RESEARCH ARTICLE OPEN ACCESS

Load Balancing Algorithms in Cloud Computing

Vinayak Shinde [1], Anas Dange**, Muhib A. Lambay [3]
HOD [1], Computer Engineering, Shree L. R. Tiwari College of Engineering, Mira Road
Lecturer [2], Computer Engineering, A. R. Kalsekar Polytechnic, Panvel
Assistant Professor [3], Computer Engineering, Theem College of Engineering, Boisar
India

ABSTRACT
Load balancing is an advanced and relatively modern technique that facilitates improvement in networks by providing an ultimate throughput with minimal response time in order to captivate the user's attention. There are large numbers of algorithm available that help in dividing the traffic load between required servers for the data being sent and received. A basic example to understand load balancing in our day-to-day life can be analogous to websites. Without this concept of load balancing, users could sense delays and lengthy system responses. Here our objective is to inspect various static and dynamic algorithms that are proposed to solve the issue of load balancing in Cloud Computing. This paper discusses and analyzes these algorithms to provide an outline of the modern approaches aimed at providing load balancing to enhance the overall performance of the Cloud in a fair manner. The paper also describes about an algorithm named honey bee behavior based load balancing (HBB-LB), whose target is to achieve equitable load across virtual machines. There is lot of enhancement in average execution time and devaluation in waiting time of tasks for efficiency of users.

Keywords :— load balancing, virtual machines

I. INTRODUCTION

Cloud Computing has gained a greater popular since the last decade and today also many of the research and information technology publications, journals, conferences, magazines and other websites are discussing about cloud computing in one way or the other. This happened due to its unique way of providing malleable and smooth method to store and access data and files, especially for making large data sets thereby involving the concept of virtualization, distributed computing, and web services. The benefit of virtualizing these applications are illustrious: minimizing the cost of hardware, reducing the need to buy software license, decreasing the implementation cost as client pays only for what is needed and globalizing the workforce which improves the accessibility there by increasing the efficiency. Today, there are more than hundred billions of computer and storage devices associated with the Internet and thousands of users access the information from these devices on a cloud at any given time. This support has driven the force to initiate many new cloud providers, ranging from private clouds, community models to the eminent publicly accessible clouds such as Google, Amazon, etc.

The word "Cloud Computing" is basically derived by combining two significant terms in the field of innovation and technology, that is Cloud and the other term is computing. The word cloud is used as a symbol for "the Internet," That is huge mesh of heterogeneous infrastructure and resources. Infrastructure refers to both the hardware such as servers, storage and system software like applications in data centres that are delivered to end users and organizations as services through the Internet on the basis of in pay-as-you-use-manner. "Computing" is based on certain parameters as specified in Service Level Agreement. Google's Eric Schmidt told attendees at a Search Engine Strategies conference in 2006, that the data services and architecture should be on servers, that is in a "cloud" somewhere.

Fig. 1 Cloud Computing Usage

And that if you have the right kind of browser or the right kind of access, it doesn't matter whether you have a PC or a Mac or a mobile phone or a BlackBerry or what have you – or new devices still to be developed – you can get access to the cloud [1]. However, The National Institute of Standards and Technology; or NIST, an agency of the U.S. Department of Commerce which was founded in 1901; and whose mission is to promote the nation's innovation and technology in science, has defined cloud computing as: "Cloud computing is 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" [2]. Thus the aim of Computation in cloud to achieve maximum resource utilization with higher availability at minimized cost will be achievable only if we consider not just of what computing service is delivered but also how it is being delivered. These features rely heavily on a very complex problem of maintaining the cohesive act of dealing with so many jobs in the cloud computing environment in a fraction of second thereby researchers receiving much more attention on load balancing.

The rest of the paper is organized as: In section II, we introduce the actual process of load balancing in cloud computing. In section V, we go through the working of various static and dynamic algorithms. In Section VII, the performance evaluation of different cloud computing algorithms have been compared and reviewed with the help of multiple parameters. Finally the paper is concluded in section VIII.

