Middle-East Journal of Scientific Research 23 (8): 1734-1740, 2015
ISSN 1990-9233
© IDOSI Publications, 2015
DOI: 10.5829/idosi.mejsr.2015.23.08.22469

An Implementation of Load Balancing Algorithm in Cloud Computing Environment

[1]B. Kalai Selvi and [2]L. Mary Immaculate Sheela

[1]Mother Teresa Women's University, Kodaikanal, India
[2]R.M.D Engineering College, Chennai, India

Abstract: Cloud Computing is a new technique in Internet era which involves virtualization, distributed computing, networking, software and web services. A cloud consists of several elements such as clients, datacenter and distributed servers. Cloud Computing includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Major area we have to concentrate on cloud computing is the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. In this paper we analyzed various load balancing algorithms like Token Routing, Round Robin, Randomized etc., The main aim of this research paper is analyzing load balancing algorithms 'round robin', 'central queuing' and 'randomized' have been executed with various combinations of millions instructions per second (MIPS) vs. VM and MIPS vs. HOST. Already we have covered basic concepts in Cloud Computing in our previous paper. In this paper mainly we are concentrating on Load Balancing techniques and implementation of load balancing algorithms only.

Key words: Cloud Computing · Systems software

INTRODUCTION

Cloud Computing is the latest technology in today's trend. Cloud Computing, is providing computer resources as a service, is a technology revolution offering flexible IT usage in a cost efficient and pay-per-use way. Cloud computing enables innovation and it alleviates the need of innovators to find resources to develop, test and make their innovations available to the user community. Cloud computing makes use of a large physical resource pool in the cloud. Cloud Computing is being transformed by a new model. In this model, data and computation are operated somewhere in a cloud, which is some collection of data centers owned and maintained by a third party. Cloud computing refers to the hardware, systems software and applications delivered as services over the Internet [1].

Cloud System Components: Cloud computing system consists of three major components such as clients, datacenter and distributed servers. These components are shown in Figure 1. Brief discussion of specific role and purpose of each component is presented in the following[2]:

Client: End users interact with the clouds to manage information related to the cloud.

Datacenter: Datacenter is nothing but collection of servers hosting different applications. An end user connects to the datacenter to subscribe different applications. A datacenter may exist at a large distance from the clients[3].

Distributed Servers: A server, which actively checks the services of their hosts, known as Distributed server. Distributed servers are the part of a cloud which is available throughout the internet hosting different applications. While using the application from the cloud, the user would feel that he /she is using this application from its own machine [4].

Corresponding Author: L. Mary Immaculate Sheela, R.M.D Engineering College, Chennai, India.

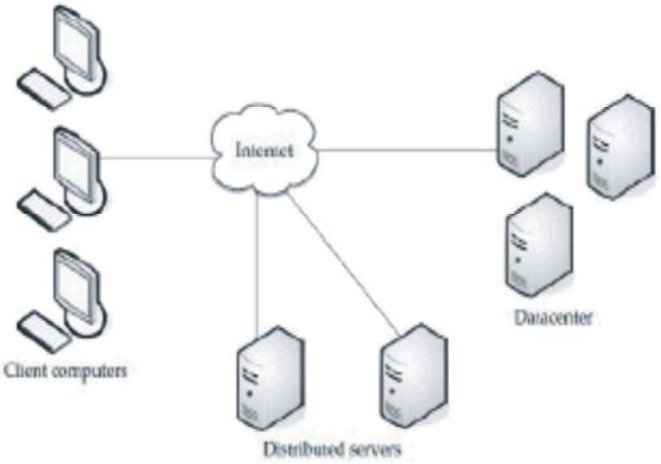

Fig. 1: Cloud System Components

Table 1: Comparative Study of Load Balancing Algorithms

Algorithm	Nature	Environment	Process Migration	Resource Utilization	Steadiness
Token Routing	dynamic	decentralized	possible	more	unstable
Round Robin	static	decentralized	Not possible	less	stable
Randomized	static	decentralized	Not possible	less	stable
Central Queuing	dynamic	centralized	Not possible	less	unstable
Least connection	dynamic	centralized	Not possible	less	stable

