International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-2, February 2015

A Load Balancing In Cloud Computing System Based on Cloud Partitioning

Mumthas B, Smita C Thomas

Abstract— **In computer science cloud computing describes various offers to the vendors. Load balancing in the cloud computing is very important factor that improves the performance. Better load balancing makes cloud computing efficient and raise user gratification. This paper present a good load balance model for the public cloud based on the cloud partitioning concept. A switch mechanism is used to choose different strategies for different situations. Apply the game theory algorithm to the load balancing strategy to improve the efficiency in the public cloud environment.**

***Index Terms*—Distributed system, Game theory, Nash equilibrium, Load balancing**

I. INTRODUCTION

Cloud computing is an interesting technology in the field of computer science. The cloud will bring changes to the IT industry. By providing users with new types of services the cloud is changing our life. Users get service from a cloud without paying any attention to the details. Many people pay attention to cloud computing. Cloud computing is efficient and stores large amount of data. One of the main issue is in the cloud computing environment is a load balancing problem.

Jobs arrive in the cloud computing environment is not predictable and each node capacity in the cloud is differ. For load balancing problem, workload control is important to improve performance of the system and maintain stability. System information not used in static schemes and complexity is less but dynamic schemes will need additional costs for the system but it can change as the system status changes.

For flexibility a dynamic scheme is used here. Goals of load balancing (i) to improve system performance gradually, (ii) to have backup plan in case of system fails even partially (iii) to maintain the stability of the system, (iv) to accommodate future modification.

II. LOAD BALANCING

Now a day load balancing in cloud computing systems is really a challenge. Always a distributed solution is required. To maintain one or more idle services just as to fulfill the required demands, it is not always practically reasonable or cost efficient. Cloud is a very complex structure and components are present throughout a wide environment so jobs can't be assigned to particular servers and clients individually for efficient load balancing. Cloud model is shown in figure 1. Here some riskiness is attached while jobs are assigned.

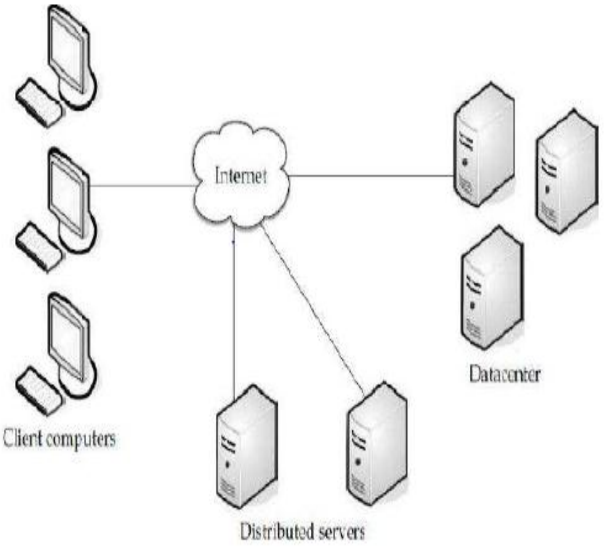


Figure 1- Cloud model

This paper introduces some of the load balancing methods in large cloud systems. Our aim is to provide a better load balancing algorithm and to improve the different performance parameters like user satisfaction, latency, efficiency etc. for the clouds of different sizes. These loads can be considered in terms of memory space used, delay or network load, load of CPU etc. Dynamic load balancing algorithm depend on present behavior of the system, it does not depend on previous state. Two different ways of dynamic load balancing techniques are distributed and non distributed.

III. SYSTEM MODEL

There are number of cloud computing categories their work is focused on a public cloud. Standard cloud computing model is the basis of public cloud, service provider provide service. Many nodes contained in large public cloud and these nodes are located in different locations based on geographical area. To manage this large cloud can use cloud partition. A cloud partition is a small part of the public cloud with divisions based on the physical locations. The architecture is shown in Fig. 2.

Manuscript received February 15, 2015.

Mumthas B, Computer Science and Engineering, M. G. University, Pathanamthitta, India, 8289962246.
Smita C Thomas, Computer Science and Engineering,, M. G. University, Pathanamthitta, India, 9995543383.

