IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 4, Issue 9, September 2016

Dynamic Load Management in Cloud Computing Environment

Ajay A. Jaiswal[1], Dr. Renu Jain[2]

Associate Professor, Dept of Computer Technology, K.D.K. College of Engineering, Nagpur, Maharashtra, India[1]

H.O.D, Computer Science & Engineering, C.S.J.M. University, Kanpur, UP, India[2]

Abstract: The hidden significance of Cloud Computing is entering or storing or sharing the data on the web. In familiar scheme, cloud computing is scattering and rising day by day and become most significant factor in business both government or private industries. Though it also has some vital issues such as load management, security and fault tolerance to be resolved in order to progress reliability of cloud environment. This paper discuss on the load balancing concern in cloud computing. Many technique to resolve this load balancing problem has been projected such as Particle Swarm Optimization, Hash method, Ant Colony optimization, Pareto Distribution, Genetic algorithms and several scheduling based algorithms. In this paper we are recommend two algorithms to provide efficient load balancing in a multiple cloud environment. Both the proposed method has been implemented using Windows Azure Framework.

Keywords: Cloud Computing, Load Balancing, Comparison of load balancing algorithms

I. INTRODUCTION

Cloud computing is novel technology which is depending on internet in which internet can represented as a cloud. Cloud is a platform that offer resources like services, applications and storage network to a computers and strategy depend on pay-per- model [1].Many Cloud computing source have setup a few data centers at various geographical places over the web as a way to serve wants of their buyers around world [2].Today's this skill rising at a quick rate.

It has also facilitate to naturally access to linked data and applications from anywhere around the world. The basic point of using this skill is to expansion the performance and efficiency and reduces the cost [3]. The increases amount of information storage quickly in cloud computing environment.

The increase data storage very fast so the load balancing is a primary concern in cloud computing. When a numbers of jobs occur equal time then load balancing is main issue. Load balancing assist to work allocate between all to be had nodes to be certain that no node is overloaded and no want is free. Load balancing benefits to use of resources which too assistance in bettering the performance of the method in cloud system.

A. Cloud computing distinctiveness:

1) Services provide on demand: - When user want to resources then the cloud provide a services on demand.

2) Rapid Elasticity: - Number of various resources in cloud is increase or decrease easily

3) Resource Pooling: - In cloud structure resources are allocated reliable with consumer obligation. The each resource is communal to serve finish users utilizing model of cloud.

4) Broad Access to Network: - The cloud resources access is possible throughout the network and used standard methods for the users to right to use the network.

5) Pay per use: - all consumers pay charges when it is usage of computing resources

Figure1. Advantages of Cloud Computing

B. Cloud Exploitation Models:

The cloud set characterizes four cloud organization models. Fig.2 represents to various sorts of cloud.

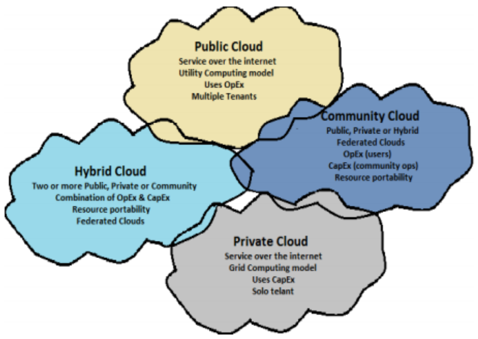

Figure 2: Cloud exploitation Models

DOI 10.17148/IJIREEICE.2016.4936

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 4, Issue 9, September 2016

1) Public Cloud: - This cloud is employ by the broad public users and the cloud service provider has the full liability for public cloud with it's possess qualities, policy, costing, profit, and charging model. Many accepted cloud services are Google App Engine, Amazon EC2and salesforce.com.

2) Private Cloud: - This cloud will be cloud bases effort for a solitary connection and give security to its resources.
3) Community Cloud: - In community cloud, cloud infrastructure which can be used through several organizations in a private community. This cloud is shared amongst many associations that have comparative cloud prerequisites.
4) Hybrid Cloud: - This cloud it exploit a blend of no under two clouds where the clouds incorporate a blend of private cloud, public cloud or community cloud.

II. TYPES OF LOAD BALANCING ALGORITHMS

Load balancing is the method of reallocating the work load among nodes of the cloud to improve both the resource consumption and the job response time. Depending on the current state of the system, load balancing algorithms can be divided into two categories as given in [7]:

A. Static Algorithm:
This static Algorithm is usually explained in implementation or design of system. This algorithm separation the traffic similar between all of the users. This algorithm needs a previous knowledge of approach resources, so that no longer rely upon the obtainable state of system for decision of shifting of the load. These are much simpler and ignore the obtainable state or the load on the node within system.

