International Journal of Physics and Applications.
ISSN 0974-3103 Volume 7, Number 1 (2015), pp. 1-8

http://www.irphouse.com

A Novel Load Balancing Model Using RR Algorithm for Cloud Computing

B.Bhaskar[1] and E. Madhusudhana Reddy[2]

Dept. of Computer Science & Engineering,
Madanapalle Institute of Technology and Science, JNTUA,
Andhra Pradesh, India
E-Mail: bskr.rdy7gmail.com, e_mreddy@yahoo.com

Abstract

Cloud Computing is computing where in several groups of servers are networked for providing online access to users. Several factors which influence cloud computing such as scalability, security etc are discussed by many researchers. Now a days usage of internet resources is widely increasing resulting in the increase of workload exponentially. In order to make cloud computing efficient and improve user satisfaction this incoming huge workload is to be handled with care. This paper introduces techniques for the public cloud based on cloud partitioning concept that makes cloud computing effective. Cloud Partitioning is an efficient method for a public cloud, where in a public cloud is divided into several sub partitions. This cloud partitioning method paves way to apply load balancing technique in a simplified manner across the multiple nodes. Load balancing is a technique that distributes workload across several nodes uniformly which results in improved performance and less response time through a right distribution strategy. This paper introduces skewness measurement technique along with load balancing using Round Robin (RR) technique in order to obtain enhanced performance which results in Green Computing.

Keywords: public cloud, cloud partition, load balancing, skewness.

I. INTRODUCTION

Cloud computing is an emerging technology in the realm of computer science. Cloud computing has distinctive characteristics. They are ubiquitous, on-demand service, availability of resources, access to a wide variety of services, convenience, user satisfaction [1][2] etc. These characteristics attracted internet [3] users to divert to

cloud computing. Even though cloud computing is efficient, processing the incoming jobs in cloud computing environment is a challenging task. The way the jobs arrive cannot be predicted and the capacities of the nodes will not be the same and differ in software and hardware configurations and so balancing the load is important to enhance system performance [4] and to maintain stability [5]. To deal this problem of imbalance of load and to increase the throughput of the system, this paper introduces techniques called cloud partitioning, load balancing [6][1] and skewness measurement. Cloud partitioning [7] is the process of dividing a huge public cloud [8] into sub partitions. Load balancing is the process of distributing the workload evenly to all the nodes that are present in the cloud [9]. The load balancing model implemented in the cloud consists of a load balancer manager which selects the appropriate cloud partition for the incoming jobs. The load balancer [10][11] present in the cloud partition selects the appropriate node by knowing its skewness. Skewness [12] measurement is the process of knowing the resource utilization [13] rate of a server which later facilitates to save energy [14].

II. RELATED WORK

Many studies were done about load balancing in cloud computing. Many tools and techniques [6] are also introduced that are commonly used for load balancing, but these techniques are not successful enough in balancing the load properly and load balancing still stands as a problem that needs new architectures to overcome. Load balancing plays a major role in improving the system performance. After a thorough go through into the comparative analysis [15] of load balancing algorithms given by Rundles we concluded that ESCE algorithm [16] and Round Robin Algorithm [17] are better by considering the factors performance, time and cost. Round Robin algorithm is used here because of its simplicity. By combining the skewness measurement technique along with the load balancing Dynamic resource concept is achieved which can guarantee in enhanced performance.

III. LOAD BALANCING

Load balancing technique ensures that no node is idle while other nodes are being utilized. A selective mechanism is implemented which is capable of turning out the incoming jobs to the nodes that are idle by preventing them from going to the nodes that are already overloaded. This helps in achieving improved resource utilization in minimum response time [18]. A public cloud is a collection of several nodes that are present in different geographical locations. Our model divides this huge cloud into several partitions. The Cloud partitioning model [17] is shown in the below figure. First step is a public cloud is partitioned into four sub partitions namely location#1, location#2, location#3 and location#4.

