would decrease the average response time and increase the availability of more VMs to allocate new jobs from requesting nodes.

Keywords: *Cloud computing, Load balancing, SJF*

1. Introduction

Cloud computing is a new emerging applied science that helps to develop a new area in education and industry. This new technology offers distributed virtualized, elastic resources as utilities to clients. It has total capability for supporting full realization of computing as a utility in the near future [1].Cloud technology supports both parallel and distributed system. This distributed architecture deploys resources to distribute services effectively to clients in different geographical channels [2]. In the distributed environment users generate request randomly in any processor. So the disadvantage of this randomness is associated with task assignment. The inconsistency in task assignment to the processor generates imbalance behavior, i.e. some of the processors are overloaded and some of them are under loaded [3]. The goal of load balancing is to transfer the load from overloaded process to under loaded process transparently. Load balancing technique decides which user would use the virtual machine and which requesting machines will be put on hold. Load balancing of the entire system can be handled dynamically by using virtualization technology where it becomes possible to remap Virtual Machines (VMs) and physical resources with respect to change in load. A VM is a software implementation of a computing environment in which an OS or program can be installed properly and run.

ISSN: 2005-4262 IJGDC

Load balancing is done with the help of load balancers where each succeeding request is redirected and transparent to users who make the request. Based on different parameters like availability of current load, the load balancer uses different scheduling algorithm to decide which server should handle and forwards the request to the selected server [5].There are different scheduling algorithm exist in load balancing like Round Robin(RR) , First-Come-First-Served (FCFS), and some other scheduling algorithm. Most of these algorithm concentrate on maximizing throughput and minimizing the turnaround time, response time, waiting time and number of context switching for a set of request. In this paper our objective is to approach a new scheduling algorithm which helps to give better performance compare to existing algorithms such as Round Robin (RR), First-Come-First-Served (FCFS), etc.

1.1. Related Works

Load balancing [6] algorithm determines the effect of balancing the server workloads. Load balancing algorithm is divided into static algorithm and dynamic algorithm [7]. The static algorithm does not take in to account the previous state or nature of the node while distributing the node. The common static algorithms are Round-Robin Scheduling Algorithm.

- **Round-Robin Algorithm** : Round-Robin Scheduling Algorithm is the simplest one which could be most easily be carried out and selects the first node randomly, then allocate jobs to all other nodes in round-robin manner. However, it is only applicable for cloud computing in that case when some nodes might heavily loaded and some are not. The good side of this algorithm is that if any node fails, it will not halt the system; it will only affect the system performance.
- **First-Come-First-Served**: First come first serve is a self-adaptive algorithm, and suitable for a great deal of requests which produce different workloads, which would be unable to be forecasted [8]. Self-adaptive load balancing system includes two processes: monitoring the load states of servers and assigning the request to the servers.

So, the ideal load balancing algorithm should achieve the following targets:

Leave the collections, computing of load node information for each node; prevent the front-end scheduler from being system bottleneck.

Reduce the disturbances of load balancing algorithm as far as possible.

2. Scheduling Criteria

For the task scheduling based on RR, FCFS the criteria include the following parameters:

- **Context Switch:** A context switch is computing process of storing and restoring state of a CPU so that execution can be resumed from same point at a later time. Context switch are usually computationally intensive, lead to wastage of time, memory, scheduler .So for optimization purpose CPU needs this switch.
- **Throughput:** The number of process completed per unit time is called throughput. Throughput will be slow in round robin scheduling implementation.
- **CPU Utilization:** We want to keep the CPU as busy as possible.
- **Turnaround Time:** Turnaround time is sum of periods spent waiting to get into memory, waiting in ready queue, executing on CPU and doing input output. If the value is less then it will give better for performance.
- **Waiting Time:** The amount of time a process has been waiting in ready queue is called waiting time. The CPU scheduling algorithm does not affect the amount of time during which a process executes or does input-output; it affects only the amount of time that a process spends waiting in ready queue.

- **Response Time:** Response time is the time it takes to start responding, not the time it takes to output the response. If the response time will high, then it will decrease the system performance, this problem will face in case of Round Robin Algorithm.

A good scheduling algorithm must possess following characteristics for improvement the overall performance.

2.1. Characteristics

1. Minimum context switches.
2. Maximum CPU utilization.
3. Maximum throughput.
4. Minimum turnaround time.
5. Minimum waiting time.
6. Minimum response time.