B. Dynamic Algorithm:
In this dynamic Algorithm measured only the present condition of the system during load balancing decision. This dynamic method is extra apposite for broadly distributed system such a cloud computing.

Dynamic Algorithm has two parts:
1) Centralized Approach: -- In this centralized approach Simplest as only node is in charge for distribution and managing within the whole system.
2) Distributed Approach:-- in this dynamic methodology each node freely constructs own load vector. Vector accumulating the load know-how of other nodes. In this all selections are made locally utilizing local load vectors. This procedure is more compatible for generally allotted systems comparable to cloud computing

III. LITERATURE SURVEY

Since dynamic load balancing can produce a better performance as it makes load balancing decisions based on the current load of the system, we will focus our attention on dynamic load balancing algorithms in this research. Dynamic load balancing can be carried out in two different schemes: distributed and non distributed.

In a distributed scheme, all nodes in the system execute the dynamic load balancing algorithm and share the responsibility of load balancing. The interaction among nodes to achieve load balancing can be cooperative where nodes work together to achieve a global objective or non-cooperative where each node works independently toward a local goal.

Thus distributed dynamic load balancing algorithms tend to generate more overhead due to large number of messages being exchanged between the nodes in order to make its load balancing decisions. An advantage Majority of dynamic load balancing algorithms proposed in the literature are distributed, which is of greater advantage when each node is given the maximum chance to act alone or to interact with as few nodes as possible.

However most proposed dynamic load balancing algorithms require full interaction among nodes of the distributed system. Hence, a distributed dynamic load balancing algorithms that call for minimum interaction among nodes is needed. Some common examples of distributed dynamic load balancing algorithms are presented in [10], [11] and [12].

In a non-distributed scheme, the responsibility of load balancing is either taken on by a single central node or some nodes but never with all nodes. Non-distributed based dynamic load balancing can take two forms: centralized and semi-distributed. In a centralized form the central node is solely responsible for load balancing of the whole distributed system and the other nodes in the distributed system react with the central node but not with each other.
Examples of centralized non-distributed load balancing algorithms is presented in [13] [14]. In a semi-distributed form nodes of the distributed system are segmented into clusters and load balancing within each cluster is centralized.

Whereas load balancing of the whole distributed system is achieved through the cooperation of the central nodes of each cluster. I. Ahmed and A. Ghafoor in their work [15] have presented an efficient Semi-Distributed Load Balancing algorithm. Centralized dynamic load balancing employs less overheads and requires fewer messages since other nodes in the system do not interact with each other, hence is mostly preferred over semi-distributed schemes.

Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma in their work [16] have presented a comparison of different load balancing algorithms as shown in table I below. Table II gives comparison of some dynamic load balancing algorithms [17].

TABLE I: Comparison of load balancing algorithms

Parameters Round	Round Robin	Random	Local Queue	Central Queue	Central Manager	Threshold
Overload Rejection	No	No	Yes	Yes	No	No
Fault Tolerant	No	No	Yes	Yes	Yes	No
Forecasting Accuracy	Large	Large	Small	Small	Large	Large
Centralized/ Decentralized	D	D	D	C	C	D
Dynamic/ Static	S	S	D	D	S	S
Cooperative	No	No	Yes	Yes	Yes	No
Resource Utilization	No	No	Yes	No	No	No
Process Migration	No	No	Yes	No	No	No

TABLE II: Comparison of load balancing algorithms [17]

Algorithm	CPU Overhead	Throughput	Turnaround time	Response time
First In First Out	Low	Low	High	Low
Shortest Job First	Medium	High	Medium	Medium
Priority based scheduling	Medium	Low	High	High
Round-robin scheduling	High	Medium	Medium	High
Deficit Round Robin	High	High	Medium	High
Earliest Deadline First	High	High	Medium	High
Multilevel Queue scheduling	High	High	Medium	Medium
FPZL	High	High	High	High

IV. PROPOSED LOAD BALANCING METHODS

The important things to reflect on while developing any load balancing algorithm are: estimation and comparison of load, stability and performance of system, interaction between the nodes, nature of work, selection of nodes, etc. In this work we have implemented two load balancing algorithm using Window Azure Framework.

