IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. XII (Mar-Apr. 2014), PP 48-53
www.iosrjournals.org

A Game Theory Modal Based On Cloud Computing For Public Cloud

[1]S.Hemachander @ Harikrishna, [2]R.Backiyalakshmi
[1]*M.Tech(CSE), Prist University, Puducherry*
[2]*A.P, Dept. of CSE, Prist University, Puducherry*

Abstract:*Cloud computing promises to increase the velocity with which applications are deployed, increase innovation, and lower costs, all while increasing business agility. This paper discusses how cloud computing Load balancing is done in public cloud. Load balancing in the cloud computing environment has an important impact on the performance. Load 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. A switch mechanism to choose different strategies for different situations. The game theory algorithm applies here to improve the efficiency of the load balancing strategy in the public cloud environment*

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

I. Introduction

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. 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. Due to the exponential growth of cloud computing, it has been widely adopted by the industry and there is a rapid expansion in data-centers.

1.1 Cloud Computing

A. Overview

Cloud computing is a on demand service in which shared resources work together to perform a task to get the results in minimum possible time by distribution of any dataset among all the connected processing units. Cloud computing is also referred to refer the network based services which give an illusion of providing a real server hardware but in real it is simulated by the software's running on one or more real machines. Such virtual servers do not exist physically so they can be scaled up and down at any point of time.

B. Cloud Infrastructure

A cloud consists of a group of processing unit put together, so we can define that the basic unit of cloud is processing units, which are grouped together to achieve same goal. The groups of processing units are connected to the master processing unit which is responsible for assigning the tasks to its slave units (single processing unit). The groups master system is again connected to the head node of the cloud which is responsible for receiving the tasks, dividing it into small tasks and then assigning it to the group masters which further assigns the tasks to its slaves. Please see the Fig.1 for the pictorial representation of the same.

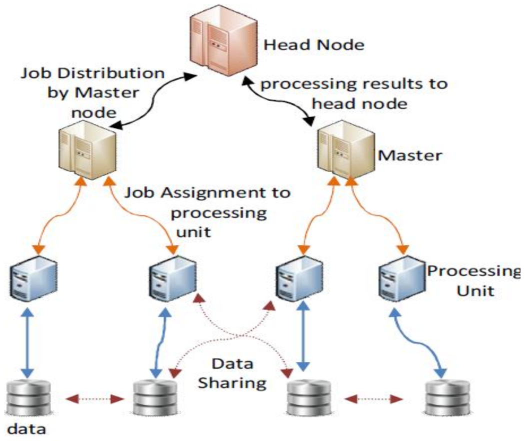

Head Node
Job Distribution by Master node
processing results to head node
Master
Job Assignment to processing unit
Processing Unit
Data Sharing
data

Fig 1. Cloud Infrastructure

II. Related Works:

Good load balancing makes cloud computing more efficient and also improves user satisfaction. 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. This paper is aimed at the public cloud which has numerous nodes. 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. It introduces a switch mechanism to choose different strategies for different situations

This paper divides the public cloud into cloud partitions and applies different strategies to balance the load on cloud. This paper gives an idea for balancing the load on clouds. It helps to avoid overloading of servers and improve response times.

In general, load balancing algorithms follow two major classifications:

- Depending on how the charge is distributed and how processes are allocated to nodes (the system load);
- Depending on the information status of the nodes (System Topology).

a) Classification According to the System Load

- Centralized approach: In this approach, a single node is responsible for managing the distribution within the whole system.
- Distributed approach: In this approach, each node independently builds its own load vector by collecting the load information of other nodes. Decisions are made locally using local load vectors. This approach is more suitable for widely distributed systems such as cloud computing.
- Mixed approach: A combination between the two approaches to take advantage of each approach.

b) Classification According to the System Topology

- Static approach: This approach is generally defined in the design or implementation of the system.
- Dynamic approach: This approach takes into account the current state of the system during load balancing decisions. This approach is more suitable for widely distributed systems such as cloud computing.
- Adaptive approach: This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms. This approach is able to offer better performance when the system state changes frequently. This approach is more suitable for widely distributed systems such as cloud computing.

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[11]. 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.

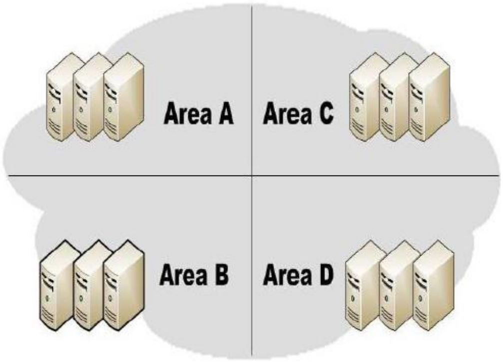


Fig. 1 Typical cloud partitioning.

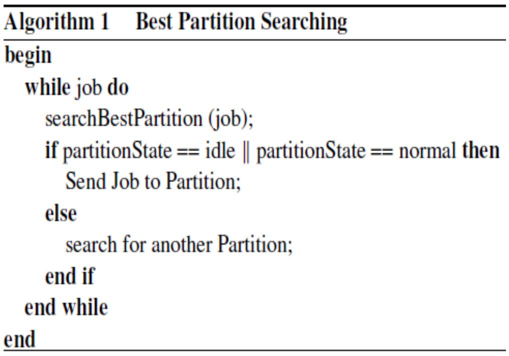
When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

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. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs.

