An Efficient Dynamic Load Balancing Mechanism for Cloud Computing Environment

Rajagopal T. K. P. (✉ tkprgrg@gmail.com)

Hindusthan College of Engineering and Technology <https://orcid.org/0000-0002-6860-9336>

Venkatesan M.

K S R Institute for Engineering and Technology

Research Article

Keywords: Cloud computing, cloud environment, load balancing, distributed solution.

Posted Date: March 24th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-230430/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

An Efficient Dynamic Load Balancing Mechanism for Cloud Computing Environment

T. K. P. Rajagopal

Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India
tkprgrg@gmail.com

Dr. M. Venkatesan

Department of CSE, K S R Institute for Engineering and Technology, Tiruchengode, India
venkatesh.muthusamy@gmail.com

ABSTRACT

Cloud computing environment provides the information technology service and resources using in the Internet. It offers a variety of services such as IAAS (infrastructure as a service), SAAS (software as a service) and PAAS (platform as a service). The scenario of cloud computing environment mainly refers to both the cloud environment service provider and the cloud environment service user. Cloud environment service provider provides cloud environment resources and services to the cloud environment service provider. The cloud environment service provider should use its resources efficiently to make a profit, but the cloud environment challenging task for them is to maximize cloud environment resources. This article describes the energy saving approach by optimizing the use of hosts and minimizing the active host using migration techniques.

Keywords

Cloud computing, cloud environment, load balancing, distributed solution.

1. INTRODUCTION

Cloud computing environment is a type of computer that is based on sharing multiple computer resources instead of using a local server or personal server to handle the application[1]. There are many issues in the cloud computing environment such as cloud security, cloud privacy, cloud compliance, cloud stability and cloud load balance. They are all load balance primary and biggest problem[1]. The biggest question is the distribution of the load on the resources shared on the Internet. Suppose two or three servers are used to complete a task, another algorithm is needed to determine which cloud server is used to assign a task, which server is busy, and which cloud server is loaded. Some cloud computing environment systems have to deal with the situation where servers are distributed over different geographical location[2]. This paper defines different possible approaches to dynamic load balance in different cloud computing environment systems.

The first method defines a nature-inspired strategy for effectively balancing loads on distributed servers.

The second method took random samples of the computer domain and designed them so that all nodes were self-organizing.

The third approach equates the node cluster with similar behaviors and the loads within them.

The Forth algorithm compares the current host load with others and balances the load according to the current system load.

The last approach calculates the load of the host variable and balances the load accordingly [13].

2. CLOUD COMPUTING ENVIRONMENT

Cloud computing environment is primarily geographically distributed and developed to implement computing within different types of resources. Cloud computing has no specific definition, but it is defined as a set of distributed computers that can provide the required calculation resources and services with the help of the Internet [1]. As previously described it provides services such as IAAS, SAAS and PAAS to a geographically diverse clientele. A well-known example is the Amazon elastic system cloud environment, which provides the virtual machine environment, different configurations of the system CPU, system processor, and internal memory [3].

3. EXISTING TECHNOLOGY

In this existing technology section the different types of schemes for cloud environment load balancing technologies are defined and explained in various technologies and their behavior.

3.1 Honeybee Technique

The self-regulatory and behavioral-based algorithm was used to dynamic load balancing for the application layer [5]. This algorithm was motivated by the Honey bees and their approach for finding food.

Two attributes are defined in their methods here, one of the methods is forerunner and the other is inspect. A method forester bee searches for a suitable food available source, and when they find a specific food source, they coming back to the hive and accomplish a "vagal dance" [5].

The suitable source is carefully chosen according to the size, quality and distance of the source. All of these parameters are observed by Forage bees, and they was show the bees through skating. The scout bees then follow the forager to harvest of their food. This method is used as a resource pursuit tool for different computer applications. There are different virtual servers and virtual server's process different requests from users [4]. Each server performances as a forerunner or scout and places the ad on the billboard after successfully completing the request. Initially, each server roughly selects a virtual server to deliver a request, which calculates the current virtual server load and compares it with the overall virtual server load. If the load on that virtual server is less than the external requests, it will be assigned to that virtual server. If the load on that server is greater than the scout reads the billboard, it will follow the virtual server listed from the ad board. The load of virtual servers is calculated according to the CPU usages [5]. The picture shows how server allocation is done.

Fig 1. Server allocation in Honeybee foraging [5]

3.2. Random pro methodology

In this method the load of the server is described permitting to its connection. Allocated links define the capacity of the server. The full work of this methodology is initiating [6]. This methodology does not use a static network to display nodes and their existing resources. That dynamic network methodology was developed, which offers the virtual machines load sharing level, resource update approach.