To understand these load balancing algorithm let us consider an example of a cloud with five servers as shown in figure 2, where we assume that each request from the client is send to any of the servers using central node or central server

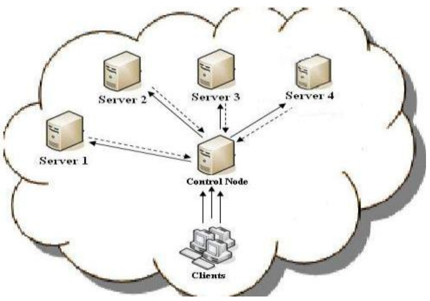

figure 3: Load Balancing in a Cloud System.

In this first proposed load balancing algorithm, consider that a new client request is received at the central server. Now the central server asks each of the servers in the cloud with their real time load. On receiving them the central server assigns this new request to the server with minimum load. In case of a tie it randomly assigns the request to any of the servers. This load balancing algorithm is a dynamic and extremely efficient, but requires for each new request the real time load to be calculated and estimated to the central server, which increases some overhead on the system.

In the second proposed load balancing algorithm, when a new request is received at the central server it asks for the real time load to each of the servers in the cloud. It then waits for the N requests to come hereafter.

The value of this window size 'N' can be changed as per the requirement of the system. After waiting for the N new requests, the central node distributes these requests equally among all the servers in the cloud depending upon their load values.

Now consider the example that the window size is 100 and real time load of the server S1, S2, S3 and S4 after receiving the 100th new request are 30, 15, 25 and 20 respectively. So these 100 new request are distributed equally among these servers such that server S1 get ((1-0.3)*100) %, S2 gets ((1-0.15)*100) %, server S3 gets ((1-0.25)*100) % and S4 gets ((1-0.2)*100) % of these requests. This load balancing algorithm is a more efficient one as it requires less computation at each server end as compared to previous one.

V. RESULTS

Figure 3(a) and (b) below shows the results of the first load balancing algorithm, in which the central server assigns this new request to the server with minimum load.

As in figure 3(a) the central server assigns the new request server 2 as it has the minimum load of 22 among the five servers in the cloud. Similarly in figure 3(b) the central server assigns the new request server 5 as it has the minimum load of 122.

