COMPARISON & ANALYSIS OF LOAD BALANCING ALGORITHMS IN CLOUD ENVIRONMENT

[1]SURABHI GUPTA, [2]UDAY MOGHE, [3]UTKARSH ATRE, [4]YASHASVI WADHWANI

[1,2,3,4]B.E., Department of Computer Science and Engineering, Acropolis Technical Campus, Indore
E-mail: sur.gupta93@gmail.com[1], udaymoghe8@gmail.com[2], utkarsh987atre@gmail.com[3], yashuwadhwani29@gmail.com[4]

Abstract: Cloud Computing is a term referred for the services provided by the third parties to the users which are flexible and on demand self services. It is basically based on the distributed system architecture where the servers included in the service, requires an efficient load balancing technique in order to provide uninterrupted responses. Load balancing algorithms decides the allocation of the server as per the process. Thus, efficient load balancing is a major challenge encountered by every cloud service provider. This load balancing can be achieved through existing scheduling algorithms. The aim of this paper is to analyze the efficiency of these existing algorithms as compared to the proposed queuing algorithm.

Keywords: Cloud Computing, Load balancing, Algorithm

I. INTRODUCTION

As computer network evolved distributed computing changed the era of IT sector. Cloud computing can be referred as the advanced version of distributed system. It provides various services such as Virtual Hardware, Software and Storage which can be accessed by anyone anywhere on leased basis. It is basically based on 'on-demand self service'. Any organization which deploys cloud has to endure many challenges such as weigh cost, availability of servers and communication overhead before launching the services publically. Google App Engine, Microsoft Azure, Aneka are some of the major cloud deployment tools provided worldwide by leading companies. Hence the availability is the major issue for a public cloud because of the numerous clients which constantly connects to the servers.

Load balancing is a technique which allocates the servers to the clients request in an orderly fashion which reduces overall load on a specific server and distributes it among all servers equally. It reduces the availability issue which may result in the 'denial of service' situation in case of an overloaded server. Thus load balancing serves two main motives, first to promote performance and second to promote availability of cloud resources.

The purpose of this paper is not only to study the existing scheduling algorithms but also to suggest and compare a new technique which can enhance the efficiency of load balancing in order to fulfill the requirement of each client.

II. ELEMENTS OF CLOUD

A cloud computing environment consists of several components. The efficiency of these components contributes in the overall efficiency of cloud system. The three major components are clients, datacenters and distributed servers as shown in the figure 1. These can be briefly described as:

- Clients: Clients are the end user who connects to the cloud via internet and use their services. Clients can be classified in the following categories:
  - Thin Clients: Thin clients do not have their own memory; hence they do not perform any computational work.
  - Thick Clients: Thick clients have their own memory, they perform computational work.
  - Mobile: Smart phones available now-a-days.
- Datacenters: The storage servers which host different application are called as datacenters. Client connects to these datacenters in order to access any data or application.
- Distributed Servers: A server which host different applications which are available throughout the internet. The user, when use this application, feels as if the application is running on his/her own environment.

