ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Load Balancing in Public Cloud

Renuka Joshi[1]
M.E. Research Student,
Computer Networking,
G.H.R.C.E.M, Ahmednagar, India

Sunita Nandgave[2]
Asst. Prof, ME Comp. Networking,
G.H.R.C.E.M, Wagholi,
Pune, India

Abstract: ***Cloud computing is the next major paradigm shift in information and communication technology (ICT). Today, contemporary society relies more than ever on the Internet and cloud computing. In the cloud computing environment Load balancing has an important impact on the performance. To make cloud computing more efficient and to improve user satisfaction good load balancing is required. In this paper a better load balancing model based on CURE clustering is introduced. This algorithm will help to create dynamic partitions. And assign the jobs to specific partition as per load. In contrast to existing system that suffers from cloud division rules in case of nodes which are present in different geographic locations.***

Keywords: Cloud computing, Load balancing, Cloud partitioning, Load status of cloud, CURE clustering.

I. INTRODUCTION

Cloud computing is most popular technology in the field of computer science. The cloud is changing our life as it provides users various new types of services. Users get these services from a cloud and not have to bother about the details. As per NIST definition of cloud computing, cloud computing is a model that enables ubiquitous and convenient network access to shared computing resources such as applications, servers, networks, storage and different services. Cloud computing also provides on-demand network access to the users. Cloud computing requires minimum management and very less interaction between user and service provider [1]. Day by day cloud computing is getting more importance in Internet world. Cloud computing is very efficient and scalable. But it is important to maintain the stability of processing huge number of jobs arriving in the cloud computing environment which is a complex problem and hence load balancing is becoming new area for research in cloud computing .

Load balancing is the technique that facilitates networks and resources by a maximum throughput with minimum response time [10]. There is a large amount of literature available on load balancing. Depending on who initiated the process, load balancing algorithms can be of three categories, as given below:

- Sender Initiated: If the load balancing algorithm is initialized by the sender.
- Receiver Initiated: If the load balancing algorithm is initiated by the receiver.
- Symmetric: It is the combination of both sender initiated and receiver initiated.

Depending on the current state of the system, load balancing algorithms can be characterized as static and dynamic [8].

- Static: Static load balancing schemes use a priori knowledge of the applications and statistical information about the system.
- Dynamic: In dynamic load balancing schemes all the decision making process are based on the current state of the system.

II. Related Work

There are different load balancing techniques proposed by the researchers over time to time which have some advantages over and vice-versa. There are many load balancing algorithms, such as Round Robin load balancing algorithm which is random sampling based. It means it selects the load randomly in case where some servers are heavily loaded or some are lightly loaded. In Throttled load balancing algorithm client first request the load balancer to check the right virtual machine which access that load easily and perform the operations which is give by the client or user. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation as it is completely based on virtual machine. In Equally Spread Current Execution Algorithm, processes are handled with priorities. Depending on the size of load it is distributed randomly to that virtual machine which is lightly loaded or can handle that task easily, take less time, and give maximized throughput .A spread spectrum technique is also used in which the load balancer spread the load of the job across multiple virtual machines.

In VectorDot algorithm the hierarchical complexity of the data centre and multidimensionality of resource load across servers, network switches, and storage in an fragile data centre that has integrated server and storage virtualization technologies is handled. VectorDot uses dot product to differentiate nodes based on the item requirements and helps in removing overload on servers, switches and storage nodes.Join-Idle- Queue load balancing algorithm can be useful for dynamically scalable web services. This algorithm makes use of distributed dispatchers. In this first idle processors are made available across all the dispatchers and then jobs are assigned to processors. It helps to reduce the average queue length of each processor. It removes the load balancing work from the critical path of request processing and effectively reduces the system load. Also there is no communication overhead at job arrivals and does not increase actual response time. Also these various load balancing algorithms can be enhanced by using Generic Gossip Protocol in a Large Cloud Environment. This protocol minimizes power consumption through server consolidation and satisfies a changing load pattern. It provides an efficient heuristic solution for load balancing. Adler [3] in his white paper has introduced Load balancing in cloud computing environment. He has also described various tools and techniques which are used for good load balancing in the cloud.

M. Randles et.al has proposed comparison of static and dynamic load balancing algorithms for cloud computing. The comparison is done by means of performance time and cost basis [9]. They concluded that as compare to the Round Robin algorithm, Equally Spread Current Execution algorithm and throttled load balancing algorithm work better. M. Randles et al. [9] also investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. In this algorithm local server actions are used to achieve global load balancing. System performance is improved with increased system diversity but throughput is not increased with an increase in system size. When there is diverse population of service types required this algorithm can work efficiently. A self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring was also studied by M. Randles et al. [9]. In this, resources were used effectively which result in enhancement of system performance and increased throughput.

