A Load Balancing Model based for Cloud Computing using Public Cloud

Pooja Khairnar
3, Shree ram Apt, Advit Colony,
Canada Corner, Nashik 2

Shiva Aher
B.E Computer
F3, Ratan App, Near
Gurudwara, Panchavati
Nashik3

Samadhan Diwane
B.E Computer
a/p: shelgaon tal, kallam
Dist: Osmanabad

ABSTRACT

It is difficult to maintain all the resources and services on single system. So by using Load balancing strategy we distribute load on single system to multiple system providing efficient response to user, resulting in improve user satisfaction. Load balancing in the cloud computing has an important impact on the performance. To improve user satisfaction there is a need of efficient cloud computing for that we need a good load balancing algorithm. This article shows a much 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. To improve the efficiency in the public cloud this algorithm applies game theory.

Keywords

Public cloud, Load Balancing Model, Cloud partition.

1. INTRODUCTION

Cloud computing is attracting computer science technology which brings the dramatic change in the world of computing [9]. Cloud computing technology provides user with the facility to access number of different services without paying attention to the background complexity [2]. Cloud computing provides: Software as a Service(SaaS), Platform as a Service(PaaS), and Hardware as a Service(HaaS).

Thus it is very important that service provided on cloud must be provide efficient result which result into user's satisfaction.

The main aim of concept of Load Balancing explained in this paper is to equalize the load on different server so that no server will be ideal. This yields higher efficiency and performance.

2. RELATED WORK

There are many inventor defined different studies on Load Balancing[3]. However there are still a new problems that can be overcome by different solutions. There are many load balancing algorithms have been developed till now. Such as, Round Robin, Ant Colony Algorithm, Equally Spread Current Execution Algorithm.

Round Robin algorithm is for equally spread execution, Ant Colony algorithm used the ant colony optimization[4] for optimizing nodes in load balancing, ESCE algorithm is much better than the Round Robin algorithm

2.1 A Fast Adaptive Load Balancing Method

In Fast Adaptive Load balancing method a binary tree structure is used to partition simulation region into subdomains [5]. The main advantage of Fast adaptive load balancing method is faster balancing speed, less elapsed time and less communication time cost of the simulation procedure. And Limitations of this method is it cannot maintain the topology that is neighboring cells cannot be maintained

2.2 Honey Bee Behavior Inspired Load Balancing

Honey bee behavior inspired load balancing as name suggest is inspired by the behavior of honeybee [6].

In bee hives there are two types of bees i.e. scout bees and forager bees. The job of scout bees is to find the food sources and when they found the source they come back to the bee hives and advertise this news by dancing. Dance is simply wiggle, vibration dance. This dance gives the idea about the quality and quantity of food available at the forged source [3]. Then the forager bees follows scout bees to the location of the forged source and then begin to collect the food.

After they return to bee hives they again dance i.e. wiggle, vibration to give idea about how much food left at the source[6]. Similarly to this when high priority task has to be submitted to virtual machine, is should consider virtual machine with minimum number of high priority tasks so that particular task will execute earlier. Current Load of all the virtual machine is calculated based on the information received from the data center. The advantage of this method is maximize throughput, minimum waiting time on task and minimum overhead. Disadvantage is if there are more high priority task in the queue then lower priority task have to stay for long time in queue.

2.3 Heat Diffusion Based Dynamic Load Balancing

Heat diffusion based load balancing proposed an efficient cell selection scheme, there are to algorithms for this method i.e. global and local diffusion. So working of this method is related to heat diffusion process. That is when two object is placed adjacently then heat is transfer from highly heated object to cold object.

Similarly Consider in the distributed virtual environment there are number of users, concurrent accessing by number of users can cause problem. This can be avoided by this algorithm. According to this the virtual environment is divided into large no of square cells and every square cell have objects associated. The working of algorithm is that every nodes in cell sends load to neighboring node in iterative fashion, and transfer is difference between load on current node to the load on neighboring node.

Advantage of this method is less communication overhead, high speed and little amount of calculation is required. Disadvantages are high network delay and waste of time because of several iterations.

Table 1: Literature Survey

Sr. No	Author	Algorithm and other technical	Advantage	Limitation
1	Dongliang Zhang, Changjun Jiang,Shu Li	2009, A fast adaptive load balancing method for parallel particle-based simulations	faster balancing speed, less elapsed time & less communication time	It cannot maintain topology i.e. neighboring cells cannot be maintained
2	Dhinesh Babu L.D, P. Venkata Krishna	2013, Honey bee behavior inspired load balancing of tasks in cloud computing environments	maximizing the throughput	If priority based queues are there, lower priority load stay continuously in queue
3	Yunhua Deng, Rynson W.H. Lau	2010, Heat diffusion based dynamic load balancing	reduce communication overhead and amount of calculation	network delay is high and several iterations are taken so there was a waste of time

3. PROPOSED SYSTEM

The main aim of proposed system is to equalize the load on different server so that no server will be ideal. This yields higher efficiency and performance [7].

This system consists of load balancer for every Cloud partition. Cloud partition will be based on the geographical location. Every load balancer is connected to the N number of nodes and gathers the information about the every node that is ‘load degree’. Load degree helps Load Balancer to keep track of status of node i.e. overloaded, ideal or normal.