3. Shortest Job First

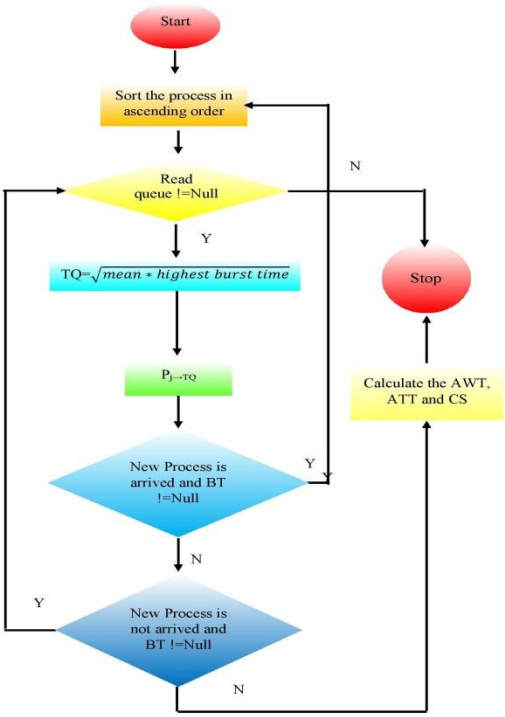
Shortest Job First (SJF) scheduling is a priority and Non-Preemptive scheduling. Non-Preemptive means, when the allotted time a processor then the processor cannot be taken the other, until the process is completed in the execution. Basically Shortest Job First is a dynamic load balancing algorithm which handles the process with priority basis. It determines the priority by checking the size of the process. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a Virtual Machine, which is lightly loaded. In that case that process size is lowest, this process will get first priority to execute whether we suppose lowest sized process executes in minimum time. The load balancer spreads the load on to different nodes known as spread spectrum technique.

The mechanism of Shortest Job First Algorithm is, to schedule the process with the shortest time to completion first, thus providing high efficiency and low turnaround time. In terms of time spent in the current program (job) began to enter in to the system until the process is finished the system, need a short time. Shortest Job First (SJF) scheduling algorithm can be said to be optimal with an average waiting time is minimal, which helps to improve the system performance.

4. Algorithm

1. Firstly start process, vmloadbalancer maintain the process by priority checking the size of the process and distribute the load to the virtual machine which is lightly loaded.
2. The vmloadbalancer, first allocate array size i.e. A [10].
3. Take number of elements to be inserted.
4. Vmloadbalancer select process which load has shortest burst time among all loads will execute first.
5. If in the process any load have same burst time length then FCFS (First come First Served) scheduling algorithm used.
6. Make average waiting time length of next process.
7. Start with first process, selection as above as shortest load come first which has minimal average time and other processes are to be in queue.
8. Calculates Burst total number of time.
9. Display the Related values.
10. Now close / Stop process.

5. Flow Chart



Flow Chart of the Proposed Algorithm

5. Simulation and Result Analysis

In every case we have compare the result of proposed SJF algorithm with the existing RR, FCFS algorithm using java language. The figure 1.Shows the comparisons of these algorithms on the basis of average waiting time. Table 1.shows the sequences according to five processes along with burst time and respective average waiting time for three respective algorithms, such as Round Robin (RR), First Come First Served (FCFS), SJF (Shortest Job First).Here we take arbitrary process burst time. Figure 2.shows the comparisons of the above three respective scheduling algorithms on the basis of average Turnaround Time. Table 2. Shows the sequences according to five processes along with burst time and respective average Turnaround Time for three respective algorithms, such as Round Robin (RR), First Come First Served (FCFS), SJF (Shortest Job First), it is said that our approach shows better performance among them.

Table 1. Three Different Scheduling Algorithms with Five Different Processes Along With Burst Time and Average Waiting Time

Sl. No.	Process Burst Time	RR	F CFS	SJF
1.	[12,15,11,25,7]	39.4	28	20
2.	[26,10,14,15,13]	49	35 .4	24.4
3.	[9,27,10,9,17]	37.4	29 .2	20
4.	[15,16,27,22,11]	56.2	36 .8	28.6
5.	[13,24,9,17,19]	49.6	31 .8	25.6
6.	[23,14,12,22,19]	59	36	30

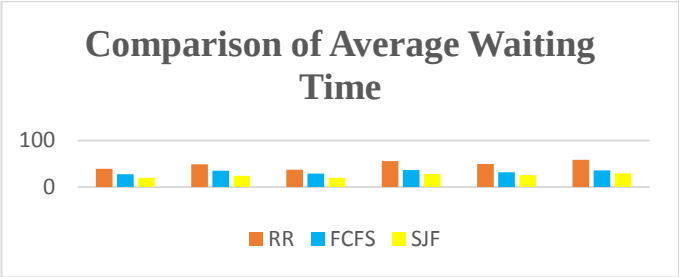


Figure 1. Comparison of Average Waiting Time with Six Different Processes of Three Different Scheduling Load Balancing Techniques

Table 2. Three Different Scheduling Algorithms with Four Different Processes Along With Burst Time and Turnaround Time

Sl. No.	Process Burst Time	RR	FCFS	SJF
1.	[12,15,11,25,7]	53.4	42	34
2.	[26,10,14,15,13]	64.6	51	40
3.	[9,27,10,9,17]	51.8	43.6	34.4
4.	[15,16,27,22,11]	74.4	55	46.8
5.	[13,24,9,17,19]	66	48.2	42
6.	[23,14,12,22,19]	77	54	48

