International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 1 ǁ January. 2013 ǁ PP.60-63

Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing

[1]Nayandeep Sran, [2]Navdeep Kaur

[1]*Information Security PEC University of Technology Chandigarh, India*
[2]*Assistant Professor Information Technology PEC University of Technology Chandigarh, India*

ABSTRACT: *Cloud computing is a new technology and it is becoming popular because of its great features. In this technology almost everything like hardware, software and platform are provided as a service. A cloud provider provides services on the basis of client's requests. An important issue in cloud is, scheduling of users requests, means how to allocate resources to these requests, so that the requested tasks can be completed in a minimum time and the cost incurred in the task should also be minimum. A good scheduling technique also helps in efficient utilization of the resources. Many scheduling algorithms have been studied like honeybee foraging algorithm, biased random sampling, active clustering, OLB + LBMM, Min-Min, Max-Min, etc. In this paper the various scheduling techniques have been discussed and their comparison has been done on the basis of metrics.*

Keywords—*Honeybee foraging algorithm, Biased random sampling, Active clustering, OLB+LBMM, Max-Min*

1. INTRODUCTION

As the IT technologies are growing day by day, the need of computing and storage are rapidly increasing. To invest more and more equipments is not an economic way for an organization to satisfy the even growing computational and storage need. So Cloud Computing has become a widely accepted paradigm for high performance computing, because in Cloud Computing all type of IT facilities are provided to the users as a service. In Cloud Computing the term Cloud is used for the service provider, which holds all types of resources for storage, computing etc. Mainly three types of services models are provided by the cloud. First is Infrastructure as a Service (IaaS), which provides cloud users the infrastructure for various purposes like the storage system and computation resources. Second is Platform as a Service (PaaS), which provides the platform to the clients so that they can develop, and deploy their applications on this platform. Third is Software as a Service (SaaS), which provides the software to the users and hence the users don't need to install the software on their machines and they can use the software directly from the cloud [6].

Cloud Computing provides many benefits [6]: it results in cost savings because there is no need of initial installation of much resource; it provides scalability and flexibility, the users can increase or decrease the number of services as per requirement; maintenance cost is very less because all the resources are managed by the Cloud providers, basically our model is a step towards green computing.

II. ISSUES

As cloud computing is in its evolving stage, so there are many problems prevalent in cloud computing [2][6]. Such as:

I. Ensuring proper access control (authentication, authorization, and auditing)
II. Network level migration, so that it requires minimum cost and time to move a job
III. To provide proper security to the data in transit and to the data at rest.
IV. Data availability issues in cloud
V. Legal quagmire and transitive trust issues
VI. Data lineage, data provenance and inadvertent disclosure of sensitive information is possible

And the most prevalent problem in Cloud computing is the problem of load balancing. Further, while balancing the load, certain types of information such as the number of jobs waiting in queue, job arrival rate, CPU processing rate, and so forth at each processor, as well as at neighboring processors, may be exchanged among the processors for improving the overall performance. For this purpose various types of algorithms have been proposed and in this paper we have tried to find the problems in the existing algorithms on the basis of some common criteria which we have termed as metrics [3]. The following figure 1 [6] shows the issues which are existing in cloud computing and we can see that the issues of performance, availability etc. are due to lack of proper load balancing algorithms.