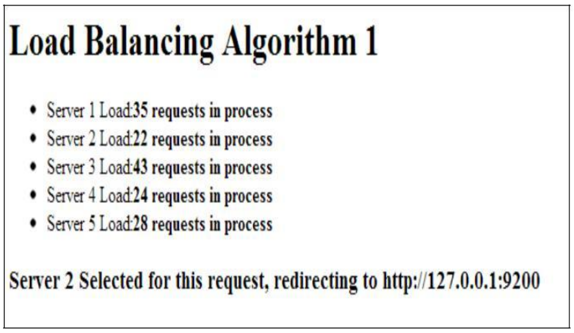

Figure 3(a): Cloud environment with five servers

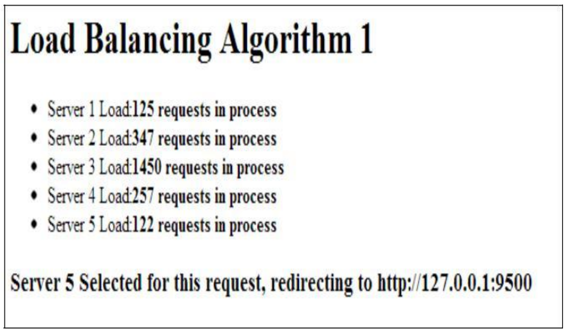

Figure 3(b): Server 1 in the cloud under normal operation

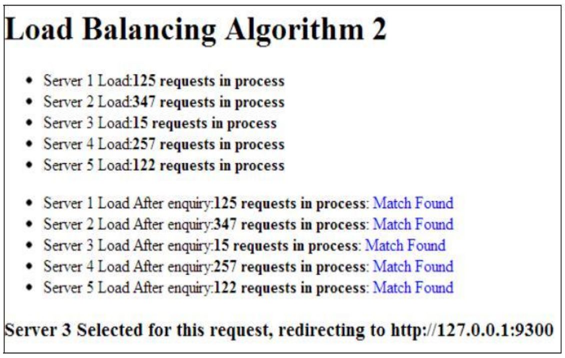

Figure 4(a): Cloud environment with five servers

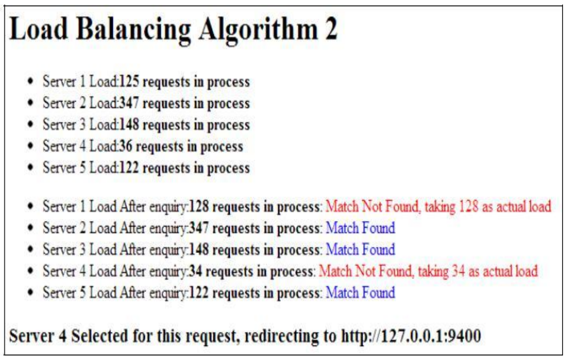

Figure 4(b): Server 1 in the cloud under normal operation

Figure 4(a) and (b) shows the results for the proposed second load balancing algorithm, in the central node asks for the current load to the which after waiting for the N new requests, the central node distributes these requests equally among all the servers in the cloud depending upon their load values. Table III below shows comparison of proposed algorithms. Both the proposed methods are dynamic ones, however the first algorithm is based on centralized distributed scheme, whereas the second is based on centralized non-distributed scheme.

As seen from the table the second algorithm thus has a low CPU overload due to minimum messages being exchanged among the nodes and has higher throughput.

TABLE III: Comparison of proposed algorithms [17]

Algorithm	CPU Overhead	Throughput	Turnaround time	Response time
Proposed Algorithm I	Medium	Low	Medium	Low
Proposed Algorithm II	Low	High	Medium	Low

VI. CONCLUSION

In this paper, two dynamic algorithms of load balancing has been proposed and implemented using Windows Azure Framework. The results show that both the algorithms are efficient and dynamic.

The proposed algorithm II has comparatively low CPU overhead as it's based on centralized non-distributed scheme and employs minimum messages being exchanged among the nodes and further has higher throughput.

REFERENCES

[1] Foster,Ian, Yong Zhao, Ioan Raicu, et al, "Cloud Computing and Grid Computing 360-Degree Compared", In Grid Computing Environments Workshop (GCE), Austin, 2008.

[2] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing. Technical Report SP 800-145 Draft, National Institute of Standards and Technology, Information Technology Laboratory, January 2011.

[3] CloudComputingDefinitionGartner.http://www.gartner.com/it/page.jsp?id=1035013

IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified

Vol. 4, Issue 9, September 2016

[4] Martin Litoiu, Murray Woodside, Johnny Wong, Joanna Ng, Gabriel Iszlai, "A Buisness Driven Cloud Optimization Architecture", Proceedings of ACM in SAC'10, pp.380 – 385.
[5] Roedig, U., Ackermann, R., Steinmetz, R.: Evaluating and Improving Firewalls for IP Telephony Environments. In: IP-Telephony Workshop (IPTel) (April 2000)
[6] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.
[7] Abirami M S, Niranjana G, "Dynamic Load Balancing in Distributed Systems", International Conference on Computing and Control Engineering (ICCCE 2012), ISBN 978-1-4675-2248-9, 2012
[8] .Kai Lu , Riky Subrata and Albert Y. Zomaya, Networks & Systems Lab, School of Information Technologies , University of Sydney , An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems considering Desirability of Grid Sites
[9] .J. A. Stankovic and I. S. Sidhu, "An Adaptive Bidding Algorithm for Processes, Cluster and Distributed IEEE Transactions on Software Engineering, Vol. SE-11, pp. 491-496, May 1985.
[10] Groups", in Proc. 4th Int. Conf. Distributed Compu Sys., pp. 49-59, 1984.
[11] R. M. Bryant and R. A. Finkel, "A Stable Distributed Scheduling Algorithm," in Proc. 2nd Int. Conf. Dist Comp., pp. 341-323, April 1981
[12] D. Grosu and A. T. Chronopoulos," Noncooperative Load Balancing in Distributed Systems," Journal of Parallel and Distributed Computing, vol. 65, no. 9, pp. 1022-1034, Sept. 2005.
[13] Y. Chow and W. Kohler, "Models for Dynamic Load Balancing in Heterogeneous Multiple Processor System," IEEE Transactions on Computers, Vol. C-28, pp. 354-361, , May 1979.
[14] L. Ni, and K. Hwang, K., "Optimal Load Balancing in a Multiple Processor System with Many Job Classes,"
[15] I. Ahmed and A. Ghafoor, "Semi-Distributed Load Balancing for Massively Parallel Multicomputers," IEEE Trans. Software Eng., vol. 17, no. 10, pp 987-1004, October 1991.
[16] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma, "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology 38 2008
[17] Abirami M S, Niranjana G, "Dynamic Load Balancing in Distributed Systems", International Conference on Computing and Control Engineering (ICCCE 2012). M. Ernst, "Self-defending software: Collaborative learning for security," 2008