OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Novel Switch Mechanism for Load Balancing in Public Cloud

Kalathoti Rambabu[1], M. Chandra Sekhar[2]

[1] *M. Tech (CSE), MVR College of Engineering and Technology, A.P., India.*

[2]*Asst. Professor, Dept. of Computer Science & Engineering, MVR College of Engineering and Technology, A.P., India.*

Abstract: *In cloud computing environment, one of the core design principles is dynamic scalability, which guarantees cloud storage service to handle the growing amounts of application data in a flexible manner or to be readily enlarged. By integrating several private and public cloud services, the hybrid clouds can effectively provide dynamic scalability of service and data migration. A load balancing is a method of dividing computing loads among numerous hardware resources. Due to unpredictable job arrival pattern and the capacities of the nodes in cloud differ for the load balancing problem. In this load control is very crucial to improve system performance and maintenance. This paper presents a switch mechanism for load balancing in cloud computing. The load balancing model given in this work is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographical areas. Thus, this model divides the public cloud environment into several cloud partitions. When the cloud environment is very large and complex, these divisions simplify the load balancing. The cloud environment has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy.*

Keywords: *Cloud computing, Load balancing, Public cloud, Switch mechanism.*

I. INTRODUCTION

Cloud Computing[1][2] is a concept that has many systems interconnected through a real time network like internet. Cloud computing enables convenient, dynamic, on-demand, and reliable use of the distributed computing resources. The cloud computing model has five main characteristics on demand service, resource pooling, broad network access, flexibility, measured service. Cloud computing is efficient and scalable but to maintain the stability and to process many jobs in the cloud computing environment is a very difficult problem. The job arrival pattern cannot be predicted and the capacities of each host in the cloud environment differ. Hence for balancing the usage of the internet and related resources has increased widely. Due to this there is tremendous increase in overall workload. So there is uneven distribution of this workload which results in severe server overloading and may crash. In such the load, it is crucial to control the workloads to improve system performance and maintain stability.

The load on every cloud is variable and dependent on several factors [3]. To handle this problem of imbalance of load in the cloud and to increase its working efficiency, this work tries to implement a switch mechanism for load balancing. Good load balancing makes cloud computing more efficient and also improves the user satisfaction. This paper is aimed at the public cloud which has numerous hosts. A system having the main controller, balancers, servers and a client is implemented in this work. It introduces a switch mechanism for balancing load to choose different strategies for different situations. The proposed work divides the public cloud into cloud partitions and applies different strategies to balance the load on cloud. Our work helps to avoid overloading of servers and improve response times. The basic designs of the system and algorithms to implement it are described in this paper. The following are the goals of Load Balancing:

- To improve the performance substantially.
- To have a backup plan in case the system fails even partially.
- To maintain the system stability.
- To accommodate future modification in the system.

II. RELATED WORK

In [4], the authors propose a binary tree structure that is used to partition the simulation area into sub-domains. The characteristics of this fast adaptive balancing technique is to be adjusted the workload between the processors from local areas to global areas. According to the difference in the workload, the arrangements of the cells are obtained. But the main workload concentrates on certain cells so that the procedure of adjusting the vertices of the network grid can be very long because of the local workload can be considered. This problem can be avoided by the fast load balancing adaptive technique. In [5], the authors propose a method named honeybee behavior inspired the load balancing algorithm. Here in this work well load balance across the virtual machines for maximizing the throughput. The load balancing cloud computing model can be achieved by modeling the foraging behavior of the honey bees. This algorithm is derived from the behavior of the honey bees that uses the method to find and reap food. In bee hives, there is a class of bees named as the scout bees and the another type was the forager bees .The scout bee which forage for the food sources, when they find the food, they come back to the beehive to advertise this news by using a dance called "vibration" dance. The purpose of this vibration dance, gives the idea of the quality and quantity of the food and also its distance from the beehive.

In [6], the authors propose a dynamic file migration load balancing method based on the distributed architecture. Considered the large file system there were various problems like dynamic file migration, algorithm based only on the centralized system and so on. So these problems are to be avoided by the introduction of the method called self acting load balancing algorithm (SALB).In the parallel file system the information is transferred between the memory and the storage devices so that the data management is an important role of the parallel file system. In [7], the authors propose an efficient cell selection scheme and two heat diffusion based method called local and global diffusion. Considered the distributed virtual environments there were various numbers of users and the load accessing by the concurrent users can cause severe problem. This can be avoided by this method. According to the heat diffusion method, the virtual environment is divided in to large number of square cells and each square cell having objects. The working of the heat diffusion method is in such a way that every node in the cell sends load to its neighboring nodes in every iteration and the transfer was the difference between the current node to that of the neighboring node. In [8], the authors addressed the concept of overlay networks for the interconnection of the machines that makes the backbone of an online environment. Virtual online world that makes the opportunities to the world for better technological advancements as well as developments. So the proposed network model that makes better feasibility and load balancing to the dynamic virtual environments. This proposed system developed Hyper verse architecture, that can be responsible for the proper hosting of the virtual environment. There were self organized load balancing technique by which the world surface is subdivided in to small cells, and it is managed by the public server. In this cells various hotspots so that the absolute mass of the object in the cell can be calculated by using the public server.

