ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
DIIF 0.876

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 622-626

Improved Dynamic Load Balance Model on Gametheory for the Public Cloud

[1] Rayapu Swathi,[2] N.Parashuram, [3]Dr S.Prem Kumar
[1](M.Tech), CSE,
[2]Assistant Professor, Department of Computer Science and Engineering
[3]Professor & HOD, Department of computer science and engineering,
G.Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.

Abstract:- Cloud computing is an enhancing technology in the field of computer science. Cloud computing is an efficient and scalable but maintaining the stability of processing several jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. Load balancing in the cloud computing surroundings has an imperative impact on the performance. Excellent load balancing makes cloud computing more efficient and improves user satisfaction. At present cloud computing is one of the utmost platforms which deliver storage of data in very lowers cost and accessible for all time over the internet. But it has more serious issue like security, load management and fault tolerance. Load balancing in the cloud computing environment has a significant influence on the presentation. The algorithm relates the game theory to the load balancing approach to increase the proficiency in the public cloud environment. This article announces an improved load balance model for the public cloud centered on the cloud segregating concept with a switch mechanism to select different approaches for different circumstances.

Keywords: Load Balancing, Cloud Partitioning, Load Balancing Models, Public Cloud.

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. The cloud is changing our life by providing users with new types of services. 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. 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.

II. RELATED WORK

In 2013, Xu, Gaochao et al [1] presented A load balancing model based on cloud partitioning for the public cloud. The load balancing model 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.There are many straightforward load balance algorithm methods such as the Weight Round Robin, the Random algorithm, and the Dynamic Round Robin [7].

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 [1].

Shivaratri et al [3] focuses on the problem of judiciously and transparently redistributing the load of the system among its nodes so that overall performance is maximized. They also discussed several key issues in load distributing for general-purpose systems, including the motivations and design trade-offs for loaddistributing algorithms. They also presented loaddistributing policies used in existing systems and draw conclusions about which algorithm might help in realizing the most benefits of load distributing. They compare various load distribution algorithms with their benefits and losses. The ability of load distributing to improve performance is intuitively obvious when work arrives at some nodes at a greater rate than at others, or when some nodes have faster processors than others. Performance advantages are not so obvious when all nodes are equally powerful and have equal workloads over the long term [3].

Zhu, Yan et al [4] suggested "Efficient provable data possession for hybrid clouds. They focused on the construction of PDP scheme for hybrid clouds, supporting privacy protection and dynamic scalability. They first provide an effective construction of Cooperative Provable Data Possession (CPDP) using Homomorphic Verifiable Responses (HVR) and Hash Index Hierarchy (HIH). This construction uses homomorphic property, such that the responses of the client's challenge computed from multiple CSPs can be combined into a single response as the final result of hybrid clouds. By using this mechanism, the clients can be convinced of data possession without knowing what machines or in which geographical locations their files reside. More importantly, a new hash index hierarchy is proposed for the clients to seamlessly store and manage the resources in hybrid clouds. Their experimental results also validate the effectiveness of our construction [6]. Lori MacVitte presented a Cloud Balancing: The Evolution of Global Server Load Balancing. Cloud balancing is still new, but the technology to add value is available today.

III. METHODOLOGY

Some of the classical load balancing methods is alike to the allocation method in the operating system. There are several load balancing algorithms, such as Round Robin, Game theory Algorithm and Ant Colony algorithm. For instance, the Round Robin algorithm and the First Come First Served (FCFS) rules. The Round Robin algorithm is used here because it is fairly simple. There have been several studies of load balancing for the cloud environment. Load balancing in cloud computing was defined in a white paper written by Adler who presented the tools and techniques generally used for load balancing in the cloud. Though load balancing in the cloud is still a problem that requests new architectures to adapt various changes. Analysis of some of algorithms in cloud computing by examining the performance time and cost.

EXISTING METHOD Load balancing systems liable, whether the system dynamics are significant and can be either static or dynamic. Static patterns do not use the system information and are less complex while dynamic patterns will carry additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility.

DISADVANTAGES Workload control is vital to improve system performance and maintain stability. Cloud computing environment is a very complex issue with load balancing receiving. The job arrival design is not predictable and the capabilities of each node in the cloud differ for load balancing problem.

PROPOSED METHOD Load balancing arrangements liable on whether the system dynamics are important can be either static or dynamic. The load balancing model is meant at the public cloud which has many nodes with dispersed computing resources in various different geographic locations. This model splits the public cloud into several cloud partitions. When the environment is very huge and complex, these separations simplify the load balancing. The cloud has a key controller that indicates the appropriate partitions for received jobs while the balancer for each cloud partition chooses the best load balancing approach.

A**DVANTAGES** When the environment is huge and compound these divisions streamline the load balancing. The role that loads balancing plays in refining refining the presentation and maintaining stability.

IV.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 [10]. 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

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 622-626

of the public cloud with divisions based on the geographic locations. The architecture is shown in Figure 2
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. 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 Fig1.

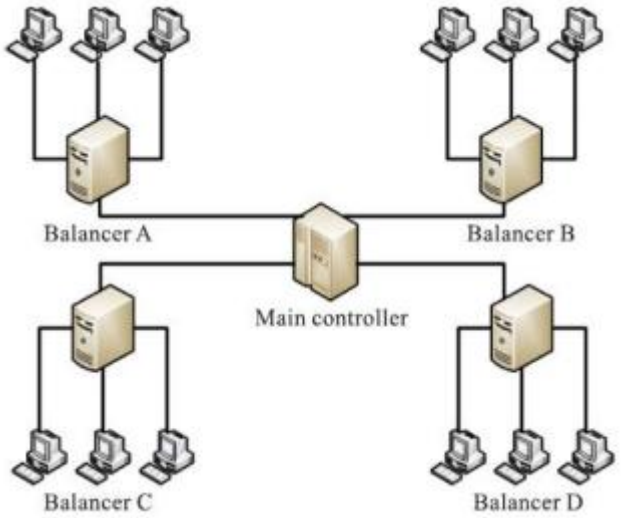

Figure 1: Relationships between the main controllers, the balancers, and the nodes.