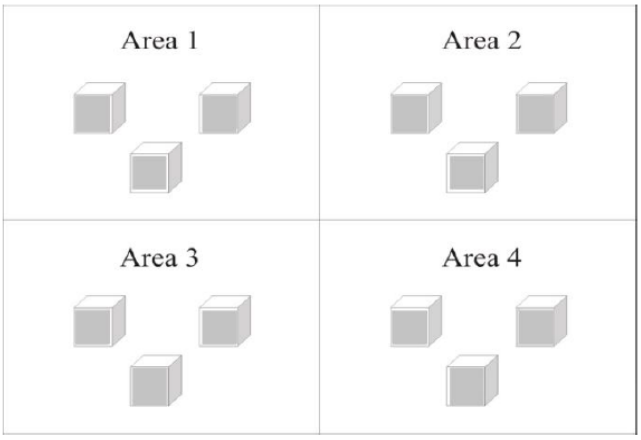


Figure 2- Cloud partitions

This load balancing system has a main controller and a balancer. Suitable partition is select using the help of main controller and provides the job to that partition. To refresh the status the main controller communicates with balancer. Balancer assigns job to suitable node. If cloud partition load balance is normal, then partitioning can be locally accomplished. If load status is not normal then job should assigned to other partition. The job assignment strategy is shown in Fig 3.

A. Assigning jobs to the cloud partition

The main controller receives jobs from different clients and they select suitable partition for the received job. When job i reaches at the system, the main controller check the status of the cloud partition, which partition the job should be assigned. If the job that is last updated, then the job is assigned to Partition1. The job is assigned to second partition, if it is an upcoming job. If it's a presently running job then it is assigned to third Partition. If the job is an outdated job then that job is assigned to fourth Partition. The Best Partition Searching algorithm is shown in Algorithm 1.

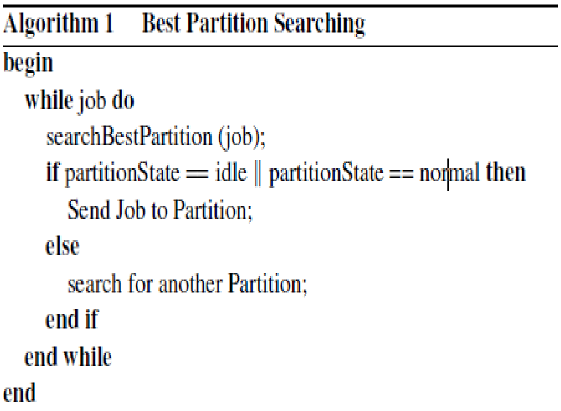
B. Cloud partition status

If the status of the cloud partition is idle or normal then job assigned to corresponding load balancer. If the status of the cloud partition is overload then that job is hand over to another partition. After assigning job to the balancer, then the balancer starts working. Cloud partition status can be determined by the status of the nodes that are included in corresponding partition.

a) Idle: when most of the nodes are in idle state.
b) Normal: some of the nodes are in idle state and others are overloaded.
c) Overload: most of the nodes are in overloaded.

C. Assigning jobs to the nodes in the cloud partition

The cloud partition balancer collects load information from each and every node to compute the status of the cloud partition. This computation of load status of node is very crucial. The first task is to define the load degree of the node.



```
Algorithm 1    Best Partition Searching
begin
  while job do
    searchBestPartition (job);
    if partitionState == idle || partitionState == normal then
      Send Job to Partition;
    else
      search for another Partition;
    end if
  end while
end
```

Algorithm: 1

The load degree for a node can be defined as follows:

$$Load\_degree\ (N) = \sum_{i=1}^{m} \alpha i F_I \quad (1)$$

Where
N - Current node
Fi- parameter either static or dynamic, Fi (1<=i<=m), m represents the total number of parameter.
αi- weights that may differ for different kinds of job for all (1<=i<=n)

Average Load Degree of the cloud partition can be calculated as follows:

$$Load\_degree_{avg} = \sum_{i=1}^{m} (load_{degree}\ (Ni))/n \quad (2)$$

A node has three load status according to the calculation of load degree. They are defined as follows:-

IDLE: When LD (N)=0
NORMAL: 0<LD (N) <=High_LD
OVERLOADED: High_LD<=LD (N)

Main controller does not selects the cloud partition having the status=HEAVY. Such nodes have Load Degree (LD) =OVERLAODED and is not eligible for processing. Only cloud partition having IDLE or NORMAL load status and Node having IDLE or NORMAL load degree are selected for scheduling and load balancing.

The cloud partition balancers create load status tables using the result of load degree. Each balancer maintains a Load Status Table and refreshes it each fixed period T. To calculate the partition status balancer uses this table.