Nowadays, define the dynamic network appliance by defining the in-degrees of the node total edges toward the node = N the free resources of that lump. When a nodule receives a request for a new tasks, it removes an incoming margin and reduces its own size. One degree deduction indicates that the total accessible resources are condensed to $N-1$ [6]. The process of removal and adding of edges is defined by the random model. The standard level defines the work attendance rate as equal to the work completion rate, so a dynamic network mechanism connecting all nodes is created.

The random pattern defines the removal and adding of edges. The random model describes how all the nodes in the network are taken approximately. The model of the boot node is fixed so that with each step it moves to the neighboring node. Therefore the removal of the tip and the adding of the edges are relational to the free sources of the tip [6].

3.3. Dynamic grouping

Related nodes or facilities are bundled composed to conserve load balance for cloud systems [8]. This procedure is well-thought-out as a self-integration procedure that transforms the network. This method can be used where nodes have same information of related types of nodes and they can act on its behalf. Each node is reactivated by dynamic clustering as follows [8]:

1. Any one type of node is the "initiator" node at random.
2. The "Matchmaker" node was assign to the launcher node.
3. The matchmaker node selects the node corresponding to the boot node and creates a connection between them.
4. The matchmaker removes the node connection between itself and the selected neighbor.

This approach has been extensively explored in [7], showing how it develops from a complex network to a stable network.

3.4 Balance Compressions

The simplest method introduced in Comparison and Equilibrium [9] is to compare the current host loads and maintain the equilibrium position by comparing the current host load with others.

The current host is randomly selected at a specific time. Using parameters such as no. Virtual machine operation is determined by the current host's CPU usage and storage usage load. The current host randomly selects other hosts from the network and compares their load to their own load. If the current host load exceeds the loads of the selected node, the current host will transfer its additional load to the selected node. All of the nodes are the system perform the same function on the network.

3.5 Dynamic comparison and load balance

This method was introduced in [10], which takes two gate values. The current host load defines the next step to be implemented. The current host load is calculated below:

$$\text{Host load} = x (\text{CPU usage}) + y (\text{RAM usage}) + z (\text{BW usage}) \quad \dots\dots\dots(1)$$

Where x, y and z are 0 [0, 1] and they are the weight coefficient.

The threshold values used here are calculated below:

Host Limit: This defines the maximum capacity of the host that the load host can perform with maximum performance.

$$\text{H_UTD (Host Top Door Value): host limit} \quad \dots\dots\dots(2)$$

β , where weight is the weight coefficient, which is selected by the seller.

H_LTD (minimum threshold value of host): host limit(3)
 α , where weight is the weight coefficient, which is selected by the seller.

This method can divide the server optimization problem into two sections. The first section distributes more loads to the current host using the load balancing method, and the second section reduces the number of active hosts on the network to support the green computing technology concept [10].

Part 1: However, there is a situation when the load of the host is higher than the upper threshold value pre-defined by the dealer, so the host is considered to have higher loads. Therefore the load balancing method is used to reduce the additional loads and transfer the additional loads to the host. Here it is done by adaptive comparison and dynamic load balance algorithm technique. It finds the host that can handle extra loads without too much load.

Part 2: Assume that there is a position where the load of the host is less than the minimum threshold value of the host, which is pre-defined by the dealer, so the load-transfer algorithm is used to transfer the additional load to the other host server. This is achieved through the migration of virtual machines. It finds the host that can handle extra loads without too much load.

Fig 2. Proposed algorithm for server optimization ^[10]

Fig 3. Result of proposed algorithm

Fig 4. Result of proposed algorithm

4. ANTICIPATED WORK

Dynamic comparison and load balancing methods does not consider server load at Virtual Machine allocation time. It simply assigns the Virtual Machine to the server, which then checks the load on the server at pre-defined intervals. Further steps are performed due to the high load or load state of the server. The load balance is used using the adaptive comparison and balance method if the server is overloaded and if the server is loaded, the server integration method is used for the shutdown server. The Virtual Machine is displaced to balance the loads after both situations. The proposed method works in estimating the host load after Virtual Machine allocation. If the server is overloaded accordingly the prediction will be made on a different host than the Virtual Machine. The proposed method predicts host loads after Virtual Machine allocation and predicts that Virtual Machine will be generated on a different host if the host has higher loads. It therefore reduces the number of migrations due to host overloading conditions. The proposed method supports Green System by reducing the number of active hosts using server integration. If the host is loaded the algorithm will move the Virtual Machine from the host and try to reduce the number of active hosts to save energy.