III. PROPOSED WORK

1. System Model

In our proposed load balancing model the public cloud is partitioned based on their geographical locations. Figure 1, shows the schematic representation of a partitioned public cloud. This proposed model divides the public cloud into several cloud partitions. When the environment is very large and complex, then these divisions simplify the load balancing technique. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy. Load balancing is based on this partitioning of the public cloud. When the jobs arrive, then the job controller selects a best partition and the jobs are processed by the servers present in that partition. This selection of best partition is done by a central module called as main controller and the distribution of jobs among the servers is done by the balancers present in every partition. Figure 2 shows main controller collecting status info from balancer and apps server. Algorithm 1 shows best partitioning search to select best partitions among the available partitions.

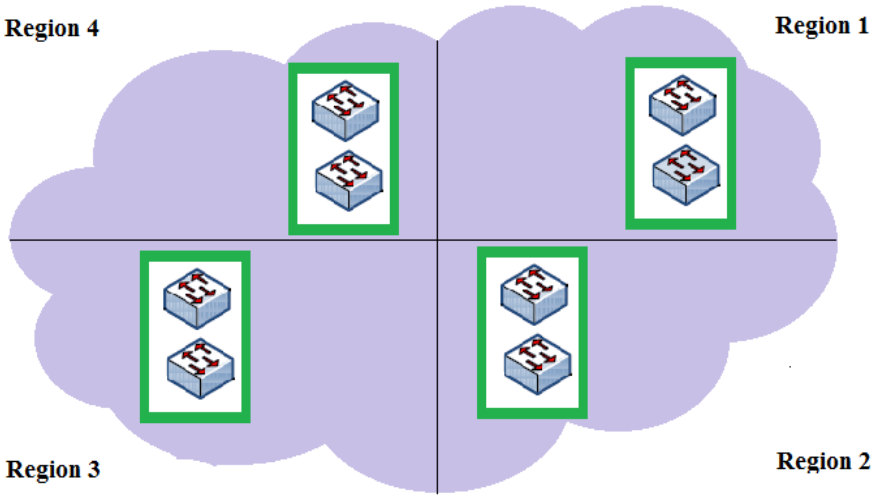

Fig. 1: Cloud Partitioning

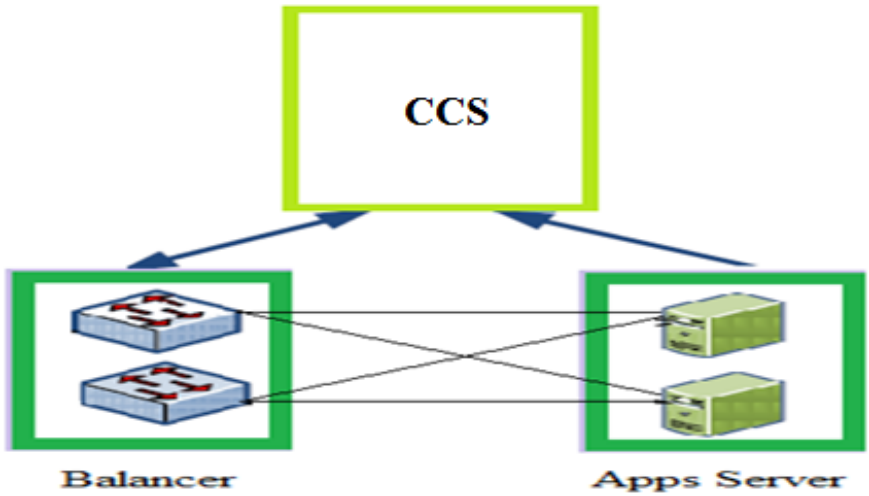

Fig. 2: Main controller **(**CCS) collecting status info from balancer and apps server.

The main controller decides the sub partition to be selected in the public cloud. The goal of the controller is to interact with the balancers and application servers (or nodes) and collect necessary status information of the system. The interaction between main controller and Balancer is a two-way interaction and that of between a controller and an Apps server is a one-way interaction. Based on the collected information and necessary calculations made, then a geographically nearest partition is selected. This selection of geographically nearest partition is to minimize and avoid the cost incurred on moving the jobs to a distant partition.

Algorithm 1: Best Partitioning Search

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

2. Assigning jobs to the cloud partition

After creating the cloud partitions, then load balancing the procedure begins as follows: The job arrives and then the partitions manager decides in which partition the job has to be executed and check is there any requirement for creation of new partition if required new partition is created or else in the existing partition the job is submitted. Then the job distributor decides how to assign the jobs to the individual nodes in that given partition. The choice of using the load balancing technique may depend on the job distributors. Even multiple strategies could be combined and used for favorable situations. Figure 3 shows the job assignment strategy.