Virtualization Concepts: Cloud providers have managed to provide many levels of virtualization in their networks. For example, OS virtualization, kernel-level virtualization, files system Virtualization and , Network or I/O virtualization. This enhanced aspect means that many types of network and software configurations/ architectures can be ported to the virtualized world. With virtualization, applications and infrastructure are independent, allowing servers to be easily shared by many applications where applications are running virtually anywhere in the world. This is possible as long as the application is virtualized. Virtualizing the application for the cloud means to package the bits of the application with everything it needs to run, including pieces such as a database, a middleware and an operating system. This self-contained unit of virtualized application can then run anywhere in the world [5].

Load Balancing in Cloud Computing: Load balancing is the process of distributing the load among various resources in any system. Thus load need to be distributed over the resources in cloud-based architecture, so that each resources does approximately the equal amount of task at any point of time. Basic need is to provide some techniques to balance requests to provide the solution of the application faster. Cloud vendors are based on automatic load balancing services, which allow clients to increase the number of CPUs or memories for their resources to scale with increased demands. This service is optional and depends on the clients business needs. So load balancing serves two important needs, primarily to promote availability of Cloud resources and secondarily to promote performance [6].

Scalability of Load Balancing: Load balancing is the key to success for cloud architectures. It is capable of distributing the working processes evenly between 2 or more computers, so that resources can be used efficiently and therefore increases performance and availability. A so-called load balancer is automatically able to deal with different amount of work capacity by adapting its distribution decisions according to the moments a request is made. A load balancing solution is often used in internet services, where the idea of load balancing is run by an application. The ability to scale out while maintaining the desired level of service is visualized through the help of load balancing. Load balancing provides availability to services within single or multi-cloud environments. They distribute the load among redundant servers based on the traffic load [7].

Objectives of Load Balancing Algorithms

- Cost effectiveness: primary aim is to achieve an overall improvement in system performance at a reasonable cost.

- Scalability and flexibility: the distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.
- Priority: prioritization of the resources or jobs need to be done on before hand through the algorithm itself for better service to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their origin [8].

Algorithms-Load Balancing: Token Routing: The main objective of the algorithm is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents can not have the enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbor's working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no communication overhead is generated [9].

Round Robin: In this algorithm, the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally [10].

Randomized: Randomized algorithm is of type static in nature. In this algorithm a process can be handled by a particular node n with a probability p. The process allocation order is maintained for each processor independent of allocation from remote processor. This algorithm works well in case of processes are of equal loaded. However, problem arises when loads are of different computational complexities. Randomized algorithm does not maintain deterministic approach. It works well when Round Robin algorithm generates overhead for process queue.

Central Queuing: This algorithm works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it. When a processor load falls under the threshold then the local load manager sends a request for the new activity to the central load manager. The central manager then answers the request if ready activity is found otherwise queues the request until new activity arrives.

Connection Mechanism: Load balancing algorithm can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it and decreases the number when connection finishes or timeout happens [11].

Simulation in Cloud: Cloudsim: Resources and software are shared on the basis of client's demand in cloud environment. Essentially, dynamic utilization of resources is achieved under different conditions with various previous established policies. Sometime it is very much difficult and time consuming to measure performance of the applications in real cloud environment. In this consequence, simulation is very much helpful to allow users or developers with practical feedback in spite of having real environment[12].

Cloud Simulator-Cloudsim: Users are capable of accessing shared resources through utilizing available public cloud platform. However, accessing real cloud environment or public cloud is not always handy. Instead of the real environment, cloud simulator could facilitate the experiments. Simulation environment allows customers or users to tune the performance bottlenecks or evaluates different kinds of features under varying load distributions.