II. LOAD BALANCING IN CLOUD COMPUTING

The real time challenge of cloud computing is load balancing. The basic reason for this requirement is the rapid increase in the number of users and their demand for cloud services. So it became impractical to control or manages one or more free service to accomplish the demand of the users. Thus, Load balancing in cloud computing provides a competent, powerful and economical solution to large number of issues lying in the environment set-up of the cloud computing. The two important functions that must be taken into account for load balancing are resource allocation and task scheduling. These will guarantee that Resources are used efficiently and economically even under the stress of high load, thereby reducing the cost of using resources. There are several characteristics of load balancing such as: equal division of work across all the nodes, facilitation in achieving user satisfaction, improve overall performance of system, reduce response time, and provide services to achieve complete resource utilization [3].

A typical basic cloud computing environment can be modelled in order to measure the efficiency of Load Balancing algorithms using following four entities:

- Datacentres,
- Hosts,
- Virtual Machines and
- Application as well as System Software

A data centre is a centralized storehouse which may exist physically or virtually used for the storage, management, and propagation of data that is responsible for providing base level Services to the Users of the Cloud computing. They act as a home to various Host Entities. Hosts are Physical Servers that are configured in advance as per their processing competence, which is expressed in

MIPS (million instructions per second) and is responsible for providing Software level service. They act as a home to one or more Virtual Machines.

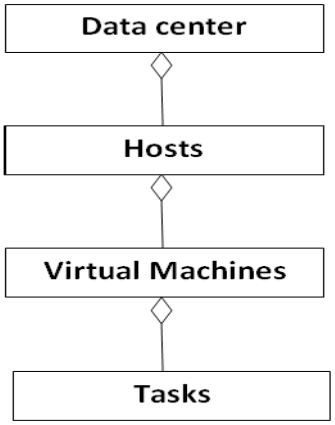

Fig. 2 Entities in Cloud Computing Environment

A virtual machine is an application environment installed on software which resembles dedicated hardware so that the end user experience same feelings as if on real hardware. For example, in a virtualized cloud, the developers will allocate new virtual machines based on demand of the users without interaction of the underlying physical equipment and users. They are mapped to a same instance of host that are balanced with their biting characteristics like storage, processing, memory, and software requirements based on its availability. Tasks are executed on Virtual Machine on-demand.

The two different types of algorithms in load balancing are static and dynamic algorithms.

A. Static Algorithms:

Static algorithms segregate the traffic equally among the virtualized servers. In this approach, the division of the traffic will be easier and consequently it will lead to imperfect circumstances. The decision to balance the load is made at compile time and hence the completion time of a task forms the basis of static algorithms. These algorithms are used in the environment where there are few load variations. It does not consider the present condition and information about the system while distributing the load thereby making things simpler. But, they are not capable to handle the load changes during run-time.

B. Dynamic Algorithms:

Dynamic algorithm constantly checks the different properties of the nodes such as its capability, network

bandwidth, processing power, memory and storage capacity and other parameters thereby assigning suitable weights to the servers. An underweight server is searched in a whole network to balance the traffic and assigned a load by this algorithm. But selecting the appropriate server needs valid and authentic communication within networks that lead to overhead in traffic inside the system. These algorithms are also known as self-adaptive algorithms.

III. PERFORMANCE MEASUREMENTS OF LOAD BALANCING IN CC

Following are the parameters used to assess various load balancing techniques to get improved distribution of resources as per demands of the cloud users.

A. Throughput: Throughput is a measure of how many units of information a system can process in a given amount of time [4].

B. Response time: The elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response; for example, the length of the time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal [4].

C. Fault tolerance: The ability of the algorithm that allows system to keep operating properly in the event of failure of one or more faults within the system.

D. Scalability: It is the ability of the algorithm to cope and perform with the growing amount of work as per the required conditions.

E. Overhead: Overhead is combo of excess computational time, memory, network, or other resources that are needed to complete a particular task.

IV. LOAD BALANCING ALGORITHMS

1. Round Robin Algorithm:

It is one of the simplest and most widely used scheduling techniques which use the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or interval. Initially, loads are equally distributed to all VMs. As the name suggests, round robin works in a circular pattern. Each node is fixed with a time slice and performs the assigned task on its turn. It is easy to implement and understand and hence less complex. As a result, at any moment some node may possess heavy load and others may have no request. However, this problem was overcome by weighted round robin algorithm.

