ADVANCES in NATURAL and APPLIED SCIENCES

ISSN: 1995-0772
EISSN: 1998-1090
2017 January 11(1): pages 13-18

Published BY**AENSI Publication**
http://www.aensiweb.com/ANAS
Open Access Journal

Efficient Load Balancing Algorithm For Cloud Computing Using Divisible Load Scheduling And Weighted Round Robin Methods

[1]S. Sankara Narayanan, [2]M. Ramakrishnan and [3]Murtaza Saadique Basha

[1]Research Scholar, Department of Computer Science & Engg., Anna University, Chennai, Tamil Nadu, India.
[2]Chairperson, School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India.
[3]Professor, Department of Computer Science & Engg., C.Abdul Hakeem College of Engg. & Tech., Melvisharam, Tamil Nadu, India.

Received 18 September 2016; Accepted 15January 2017; Available online 29 January 2017

Address For Correspondence:
S. Sankara Narayanan, 1Research Scholar, Department of Computer Science & Engg., Anna University, Chennai, Tamil Nadu, India

ABSTRACT

Cloud computing is the new architecture that fulfils the dreams of several researchers on centrally organized and accessed resources. When a request is made to the cloud service provider, it has to be executed by one among several available servers and load on servers need to be balanced. This paper provides efficient load balancing algorithm for cloud environments, and it combines the merits of divisible load balancing algorithm and weighted round robin algorithm. The simulation result shows that this method is efficient and the values of processing time and response time yields low values compared to other methods. Additionally this method also removes the drawbacks of traditional round robin methods.

KEYWORDS: Cloud Computing, Load Balancing, Divisible Load Scheduling, Round Robin, Resource Management

INTRODUCTION

An emerging commercial infrastructure paradigm that is internet based and can be utilized over internet is the cloud computing [1]. It is a distributed computing standard that allows users to access their data, applications and other services over internet using their browsers and it is widely accepted by many organizations [2]. Cloud computing is a framework that enables suitable, on-demand network access to a pool of shared computing resources [3]. These computing resources can be made available in internet with a minimum effort and with less service provider interaction. Now days, internet is represented as a cloud. Generally cloud architecture is classified into three basic types [4]. Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS)[5,6]. Apart from these common architectures, we have Data as a Service (DaaS), Storage as a Service (STaaS), Security as a Service (SECaaS)[6].

Since the cloud network is large, it has to be highly reliable and scalable. For an incoming request, resource allocation has to be performed without any bias. This allocation should be dynamic so that based on the number of requests available, resource allocation can be performed. This requirement of dynamism needs efficient and accurate load balancing mechanisms [7].

ToCite ThisArticle: S. Sankara Narayanan, M. Ramakrishnan and Murtaza Saadique Basha., Efficient Load Balancing Algorithm For Cloud Computing Using Divisible Load Scheduling And Weighted Round Robin Methods. **Advances in Natural and Applied Sciences**. 11(1); Pages: 13-18

Load balancing techniques facilitates effective utilization of cloud resources thereby providing maximum throughput with minimum response time. Roughly dividing algorithm cannot be applied to manage the incoming requests. Load balancing algorithms provides users with minimum delay, timeouts and avoids long system responses[8]. Load balancing algorithms distributes the communication traffic (incoming requests) in an amicable manner that resource availability can be settled conclusively. Two broad categories of load balancing algorithm exists viz. static and dynamic algorithms [8].

The primary objective of load balancing algorithm is to quickly execute applications on resources where workload varies in an unpredictable way [10]. Dynamic load balancing algorithms are widely used in balancing heterogeneous resources. Divisible load scheduling algorithm assumes that computation can be divided into small pieces of arbitrary size and each partition can be executed by one processor independently [9]. Divisible load scheduling algorithm is widely used in processing intensive data applications, data grid applications, image processing, etc. It will be a good idea to use this algorithm to balance cloud resources [11]. Hence we present a novel load balancing algorithm using divisible load scheduling algorithm which is dynamic in nature. The paper is organized as follows: paper starts by providing basics about cloud computing and need for load balancing in cloud environments in section 1. Related works carried out by various researchers is presented section 2. Introduction to divisible load scheduling scheme and round robin scheduling, its advantages and disadvantages, usage of these schemes in cloud environments are discussed in section 3. Proposed method is elaborately discussed in section 4. Section 5 lucidly presents the experimental results and discussion. Paper ends by giving conclusion remarks in section 6.