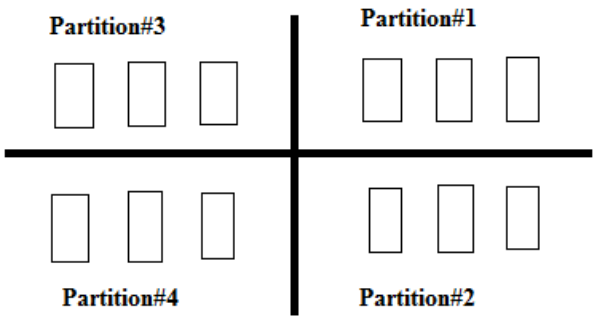

Fig. 1. a public cloud divided into 4 partitions

In our proposed system load balancing is done by load balancer controller and load balancer. Each cloud partition has a group of nodes with load balancer as its manager. The entire public cloud will have a controller called load balancer controller which is responsible for handling the load balancers present in each cloud partition. The Load balancer controller is a vital component connected to all load balancers and will be regularly communicating with them at certain time intervals. After partitioning the cloud then starts the task of load balancing. Here load balancing is done by load balancer controller. The load balancer controller serves as an interface between users and data source, besides it supplies the load balancers with jobs evenly. The whole incoming load is passed to load balancer controller which then searches for the partitions that contain idle nodes. Load balancers contain the status information of all the nodes connected to it such as how many nodes are idle, how many nodes are busy, how long a particular node is engaged with job etc.The evaluation of load status of each node is crucial. This can be known by computing the load degree of each node. The load degree is computed on the basis of static and dynamic parameters of a node such as number of CPU's, processing speed, memory size, memory and cpu utilization ratio, bandwidth of a network etc. The Load balancers share the status information with load balancer controller which then decides to which partition the job is to be forwarded, thereby avoiding the load imbalance problem. Once the job is arrived to a partition corresponding load balancer chooses the most appropriate node for processing the job.

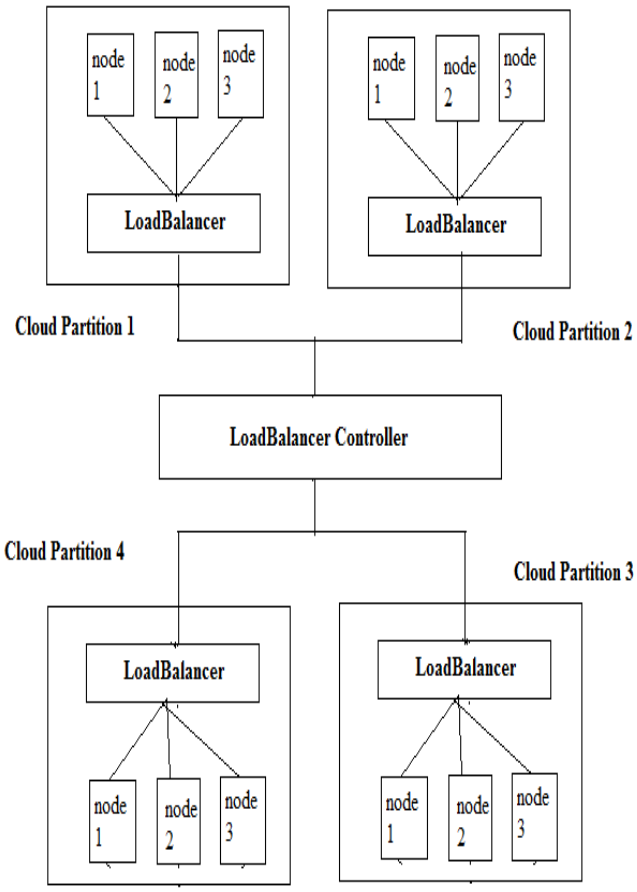

Fig. 2 Relationship between load balancer controller load balancer and nodes

A. Working of Load Balancer Controller:

- Receive jobs from end users.
- Select appropriate partition in the public cloud for job processing.
- Know the status of cloud partition (idle, normal or overloaded).
- If partition status is overloaded then job is not forwarded to that particular partition.
- If idle or normal then job is forwarded to respective load balancer which is then activated to proceed further.

B. Cloud Partition Status:

- Idle represents that nodes are idle
- Normal represents nodes are neither idle nor overloaded
- Overloaded represents nodes are busy and overloaded.