Nishant et al.[6] has used a load balancing mechanism which is based on ant colony optimization technique and complex network theory in cloud computing federation in which the characteristic of Complex Network are taken into consideration. Penmatsa and Chronopoulos[13] has proposed a static load balancing strategy. The approach is based on game theory for distributed systems.

V.Nae et al.[4] has proposed an event-driven load balancing algorithm for real time massively multiplayer online games (MMOG).In this algorithm first capacity events is given as input and then analyzes its components in context of the resources and the global state of the game session. It generates the game session load balancing actions. The algorithm can scale up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.

Liu et al. [5] has proposed a lock-free multiprocessing load balancing solution. It can avoid the use of shared memory in unlike to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. In this Linux kernel is modified. In a multi-core environment by running multiple load-balancing processes in one load balancer this solution can improve the overall performance of load balancer.

Grosu et al.[11] has proposed load balancing strategy for the distributed systems which is based on game theory. In this non-cooperative game is used as distributed structure. The algorithm is compared with other traditional methods and the comparison proves that the algorithm is less complex and gives good performance result. Aote and Kharat [12] has proposed dynamic load balancing model which uses game theory. In this model the dynamic load status of the system is considered as the user who is making the decision in a non-cooperative game.

III. PROPOSED WORK

Based on the domain or environment in which clouds are used, clouds can be divided into 3 categories:

- Public Clouds
- Private Clouds
- Hybrid Clouds (combination of both private and public clouds)

The load balancing model for the public cloud has to balance number of nodes sharing distributed computing resources located at different geographic locations. In this model the public cloud is divided into several cloud partitions. The cloud environment is very large and complex; and this kind of cloud divisions helps to simplify the process of load balancing. A cloud partition is nothing but a small part of the public cloud obtained by dividing the public cloud on the basis of geographic locations. The public cloud consisting of small partitions is shown in Fig.1. In load balancing the incoming network traffic is distributed across multiple virtual machine instances. Load balancing is useful in following ways:

- Scale your application
- Support heavy traffic
- Detect unhealthy virtual machines instances
- Balance loads across regions
- Route traffic to the closest virtual machine

The main controller and the balancers in the cloud give the load balancing solution. Each cloud partition has a Load balancer (LB) associated with multiple nodes. There is a main controller system which manages all the load balancer called Load Balancer Manager (LBM). After partitioning the public cloud into different partitions, load balancing starts.

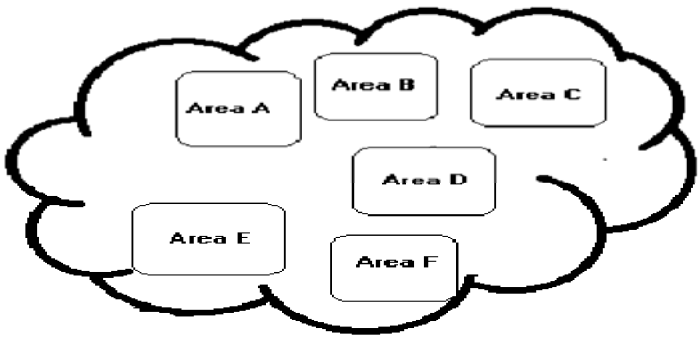

Fig.1 Different cloud Partitions in a big public cloud

Load Balancer Manager (LBM) is responsible for the following task:

- Receives the jobs from different end users.

- Choose a specific partition for the received jobs.
- Check the status of the cloud partition
- If the partition is heavily loaded by jobs the allocation will not be done, it means all nodes are overloaded already.
- The balancers present in each cloud partition collect the load status information from every node and then select the right strategy to distribute the jobs among the nodes.

The status of a cloud partition can be divided into three types:

- IDLE: In this, most of the nodes are in idle state.
- NORMAL: In this status, some of the nodes are in idle status while some others are overloaded.
- HEAVY: In this status of the cloud partition, most of the nodes are overloaded.

The main controller updates the load status information of the node by communicating with the balancers regularly after certain period of time. This evaluation of each node's load status is very important. First it is important to define the load degree of each node. The node load degree depends on various static parameters like the number of CPU's, the memory size, the CPU processing speeds, etc and dynamic parameters like network bandwidth, the CPU utilization ratio, memory utilization , etc. The load degree is calculated by using these parameters. By comparing with this load degree the status of a particular node is evaluated.