Related Work:

Dhinesh Babu L D and Venkata Krishna R[12] presented a paper on load balancing in cloud computing environments using honey bee behaviour. The proposed method achieves good load balancing across virtual machines which maximizes the throughput. Load is balanced according to the priorities of tasks so that waiting time of tasks in the queue is minimal. Compared to other traditional methods, this method is effective and overall execution time is improved. This method is well suited for heterogeneous cloud computing environments on non-preemptive independent tasks.

Brototi Mondal et al[13] proposed stochastic hill climbing based load balancing algorithm in cloud computing environments. It is a soft computing based load balancing algorithm in which resources allocation for incoming jobs either to servers or to virtual machines are allocated based on local optimization properties of stochastic hill climbing policy. This method is better compared to round robin and first come first server algorithms.

Koushik Dasgupta et al[14] presented a paper on efficient load balancing strategy in cloud computing using genetic algorithm. The usage of genetic algorithm allows balancing of load in cloud environment and minimizes the make span of given tasks. The results of this method are compared with traditional techniques viz. First Come First Serve, Round Robin and Local Search Stochastic Hill Climbing algorithm. The simulation results of this method outperform the traditional methods. This method also guarantees the QoS requirements of the user request.

Kun Li et al [15] proposed cloud load balancing algorithm using ant colony optimization techniques. Each ant chooses a virtual machine for its task. Local pheromone is updated whenever an ant completes the tour. The current optimal solution can be used to update the global pheromone and it is continued till all the ants finish the tour. The proposed method is compared with FCFS and ACO algorithms, and it shows that this method can handle different task conditions.

Shu-Ching Wang et al [16] proposed load balancing algorithm for three-level cloud computing network. This method combines scheduling algorithms. OLB algorithm makes every node in working state and LBMM algorithm provides minimum execution time on the node of each task and minimum completion time. This load balancing algorithm provides better manipulation of cloud resources and this algorithm is dynamic.

From the above, it is clear that already enough research work is going on in load balancing schemes in cloud computing environments. Any method proposed further must be efficient compared with the above methods atleast.

Divisible Load Scheduling:

Divisible Load Scheduling (DLT) is a mathematical tool. It allows traceable performance analysis of systems involved in computation and communication parameters, generally seen in parallel and distributed computing environments [17]. DLT follows linear mathematical concept with rich features of easy computation, schematic language, easy network modeling, etc.

DLT assumes that computation and communication loads can be divided and passed arbitrarily to a number of processors, links [18]. Since mathematical formulation is used and there are no strong relationships among data, loads can be arbitrarily assigned to links and processors in the network. DLT is fundamentally deterministic, and concepts are applied directly to parallel and distributed processing [19].

In cloud computing, end users are not the direct owners of any resource or infrastructure. All the incoming requests are handled in parallel and care must be taken that load on various cloud infrastructure needs to be balanced. Here we are using DLT scheduling concept to distribute the load optimally among various service providers in the cloud.

3.1 Round Robin Load Balancing:

It is one of the best and simple methods of distributing client request across a group of servers. Round-Robin load balancing algorithm allocates the client request to each server in turn [20]. Algorithm allocates load to each server and when it reaches the end of the server list, it moves to top once again.

One of the variants of round robin method is weighted round robin method which assigns weights to each server based on certain criterion. The server with higher weight has larger portion of client requests it receives [21]. Another variant is dynamic round robin in which weights are assigned for servers dynamically using real time data about current server's load and free time [22].

Major advantage of round robin load balancing algorithm is that it can be implemented easily [23]. However, these algorithms may not result in most accurate and efficient distribution of workloads as they assume that all the servers are same in configuration, can handle the same load, same computing capacity with same storage[23]. This drawback leads to poor allocation of loads to the servers.