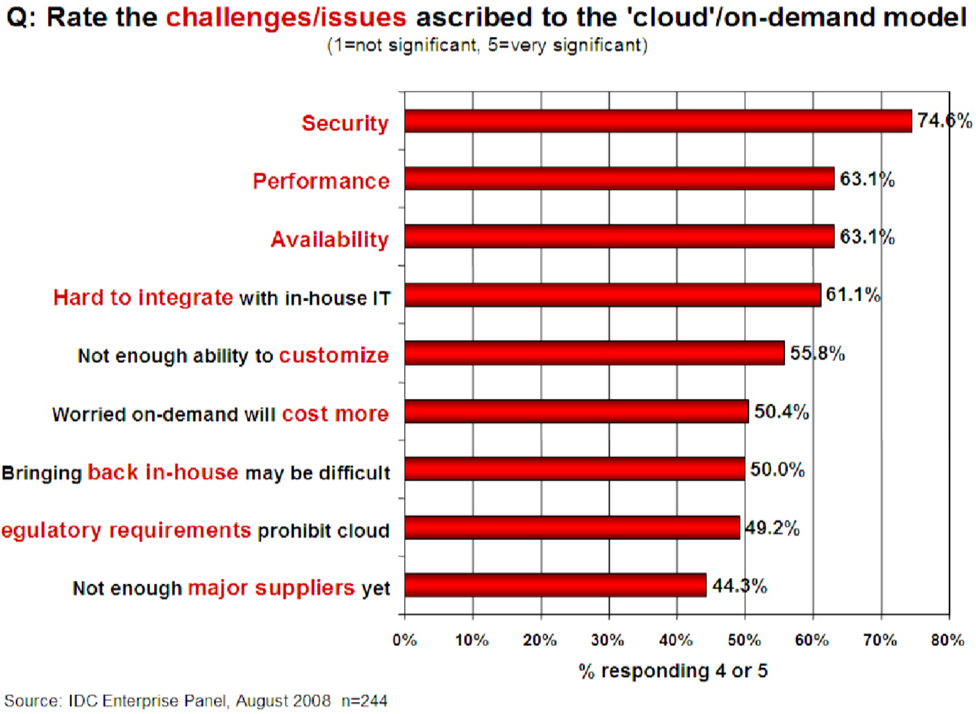

Figure 1. Issues in cloud computing

2. Metrics Of Load Balancing

In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning etc[4][5]. In this paper we have considered various metrics in existing load balancing techniques in cloud computing, which are discussed below:

3.1 Throughput is used to calculate the no. of tasks whose execution has been completed.
3.2 Overhead Associated determines the amount of overhead involved while implementing a load-balancing algorithm. It includes overhead due to movement of tasks, inter-processor and inter-process communication.
3.3 Fault Tolerance is the ability of an algorithm to perform uniform load balancing in case of link failure. The load balancing should be a good fault-tolerant technique.
3.4 Migration time is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.
3.5 Response Time is the amount of time taken to respond by a particular load balancing algorithm in a distributed system.
3.6 Resource Utilization is used to check the utilization of resources.
3.7 Scalability is the ability of an algorithm to scale according to the requirement.
3.8 Performance is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

III. SCHEDULING TECHNIQUES

In this paper we will discuss the various algorithms in chronicle order [2] and then we will study the difference between these algorithms on the basis of above metrics involved in load balancing.

IV. HONEYBEE FORAGING ALGORITHM

The main idea behind the algorithm is derived from the behavior of honey bees for finding and reaping food. M. Randles et al. [1][5] proposed a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. In this case the servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each Server processing a request from its queue calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance. If this profit was high, then the server stays at the current virtual server otherwise then the server returns to the forage. The algorithm performs as the system diversity increases. But it has a big disadvantage that it does not increase the throughput as the system size increases.

4.2 Biased Random Sampling

M. Randles et al. [2] investigated a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. Here a virtual graph is constructed, with the connectivity of each node (a server is treated as a node) representing the load on the server. Each server is symbolized as a node in the graph, with each indegree directed to the free resources of the server. The load balancing scheme used here is fully decentralized, thus making it apt for large network systems like that in a cloud. The performance is degraded with an increase in population diversity.

4.3 Active Clustering

Active Clustering works on the principle of grouping similar nodes together and working on these groups. The performance of the system is enhanced with high resources thereby in-creasing the throughput by using these resources effectively. It is degraded with an increase in system diversity [2].

4.4 OLB + LBMM

S.-C. Wang et al. [4] proposed a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. This combined approach helps in an efficient utilization of resources and enhances the work efficiency. It gives the better results than the above discussed algorithms.

