Analysis of Load Balanced Model for Public Cloud Based on Cloud Partition Method

K. Venkateswarlu[1], Y.Leela krishna[2],
[1] Pursuing M.Tech in CSE, E.V.M College of Engineering & Technology, Andhra Pradesh, India
[2] Asst. Professor, Department of CSE, E.V.M College of Engineering & Technology, Andhra Pradesh, India
[1]venki0525.kotha@ gmail.com, leelakrishnayalavarthi@gmail.com [2]

Abstract: As we know Cloud computing is a concept that has many computers interconnected through a real time network like internet. Cloud computing means distributed computing. Cloud computing enables convenient, on-demand, dynamic and reliable use of distributed computing resources. Load balancing in the cloud computing environment has an important impact on the performance. Goodload balancing makes cloud computing more efficient and improves user satisfaction. This paper introduces a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applies the game theory to the load balancing strategy to improve the efficiency in the public cloud environment.

Key words: load balancing model; public cloud; cloud partition; game theory

I. INTRODUCTION

Cloud computing is an attracting technology in the field of computer science. In Gartner's report, it says that the cloud will bring changes to the IT industry. Generally as we know Cloud computing is a concept that has many computers interconnected through a real time network like internet. Cloud computing means distributed computing. Cloud computing enables convenient, on-demand, dynamic and reliable use of distributed computing resources.

The cloud computing model has five main characteristics

- on-demand service
- broad network access
- resource pooling
- flexibility
- measured service.

Users get service from a cloud without paying attention to the details. NIST gave a definition of cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. More and more people pay attention to cloud computing. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers.

Since the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is crucial to improve system performance and maintain stability. Load balancing schemes depending on whether the system dynamics are important can be either static or dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility. The model has a main controller and balancers to gather and analyze the information. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right load balancing strategy.

1.1 Proposed Work

The load balancing model given in this article is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. 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.

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in Different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. The load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions, the load balancing then starts: when a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally.

II. LITERATURE SURVEY

Challenges on Cloud Computing Very difficult problem to maintain the stability of processing many jobs in the Cloud Computing. The job arrival pattern cannot be predicted and the capacities of each node in the cloud differ. Hence for balancing the load, it is crucial to control workloads to improve system performance and maintain stability.

2.1 Challenges for Load Balancing

There are some qualitative metrics that can be improved for better load balancing in cloud computing.

Throughput: It is the total number of tasks that have completed execution for a given scale of time. It is required to have high through put for better performance of the system.

Associated Overhead: It describes the amount of overhead during the implementation of the load balancing algorithm. It is a composition of movement of tasks, inter process communication and inter processor. For load balancing technique to work properly, minimum overhead should be there.

Fault tolerant: We can define it as the ability to perform load balancing by the appropriate algorithm without arbitrary link or node failure. Every load balancing algorithm should have good fault tolerance approach.

Migration time: It is the amount of time for a process to be transferred from one system node to another node for execution. For better performance of the system this time should be always less.

Response time: In Distributed system, it is the time taken by a particular load balancing technique to respond. This time should be minimized for better performance.

Resource Utilization: It is the parameter which gives the information within which extant the resource is utilized. For efficient load balancing in system, optimum resource should be utilized.

Scalability: It is the ability of load balancing algorithm for a system with any finite number of processor and machines. This parameter can be improved for better system performance.

Performance: It is the overall efficiency of the system. If all the parameters are improved then the overall system performance can be improved.

There have been many studies of load balancing for the cloud environment. Load balancing in cloud computing was described in a white paper written by Adler who introduced the tools and techniques commonly used for load balancing in the cloud. However, load balancing in the cloud is still a new problem that needs new architectures to adapt to many changes. Chaczko et al. described the role that load balancing plays in improving the performance and maintaining stability.

There are many load balancing algorithms, such as Round Robin, Equally Spread Current Execution Algorithm, and Ant Colony algorithm. Nishant et al. used the ant colony optimization method in nodes load balancing. Randles et al. gave a compared analysis of some algorithms in cloud computing by checking the performance time and cost. They concluded that the ESCE algorithm and throttled algorithm are better than the Round Robin algorithm. Some of the classical load balancing methods is similar to the allocation method in the operating system, for example, the Round Robin algorithm and the First Come First Served (FCFS) rules. The Round Robin algorithm is used here because it is fairly simple.

III. SYSTEM MODEL

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. The architecture is shown in Fig.1.