Proposed Method:

As stated above, divisible load balancing algorithms are good load balancers. On the other hand, weighted round robin method eliminates the starvation problem and implements priority based technique to allocate the load among available servers. Hence we have presented here the combination of DLT technique and weighted round robin technique names as Divisible Weighted Round Robin (DWRR) as proposed method in this paper. This method is implemented by dividing the task in to arbitrary sub tasks. These sub tasks can be executed sequentially and few are executed parallel. By this way, performance of network is increased and completion time of a request is decreased.

We assume that load balancer is connected too much number of servers and each server is assigned with the weight. The servers are having different computational speeds, configuration. The tasks are assigned to the powerful servers first and then to the subsequent servers with less weight. The servers are also identified using their previous allocation of tasks.

The proposed method works as follows: The request is arbitrarily divided into many sub tasks. Each server is assigned with their weights and previous allocation of tasks. The status of all the servers is made READY. These steps are preliminary done before getting customer request. When a request arrives to the cloud server, it is passed to the load balancer. Load balancer first divides the tasks to many number of arbitrary sub tasks using divisible load balancing scheme.

The load is assigned to the server with higher weight for all the sub tasks and status of the server is changed as BUSY. Now the server executes the task and returns back the results to load balancer. The status of server is once again changed to READY and weight of the server is updated.

Change of status from READY to BUSY helps that the particular server is no longer made available in the ready server list. By this way, overlapping of tasks is avoided. The server once again joins the list when it finishes the job. This is fair in the sense that each server will get a change as round robin method is applied.

Let *'t'* represents time. When *t=0*, all the servers are in idle state and when a request is supplied to load balancer, now *t=t1*. Let the reporting time of each server be T_{k_i} and load assigned to the sever is β_{k_i} constant that is inversely proportional to speed of server a_{k_i} communication constant as b_{k_i}

The total reporting time of a server is defined as

$$T_{1N} = t_1 + \beta_{1N} a_{1N} T_{ms} + \beta_{1N} b_{1N} T_{cm}$$

Where T_{ms} is the measurement intensity constant, T_{cm} is the total communication cost. The total measurement time for all the servers would be assumed a normalize unit load. If a server can handle $(1/k)$unit of load, then the total measurement time is expressed as

$$\beta_{11} + \beta_{12} + \ldots\ldots \beta_{1n} = \frac{1}{k}$$

4

Using equation (1) ,(2) can be rewritten as

$$\beta_{11} a_{11} T_{ms} = \beta_{12} a_{12} T_{ms} + \beta_{12} b_{12} T_{cm}$$

$$\beta_{12} a_{12} T_{ms} = \beta_{13} a_{13} T_{ms} + \beta_{13} b_{13} T_{cm}$$

A general expression for the above equations can be given as

$$\beta_{1j} = \prod_{j=2}^{i} S_{1_j} \beta_{11}$$

$$\beta_{11} + \sum_{i=2}^{N} \prod_{j=2}^{i} S_{1\ j}\ \beta_{11} = \frac{1}{k}$$

Substituting (3) and (4), we get

$$\beta_{1i} = \frac{\prod_{j=2}^{i} S_{1,j}}{k\left(1 + \sum_{i=2}^{N} \prod_{j=2}^{i} S_{1,j}\right)}$$

Hence for a load balancer, the load assigned to each of the server is represented as

$$T_{f_r} = t_1 + \frac{\left(a_{r_1} T_{ms} + b_{r_1} T_{cm}\right)}{k\left(1 + \sum_{i=2}^{N} \prod_{j=2}^{i} S_{1,j}\right)}$$

The minimum reporting time of a sever to a client request is calculated as

$$T_{f_1} = t_1 + \frac{\left(a_{r_1} T_{ms} + b_{r_1} T_{cm}\right)(1 - S_1)}{k(1 - S_1, N)}$$

Experimental Results & Discussion:

To analyze the efficiency of the proposed method, a simulation is performed using CloudSim tool. The parameters we consider for experiment are as follows:

Table 1: Parameters and their values

Parameters	Value
Number of Servers	08
Speed of Server	100MIPS
Number of Processor per Server	05
Memory of Each Server	2GB
Primary Memory of Each Serve	512MB
Architecture	X86
Operating System	Windows XP

For a good load balancing algorithm, it is important to test its performance parameters viz. processing time and overall response time, and compare it with the traditional methods. The elapsed time between the end of an inquiry and beginning of response is said to be overall response time. Generally it is expected that response time should be minimal. For our method also, the above performance parameter values are measured. Table 2 shows the response time of the proposed method

Table 2: Response Time

Method	Average (ms)	Min (ms)	Max (ms)
Round Robin	608.53	46.20	29,887.38
Weighted Round Robin	653.67	38.46	36,203.93
Our Method	563.08	45.78	28,146.31

The elapsed time between the submission of an inquiry and getting the result in cloud computing is said to be overall processing time. The table 3 represents the comparison of overall processing time of our method with traditional methods.

Table 3: Processing Time

Method	Average (ms)	Min (ms)	Max (ms)
Round Robin	288.17	2.97	38,930.83
Weighted Round Robin	258.85	1.26	42,632.49
Our Method	249.7	1.38	33,086.29

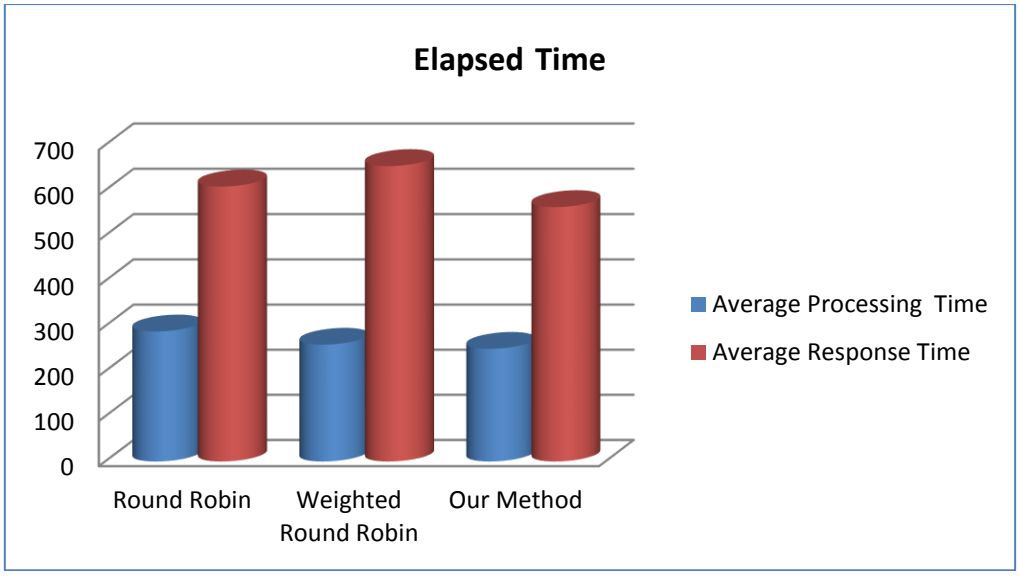

Fig. 1: Performance in time

The above graph shows the comparison of average response time and average processing time. It is viewed that the peaks representing our method are little bit less compared to other two methods. The proposed method is also tested for its efficiency with other performance metrics viz. overhead cost, throughput, performance and resource utilization. It is concluded that the proposed method is good and efficient.

Conclusion:

Cloud computing is perceived as the biggest part of today's information technology landscape. Apart from various advantages of cloud computing, the load on cloud server needs to be balanced such that no server should be overwhelmed with the load as well as without load for a long time. It is necessary for cloud storage to have load balancing mechanisms. Here we have presented a load balancing algorithm called Divisible Weighted Round Robin method which is better than other traditional methods viz. round robin and weighted round robin. DWRR is a combination of divisible load balancing and round robin methods. The efficiency of the proposed method is tested using CloudSim tool and it is observed that this method outperforms the other methods. The average response time and average processing time of this method is significantly less than other methods. The major contribution of this proposed method is to introduce a new algorithm for cloud load balancing using response time and processing time. In future work, we are going to test the same method with other efficiency parameters such as throughput and scalability.