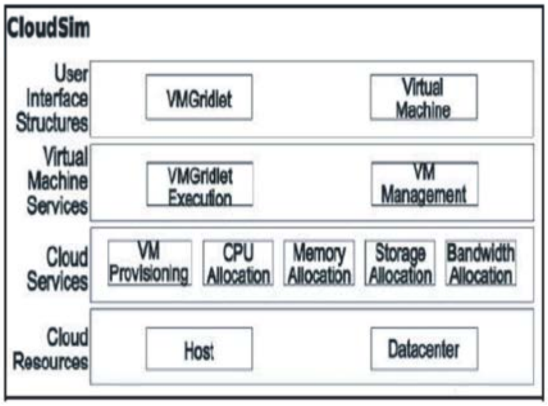

Fig. 2: CloudSim Architecture

The architecture of CloudSim comprises of four layers, as shown in Figure 2. At the bottom most layer, cloud resources are managed. During the simulation period, these core entities are instantiated and executed. On top of this layer, cloud services, like allocation of CPU, memory, storage and bandwidth are provided as dedicated management interfaces. Another two top most layers are virtual machine services and user interface structures. Virtual machine in user interface structures layer is responsible for physical host specifications such as number of machines and their configurations. CloudSim facilitates a specific host to be concurrently shared among different VMs based on user-defined QoS specifications. Next section presents proposed execution environment in order to analyse execution performance of existing load balancing algorithms[13].

Proposed Execution Environment: Equal load distribution may improve resource utilization by transferring load from heavily loaded server to the idle server. Existing scheduling algorithms estimate system parameters such as the job arrival rate, CPU processing rate and load on the processor for migrating jobs into least loaded processors in order to balance load. This research work considers Datacenter, Virtual Machine (VM), host and Cloudlet components from CloudSim for execution analysis of a few algorithms. Datacenter component is used for handling service requests. VM consist of application elements which are connected with these requests, so Datacenter's host components should allocate VM process sharing. VM life cycle starts from provisioning of a host to a VM, VM creation, VM destruction and VM migration. A brief description of these components and the working relationship between them is presented in the following:

Datacenter: Datacenter encompasses a number of hosts in homogeneous or heterogeneous configurations (memory, cores, capacity and storage). It also creates the bandwidth, memory and storage devices allocation.

Virtual Machine (VM): VM characteristics comprise of memory, processor, storage and VM scheduling policy. Multiple VM can run on single hosts simultaneously and maintain processor sharing policies [14].

Host: This experiment considers VM need to handle a number of cores to be processed and host should have resource allocation policy to distribute them in these VMs. So host can arrange sufficient memory and bandwidth to the process elements to execute them inside VM. Host is also responsible for creation and destruction of VMs.

Cloudlet: Cloudlet is an application component which is responsible to deliver the data in the cloud service model. So the length and output file sizes parameter of Cloudlet should be greater than or equal to 1. It also contains various ids for data transfer and application hosting policy. Experimental results of executing task in CloudSim are represented in the next section [15].

Cloudsim Execution: This research uses CloudSim-3.0 as a framework in the simulator environment. Implementation has been started with installation of simulation package CloudSim-3.0 on Windows XP (Service pack 3). There after Java version 7 is installed and classpath along with other necessary execution setup requirement is fulfilled. The minimum requirement of this experiment is VM memory of 1GB, VM bandwidth of 1000 and local operating system used as a host. In this simulation setup, three well-known load balancing algorithms 'round robin', 'central queuing' and 'randomized' have been executed with various combinations of millions instructions per second (MIPS) vs. VM and MIPS vs. HOST. Analysis is being carried out with respect to the response time as output. Figure 3, Figure 4 and Figure 5 represent various response time based with different combinations of MIPS vs. VM and MIPS vs. Host. It is observed for all the cases, response time is inversely proportionate with MIPS vs. VM and MIPS vs. Host. But optimum response time is achieved with same value of MIPS vs. VM and MIPS vs. Host. This execution analysis illustrates that nature of each simulation results are similar as this research currently concentrates on improvement of response time with similar setup.

