IJSEAT International Journal of Science Engineering and Advance Technology

A Load Harmonizing Standard Based On Cloud Segregating For The Public Cloud

Dr.D. Srujan Chandra Reddy#1, Kandagadla Nishitha#2
#1 Department of CSE, PBR Visvodaya Institute of Technology & Science, Kavali.
#2student, M.Tech (C.S) and Department of CSE, PBR Visvodaya Institute of Technology & Science,Kavali.

ABSTRACT:

Concept of Harmonizing Load in cloud computing has an important effect on the performance. A cloud computing system which does not use load Harmonizing has numerous drawbacks. Now-a days the usage of internet and related resources has increased widely. Due to this there is tremendous increase in workload. So there is uneven distribution of this workload which results in server overloading and may crash. In such systems the resources are not optimally used. Due to this the performance degrades and efficiency reduces. Cloud computing is made more efficient by better load Harmonizing methods. User satisfaction also improves. This paper introduces a better load Harmonizing Standard for the public cloud based on the cloud segregating concept. A switch mechanism is introduced here to choose different strategies for different situations. The public cloud is divided into cloud partitions and different strategies are applied to balance the load on clouds. This paper introduces a system which has main controller, balancers and servers. The main controller selects the appropriate balancer for a particular job. The balancer further selects the server having minimum load. Hence, this system will help dynamically allocate jobs (data) to the least loaded server which will result in an efficiently balanced cloud system.

Key words: load Harmonizing Standard; public cloud; cloud partition; game theory

1 INTRODUCTION

Cloud computing is an attracting technology in the field, it says that crucial to improve system performance and maintain of computer science. In Gartner's report stability. Load Harmonizing

Schemes depending on the cloud will bring changes to the IT industry. The whether the system dynamics are important can be cloud is changing our life by providing users with new either static or dynamic [6]. Static schemes do not use types of services. Users get service from a cloud. NIST gave a the system information and are less complex while without paying attention to the details definition of cloud computing as a Standard for dynamic schemes will bring additional costs for the enabling ubiquitous, convenient, on-demand network system but can change as the system status changes. An access to a shared pool of configurable computing dynamic scheme is used here for its flexibility.

The resources (e.g. networks ,servers ,storage ,Standard has a main controller and balancers to gather applications, and services) that can be rapidly and analyse the information. Thus, the dynamic control provisioned and released with minimal management. More and more has little influence on the other working nodes. The effort or service provider interaction. Cloud system status then provides a basis for choosing the people pay attention to cloud computing right load harmonizing strategy. Computing is efficient and scalable but maintaining the load Harmonizing Standard given in this article is stability of processing so many jobs in the cloud aimed at the public cloud which has numerous nodes computing environment is a very complex problem with distributed computing resources in many different with load Harmonizing receiving much attention for geographic locations. Thus, this Standard divides the researchers. Public cloud into several cloud partitions. When the since the job arrival pattern is not predictable and environment is very large and complex, these divisions the capacities of each node in the cloud differ, for load simplify the load Harmonizing. The cloud has a main Harmonizing problem, workload control is controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load harmonizing strategy.

2 RELATED WORK

There have been many studies of load harmonizing for the cloud environment. Load Harmonizing in cloudcomputing was described in a white paper written by Adler[7] who introduced the tools and techniques commonly used for load Harmonizing in the cloud. However, load harmonizing in the cloud is still a new problem that needs new architectures to adapt too many changes. Chaczko et al. [8] described the role that load Harmonizing plays in improving the performance and maintaining stability.

There are many load harmonizing 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 Harmonizing. 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 harmonizing methods are 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.

3 SYSTEM STANDARD

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing Standard, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud Segregating 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 harmonizing strategy is based on the cloud Segregating concept. After creating the cloud partitions, the load Harmonizing then starts: when

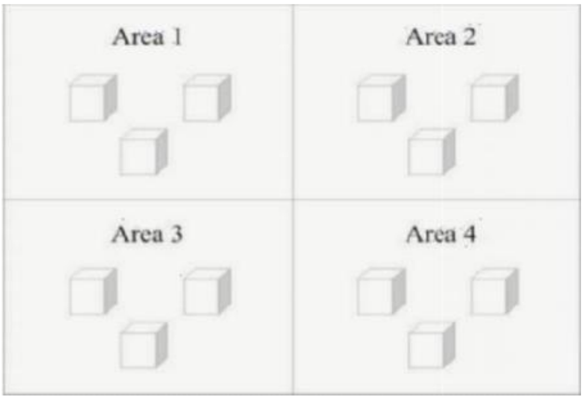


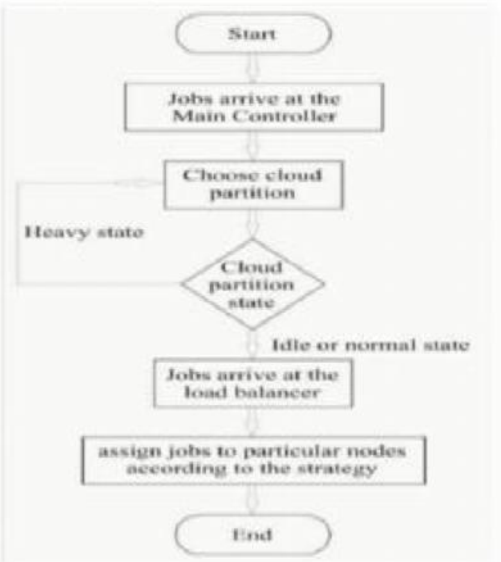
Fig. 1 Typical cloud Segregating.