REFERENCES

1. Xu, X., 2012. From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing, 28(1): 75-86.
2. Foster, I., C. Kesselman and S. Tuecke, 2001. The anatomy of the grid: Enabling scalable virtual organizations. International journal of high performance computing applications, 15(3): 200-222.
3. Calheiros, R.N., R.R anjan, A. Beloglazov, C.A. De Rose and R. Buyya, 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1): 23-50.
4. Zhu, W., C. Luo, J. Wang and S. Li, 2011. Multimedia cloud computing. IEEE Signal Processing Magazine, 28(3): 59-69.
5. JAMES, B., 2010. Security and privacy challenges in cloud computing environments.
6. Kaushik, A., 2013. Libraries Perception Towards Cloud Computing: A Survey. World Digital Libraries-An international journal, 6(1): 13-24.
7. Randles, M., D. Lamb and A. Taleb-Bendiab, 2010. April. A comparative study into distributed load balancing algorithms for cloud computing. In Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on (pp. 551-556). IEEE.
8. Zaharia, M., D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker and I.S toica, 2010. April. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In Proceedings of the 5th European conference on Computer systems (pp. 265-278). ACM.

9. Berlińska, J. and M. Drozdowski, 2011. Scheduling divisible MapReduce computations. Journal of Parallel and Distributed Computing, 71(3): 450-459.
10. Reddy, V.K., B.T. Rao and L.S.S. Reddy, 2011. Research issues in cloud computing. Global Journal of Computer Science and Technology, 11(11).
11. Chen, W.H., Y.C. Fang, J.F. Yao and W. Zhang, 2010. Multi-core based parallel computing technique for content-based image retrieval. Journal of Shanghai University (English Edition), 14: 55-59.
12. Krishna, P.V., 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing, 13(5): 2292-2303.
13. Mondal, B., K. Dasgupta and P. Dutta, 2012. Load balancing in cloud computing using stochastic hill climbing-a soft computing approach. Procedia Technology, 4: 783-789.
14. Hu, J., J. Gu, G. Sun and T. Zhao, 2010. December. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In 2010 3rd International symposium on parallel architectures, algorithms and programming (pp. 89-96). IEEE.
15. Li, K., G. Xu, G. Zhao, Y. Dong and D. Wang, 2011. August. Cloud task scheduling based on load balancing ant colony optimization. In 2011 Sixth Annual ChinaGrid Conference (pp. 3-9). IEEE.
16. Wang, S.C., K.Q. Yan, W.P. Liao and S.S. Wang, 2010. Towards a load balancing in a three-level cloud computing network. In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on 1: 108-113.
17. Drozdowski, M. and P. Wolniewicz, 2003. Out-of-core divisible load processing. IEEE Transactions on Parallel and Distributed Systems, 14(10): 1048-1056.
18. Mingsheng, S., 2008. Optimal algorithm for scheduling large divisible workload on heterogeneous system. Applied mathematical modelling, 32(9): 1682-1695.
19. Shokripour, A. and M. Othman, 2009. April. Survey on divisible load theory. In Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09. International Association of pp: 9-13.
20. Coulouris, G.F., J. Dollimore and T. Kindberg, 2005. Distributed systems: concepts and design. pearson education.
21. Acharya, Y.R., Advanced Micro Devices, 2006. System and method for dynamically updating weights of weighted round robin in output queues. U.S. Patent, 7,110,359.
22. Mangipudi, K. and V. Basani, Network Appliance, 2006. Method and apparatus for policy based class service and adaptive service level management within the context of an internet and intranet. U.S. Patent 7,124,188.
23. Chaczko, Z., V. Mahadevan, S. Aslanzadeh and C. Mcdermid, 2011. September. Availability and load balancing in cloud computing. In International Conference on Computer and Software Modeling, Singapore 14.