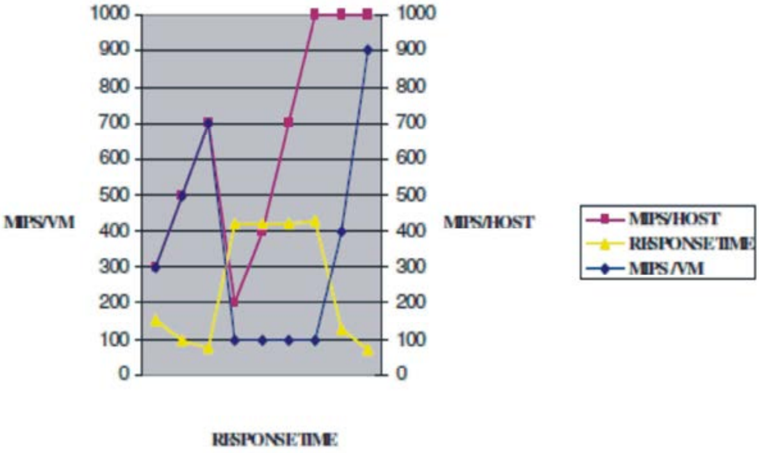

Fig. 3: Variation of response time in round robin algorithm

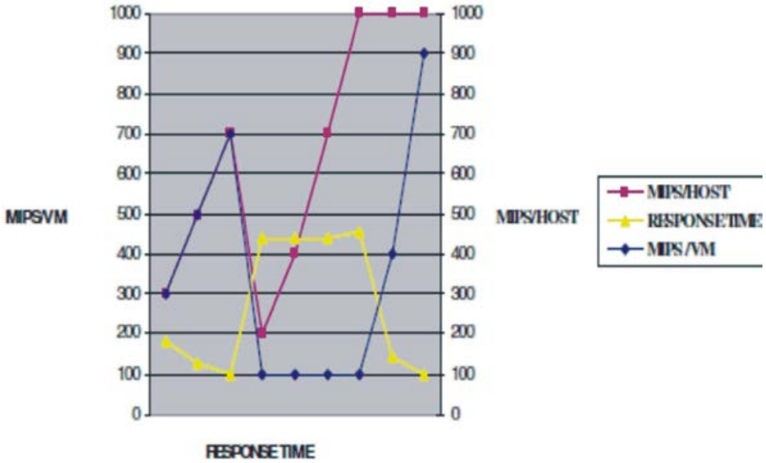

Fig. 4: Variation of response time in central queuing algorithm

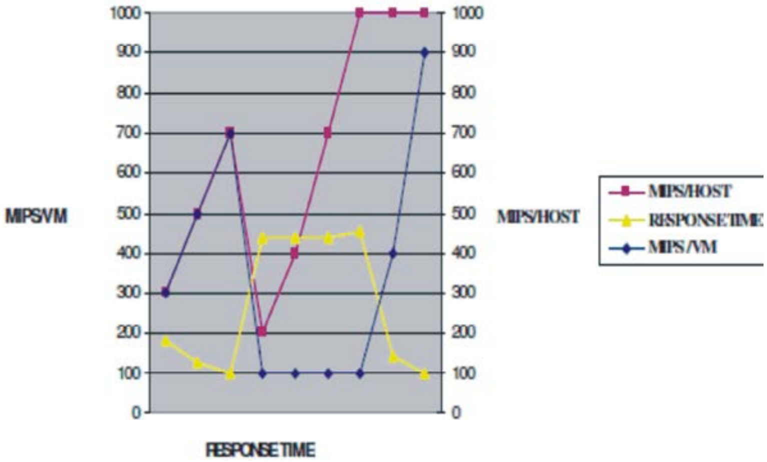

Fig. 5: Variation of response time in randomized algorithm

Related Work: Several algorithms have been developed and simulated in CloudSim environments for balancing and maintaining load in the cloud computing environment. Many of the existing researchers concentrate on virtual machine, hosts and response time specifically in order to balance load in cloud. These research works primarily focus on improving response time activity CloudSim environment mostly. The result indicates that the time and memory requirement is linear. Jaspret Kaur emphasizes a model to find the suitable VM within very short period of time when any request arrives. He suggests that the least loaded VM would be selected to handle the request and the id of that VM would be sent to the datacenter controller for further information processing requirement. In, Hsu has discussed an algorithm called active vm load balancer algorithm to find the suitable VM in a short time period.. He has stressed to count the maximum length of VM for the allocation of new request. If the length of the vm is not sufficient then a new VM would be added. After that all the VM's load needs to be counted and least loaded VM would be selected to handle the new request. Foster deals with the issues in a simulation environment efficiently and significantly outperform the existing approaches through experimental results. Their results demonstrate that load balancing algorithm proposed in can possibly improve the response time in order of magnitude with respect to number of VMs in Datacenter.