When a job arrives at the public cloud, the first step is to choose the right partition. The cloud partition status can be divided into three types: **(1) Idle:** When the , change to idleαpercentage of idle nodes exceeds status. (2) **Normal:** When the percentage of the normal , change to normal load status. (3)βnodes exceeds **Overload:** When the percentage of the overloaded nodes , change to overloaded status. The parametersγexceeds are set by the cloud partition balancers. Theγ, and β, α main controller has to communicate with the balancers frequently to refresh the status information. 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 nodes. 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 [1].
When the cloud partition is inactive, many computing resources are accessible and relatively few jobs are incoming. In this phase, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be employed.

V. CLOUD PARTITION LOAD BALANCING STRATEGY

Motivation Good load balance will improve the performance of the entire cloud. However, there is no been developed in improving existing solutions to resolve new problems? Each particular method has advantage common method that can adapt to all possible different situations. Various methods have in a particular area but not in all situations. Therefore, the current model integrates several methods and switches between the load balance methods 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.
Load balance strategy for the idle status
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". The algorithm is still fairly simple. Before the Round Robin step, the nodes in the load balancing table are ordered based on the load degree from the lowest to the highest. The system builds a circular queue and walks through the queue again and again. Jobs will then be assigned to nodes with low load degrees. The node or-

der will be changed when the balancer refreshes the Load Status Table. However, there may be read and write inconsistency at the refresh period T . When the balance table is refreshed, at this moment, if a job arrives at the cloud partition, it will bring the inconsistent problem. The system status will have changed but the information will still be old. This may lead to an erroneous load strategy choice and an erroneous nodes order. To resolve this problem, two Load Status Tables should be created as: Load Status Table_1 and Load Status Table_2. A flag is also assigned to each table to indicate Read or Write. When the flag = "Read", then the Round Robin based on the load degree evaluation algorithm is using this table. When the flag = "Write", the table is being refreshed, new information is written into this table. Thus, at each moment, one table gives the correct node locations in the queue for the improved Round Robin algorithm, while the other is being prepared with the updated information. Once the data is refreshed, the table flag is changed to "Read" and the other table's flag is changed to "Write". The two tables then alternate to solve the inconsistency. The process is shown in Fig.4. Load balancing strategy for the normal status 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[13] 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. Fig. 2 The solution of inconsistently problem. There have been many studies in using game theory for the load balancing. Grosu et al.[14] proposed a load balancing strategy based on game theory for the distributed systems as a non-cooperative game using the distributed structure. They compared this algorithm with other traditional methods to show that their algorithm was less complexity with better performance. Aote and Kharat[15]gave a dynamic load balancing model based on game theory. This model is related on the dynamic load status of the system with the users being the decision makers in a non-cooperative game. Since the grid computing and cloud computing environments are also distributed system, these algorithms can also be used in grid computing and cloud computing environments.

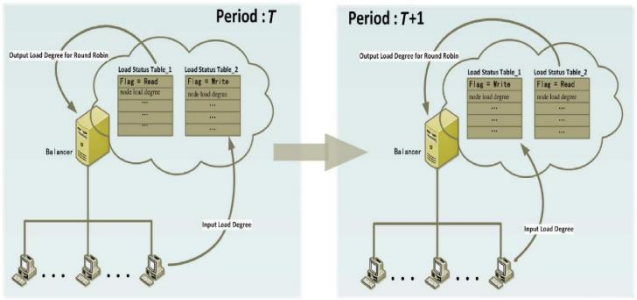

Fig 2.The solution of inconsistently problem

VI. CONCLUSION

Load balancing is the utmost essential issue in the system to allocate load in well-organized manner. It also confirms that each computing resource is dispersed efficiently and objectively. Existing load balancing method have been studied and mostly focus on reducing overhead, reducing migration time and improving performance. The response time and data transfer cost is a challenge of every engineer to progress the products that can upsurge the business performance and high customer satisfaction in the cloud based sector. Cloud computing system has broadly been implemented by the industry however there are many existing problems like load balancing, migration of virtual machine, server unification which have been not yet completely addressed.

REFERENCES

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

[2] 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.

[3] 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.

[5] Google Trends, Cloud computing, http://www.google.
com/trends/explore#q=cloud%20computing, 2012.
[6] N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.
[7] B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/whitepapers/ Load-Balancingin-the-Cloud.pdf, 2012
[8] 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.
[9] 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
[10] 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 InformationNetworking and Applications, Perth, Australia, 2010, pp. 551- 556.
[11] A. Rouse, Public cloud, http://searchcloudcomputing.techtarget.com/d efinition/public-cloud, 2012. [12] D. MacVittie, Intro to load balancing for developers —The algorithms, https://devcentral.f5.com/blogs/us/introtoload-balancing-for-developers-ndash-the- algorithms, 2012.
[13] S. Penmatsa and A. T. Chronopoulos, Gametheoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol. 71, no. 4, pp. 537-555, Apr. 2011.
[14] 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
[15] . 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. Gaochao Xu received his BEng degree