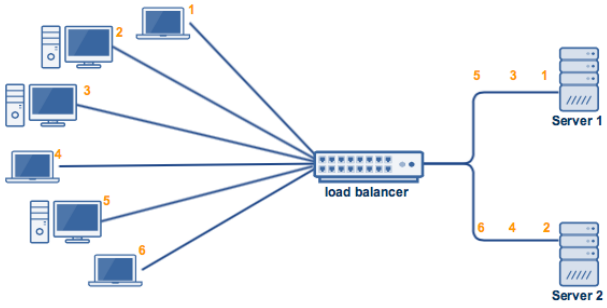

Fig. 3 Example of RR algorithm

2. Weighted Round Robin Algorithm:

The Weighted Round Robin is similar to the Round Robin in a sense that the method by which requests are assigned to the nodes is still circular, but with a small twist. The node with the higher capability and specification will be able to handle a greater load or larger number of requests. The weights are assigned to each node during the set-up of the load balancer. Consider the situation in which the capacity of Server 1 is 5 times more than the other server, then the weights of Server 1 and 2 are 5 and 1 respectively.

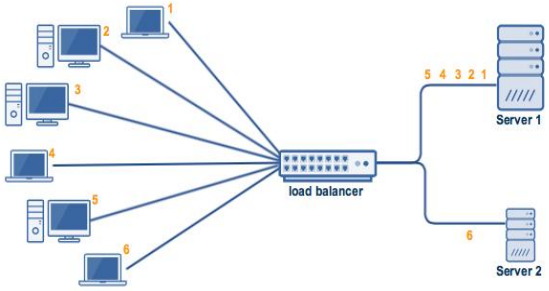

Fig. 4 Example of weighted RR algorithm

3. Max-Min Algorithm:

Max-Min begins with the set of all the submitted tasks in the task-set that are unassigned. This algorithm works in two stages. First, the expected execution time and completion time for all the tasks is calculated on all the machines. In the second phase, the task with the maximum expected completion time is selected and assigned to the resource with minimum execution time. This task is then removed from the task-set and process is repeated until the task-set is empty. The step wise algorithm is discussed below:

1. For all submitted tasks in task-set; Ti and for all resources; Rj i. Calculate: Cij = Eij + rj
where,
rj represents the ready time of resource Rj to execute a task, Cij represents expected completion time of task,

Eij represents expected execution time of task.

2. While task-set is not empty
i. Find task T consumes maximum completion time.
ii. Assign T to the resource R which gives minimum execution time
iii. Remove T from task-set

Thus, larger tasks will be executed first, while the smaller tasks have to wait for long time, which will finally lead to starvation. The algorithm works better in situations where small tasks are greater in number than the large tasks.

4. Min-Min Algorithm:

Min-Min is a simple and fast algorithm capable of providing improved performance. This algorithm also works in two phases. First, the expected execution time and completion time for all the tasks is calculated on all the machines. In the second stage, the task with the minimum expected completion time is selected and assigned to the resource with minimum execution time. This task is then removed from the task-set and process is repeated until the task-set is empty. The descriptive flow of the algorithm is presented in the below figure.

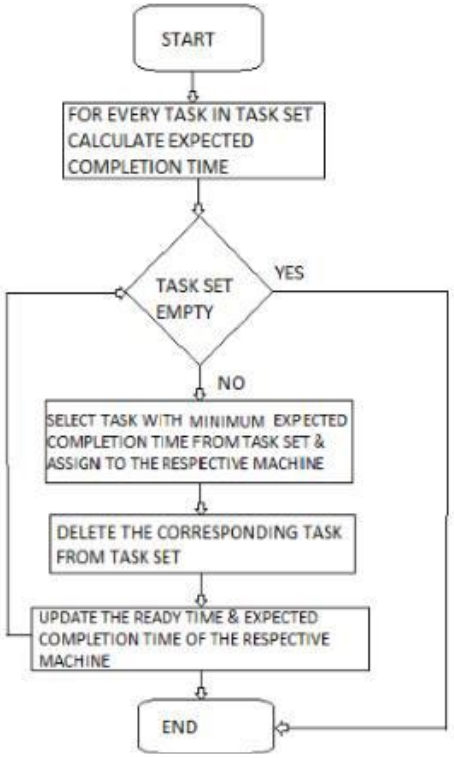

Fig. 5 Flowchart for Min-Min algorithm

5. Throttled Algorithm:

This dynamic algorithm is based on the entity called Throttled Load Balancer (TLB) which monitors the loads on all the Virtual Machines. Each VM is entrusted with only one job at a time and can be assigned another task when the present task has been completed successfully. In this algorithm, first the client requests for a suitable Virtual machine from the load balancer in order to perform the required operation [5]. Further procedure for the algorithm is discussed in the following steps as shown in the figure.