3.2 Assigning jobs to the cloud partition

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 percentage of idle nodes exceeds alpha, change to idle status.
(2) **Normal**: When the percentage of the normal nodes exceeds beta, change to normal load status.
(3) **Overload**: When the percentage of the overloaded nodes exceeds gamma, change to overloaded status.



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
```

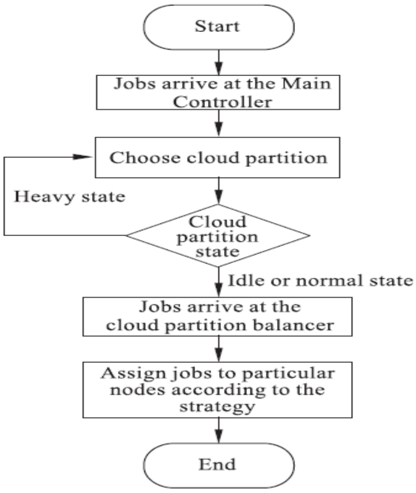

Start
Jobs arrive at the Main Controller
Choose cloud partition
Heavy state
Cloud partition state
Idle or normal state
Jobs arrive at the cloud partition balancer
Assign jobs to particular nodes according to the strategy
End

Fig. 2 Job assignment strategy.

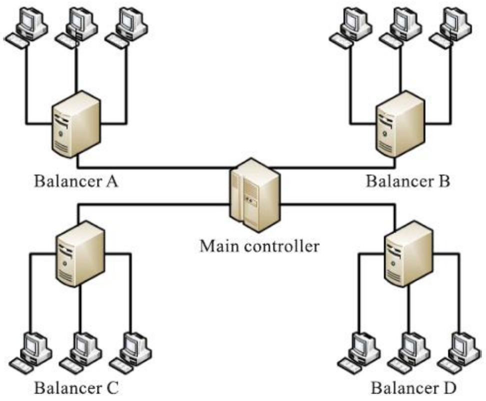


Fig. 3 Relationships between the main controllers, the balancers, and the nodes.

3.3 Assigning jobs to the nodes in the cloud partition

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.

Step 1 Define a load parameter set: F ={F1; F2;…; Fm) with each parameter being either static or dynamic. m represents the total number of the parameters.
Step 2 Compute the load degree as:

$$\text{Load\_degree}(N) = \sum_{i=1}^{m} \alpha_i F_i ,$$

$\alpha_i (\sum_{i=1}^{n} \alpha_i = 1)$ are weights that may differ for different kinds of jobs.N represents the current node.
Step 3 Define evaluation benchmarks. Calculate the average cloud partition degree from the node load degree statistics as:

$$\text{Load\_degree}_{\text{avg}} = \frac{\sum_{i=1}^{n} \text{Load\_degree}(N_i)}{n}.$$

The bench mark Load degreehigh is then set for different situations based on the Load degreeavg.
Step 4 Three nodes load status levels are then defined as:

- **Idle** When

Load degree. (N) = 0;
there is no job being processed by this node so the status is charged to Idle.

- **Normal** For

0 <Load_degree.(N)≤ Load_degreehigh;
the node is normal and it can process other jobs.

- **Overloaded** When

Load_degree high ≤ Load_degree. (N);
The node is not available and cannot receive jobs until it returns to the normal. 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.

IV. Cloud Partition Load Balancing 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.

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.

4.2 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.

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".

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.

4.3 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.

They compared this algorithm with other traditional methods to show that their algorithm was less complexity with better performance. Aote and Kharat 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 computingenvironments are also distributed system, thesealgorithms can also be used in grid computing and cloud computing environments. Previous studies have shown that the load balancing strategy for a cloud partition in the normal load status can be viewed as a noncooperative game, as described here.

V. 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 period.

(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.

VI. Conclusion

Cloud Computing has widely been adopted by the industry, though there are many existing issues like Load Balancing,Virtual Machine Migration, Server Consolidation, Energy Management, etc. which have not been fully addressed. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole Cloud to achieve a high user satisfaction and resource utilization ratio. It also ensures that every computing resource is distributed efficiently and fairly. This paper presents a concept of Cloud

References

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

[2]. R. X. T. and X. F. Z.. "A Load Balancing Strategy Based on the Combination of Static and Dynamic, in Database Technology and Applications (DBTA)", 2010 2nd International Workshop (2010), pp. 1-4

[3]. B. Adler, Load balancing in the cloud: Tools, tips and techniques,http://www.rightscale.com/infocenter/whitepapers/Load-Balancing-in-the- loud.pdf, 2012

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

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

[6]. Rouse, Public cloud, http://searchcloudcomputing. techtarget.com/definition/public-cloud, 2012.

[7]. 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.

[8]. 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

[9]. 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.

[10]. T. Kokilavani J.J. College of Engineering & Technology and Research Scholar, Bharathiar University, Tamil Nadu, India" Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing" International Journal of Computer Applications (0975 – 8887) Volume 20– No.2, April 2011.

Hemachander@Harikrihsna.S
M.tech, 1st year II sem ,Prist university, puducherry

R.Backiyalakshmi, A.P dept of CSE, Prist university, Puducherry