Indian Journal of Science and Technology, *Vol 9(20), DOI: 10.17485/ijst/2016/v9i20/85866, May 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Comparative Study of Load Balancing Algorithms in Cloud Computing Environment

R. Rajeshkannan[1]* and M. Aramudhan[2]

[1]Faculty of Computing Science and Engineering, VIT University, Vellore - 632014, Tamil Nadu, India;
rajeshkannan.r@vit.ac.in
[2]PKIET, Karaikal - 609603, Puducherry, India; aranagai@yahoo.co.in

Abstract

Background: Cloud computing is an emerging technology in a business. It is used to access the application or services and infrastructures at anywhere any time. Load balance means that to share the work across the multiple computing resources to serve higher for the user and utilize the resource with efficiency for reach the good performance of the application. These ideas are enforced with software system, hardware or each. **Statistical Analysis:** The various load balancing algorithms are compared with quality of service parameters in a cloud network. This analysis helps to identify the effective load balancing algorithm for optimizes resource use, maximizes throughput, minimizes response time, and avoids overload. **Findings:** The load balancer is indorsed in all situations for to supply service continuity and handling additional traffic. Therefore the effective load balancing algorithms needed to form economical resource utilization by provisioning of resources to cloud user's on-demand basis. **Application:** This paper discusses numerous load balancing algorithms so as to improve resource utilization and quality of services in cloud computing environment.

Keywords: Cloud Computing Model, Cloud Computing Characteristics, Load Balancing, Task Scheduling, Virtual Machine

1. Introduction

Cloud computing is that the next stage in evolution of the net. The cloud in cloud computing provides the means that through that everything from computing power to computing infrastructure, applications, business processes to non-public collaboration will be delivered to you as a service where and whenever you wish[1,7]. The cloud computing provides to the users by attracting typical services like software System As A Service (SAAS) wherever finish users will avail software system or services provided by SAAS while not getting and maintaining overhead, Platform As A Service (PAAS) wherever finish users will run and deploy their applications a lot of simply which has OS support and software system development and last however not the list Infrastructure As A Service (IAAS) that demands provisioning of infrastructural resources, sometimes in terms of virtual machines[2].

In this paper we present a survey of the current load balancing algorithms developed specifically to suit the Cloud Computing environments. We provide an overview of these algorithms and discuss their properties. In addition, we compare these algorithms based on the following properties: environment, limitation and time series. The rest of this paper is organized as follows. We discuss the load balancing concepts and environment in chapter 2. Then, Chapter 3 we go over the current literature and discuss the algorithms proposed to solve the load balancing issues in Cloud Computing. After that, we discuss and compare the relevant approaches in Chapter 4. We then conclude the paper and show possible areas of enhancement and our future plan of improving load balancing algorithms in Chapter 5.

** Author for correspondence*

2. Essential Characteristics of Cloud Computing

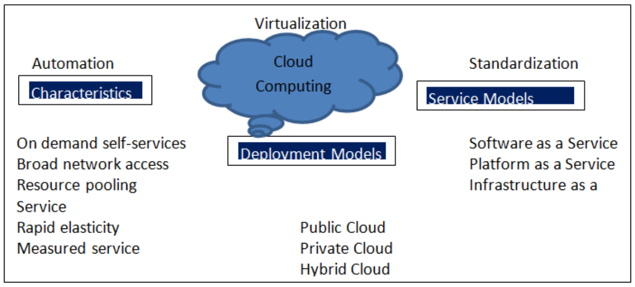

Figure 1. Cloud computing models and characteristics.

2.1 On Demand Self-Services

A business application should be secure through cloud service provider. The consumer will access to cloud services and he/she will amendment the services through on-line. The cloud application services are provided like email, software applications as required, with not requiring human interaction with each service provider. Cloud service providers providing on demand self-services include Amazon Web Services (AWS), Microsoft, Google, IBM and Salesforce.com[1,4].

2.2 Broad Network Access

The business management services using their smartphones, tablets and laptops. They will access the cloud services wherever they located using access point. The employee will work cloud project altogether time with help of mobility features so that the business will achieve good revenue and services. Cloud services are obtainable over the network and accessed through standard mechanisms of various cloud models like private, public, and hybrid for deployment[1].

2.3 Resource Pooling

When the consumer demand enlarged the computing resources are shared together and virtual resources are dynamically allocated for higher service continuity to the consumer. The resources are processing, memory, network bandwidth, application services and etc[1].