Idle:

If the Load degree of the particular node is zero then it means that the node is not processing any job so its status is Idle.

Normal:

If the Load degree of the particular node is greater than zero but less than the highest load degree which is set benchmark for different situations then the load status of that node is normal and it is capable of processing other jobs arriving in the cloud environment.

Overloaded:

If the Load degree of the particular node is greater than the highest load degree the node is not available and cannot receive jobs for processing until its load status becomes normal.

The balancers use these load degree results to calculate the partition status. Depending on load status different load balancing solution is provided to each partition. On the basis of current load strategy of a node the balancers will assign the job to it whenever a job arrives at a cloud partition. This strategy is continuously kept on changing by the balancers as the status of cloud partition changes.

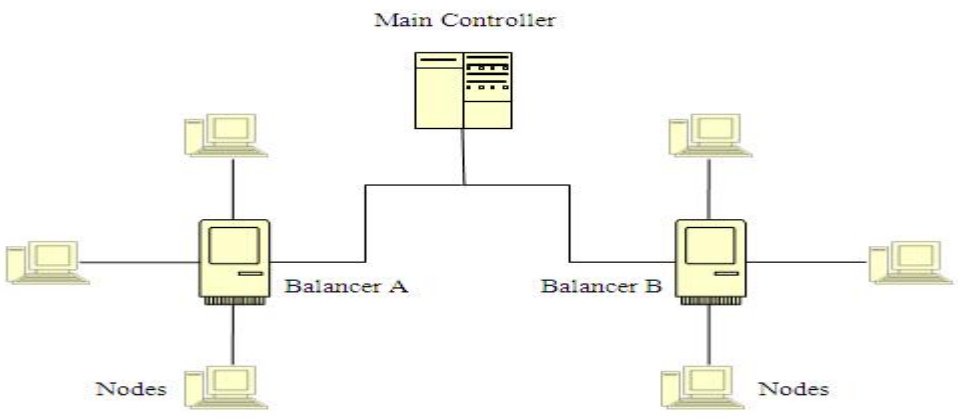

Fig. 2 Relationships between the main controllers, balancers, and the nodes

a) Load Balancing Strategy

If load balancing is good the performance of the entire cloud is improved. But, there is no common method that can be used for all possible different situations. For a partition which is in idle state simple method can be used and more complex method for the partition with normal state.

1) For the idle status:

In idle cloud partition many computing resources are free and heavily available but the job arriving frequency is low. So the cloud partition can process jobs quickly. In this case a simple load balancing method can work well. Different simple load balance algorithm such as the Weight Round Robin, the Dynamic Round Robin and the Random algorithm, can be used in this case. Round Robin is a simple algorithm. In this every new arriving request is passed to the next server present in the queue. The status of each connection is not recorded. In this algorithm there is a possibility that every cloud can be selected. But it do not work in case of Public cloud. In public cloud every node has different configuration and performance. Use of this algorithm in such case may overload some nodes resulting in inefficient performance of system. To overcome this problem an improved form of Round Robin algorithm is used, which is based on the load degree evaluation as explained above.

2) For the normal status:

In normal cloud partition jobs arriving frequency is more than in the idle state.This situation is more complex, hence for the load balancing different strategy is used. Each jobs is to be completed in the shortest time, so it is necessary to have a method that can complete all the jobs with reasonable response time as expected by the users.In cloud environment as imlementation of distributed system, the load balancing can also be imagined as a game.Game theory consist of cooperative games and non-cooperative games.In case of cooperative games decisions are made by comparing notes with each others.While in non cooperative games decisions are made only for the benefit of that particular decision maker. Dynamic load balancing model which is based on game theory can be used in case of non cooperative decision making problem.In this the dynamic load status of the system is considered as the user who is making the decision in a non-cooperative game.

IV. FUTURE WORK

For good load balancing cloud division is an important factor. It is not a simple problem to solve. The division rule should be based on the geographic location of nodes. A good cloud division methodology such as CURE clustering can used. A public cloud covers large geographic location. The nodes in particular a cluster may be far away from other nodes or there may be some more clusters present in the same geographic area but still very far apart.