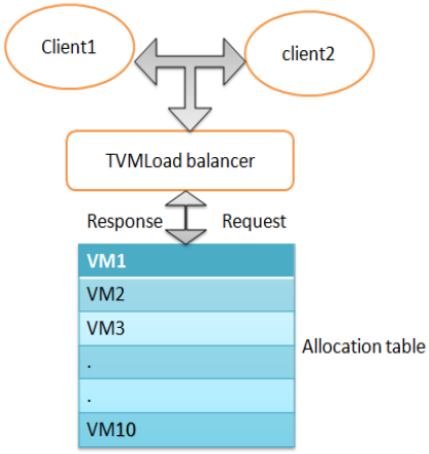

Fig. 6 Working of Throttled LB algorithm

1. TLB maintains an index table of all VMs and the state of the VM (BUSY/AVAILABLE). All the machines are available in the beginning.

2. The Datacenter Controller (DC) receives a new request from the client for the allocation of VM.

3. DC queries the TLB for the proper VM allocation.

4. TLB scans the allocation table from top to bottom until the available VM is found.
If found:-

i. TLB returns the id of VM to the DC.
ii. DC acknowledges TLB about the new allocation of VM.
iii. DC then communicates the request to the VM identified by that id.
iv. TLB updates the allocation table by increasing the allocation for that VM by 1.

If not found:-
i. TLB returns null value to DC.
ii. The DC queues the request until the next availability of VM.

5. When the VM finishes processing the request, it sends results to the DC and acknowledges TLB of the VM de-allocation.

6. TLB updates the allocation table by decreasing the allocation for the VM by 1.
7. DC then checks for the requests in the waiting queue. If the queue is not empty then it goes to step 3.

6. Ant Colony Optimization

Dorigo Marco introduced the ant algorithm based on the behavior of real ants in 1996 [6], which is a new heuristic algorithm based on the ability of ants to find an optimal path from nest to food source. ACO algorithm is one of the most outstanding strands of swarm intelligence. Ants lay some pheromone on the ground through their gesture while moving which is secreted by pheromone gland; while an isolated ant encounter a previously laid trail, this ant can detect it and decide with high probability to follow it. The more ants select a particular way, that way has denser pheromone, and the heavier pheromone attracts more ants. Thus ant can find an optimal way through this positive feedback mechanism.

Ants originate from the root or head node and traverse the whole network in such a way that they know each location of under loaded node and overloaded node. When these ants traverse the network they update the pheromone table which stores the information about each node's utilization. Ant move in two ways:

1) Forward movement-The ants continuously move in the forward direction in the cloud encountering overloaded node or under loaded node.
2) Backward movement-If an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also possible.

V. HONEY BEE LOAD BALANCING ALGORITHM

The artificial bee colony contains following three groups of bees [7]:
• Scout bees:-The bee carrying out random search is known as scout.
• Forager bees: - The bee which is going to the food source which is visited by scout bees previously is forager bee.
• Onlooker bees: - The bee waiting on the dancing area is an onlooker bee.

Scout bees are sent for search of suitable food sources; when found, they return to the hive to advertise this using a display dance known as a "waggle dance". The suitability of the food source and its distance from the hive is communicated through the waggle dance display. Forager bees then follow the scout bees back to the discovered food source and begin to harvest it. Upon the bees return to the hive again, the remaining quantity of food available is reflected in their waggle dances, allowing more bees to be sent to a source.

TABLE I
MAPPING OF HONEY BEE BEHAVIOUR TO A CLOUD ENVIRONMENT

Honey bee Hive	Cloud Environment
Honey bee	Task also called as cloudlet
Source of the food	Virtual Machine
Honey bee foraging a food source	Task is being loaded to a VM
Honey bee reduction at food source	VM is overloaded
Finding new food source	Removed task scheduling to under loaded VM

First of all we calculate the capacity and load of all VMs and then group them into three categories as under:-

1. Overloaded Virtual Machine (OVM),
2. Underloaded Virtual Machine (UVM) and
3. Underloaded Virtual Machine (BVM)

If there is more than one under loaded VM in UVM list, select a VM in such a way that it has minimum objective function value so that the currently selected task will get executed faster.