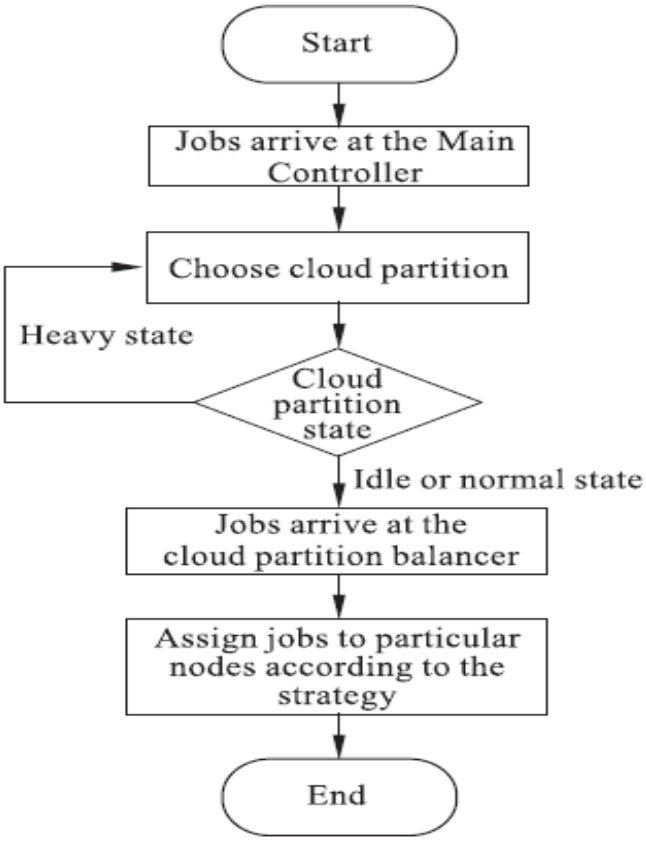

Fig. 3: Job assignment strategy

3. Assigning Jobs To The Nodes In The Cloud Partition

Cloud partition job distributor gathers load information from every node to calculate the cloud partition status. This calculation of each node's load status is very essential. The first task is to evaluate the load degree of each node. The node load degree is related to various static and dynamic parameters. The static parameters include the number of processing units, the memory size, the CPU processing speeds etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc. Based on the parameters the jobs are assigned to the nodes in the selected partitions. The Load Degree (LD) of a given node in any cloud partition is calculated from following equation:

$$LD\ (N) = \sum_{i=1}^{m} Xi * Fi$$

Here, N=Current Node, Fi are the parameter either static or dynamic where Fi(1<=i<=m), m represents the total number of parameter . Xi are weights that may differ for different kinds of job for all (1<=i<=n). Average Load Degree (LD) of the cloud partition will be calculated as:

$$Avg\_LD = \sum_{i=1}^{n} LD\ (Ni)/n$$

According to the calculation of load degree for the given node three load status of the node are defined as follows:

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

Any cloud partition having the status=HEAVY is not selected by the balancing manager and likewise any given node having the Load Degree (LD) =OVERLAODED is not eligible for the processing. Only cloud partition having IDLE or NORMAL load status and the node having IDLE or NORMAL load degree are selected for scheduling and load balancing.

IV. CONCLUSION

Load balancing in the cloud environment differs from classical thinking on load balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, and also presenting its own unique set of challenges. This paper proposes load balancing model based on the cloud partitioning concept. This model uses the switch mechanism depending upon the load status at the cloud partition when the request is made. Switch mechanism based load balancing allows choosing different strategies in different situations. This load balancing model applies to the public cloud to improve efficiency in the public cloud environment.

REFERENCES

[1] P.T.Jaeger, J.Lin, and M. grimes, Cloud computing and information policy: Computing in a policy cloud? Journal of Information Technology and politics, 2009.
[2] B.P.Rimal, E.Choi, and I.Lumb. A taxonomy and survey of Cloud Computing Systems. in Networked Computing and Advanced Information Management, International Conference. 2009.
[3] Z. Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid. "Availability and Load Balancing in Cloud Computing" 2011 International Conference on Computer and Software Modeling.
[4] Dongliang Zhang, Changjun Jiang,Shu Li, "A fast adaptive load balancing method for parallel particle-based simulations", Simulation Modelling Practice and Theory 17 (2009) 1032–1042.
[5] Dhinesh Babu L.D, P. VenkataKrishna, "Honey bee behaviour inspired load balancing of tasks in cloud computing environments", Applied Soft Computing 13 (2013) 2292–2303.
[6] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, "A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers", J. Parallel Distribution Computing. 72 (2012) 1254–1268.
[7] Yunhua Deng, Rynson W.H. Lau, "Heat diffusion based dynamic load balancing for distributed virtual environments", in: Proceedings of the17th ACM Symposium on Virtual Reality Software and Technology, ACM, 2010, pp. 203–210.
[8] Markus Esch, Eric Tobias, "Decentralized scale-free network construction and load balancing in Massive Multiuser Virtual Environments", in: Collaborative Computing: Networking, Applications and Worksharing, Collaborate Com, 2010, 6th International Conference on, IEEE, 2010, pp. 1–10.