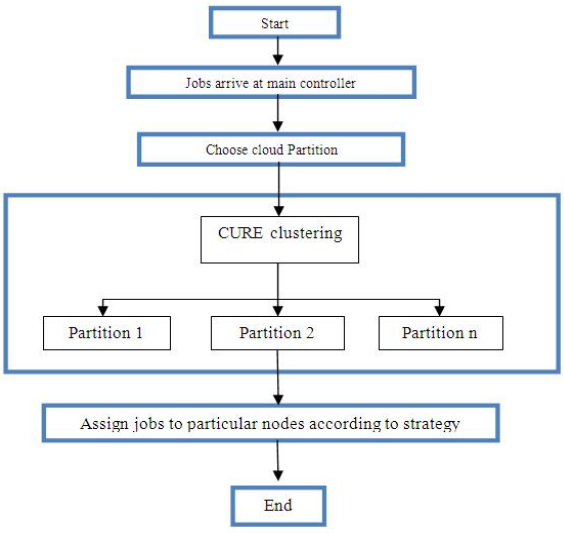

Fig.3 Load Balancing in cloud using CURE clustering.

In CURE a set of nodes in a same geograhic location is considered as a cluster. Randomly a cluster is selected and considered as a center. Each cluster is represented by a certain fixed number of points which are computed by selecting scattered points from the cluster.Then they are shrinked toward the cente of the cluster.These shrinked clusters are then used as representatives of the cluster. Shrinking helps to reduce the effects of outliners. In CURE algorithm combination of random sampling and partitioning is used. A random sample drawn from the node set is partitioned first and then each partition is clustered partially. The remaining partial clusters are then clustered in a second round to obtain the desired clusters.After this partitioning main controler will select the appropriate partition for arriving jobs while the balancer will choses the best load balancing strategy for each cloud partition.

V. CONCLUSION

In this paper we have studied the working of switching mechanism used for load balancing in public cloud. The system have cloud division problem which can be solved by introducing CURE clustering algorithm. Also various load balancing algorithms and techniques are given in this paper.

ACKNOWLEDGEMENT

We would like to thank all the authors of different research papers referred during writing this paper. It was very knowledge gaining and helpful for the further research to be done in future.

References

1. P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/ publications/nistpubs/800-145/SP800-145.pdf, 2012.
2. Gaochao Xu, Junjie Pang, and Xiaodong Fu "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" Tsinghua Science And Technology ISSN 1007 - 0214 04 /12 Volume 18, Number 1, February 2013, pp 34-39
3. B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/whitepapers/ Load-Balancing-in-the-Cloud.pdf, 2012
4. Nae V., Prodan R. and Fahringer T. (2010) 11th IEEE/ACM International Conference on Grid Computing (Grid), 9-17.
5. Liu Xi., Pan Lei., Wang Chong-Jun. and Xie Jun-Yuan. (2011) 3rd International Workshop on Intelligent Systems and Applications, 1-4.
6. K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, Load balancing of nodes in cloud using ant colony optimization, in Proc. 14th International Conference on Computer Modeling and Simulation (UKSim), Cambridge shire, United Kingdom, Mar. 2012, pp. 28-30.
7. S. Penmatsa and A. T. Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol. 71, no. 4, pp. 537-555, Apr. 2011.
8. T. Casavant, J.G. Kuhl, A taxonomy of scheduling in general-purpose distributed computing systems, IEEE Trans. Software Eng. 14 (2) (February 1988) 141–154.
9. 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 Applications Workshops, 2010, pp. 551–556.
10. R. Shimonski. Windows 2000 & Windows Server 2003 Clustering and Load Balancing. Emeryville. McGraw-Hill Professional Publishing, CA, USA (2003), p 2, 2003.
11. D. Grosu, A. T. Chronopoulos, and M. Y. Leung, Load balancing in distributed systems: An approach using cooperative games, in Proc. 16th IEEE Intl. Parallel and Distributed Processing Symp., Florida, USA, Apr. 2002, pp. 52-61.
12. S. Aote and M. U. Kharat, A game-theoretic model for dynamic load balancing in distributed systems, in Proc. The International Conference on Advances in Computing, Communication and Control (ICAC3 '09), New York, USA, 2009, pp. 235-238.
13. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shims, "CURE: AN EFFICIENT CLUSTERING ALGORITHM FOR LARGE DATABASES", *Information Systems* Vol. 26, No. 1, pp. 35-58,copyright 2001 Published by Elsevier Science Ltd. Printed in Great Britain 0306-4379/01.

AUTHOR(S) PROFILE

Renuka Joshi, M.E. Research Student of Computer Networking, from G.H.R.C.E.M, Ahmednagar, India, for academic year 2014-15. Presently working on new public cloud load balancing technique.

Prof. Sunita Nandgave, ME degree in Computer Networking, working as a Assistant Professor, in G.H.R.C.E.M, Wagholi, Pune, India from 2008 till date.