2.4 Rapid Elasticity

The cloud services are flexible and it can be rapidly and elastically provisioned to the customers. The customers will use the services at anywhere any time. The services are easily scaled based on the consumers[1].

2.5 Measured Service

Cloud computing resources are monitored, measured, controlled and reportable transparently to the consumer by using resource utilization. Therefore the computing resources are optimized for the economical resource utilization[1].

2.6 Multi Tenacity

Cloud computing advocated by the Cloud Security Alliance. It refers to the requirement for policy-driven enforcement, segmentation, isolation, governance, service levels, and chargeback/billing models for various client constituencies. Consumers may utilize a public cloud provider's service offerings or truly be from the identical organization, like completely different business units instead of distinct structure entities, however would still share infrastructure[1,4].

3. Load Balancing

Load balancing is a technique which is used to sharing the work load among the virtual machines and completed the task. The reason behind using these techniques is to serve better to the user without any service breaking. The main benefits of using cloud computing is cost, flexibility, scalability and availability to the end-users. As a major concern in these benefits, load balancing manage to scale up to increasing demands by dynamic work allocation to any or all nodes.

3.1 Hardware vs. Software Load Balancing

The load balancers are classified in two ways. The one is called software based load balancing and another is called hardware based load balancing. In software based load balancing, the service runs on each machine in a cluster. If anyone of the machine goes down, the other machines in the cluster are alerted by communication among the machines and they act to engage the additional load. The server machine would possibly act as agent and it eliminate a single point failure of a cluster[3].

Hardware-based load balancers placed in front of the servers and route all requests to the servers. The load balancing hardware manages the route based on system performance for instance CPU, memory and virtual machine utilization. This routing management provides distribute the server load based on server resources. These load balancers afford more expensive, single point failure and it added layer of security[3].

3.2 Load Balancing Environment

The load balancing is one of the scheduling types which are either static or dynamic environment with respect to cloud configuration. The static algorithm divided the traffic consistently to the servers. This algorithm requires a prior knowledge of system resources. So the decision not depends on the current state of the system. The dynamic algorithms taking decision based on actual current system state and allowed to move from over load machine to underutilized machine in real time. Mixed Load Balancing Algorithm focuses on symmetrical distribution of assigned computing task and reducing communication cost of distributed computing nodes[4,1].

We can clearly perceive that cloud computing falls under dynamic environment in order to we would like to focus dynamic load balancing algorithms. This algorithm can be classified in to two flavours. The one is called batch mode scheduling which means after collecting arrival of tasks are assigned to suitable resources and another one is termed immediate mode scheduling which suggest the tasks are assigned to the resources immediately based on minimum completion time and minimum execution time[4]. A good scheduling algorithm should influence the following characteristics:

- Minimum waiting time
- Minimum response time
- Maximum throughput
- Maximum CPU utilization

4. Related Work to Load Balancing Algorithms

4.1 Min-Min Load Balancing Algorithm

This is simple static algorithm and offers excellent performance in task scheduling. The cloud service manager find the completion time of every task. The new task has been waiting in a queue for execution. This algorithm assigns the task to the resource based on which task has minimum execution time to complete. The pseudo code is following

```
Procedure Minmin(Task Ti)
  {
         Find execution_Completion_Time of each task
         Store the execution_Completion_Time of task Ti
         in orderQueue
         repeat
    {
          for each task Ti in orderQueue
    {
    obtain minimum completionTime from orderQueue;
      assign task to vm;
      update the execution_Completion_Time;
    }
   }
    Until orderqueue empty;
  }
```

This algorithm works well when the task has minimum execution time however if task has maximum execution time then the task must be wait with undefined time. This will lead the starvation problem. This algorithm is best in the situations where the number of tasks with minimum completion time[5,6].

4.2 Max-Min Load Balancing Algorithm

This algorithm is following identical procedure of Min-Min algorithm. This algorithm calculates the execution completion time of all tasks. The maximum completion time is taken and assigned to the corresponding resources. This algorithm is best in the situations where the amount of tasks with maximum completion time and it take away the starvation. The task minimum completion time has been waiting in ordered queue until the other maximum completion time task must be completed. Here we can understand that this algorithm performs well in a static environment and both the algorithm has their merits and demerits based on the environments. The performance doesn't depend on the algorithm chosen but indeed the environment taken. Min-Min and Max-Min algorithms are equally performed on the static cloud environment[7,8].