Based on this information of nodes partition status of cloud is define. Every Load balancer is connected to the main controller. The job of main controller is to select the particular cloud partition based on the partition status and redirect the job request to Load balancer of that partition then load balancer of that partition will assign the job to particular node based on status of nodes.

4. METHODOLOGY

Load degree of node is related to various parameters, i.e. static and dynamic parameters. Static parameter refers CUP speed, Memory, Number of CPU’s. Dynamic parameter refers Network bandwidth, utilization ratio, etc.

Step1: Define a load parameter set:

F={F1, F2,…., Fm}

with each Fi (1<= i <= m, Fi € [0, 1]) parameter being either static or dynamic. m represents the total number of the parameters.

Step 2: Compute the load degree as:

$$\text{Load degree(N)} = \sum_{i=1}^{m} \alpha i \, Fi,$$

$\alpha i(\sum_{i=1}^{n} \alpha i = 1)$ 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:

$$\text{Load degree(avg)} = \sum_{i=1}^{n} load_{degree(Ni)}/_{n}$$

Step 4: Three nodes load status levels are then Defined as:

1) Idle When

Load_degree(N)= 0

2) Normal For

0 < Load degree(N) <= Load_degree(high)

3) Overloaded When

Load_degree(high) <= Load_degree(N)

5. ARCHITECTURE

As shown in the architecture when job ‘I’ arrive at node A it will be given to the main controller. Main controller contains the database consisting the status of all the cloud partitions. The job of main controller is to select particular partition based on the status and assign the job to balancer of that partition. Balancer of partition consist database containing Load degree of every node. Job will be assign to the appropriate node having status ideal or normal. At every short time interval balancer communicate with the every node to retrieve load degree and update the status in the database.

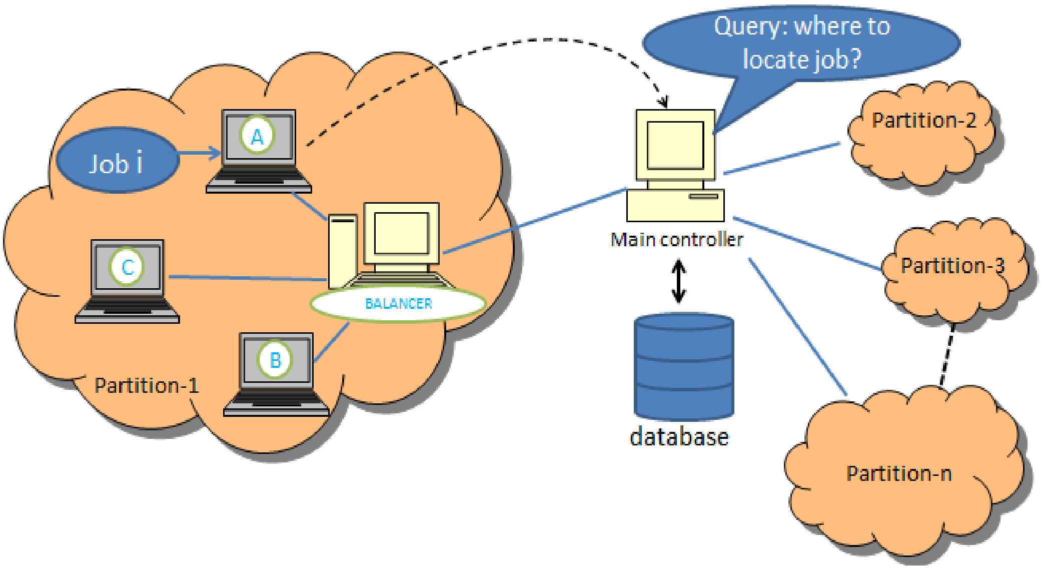

Fig 1: System Architecture

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template and the reviewers for there valuable comments and suggestions.

7. CONCLUSION & FUTURE SCOPE

Since this work is just a conceptual framework, more work is needed to implement the framework and resolve new problems. Cloud Computing has widely been adopted by the industry, even though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, etc. which have not been fully addressed. Load balancing in the cloud computing environment is required to distribute the excess workload evenly to all the nodes in the whole Cloud to achieve better resource utilization and ensure that every computing resource is distributed efficiently and fairly. The load balancing algorithms were first classified as static and dynamic. Various algorithms under each class along with their variations were also studied. Static Load Balancing algorithms attempt to achieve optimal utilization of resources by considering the size of the tasks and the machines. However, such information may not always be available at hand. Moreover some static load balancing algorithms like the min algorithm may cause heavy tasks to starve. This paper presents a System which successfully balances the load on the cloud.

8. REFERENCES

[1] Tsinghua Science And Technology Issn 1007-0214 04/12 pp34-39 Volume 18, Number 1, February 2013 Link:ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6449405

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

[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] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh,N. Nitin, and R. Rastogi, Load balancing of nodes in cloud using ant colonyoptimization, in Proc. 14th International Conference on Computer Modeling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.

[5] 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.

[6] Dhinesh Babu L.D, P. VenkataKrishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing 13 (2013)

[7] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[8] J.R. Koza, Gentic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.

[9] W.Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone, Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers, 1998.

[10] H.M. de Almeida, M.A. Gonc¸alves, M. Cristo, and P. Calado, "A Combined Component Approach for Finding Collection-Adapted Ranking Functions Based on Genetic Programming," Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 399-406, 2007.