4.5 Join-Idle-Queue

This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. Y. Lua et al.[3] proposed a Join-Idle-Queue load balancing algorithm for dynamically scalable web services. It effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time. It can perform close to optimal when used for web services. However, it cannot be used for today’s dynamic-content web services due to the scalability and reliability.

4.6 Min-Min Algorithm

It begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources. But this approach has a major drawback that it can lead to starvation [7].

4.7 Max-Min Algorithm

Max-Min is almost same as the min-min algorithm except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. Then according to that maximum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines[7].

Based on metrics discussed in section 3, the existing load balancing techniques have been compared in figure 1

Metrics	Honeybee Scheduling	Biased random Sampling	Active clustering	OLB+ LBMM	Join Idle Queue	Min-min	Min-max
Throughput	**No**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
Overhead	**No**	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**
Fault tolerance	**No**	**No**	**No**	**No**	**No**	**No**	**No**
Migration Time	**No**	**No**	**Yes**	**No**	**No**	**No**	**No**
Response Time	**No**	**No**	**No**	**No**	**Yes**	**Yes**	**Yes**
Resource utilization	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**
Scalability	**No**	**No**	**No**	**No**	**No**	**No**	**No**
Performance	**No**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**Yes**

Figure 1 comparison of existing load balancing techniques

V. CONCLUSION & FUTURE WORK

Load balancing is one of the main challenges in cloud computing. It is required to distribute the dynamic local workload evenly across all the nodes to achieve a high user satisfaction and resource utilization ratio by making sure that every computing resource is distributed efficiently and fairly. So in this paper we have compared various algorithms of load balancing in Cloud Computing. And we have concluded that we can use a particular algorithm according to our requirement/need. But as we know that the Cloud Computing covers a very vast area, it is applicable to both small and large scale area but as we have concluded that none of the above algorithms satisfies the criteria. So there is a need to develop an adaptive algorithm which is suitable for heterogeneous environment and should also reduce the cost.

ACKNOWLEDGEMENT

First of all we would like to express our profound sense of gratitude towards my guide Mrs. Navdeep Kaur Asst. Professor, Department of Information Technology and Engineering, for her able guidance, support and encouragement throughout the period this work was carried out. Her readiness for consultation at all times, her educative comments, her concern and assistance even with practical things have been invaluable. We would also like to convey our sincerest gratitude and indebtedness to our entire faculty members and staff of the Department of Computer Science and Engineering, PEC University Of Technology, Chandigarh, who bestowed their efforts and guidance at appropriate times without which it would have been very difficult on my part to finish this work. A vote of thanks to my fellow batch mates for their friendly co-operation and suggestions.

REFERENCES

[1]. Ram Prasad Padhy, P Goutam Prasad Rao “LOAD BALANCING IN CLOUD COMPUTING SYSTEMS” Thesis from National Institute of Technology, Rourkela-769 008, Orissa, India May, 2011.

[2]. T.R.V. Anandharajan, Dr. M.A. Bhagyaveni” Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[3]. Ruchir Shah, Bhardwaj Veeravalli, Senior Member, IEEE, and Manoj Misra, Member, IEEE “On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 12, DECEMBER 2007.

[4]. A Book by O’ Reilly on “Cloud Security And Privacy”.

[5]. Sri Varsha Gorge, Virajith Jalaparti and Harini Vaidhyanathan “Multi-Tier Distributed Load Balancing” CS598RHC Literature Survey.

[6]. Wayne Jansen Timothy Grance” Guidelines on Security and Privacy in Public Cloud Computing” NIST Draft Special Publication 800-144.

[7]. T. Kokilavani J.J. College of Engineering & Technology and Research Scholar, Bharathiar University, Tamil Nadu, India” Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing” International Journal of Computer Applications (0975 – 8887) Volume 20– No.2, April 2011.