Fig 5. Proposed method

Fig 6. Result of proposed algorithm

Fig 7.Result of proposed algorithm

Fig 8. Result of proposed algorithm

5. RESULTS INVESTIGATION & EVALUATION

All tests are accompanied in the cloud simulator tool kit using cloudsim tool. The results evidently show that the proposed algorithm works better than our proposed algorithm. The proposed method minimizes migration by verifying host usage before Virtual Machine allocation. The proposed method supports green cloud computing by reducing the number of active hosts in the cloud computing environment.

This method decreases due to relocation and idle time during relocation. The proposed algorithm minimizes migration due to high load on the host, checking in advance whether the host is suitable for VM or not? If the host has enough capacity to create and run the new Virtual Machine without overloading the new Virtual Machine, it will be assigned a new Virtual Machine so that the host overload level cannot occur. The proposed method moves the Virtual Machine from the least loaded host to the ideal host. The results clearly show that the proposed method can save more energy with less time for our proposed method.

Declarations

I declare that I shall not submit the paper for publication in any other Journal or Magazine till the decision is made by journal editors. If the paper is finally accepted by the journal for publication, I confirm that I will either publish the paper immediately or withdraw it according to withdrawal policies.

***Funding** : Not Applicable

***Conflicts of interest/Competing interests** : Mobile computing and networking, Network management and operations, Traffic modeling.

***Availability of data and material** : 60% Transparency

***Code availability** : Custom code

4. REFERENCES

- [1] Martin Randles, A. Taleb-Bendiab and David Lamb, Cross Layer Dynamics in Self-Organizing Service Oriented Architectures. IWSOS, Lecture Notes in Computer Science, 5343, pp. 293-298, Springer, 2008.
- [2] S. Nakrani and C. Tovey, On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers. Adaptive Behavior 12, pp: 223-240 (2004).
- [3] F. Saffre, R. Tateson, J. Halloy, M. Shackleton and J.L. Deneubourg, Aggregation Dynamics in Overlay Networks and Their Implications for Self-Organized Distributed Applications. The Computer Journal, March 31st, 2008.
- [4] Martin randles, David lamb and taleb-bendiab, A comparative study into distributed load balancing algorithms for cloud computing, 2009 Second International Conference on Developments in E-Systems Engineering
- [5] Yi Zhao, Wenlong Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud" Fifth International Joint Conference on INC, IMS and IDC, IEEE, 2009, pp: 6/09.
- [6] Yatendra sahu, R.K.Pateriya, Rajeev kumar gupta, Cloud Server Optimization with Load Balancing and Green Computing Techniques Using Dynamic Compare and Balance Algorithm, 2013 5th International Conference on Computational Intelligence and Communication Networks
- [7] E. Di Nitto, D.J. Dubois, R. Mirandola, F. Saffre and R. Tateson, Applying Self-Aggregation to Load Balancing: Experimental Results. In Proceedings of the 3rd international Conference on Bio-inspired Models of Network, information and Computing Systems (Bionetics 2008), Article 14, 25 – 28 November, 2008.
- [8] A detailed analysis of distributed load balancing algorithm, Akbarali Kharodia,Rutvik Mehta,Miren Karamta,Dr.M.B.Potdar, International Journal of Scientific & Engineering Research, Volume 5, Issue 12, December-2014
- [9] Rajagopal, T K P, Venkatesan, M & Rajivkannan, A, (2019), ‘An Improved Efficient Dynamic Load Balancing Scheme Under Heterogeneous Networks in Hybrid Cloud Environment’, Wireless Personal Communications, Springer International Publication, vol. 111, Issue 3, pp. 1837-1851.
- [10] Rajagopal, T K P, Venkatesan, M & Rajivkannan, A (2018), ‘A Hybrid Dynamic Load Balancing for Energy-Efficient of Multi Access in Heterogeneous Wireless Networks’, TAGA Journal of Graphic Technology Technical Association of the Graphic Arts, Published by Swansea Printing Technology Ltd Sketty, Swansea, SA20QG, United Kingdom, vol. 14, pp. 1687-1700.
- [11] M.Kamarunisha, S.Ranichandra, T.K.P.Rajagopal,(2011) “Recitation of Load Balancing Algorithms In Grid Computing Environment Using Policies And Strategies An Approach”, International Journal of Scientific & Engineering Research Volume 2, Issue 3.

Figures

Figure 1

Server allocation in Honeybee foraging [5]

Figure 2

Proposed algorithm for server optimization [10]

Utilization at time frame 5.10

Figure 3

Result of proposed algorithm

Utilization at time frame 20.10

Figure 4

Result of proposed algorithm

Figure 5

Proposed method

Figure 6

Result of proposed algorithm

Figure 7

Result of proposed algorithm

Figure 8

Result of proposed algorithm