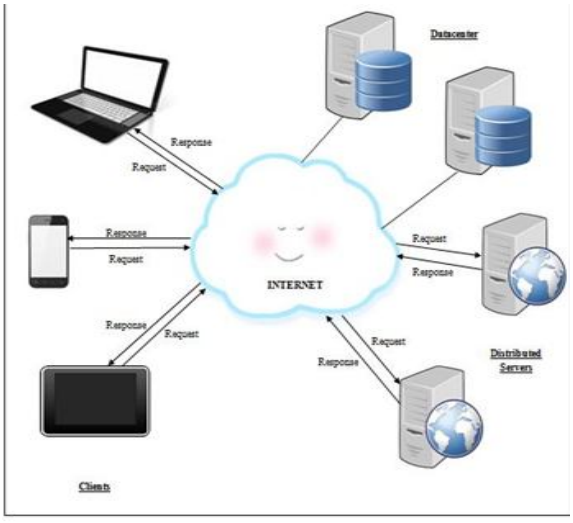

Figure 1: Elements Of Cloud

III. CLOUD DEPLOYMENT MODEL

- Public Cloud: The resources offered as service, over an internet connection for a pay-per usage fee, comes under Public cloud. Clients need not purchase any hardware to get these services.
- Private Cloud: The resources which are managed by an organization and deployed inside a firewall come under a private cloud. The organization owns the hardware and controls access to its resources.
- Community Cloud: The resources offered as services which are of a common interest for a group of organization comes under a community cloud.
- Hybrid Cloud: The combination of public cloud and private cloud is termed as hybrid cloud.

IV. CLOUD REFERENCE MODEL

- Software-as-a-Service (SaaS): The applications provided to users which run on the service providers environment comes under software as a service. Google Apps are one of the examples of SaaS providers.
- Platform-as-a-Service (PaaS): The platform provided on which the clients create the software using tools and libraries comes under platform as a service. For example clients run their applications on Google App Engine.
- Infrastructure-as-a-Service (IaaS): The infrastructure provided to the clients over the internet that is CPU cycles and storage comes under infrastructure as a service. Amazon Elastic Compute Cloud (EC2) is one of the examples of IaaS provider.

V. TOOLKIT: CLOUDSIM

Mimicking the actual environment is called a simulation. It is used for the benefit of the research. The users analyze the proposed design or algorithm through simulation.

CloudSim is a simulation platform which provides a fake cloud environment instead of the real cloud, which could facilitates the experiments. The different kinds of functionalities of CloudSim are as follows:

- Support for modeling and simulation of large scale Cloud computing data centers
- Support for modeling and simulation of virtualized server hosts, with customizable policies for provisioning host resources to virtual machines
- Support for modeling and simulation of energy-aware computational resources
- Support for modeling and simulation of data center network topologies and message-passing applications
- Support for modeling and simulation of federated clouds
- Support for dynamic insertion of simulation elements, stop and resume of simulation
- Support for user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines

VI. EXISTING ALGORITHMS

- Token Routing: In this algorithm tokens are moved around the system in order to minimize the system cost . Heuristic approach can be used to remove the drawback of token routing algorithm. It is based on load balancing. Token routing algorithm provides fast and efficient routing decision. Hence no communication overhead is generated in this approach.
- Round Robin: All the processes are divided between all the processors in this algorithm [5]. Each process is allocated to the processor in a round robin fashion. The job processing time is not same for different processes so some nodes may be heavily loaded and others remain ideal at any point of time.
- Randomized: A particular node ‘n’ with a probability ‘p’ can handle a process in this algorithm [5]. It is static in nature. Independent allocation from remote processor, the process allocation is maintained for each process. It is best suited when round robin algorithm generates overhead for process queue.
- Central Queuing: Dynamic distribution is the basic principle of this algorithm [6]. Each new process arriving at the manager is inserted into the queue. The activities are allotted to the servers in a first come first serve fashion.
- Connection Mechanism: This is based on dynamic scheduling algorithm [7]. It estimates the load over a server by counting number of connections dynamically. It maintains this count in a record and allocates the process henceforth.

VII. PROPOSED ALGORITHM

There are many algorithms used as load balancers in order to distribute traffic among the servers in a cloud. But most commonly only one algorithm is used at a time. This paper proposes a technique which is a combination of three algorithms which in turn could enhance the performance of a load balancer. Resource utilization may be improved by equal load distribution.

This can be done by transferring the load to the idle server from the heavily loaded server. The existing parameters such as CPU processing rate, job arrival rate and load on the processor. This research work shows an efficient use of three existing algorithm which altogether would distribute load among the servers very accurately.

In this strategy we used the concept of Multi Level Queue. A multi level queuing algorithm partitions the queue into several separate queue applying different algorithms over them. Each queue has its own priority over other containing its own scheduling algorithm. The processes are given to these queues in a First-come-first-serve fashion.