The load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions, the load balancing then starts: when a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the

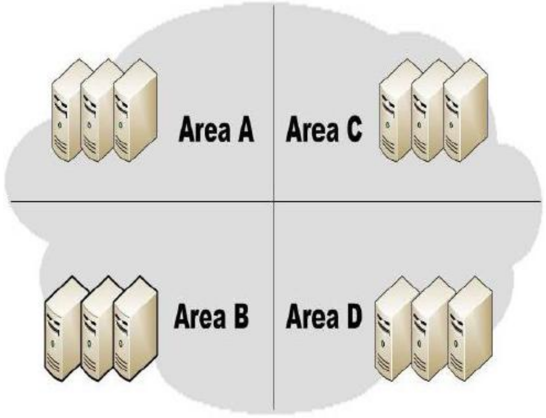

Fig. 1 typical cloud partitioning.

cloud partition load status is not normal, this job should be transferred to another partition. The whole process is shown in Fig.2.

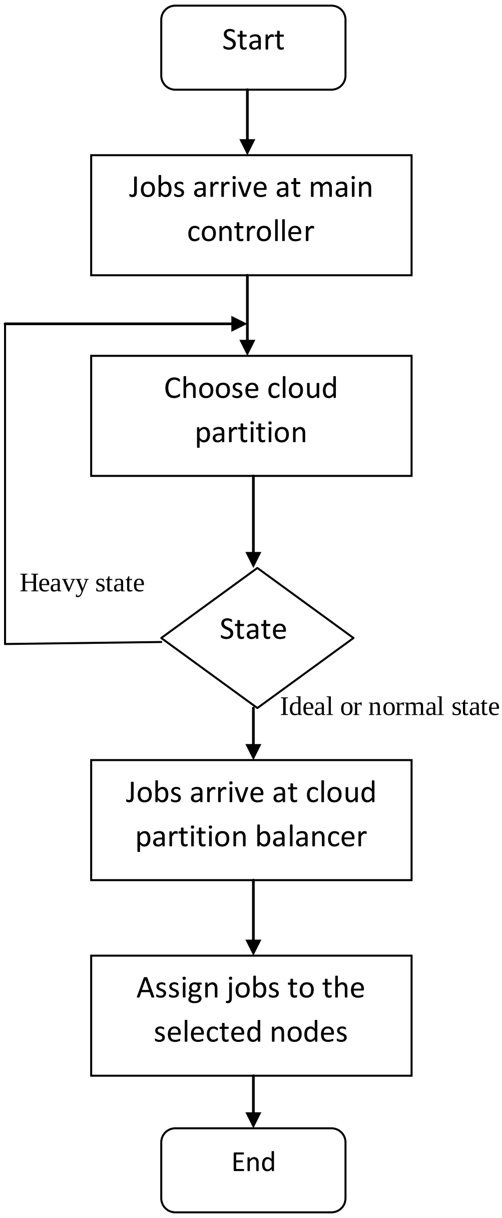

Fig. 2 Strategy for job assignment

3.1 Main controller and balancers

The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs. The relationship between the balancers and the main controller is shown in Fig.3.

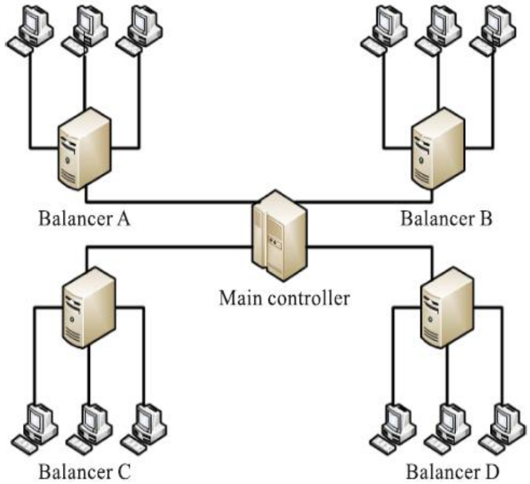

Fig. 3 Relationships between the main controllers, the balancers, and the nodes.

3.2 Allocating jobs to the cloud partition

When a job arrives at the public cloud, the first step is to choose the right partition. The status of the cloud partition can be broadly divided into three types, they are:

I. Ideal

II. Normal

III. Overload

(1) **Idle:** When the percentage of idle nodes exceeds α, change to idle status.