4.3 Round - Robin Algorithm

It is the static load balancing algorithm and elects the node randomly for allocate a job. This algorithm assigned the resource in circular order and without using priority of the task. It's not suitable if any virtual machine is heavy loaded and some virtual machines are lightly loaded because the running time of all processes is not aware in advance. This algorithm is not preferred because prior prediction of execution time is not possible. The round robin algorithm is given below[9,10].

```
Create Q1, Q2;
        Q1 used to store ready process
        Q2 used to store blocked process
        New process place to end of Q1
        If task time interval finished then
                Move to end of Q1.
If I/O request or swapped out request is made by process then
Move process from Q1 to Q2.
If I/O operation is completed or ready to move from blocked processes then
        Move process from Q2 to Q1.
```

4.4 Genetic Load Balancing Algorithm

This algorithm implement in dynamic cloud environment and it used soft computing approach. This algorithm is experimental from the natural development. This algorithm provides better performance compare to RR and FCFS algorithm. The advantage of this algorithm is easily handle a vast search space, applicable to complex objective function and may avoid being trapping into local optimal solution[11].

GA's implementation mechanism is based on three steps.

- Selection Operator: Selects the initial population randomly.
- Crossover Operator: Find fitness pair of individuals for crossover.
- Mutation Operator: A small or low probability value is called mutation value. These bits are toggled from 0s to 1s or 1s to 0s. The output is new pool of individuals ready for crossover.

4.5 HBB-LB Algorithm

This is nature inspired algorithm for self-organization. HB consists of a queen and foragers, where forager is of two types; employed and unemployed. The foragers are informed about the food available nearby by waggle dance by scout bee (unemployed), the dance is to give the information to the other foraging bees about the distance, quality, direction and other information which is useful in getting the food[12]. This algorithm has similar principle in balance the work of the virtual machine. The HBB algorithm calculates the virtual machine workload then it decides whether it is overloaded, light weighted or balanced. The high priority of the task is off from the overload virtual machine and tasks are waiting for the light weight virtual machine. These tasks are known as scout bee in the next step. Honey Bee Behaviour inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task[13-15].

4.6 Ant Colony based Algorithm

The ant colony algorithm is based on the behaviour of the real ants. The ant can notice the optimal path where the source food is available. The Ants while seeking a path from their colony in search of food secrete a chemical called pheromone on the ground thus leaving a trail for other ants to follow the path. But this chemical evaporates with time. This approach aims to give efficient distribution of workload among the node. The ant maintains the record of every visited node for better making decision in future. The ant would deposit pheromones during their movement for other ant to select next node. The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, the solution set is continuously updated[16-18].

The basic algorithm is given below:

```
Step 1: Initialize the pheromone.
Step 2: Placing all the ants at the beginning of the VMs.
Step 3: Until all the ants have found food (solution) DO.
    •   Each ant should follow Do
                Choose a VM for new task
                Check for the pheromone intensity
                End Do
            End Do
    •   Find the best so far
    •   Update the pheromone

Step 4: End
```

4.7 Opportunistic Load Balancing (OLB) Algorithm

It is the static load balancing algorithm and this algorithm does not consider the current workload of the virtual machine. It keeps each node busy although they are already loaded with the task. This algorithm schedules the new task in randomly available virtual machine without checking workload of that machine. It provides load balance schedule without good result. The task executed slowly because it does not calculate execution time of the VM. This algorithm is not suitable in improving resource utilization[19].

4.8 Game Theory Algorithm

This dynamic algorithm works in public cloud environment. This algorithm is partition the cloud into three class namely idle, normal and overload based on the load degree. The public cloud includes many nodes and it located at different places. This partition helps us to manage the large cloud. The load balancing started after the portioning, with the main controller deciding which cloud partition should receive the job and partition load balancer formulates the job assigning to the nodes. If the cloud partition is normal then task complete locally and if the cloud partition load status is not normal, this job should be transferred to another. When the environment is huge and compound these divisions streamline increase the proficiency in the public cloud environment. This load balancing refining the presentation and maintain stability. The challenge of this algorithm is to predict the job arrival, capabilities of each node in the cloud[20,21].

4.9 Stochastic Hill Climbing Algorithm

A variant of Hill Climbing algorithm Stochastic Hill Climbing (SHC) and it gives the solution for optimization problem. This procedure classified into two methods called complete and incomplete. Complete method which guarantees a correct answer either by proving that no such assignment exists or by finding a possibly valid assignment to the variable. On the other hand, incomplete method doesn't guarantee correct answers for all the given inputs. A Hill Climbing algorithm- Stochastic Hill Climbing algorithm is based on the incomplete method for solving optimization problems. SHC is a local optimization algorithm that continuously moves in the upward direction for increasing the value. If no neighbour has a higher value then it will automatically stop. This basic idea of this operation is repeat the solution until found or stopping no neighbour has a high value. So it has two main components a candidate generator that maps one solution candidate to a set of possible successors, and evaluation criteria which ranks each valid solution such that improving the evaluation leads to better solutions[21,5].