Using this concept we arranged two queues with two different algorithms (Randomize and Connection Mechanism respectively). Any process or activity when enters in a cloud environment, it is given to one of the above two queues as per the FCFS (First come first serve) algorithm. These two queues then allocate the server to these activities as per their protocols. This strategy is more efficient than the individual algorithms.

Virtual Machine (VM), Datacenters, cloudlet and host are some of the components of the CloudSim which were used during this research work. These can be briefly described as follows:

- Virtual Machine (VM): Virtual machine characteristics comprises of storage, processor, memory and VM scheduling. A single host can run multiple virtual machines simultaneously creating a cloud environment.
- Datacenters: Hosts in heterogeneous and homogeneous configuration encompasses in Datacenters. It also creates storage device, memory and bandwidth allocation.
- Cloudlet: The data in the cloud service model is delivered with the help of an application component known as cloudlet. It also contains ids for various applications hosting policy and data transfer.

A comparative study has been done on some of the algorithm on the basis of their nature, environment, process migration, resource utilization and steadiness. These algorithms were simulated on CloudSim tool. Our proposed algorithm is expecting the results, depicted in a table below:

Load balancing algorithm	Nature	Environment	Process migration	Resource utilization	Steadiness
Multi level Queuing	Dynamic	Decentralized	Not Possible	More	Stable
Token Ring	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

VIII. RELATED WORK

As previously proposed algorithms for load balancing, most of them emphasized on equal distribution of load which may improve utilization of resource by transferring load among servers from heavily loaded server to idle server. Existing algorithms analyze system on some parameters like job arrival rate, CPU processing rate, migrating load for equally balancing load on all servers.

In, Rodrigo presents an algorithm which results in increase in time delay as the number of clients' increases. Warstein suggested a model which finds a best suited VM within short period of time when any client request arrives. Kaur gave the same algorithm but focused on counting the length of VM for allocation of new request. If length of VM is not sufficient then a new VM is added and the least loaded VM handles the new request.

CONCLUSION AND FUTURE WORK

The aim of this research was to analyze the challenges in the load balancing in cloud computing. Major focus was on the study of load balancing algorithms, followed by a proposed strategy to enhance the efficiency of load balancing. This paper also shows the comparative study of these algorithms on the basis of key features like resource utilization and process migration. The analysis on the basis of the simulation shows that the proposed strategy is more stable and provides more resource utilization as compared to individual algorithms. Also the observation shows that the load balancing algorithm works on the situation assigned workload design.

REFERENCES

[1] Michael Armbrust, Armando Fox, Gunho Lee, Ion Stoica(2009) "Above the Clouds :A Berkeley View of Cloud Computing" University of California at Berkeley Technical Report No. UCB/EECS-2009-28

[2] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, (2011) available from: http://www.cloudbus.org/cloudsim/

[3] Carnegie Mellon, Grace Lewis(2010) "Basics About Cloud Computing" Software Engineering Institute September

[4] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid (2011)"Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press,Singapore 2011

[5] Zhong Xu, Rong Huang,(2009)"Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[6] M.Beltran, A.Guzman and J.L. Bosque(2011), "Dealing withheroginity in clusters" in proceeding of the Fifth

International Symposium on Parallel and Distributed Computing, ISPDC.

[7] P.Warstein, H.Situ and Z.Huang(2010), "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE

[8] http://siber.cankaya.edu.tr/OperatingSystems/ceng328/node126.html

[9] Jaspreet kaur (2012),"Comparison of load balancing algorithms in a Cloud" International Journal ofEngineering Research and Applications(IJERA) ISSN: 2248-9622 www.ijera.comVol. 2, Issue3,pp.1169-1173

[10] Rodrigo N. Calheiros Rajiv Ranjan, César A. F. De Rose and Rajkumar Buyya1(2011) : CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services Grid Computing and Distributed Systems (GRIDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia.

★★★