(2) **Normal:** When the percentage of the normal nodes exceedsβ, change to normal load status.

(3) **Overload:** When the percentage of the overloaded nodes exceedsγ, change to overloaded status.

The parameters α,β and γ are set by the cloud partition balancers. The main controller has to communicate with the balancers frequently to refresh the status information. The main controller then dispatches the jobs using the following strategy: When job i arrives at the system, the main controller queries the cloud partition where job is located. If this location's status is idle or normal, the job is handled locally. If not, another cloud partition is found that is not overloaded.

3.3 Assigning jobs to the nodes in the cloud partition

The cloud partition balancer gathers load information from every node to evaluate the cloud partition status. This evaluation of each node's load status is very important. The first task is to define the load degree of each node. The node load degree is related to various static parameters and dynamic parameters. The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc.

IV. CLOUD PARTITION LOAD BALANCING STRATEGY

4.1 Background

Good load balance will improve the performance of the entire cloud. However, there is no common method that can adapt to all possible different situations. Various methods have been developed in improving existing solutions to resolve new problems. Each particular method has advantage in a particular area but not in all situations. Therefore, the current model integrates several methods and switches between the load balance method based on the system status .A relatively simple method can be used for the partition idle state with a more complex method for the normal state. The load balancers then switch methods\as the status changes. Here, the idle status uses an improved Round Robin algorithm while the normal status uses a game theory based load balancing strategy.

Here we need to apply load balancing strategy for both when the system is in ideal state and normal state. Let see how we can apply these strategies in detailed.

4.2 Load balancing strategy for Ideal case

When the cloud partition is idle, many computing resources are available and relatively few jobs are arriving. In this situation, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be used. There are many simple load balance algorithm methods such as the Random algorithm, the Weight Round Robin, and the Dynamic Round Robin. The Round Robin algorithm is used here for its simplicity. The Round Robin algorithm is one of the simplest load balancing algorithms, which passes each new request to the next server in the queue. The algorithm does not record the status of each connection so it has no status information. In the regular Round Robin algorithm, every node has an equal opportunity to be chosen. However, in a public cloud, the configuration and the performance of each node will be not the same; thus, this method may overload some nodes. Thus, an improved Round Robin algorithm is used , which called "Round Robin based on the load degree evaluation".

4.3 Load balancing strategy for normal case

When the cloud partition is normal, jobs are arriving much faster than in the idle state and the situation is far more complex, so a different strategy is used for the load balancing. Each user wants his jobs completed in the shortest time, so the public cloud needs a method that can complete the jobs of all users with reasonable response time. Penmatsa and Chronopoulos proposed a static load balancing strategy based on game theory for distributed systems. And this work provides us with a new review of the load balance problem in the cloud environment. As an implementation of distributed system, the load balancing in the cloud computing environment can be viewed as a game. Game theory has non-cooperative games and cooperative games.

In cooperative games, the decision makers eventually come to an agreement which is called a binding agreement. Each decision maker decides by comparing notes with each others. In non-cooperative games, each decision maker makes decisions only for his own benefit. The system then reaches the Nash equilibrium, where each decision maker makes the optimized decision. The Nash equilibrium is when each player in the game has chosen a strategy and no player can benefit by changing his or her strategy while the other players strategies remain unchanged.

V. CONCLUSION & FUTURE SCOPE

Load balancing in the cloud computing environment has been an important impact on the performance. Good load balancing makes cloud computing more efficient and improves user satisfaction. This article introduced a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applied the game theory to the load balancing strategy to improve the efficiency in the public cloud environment.

In future study we will try to find other load balance strategy because other load balance strategies may provide better results, so tests are needed to compare different strategies. Many tests are needed to guarantee system availability and efficiency. Also we will address the development of game theoretic models for load balancing in the context of uncertainty as well as game theoretic models for dynamic load balancing in future. We also plan to develop dynamic load balancing schemes based on dynamic game theory that provide fairness by taking the current system load into account and also consider other aspects of heterogeneity.

REFERENCES

[1] 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 Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.

[2] 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.

[3] 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.

[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] 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.

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

[7] 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.

[8] P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/ publications/nistpubs/800- 145/SP800-145.pdf, 2012. [4] Microsoft Academic Research, Cloud computing, http:// libra.msra.cn/Keyword/6051/cloud computing?query= cloud%20computing, 2012.

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

[10] N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.

[11] 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.

[12] 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.

[13] 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 Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.