4.10 A2LB Algorithm

The dynamic A2LB algorithm addresses the resource utilization, maximum throughput and minimum response time issues. The overloaded virtual machine was distributed by cloud service provider to available resource for utilize in a right manner. This approach helps to balance the workload of all virtual machines. A2LB consists of three agents called load, channel and migration agent. Load and channel agents are static agents whereas migration agent is an ant, which is a special category of mobile agents. The reason behind deploying ants is their ability to choose shortest/best path to their destination. The ant start searching a food randomly, thus they may follow different paths to the same source, however with passage of time, density of pheromone on the shortest path increase and thus all follower ants start following that path resulting in increase of pheromone density even further. The ant moves from source to destination for collecting necessary information or carryout a task. It is not necessary come back to their source rather they destroy themselves at the destination only thereby reducing unnecessary traffic on the network. This algorithm would require searching for under loaded servers and resources, ant agents suit the purpose and fulfil it appropriately without putting additional burden on network[4,1].

4.11 Firefly Algorithm

The dynamic firefly scheduling algorithm is relies on flashing characteristics of fireflies and its application in optimizing the schedule process to the cloud network. This approach concerned about workload balance, thus they used following three rules in the cloud system.

- All fireflies are unisex so that one firefly can attracted to other fireflies of their sex[6].
- Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases[6].
- The brightness of a firefly is affected or determined by the landscape of the objective function.
- The brightness proportional to the value of the objective functions in maximization problem.

The load balancing operation going to be initiated effectively based on load index value which was calculated from memory usage, processing power and access rate. The CPU rate (represented as C), Memory rate (represented as M) and Processing time (represented as P) are considered as the loads to the nodes. The load balancing parameter value calculated from above rates, thus the node will be elected with minimum load weightage. Thus, according to the definitions of firefly algorithm, the equation to the attraction is defined:

$$\text{attr}(n_i) = p_i / (\text{cpu}_i + \text{mem}_i)$$

Here, $\text{attr}(n_i)$ represents the attraction between the node and the request as the node will be considered for the request if the attraction is high. p_i represents the processing time for the particular node, cpui represents the cpu rate of the node and memi represents the memory rate of the nodes[2]. The load index is derived from the above formulae:

$$LI = \sum_{i=1}^{n} pi/(cpu_i + mem_i)$$

The LI is the load index The SI is the scheduling index and it is the total sum of the nodes in a particular scheduling queue. According to node load index value the current load will be shared to available virtual machines[22,23]. Table 1 provides a comparison of different load balancing algorithms.

5. Conclusion

Cloud computing offered a service over a network. The important issue of cloud computing is load balancing. The overloaded system provides very poor performance and it does not offer service effectively. Therefore the scenario needed economical load balancing algorithm to produce a service while not breaking. This paper reviewed on a methodological analysis of various load balancing algorithms for services in a cloud network by concentrating on balancing the loads. The various load balancing algorithms are also being compared here on the

Table 1. Comparison of different load balancing algorithms

Algorithm	Environment	Parameters	Challenge(s)	Advantages
Min - Min	Static	Response time	Starvation	Out performs when small tasks are greater in number.
Max - Min	Static	Waiting time	Starvation	Tasks with MCT is executed first
Round Robin	Static	Waiting time	Less resource utilization.	Reduced the response time
Genetic Algorithm	Dynamic	Process utilization	Assumes that the jobs are of same priority	Find best fit solutions
Honey Bee Behaviour	Dynamic	Throughput	Take some more time to allocate high priority tasks finds the VM which has less number of high priority tasks	Reduced waiting time
Ant Colony	Dynamic	Throughput	Probability distribution changes by iteration and collective interaction of population of agents.	Greedy heuristic helps find acceptable solution in early stages
Opportunistic Load Balancing	static	Waiting time	Task processed in slow in manner because it does not calculate the current execution time of the node.	unexecuted tasks allocated very quickly without checking current workload
Game Theory	Dynamic	Reliability	predict the job arrival, capabilities of each node	Suitable for public cloud environment
Stochastic Hill Climbing	Dynamic	Response time	Still improvement needed	Better than RR and FCFS algorithms
A2LB	Dynamin	Scalability	Need to know the information of the candidate from other DC to get efficient result.	Proactive calculation of the VM in data centers
Firefly	Dynamic	Resource utilization	Exploration problem.	good balanced load among all the resources in cloud servers

basis of different types of parameter. The authors identified a swarm based algorithm to satisfy the user requirement for continuing the service by distributing the workload in balance manner to acquire maximum resource utilization and reduce system idle time.