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 Segregating can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition. The whole process is shown in Fig.2.

3.1MAIN 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 exceeds ‸, change to idle status.



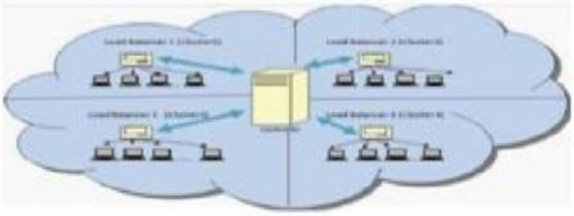


Fig. 2 Process workflow & System
(1)Idle: When the percentage of idle nodes
architecture

3.2 ASSIGNING JOBS TO THE CLOUDPARTITION

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:
(2)Normal: When the percentage of the normalnodes exceeds ˇ, change to normal load status.
(3)Overload: When the percentage of the overloaded nodes exceeds, change to overloaded status.

The parameters ‸, ˇ, and 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 is patches 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.

The load degree results are input into the Load Status Tables created by the cloud partition balancers. Each balancer has a Load Status Table and refreshes it each fixed period T. The table is then used by the balancers to calculate the partition status. Each partition status has a different load harmonizing solution. When a job arrives at a cloud partition, the balancer assigns the job to the nodes based on its current load strategy. This strategy is changed by the balancers as the cloud partition status changes.

4 CLOUD PARTITION LOADHARMONIZING STRATEGY

4.1 MOTIVATION

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 Standard 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 harmonizing strategy.

4.2LOAD 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 harmonizing 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

ISSN 2321-6905

Round Robin algorithm is used here for its simplicity.

The Round Robin algorithm is one of the simplest load harmonizing 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 Robinbasedon the load degreeevaluation".

The algorithm is still fairly simple. Before the Round Robin step, the nodes in the load harmonizing 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 order 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 and Load Status. 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 beingrefreshed, 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 Load Harmonizing 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 Harmonizing. 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 harmonizing 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 harmonizing 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 other's. In noncooperative 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 player's strategies remain unchanged.

There have been many studies in using game theory for the load Harmonizing. Grosu et AL proposed a load harmonizing 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 Kharatgave a dynamic load Harmonizing Standard based on game theory. This Standard 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. Previous studies have shown that the load harmonizing strategy for a cloud partition in the normal load status can be viewed as a non-cooperative game.In this Standard, the most important step is finding the appropriate value of sji. The current Standard uses the method of Grosu et al.called "the best reply" to calculate sji of each node, with a greedy algorithm then used to calculate sji for all nodes. This procedure gives the

Nash equilibrium to minimize the response time of each job. The strategy then changes as the node's statuses change.

5 FUTURE WORK

Since this work is just a conceptual framework, more work is needed to implement the framework and resolve new problems. Some important points are:

(1)Cloud division rules: Cloud division is not a simple problem. Thus, the framework will need a detailed cloud division methodology. For example, nodes in a cluster may be far from other nodes or there will be some clusters in the same geographic area that are still far apart. The division rule should simply be based on the geographic location (province or state).

(2)How to set the refresh period: In the data statistics analysis, the main controller and the cloud partition balancers need to refresh the information at a fixed period. If the period is too short, the high frequency will influence the system performance. If the period is too long, the information will be too old to make good decision. Thus, tests and statistical tools are needed to set a reasonable

refresh periods.

(3)A better load status evaluation: A good algorithm is needed to set Load degree high and Load degree low, and the evaluation mechanism needs to be more comprehensive.

(4)Find other load balance strategy: 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.

REFERENCES

[1] R. Hunter, The why of cloud, http:// www.gartner.com/ Display Document? Doc 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 /cloudcomputing? 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 harmonizing in the cloud:Tools, tips and techniques, http://www.rightscale. Com/info center/ whitepapers/ Load Harmonizingin-theCloud.pdf, 2012

[8] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, Availability and load Harmonizing in cloud computing, presented at the 2011 International Conference on Computer and Software Standarding,Singapore, 2011.

Dr.D. Srujan Chandra Reddy,
Assoc .Professor,
Department Of CSE, PBR VISVODAYAINSTITUTE OF TECHNOLOGY & SCIENCE, KAVALI.

Kandagadla Nishitha,
M.Tech(CS), Student,
Department Of CSE, PBR VISVODAYA INSTITUTE OF TECHNOLOGY&SCIENCE, KAVALI.