The Load balancer controller forwards the jobs to cloud partitions which are idle or normal. If a cloud partition is heavy then jobs are not sent to that partition instead to the partitions that are idle. Once the load balancer controller forwards the

jobs to a particular cloud partition, the corresponding load balancer is activated. Load balancer is responsible for assigning a job to a suitable node. The load balancer in each partition gets the status information from each node and calculates the load degree of nodes and based upon it forwards the task to suitable node. The load degree of a node can be computed by using the formula

$$LOADDEGREE\ (N) = \sum_{i=1}^{n} X_i F_i$$

Where N is the current node, F_i is a $F_i(1<=i<=n)$, n is total number of parameters, x_i is the priority given to the jobs.

Average load degree of a node is computed by

$$AVG\_LOADDEGREE = \sum_{i=1}^{n} LOADDEGREE(N_i)/n$$

C. Possible load status of a node

Along with evaluating load degree of nodes the load balancer also computes the skewness of each node.

LOAD DEGREE	**STATUS**
LOADDEGREE(N)=0	IDLE
0<LOADDEGREE(N)<=High_LOADDEGREE	NORMAL
High_LOADDEGREE<=LOADDEGREE(N)	OVERLOADED

IV. SKEWNESS

Skewness concept is used to measure the utilization rate of a node. In a cloud partition containing some number of nodes, one node might be working for a long time while other nodes are sitting idle [19]. Despite a node being utilized for a long time the cloud partition status will be showing normal. This is because of the remaining nodes in that particular partition are sitting idle. This very same node working for a long time might lead to temporary fluctuation of application resource demands and system hangs. In this situation the load balancer will define a threshold value and regularly checks that the skewness values of all the nodes doesn't exceed this threshold value. By following this procedure we make sure that the node being utilized for a long time should be freed and the workload is to be passed on to the nodes that are idle for a long time thereby reducing the temperature in surroundings. If the work load is minimum and some nodes are idle then those nodes can be turned off temporarily thereby saving the energy. Thus by reducing the temperature and saving the energy green computing is achieved. Green Computing tends to attain economic viability to improve the way the computing devices are used. It is environmentally responsible and eco friendly use of computers and resources. When the servers resource utilization is less, then they are turned off.

The skewness of a node x can be computed by using the below formula

$$SKEWNESS\ (N) = \sqrt{\sum_{i=1}^{m} \left(\frac{ri}{\bar{r}} - 1\right)^2}$$

Where *N* denotes the current node, $\bar{r}$ is the average utilization of resources of node N

If the skewness of a node exceeds the threshold value, then we define that particular node as a hot spot. Hot spot denotes that particular node is having relatively higher temperature than the surroundings and it is being utilized for a long time. Cold spot denotes that node with ambient temperature. If any node becomes a hot spot then the workload is migrated to idle nodes that are cold spot.

V. ALGORITHM

Load balancing using RR algorithm

```
BEGIN
        WHILE JOB
        DO
        SELECT NODE(JOB)
        IF(CP==IDLE OR CP==NORMAL)
        THEN FORWARD JOB TO CP
        WHILE JOB
        DO
        SELECT NODE(JOB)
        IF(SKEWNESS(NODE)<=THRESHOLD)
        FORWARD JOB TO NODE
        ELSE
        GO FOR ANOTHER NODE
        END IF
        END WHILE
        ELSE
        SEARCH FOR ANOTHER CP
        END WHILE
END
```

VI. PERFORMANCE EVALUATION

By observing the Table 1 and figure 3, it is clearly understood that the performance is high when these techniques cloud partitioning, load balancing and skewness measurement are implemented. As the rate of incoming jobs is increased the cloud computing performance is gradually decreased when no mentioned techniques are followed. The computing performance is efficiently maintained by following the techniques. The performance is maintained to its optimum even though the rate of incoming jobs is at full pace.