6. References

1. Hurwitz J, Bloor R, Kaufman M, Halper F. Cloud computing for dummies. John Wiley and Sons; 2010 Jan 19.
2. Rajkumar B, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms. Future Generation Computer Systems. Elsevier Press, Inc.; 2009.
3. Hardware, software load balancing in cloud environment, Macromedia Press; 2003. Available from: http://www.adobepress.com/articles/article.asp?p=31089&seqNum=5. 28/02/2003
4. Singh A, Juneja D, Malhotra M. Autonomous agent based load balancing algorithm in cloud computing. Procedia Computer Science. 2015 Dec 31; 45:832-41.
5. Liu G, Li J, Xu J. An improved min-min algorithm in cloud computing. Proceedings of the 2012 International Conference of Modern Computer Science and Applications; Berlin Heidelberg: Springer. 2013. p. 47-52.
6. Kokilavani T, Amalarethinam DD. Load balanced min-min algorithm for static meta-task scheduling in grid computing. International Journal of Computer Applications. 2011 Apr; 20(2):43-9.
7. Bhoi U, Ramanuj PN. Enhanced max-min task scheduling algorithm in cloud computing. International Journal of Application or Innovation in Engineering and Management. 2013 Apr; 2(4):259-64.
8. Balaji N, Umamakeshwari A. Load balancing in virtualized environment - A survey. Indian Journal of Science and Technology. 2015 May 1; 8(S9):230-4.
9. Samal P, Mishra P. Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing. IJCSIT. 2013; 4(3):416-9.
10. James J, Verma B. Efficient VM load balancing algorithm for a cloud computing environment. International Journal on Computer Science and Engineering. 2012 Sep 1; 4(9):1658-63.
11. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A Genetic Algorithm (GA) based load balancing strategy for cloud computing. Procedia Technology. 2013 Dec 31; 10:340-7.
12. LD DB, Krishna PV. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing. 2013 May 31; 13(5):2292-303.
13. Anju Baby J. A survey on honey bee inspired load balancing of tasks in cloud computing. International Journal of Engineering Research and Technology. 2013 Dec 18; 2(12):1442-5.
14. Nakrani S, Tovey C. On honey bees and dynamic server allocation in internet hosting centers. Adaptive Behavior. 2004 Dec 1; 12(3-4):223-40.
15. Chakaravarthy T, Kalyani K. A brief survey of honey bee mating optimization algorithm to efficient data clustering. Indian Journal of Science and Technology. 2015 Sep 16; 8(24):1-7.
16. Mishra R, Jaiswal A. Ant colony optimization: A solution of load balancing in cloud. International Journal of Web and Semantic Technology. 2012 Apr 1; 3(2):33-50.
17. Li K, Xu G, Zhao G, Dong Y, Wang D. Cloud task scheduling based on load balancing ant colony optimization. 6th Annual IEEE Chinagrid Conference (China Grid); 2011 Aug 22. p. 3-9.
18. Sakthipriya N, Kalaipriyan T. Variants of ant colony optimization-a state of an art. Indian Journal of Science and Technology. 2015 Nov 14; 8(31):1-15.
19. Hung CL, Wang HH, Hu YC. Efficient load balancing algorithm for cloud computing network. International Conference on Information Science and Technology (IST 2012); 2012 Apr 28. p. 28-30.
20. Xu G, Pang J, Fu X. A load balancing model based on cloud partitioning for the public cloud. Tsinghua Science and Technology. 2013 Feb; 18(1):34-9.
21. Mondal B, Dasgupta K, Dutta P. Load balancing in cloud computing using stochastic hill climbing - A soft computing approach. Procedia Technology. 2012 Dec 31; 4:783-9.
22. Florence AP, Shanthi V. A load balancing model using firefly algorithm in cloud computing. Journal of Computer Science. 2014 Jul 1; 10(7):1156-65.
23. Vardhini KK, Sitamahalakshmi T. A Review on nature-based Swarm intelligence optimization techniques and its current research directions. Indian Journal of Science and Technology. 2016 Mar 16; 9(10):1-13.