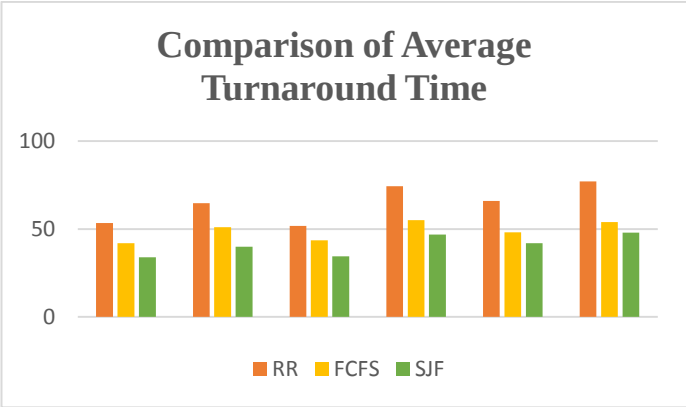


Figure 2. Comparison of Average Turnaround Time with Six Different Processes of Three Different Scheduling Load Balancing Techniques

6. Conclusion and Future Plan

The result shows that both average waiting time and average Turnaround Time are very less in case of SJF scheduling which helps to improve the performance of the system and maintain an efficient load balancing with very fast manner. But SJF faces some difficulties, such as the burst time of the process have to predict before CPU start the execution. But it is quite impossible to predict the burst time of the process complete

execution and if processes did not finish their execution in predicted burst time then in which algorithm process will execute, come this question.

For this solution we do experiment and extend our study with Dynamic Time Quantum SJF schedule in future to get fur better performance in Load Balancing Technique.

Acknowledgment

We would like to express our gratitude to Dr. Kalyani Mali, Head of Department, Computer Science and Engineering of University of Kalyani. Without her assistance and guidance, we would not have been able to make use of the university's infrastructure and laboratory facilities for conducting our research.

References

[1] G. Pallis, "Cloud Computing: The New Frontier of Internet Computing", IEEE Journal of Internet Computing, vol. 14, no. 5, **(2010)** September/October, pp. 70-73.
[2] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, vol. 13, no. 5, **(2009)** September/October, pp, 10-13.
[3] A. Khiyaita, H. EI Bakkli, M. Zbakh and D. E. Kettani, "Load Balancing Cloud Computing, State Of Art", **(2010)**, IEEE.
[4] J. Sahoo, S. Mohapatra and R. lath, "Virtualization: A survey on concepts, taxonomy and associated security issues" computer and network technology (ICCNT), IEEE, **(2010)** April, pp. 222-226.
[5] T. Sharma and V. K. Banga, "Efficient and Enhanced Algorithm in Cloud Computing", *International Journal of Soft Computing and Engineering (IJSCE) ISSN* **(2013)**, pp. 2231-2307.
[6] R. Shimonski, "Windows 2000 & Windows server 2003 clustering and load balancing", Emeryville McGraw-Hill Professional publishing, CA,USA **(2003)**, pp. 2.
[7] R. X. T. and X. F. Z., "A load balancing strategy based on the combination of static and dynamic, in database technology and applications (DBTA)", 2010 2nd international workshops, **(2010)**, pp.1-4.
[8] W. Li and H. Shi, "Dynamic Load Balancing Algorithm Based on FCFS" IEEE, **(2009)**, pp. 1528-1531.
[9] R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit Challenges and opportunities", High Performance Computing & Simulation, 2009 HPCS'09 International Conference on IEEE, **(2009)**.
[10] J. Li, M. Qui, J. –W. Niu, Yuchen and Z. Ming, "Adaptive resource allocation for preemptable jobs in cloud system", IEEE International Conference on intelligent system design and applications, **(2010)**, pp. 31-36.
[11] R. P. Pandhy (107CS046) and P. G. P. Rao (107CS039), "Load balancing in cloud computing system" Department of computer science and engineering National Institute of Technology Rourkela, Rourkela-769008, Orissa, India **(2011)** May.
[12] M. Randles, D. Lamb and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," 2010 IEEE 24th international conference on advanced information networking and application workshops, **(2010)**, pp. 551 - 556.
[13] J. Kaur "Comparison Load Balancing Algorithms In A Cloud", International Journal Of Engineering Research And Applications, **(2012)**, pp. 1169 - 1173.

Authors

Ranjan Kumar Mondal, received his M.Tech in Computer Science and Engineering from University of Kalyani, Kalyani, Nadia; and B.Tech in Computer Science and Engineering from Government College of Engineering and Textile Technology, Berhampore, Murshidabad, West Bengal under West Bengal University of Technology, West Bengal, India. At present, he is a Ph.D research scholar in Computer Science and Engineering from University of Kalyani. His research interests include Cloud Computing, Wireless and Mobile Communication Systems.

Enakshmi Nandi, received her M. Tech in VLSI and Micro-electronics from Techno India, Salt Lake, West Bengal and B. Tech in Electronics and Communication Engineering from JIS College of Engineering, West Bengal under West Bengal University of Technology, West Bengal, India. At present, she is Research scholar in Computer Science and Engineering from University of Kalyani. Her research interests include Cloud Computing, Mobile Communication system, Device and Nanotechnology.

Debabrata Sarddar, is an Assistant Professor at the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He completed his PhD from Jadavpur University. He did his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published more than 75 research papers in different journals and conferences. His research interests include Wireless and Mobile Systems and WSN, and Cloud computing.