Table.1: Comparision of performance with no load balancing and load balancing

Number of Jobs	Performance	
	No Load Balancing	Load Balancing applied
No. Of jobs=0	0	0
No. Of jobs=10	0.5	1
No. Of jobs=20	1	2
No. Of jobs=30	1	2.5
No. Of jobs=40	1.5	4
No. Of jobs=50	2	5

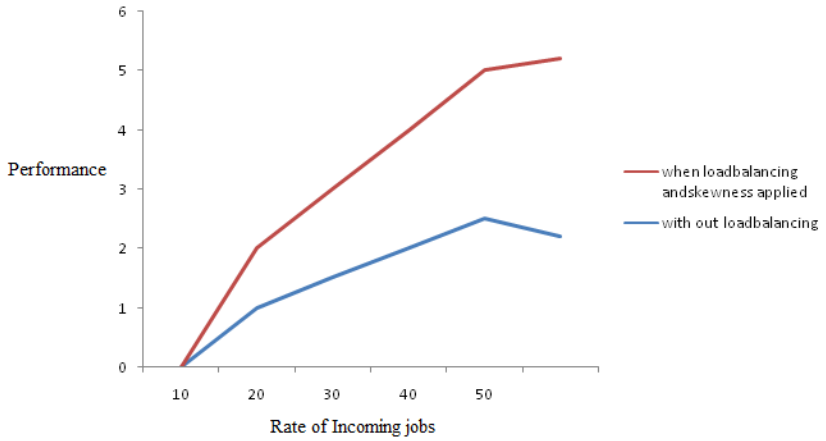

Figure 3: Figure analyzing the performance

VII. CONCLUSION

The overall goal of this paper is to make cloud computing effective by implementing the techniques that achieve the same. Load balancing enhances the performance of cloud services substantially. It prevents the overloading of servers which otherwise would degrade the performance. The proposed load balancing using RR algorithm balances the incoming load equally to all partitions and reduces the response time by distributing the jobs to the nodes uniformly and improves the throughput of the system. The energy is also saved by turning off a particular node that is idle for a long time. Hence green computing concept is also followed.

REFERENCES:

[1] Peter Mell, Timothy Grance, "National Institute of Standards and Technology Special Publication 800-145: The NIST Definition of cloud computing", September 2011.

[2] Rashmi. K. S, Suma. V ,Vaidehi. M,"Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud" Special Issue of International Journal of Computer Applications (0975 – 8887)

[3] K.D. Devine, E.G. Boman , R.T. Hepahy, B.A.Hendrickson, J.D. Teresco, J. Faik,J.E. Flaherty,L.G. Gervasio," New Challenges In Dynamic Load Balancing, Applied Numerical Mathematics,52(2005)133-152.
[4] 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.
[5] N. G. Shivaratri, P. Krueger, and M. Singhal, "Load distributing for locally distributed systems", Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.
[6] 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.
[7] Gaochao Xu, Junjie Pang, Xiaodong FuJaya, Bharathi Chintalapati,Srinivasa Rao T.Y.S. " A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" IEEE transactions on cloud computing year 2013.
[8] M. Rouse, Public cloud, http://searchcloudcomputing, techtarget.com/definition/public-cloud, 2012
[9] Doddini Probhuling L." Load balancing algorithms in cloud computing" International Journal of Advanced Computer and Mathematical Sciences
[10] HaProxy, "HaProxy load balancer", http://haproxy.1wt.eu/.
[11] Nginx,"Nginx web server and load balancer", http://nginx.net/.
[12] T. Das, P. Padala, V. N. Padmanabhan, R. Ramjee, and K. G. Shin, "Litegreen: saving energy in networked desktops using virtualization," in Proc. Of the USENIX Annual Technical Conference, 2010. 19.222
[13] Scalability resource utilization "Analysis of load balancing techniques in cloud computing" 2011 International Conference on Computer and Software Modelling.
[14] R. Buyya, A. Beloglazov, J. Abawajy, Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges, in: Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2010, Las Vegas, USA, 2010.
[15] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, Availability and load balancing in cloud computing, presented at the 2011 International Conference on Computer and Software Modeling", Singapore, 2011.
[16] M. Randles, D. Lamb, and A. Taleb-Bendiab, A comparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.
[17] Gaochao Xu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, IEEE Transactions on Cloud Computing Year-2013.
[18] Nginx,"Nginx web server and load balancer", http://nginx.net/.
[19] Kumar Nishant, Pratik Sharma,Vishal Krishna "Load Balancing of Nodes in Cloud Using Ant Colony Optimization" 2012 14th International Conference on Modelling and Simulation