IV. CLOUD PARTITION LOAD BALANCING STRATEGY

A. Motivation

The performance of the entire cloud will improve when using good load balancing. But, there is no standard method that can adapt to all possible different situations. To resolve new problems there is several methods in improving existing solutions. For idle state a simple method can be used that is improved round robin algorithm and more complex method is used for normal state that is game theory.

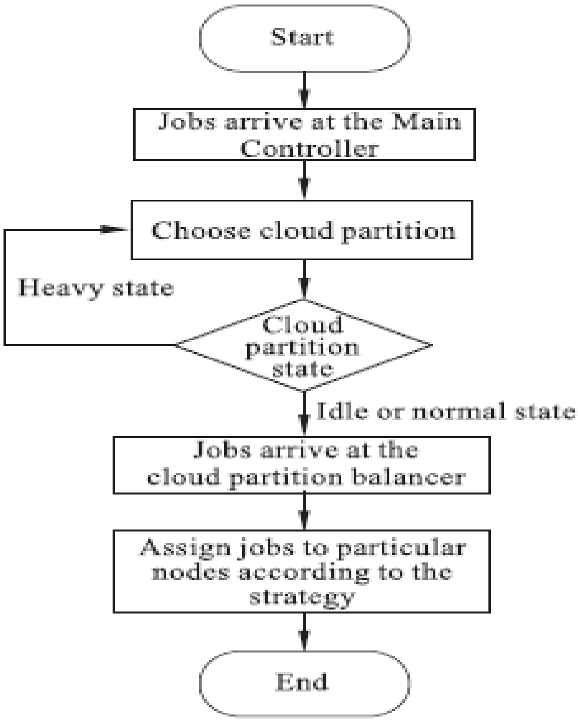

Start
Jobs arrive at the Main Controller
Choose cloud partition
Heavy state
Cloud partition state
Idle or normal state
Jobs arrive at the cloud partition balancer
Assign jobs to particular nodes according to the strategy
End

Fig 2: Job assignment strategy

B. *Load balance strategy for the idle status*

For cloud partition has idle status, there is very simple method can be used because this partition process jobs as speedily as possible. There are varieties of load balancing algorithm to process the job like the Random algorithm, and the Dynamic Round Robin. The Round Robin is very simple method for load balancing so it is used here. It has no status information since this algorithm does not keep the status of each connection. In a public cloud, each node has the different configuration and performance; thus, this method may overload some nodes. So there can be use an improved Round Robin algorithm, which called “Round Robin based on the load degree evaluation”.

C. *Load balancing strategy for the normal status*

For cloud partition has idle status, there is very simple method can be used because this partition process jobs as speedily as possible. There are varieties of load balancing algorithm to process the job like the Random algorithm, and the Dynamic Round Robin. The Round Robin is very simple method for load balancing so it is used here. It has no status information since this algorithm does not keep the status of each connection. In a public cloud, each node has the different configuration and performance; thus, this method may overload some nodes. So there can be use an improved Round Robin algorithm, which called “Round Robin based on the load degree evaluation”.

V. CONCLUSION

In the cloud computing environment load balancing has been strong effect on the performance. This method will prevent overloading of the servers which would else degrade the efficiency. Public cloud is divided into several parts and each part is located based on physical locations. Performance and user satisfaction can be improved using better load balancing strategies. This method has a main controller that chooses right partition for the arrived job and has balancer that determines the node which has not overloaded. Game theory is applied to improve the performance.

ACKNOWLEDGEMENT

I would like to extend my gratitude to the reference authors, review committee and my guide Ms. Veena Ramachandran L

REFERENCES

[1] 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 pp 34-39 Volume 18, Number 1, February 2013.

[2] 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. 14thInternational Conference on Computer Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.

[3] 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.

[4] Hunter, The why of cloud, http://www.gartner.com/ DisplayDocument ?doc cd=226469&ref= g noreg, 2012.

[5] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, Cloud computing: Distributed internet computing for IT and scientific research, Internet Computing, vol.13, no.5, pp.10-13, Sept.-Oct. 2009.

[6] P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/ publications/nistpubs/800- 145/SP800-145.pdf, 2012.

[7] Microsoft Academic Research, Cloud computing, http:// libra.msra.cn/Keyword/6051/cloud-computing?query=cloud20computing , 2012.

[8] Google Trends, Cloud computing, http://www.google. com/trends/explore#q=cloud%20computing, 2012.