CONCLUSION

Cloud computing is a powerful new abstraction for large scale data processing systems which is scalable, reliable and available. In cloud computing, there are large self-managed server pools available which reduces the overhead and eliminates management headache.This paper presents a concept of Cloud Computing along with research challenges in load balancing. It also focus on merits and demerits of the cloud computing. Major thrust is given on the study of load balancing algorithm, followed by a comparative survey of these above mentioned algorithms in cloud computing with respect to stability, resource utilization, static or dynamicity, cooperative or non-cooperativeness and process migration. This paper aims towards the establishment of performance qualitative analysis on existing VM load balancing algorithm and then implemented in CloudSim and java language. Execution analysis of the simulation shows that change of MIPS will effect the response time. Increase in MIPS vs. VM decreases the response time. It is observed with thorough study that, load balancing algorithm works on the principle on which situation workload is assigned, during compile time or run time. Cloud computing services can also grow and shrink according to need. Cloud computing is particularly valuable to small and medium businesses, where effective and affordable IT tools are critical to helping them become more productive without spending lots of money on in-house resources and technical equipment. Also it is a new emerging architecture needed to expand the Internet to become the computing platform of the future.

REFERENCES

1. Alstom, 2010. About Us. Alstom Homepage. Available at: http://www.alstom.com/. Accessed on: 5 April 2010.
2. Amazon, 2009. About Amazon Web Services. Amazon Web Services Homepage. Available at: http://aws.amazon.com/. Accessed on: 25 February 2010.
3. Armbrust, M., A. Fox and R. Griffith, 2009. Above the Clouds: A Berkeley View of Cloud Computing. Electrical Engineering and Computer Sciences University of California at Berkeley, pp: 1-8.
4. Bhathiya and Wickremasinghe, 2010. Cloud Analyst: A Cloud Sim-based Visual Modeller for Analysing Cloud Computing Environments and Applications.
5. Hsu, C.H. and J.W. Liu, 2010. Dynamic Load Balancing Algorithms in Homogeneous Distributed System, Proceedings of The 6th International Conference on Distributed Computing Systems, pp: 216-223.
6. CloudSim, 2011. A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, () available from: http://www.cloudbus.org/cloudsim.
7. Jaspreet kaur, 2012. Comparison of load balancing algorithms in a Cloud, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com, 2(3): 1169-1173.
8. Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud Computing and Grid Computing 360-Degree Compared, University of Chicago, pp: 10-56.
9. Gartner, 2008. Gartner Says Cloud Computing Will Be As Influential As E-business. Gartner press release, 26 June 2008. http://www.gartner.com/it/page.jsp?id=707508. Retrieved 3rd May 2010.
10. Jensen, M., J.O. Schwenk, N. Gruschka and L.L. Iacono, 2009. On Technical Security Issues in Cloud Computing. In IEEE International Conference on Cloud Computing (CLOUD-II 2009), Bangalore, India, September 2009, pp: 109-116.

11. Qamar, S., N. Lal and M. Singh, 2010. Internet Ware Cloud Computing: Challenges. (IJCSIS) International Journal of Computer Science and Information Security, 7(3).
12. Mladen A. Vouk, 2008. Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces.
13. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, 2010. Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition.
14. Martin Randles, David Lamb and A. Taleb-Bendiab, 2010. A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.
15. Caryer, G., T. Rings, J. Gallop, S. Schulz, J. Grabowski, I. Stokes-Rees and T. Kovacikova, 2009. Grid/cloud computing interoperability, standardization and the Next Generation Network (NGN), in 13th IEEE International Conference on Intelligence in Next Generation Networks, pp: 1-6.