Step1: Start

Step 2: Find capacity of all VMs.
$Ci = P_{ni} \times P_{mpi} + VM_{bwi}$

Where P_{ni} : No. of processors in VMi,
P_{mpi} : no. of instructions of all processors in VMi in millions per second and
VM_{bwi} : communication bandwidth ability of VMi

Capacity of all VMs,

C = C1+C2+.....Cn

Step 3:
Find load of all VMs. Total length of tasks that are assigned to a VM is called load,

$$Lvm_{i,t} = N(T,t)/S(vm_{i,t})$$

Load of a VM at time t can be calculated as the Number of tasks at time t on service queue of VM_i divided by the service rate of VM_i at time t. Load of all VMs,

$L = Lvm_1+Lvm_2+.....Lvm_n$

Processing time of a VM m is: $PT_i = Lvm_i/C_i$

Processing time of all VMs: PT = L/C

Standard deviation of load:

$$\sigma = \sqrt{1/m \sum_{i=1}^{m} (PTi - PT)^2}$$

Step 4:
Load balancing is possible only when the capacity of the datacenter is greater than the current load. If the standard deviation of the VM load is less than or equal to the threshold condition then the system is balanced. Otherwise system is in an imbalance state. If balanced then exit.

Step 5:
VMs are grouped based on their loads, i.e., Overloaded VMs (OVM), under loaded VMs (UVM) and balanced VMs (BVM). The VMs whose SD values greater than threshold is considered as the overloaded VMs and the VMs whose standard deviation values less than threshold is considered as under loaded VMs.

Step 6:
Perform Load balancing

- Find supply of each VM in UVM
- Find demand of each VM in OVM
- Sort VMs in OVM and UVM
- For each task in each overloaded VM find a suitable under loaded VM.
- If there are more than one VM in UVM list

  VM_d=Call Pareto optimal scheduling

Pareto dominance means a vector J1 = (j1, j2…jn) is said to dominate J2 = (k1, k2 ….kn) if and only if J1 is partially less than or equal to J2. That is J1i <= J2i for all i ϵ {1, 2, 3…k}.

Step 6.A: Find cost of all VMs using Fixed Price Allocation Scheme.

Step 6.B: For each task ti, For each VM mj, Calculate cost of executing ti on mj using:

$C(ti,mj) = \sigma * PT(ti,mj) * Vco * (C_{mj} / C_{lcap})$

C_{lcap} denotes the capacity for the VM having less capacity and Vco is the cost of that VM which is calculated through fixed price allocation mechanism.

σ is the random variable.

Most of the cloud providers assign virtual machine instances to their users by using technique of fixed-price allocation schemes. It means that users have to pay flat prices per unit of time for using those assets.

Calculate the minimization function F

$F = w1*T(i,j) + w2*C(i,j)$

Where, w1+w2=1
and $T(i,j) = (PT(ti,mj)-t_{min})/(t_{max}-t_{min})$
and $C(i,j) = (C(ti,mj)-C_{min})/(C_{max}-C_{min})$

t_{max} and t_{min} are the minimum and maximum execution time respectively and C_{max} and C_{min} are maximum and minimum cost of any task.

Select mj having minimum F

Step 6.C: Return mj.

Step 7: Update the overloaded and under loaded VM sets and go to step 3.

Step 8: Stop.

VI. CONCLUSION

In this paper, we have described and compared different static and dynamic load balancing algorithms for cloud computing such as, round robin (RR), Min-Min, Max-Min, Throttled, Ant colony, etc. considering the characteristics like throughput, fault tolerance, overhead, speed, and complexity. The key part of the paper is the proposed algorithm which follows the foraging behavior of honey bees for loading tasks to the VMs and uses preemptive task scheduling. Also, Pareto dominance concept is not only used for selecting optimal VM but also for setting the priorities of the tasks. The remaining expected completion time of the tasks is considered while preempting tasks so that it improves the performance of the virtual machines. So this algorithm can be strengthened in future by considering other QoS factors for tasks like VM processing speed, network bandwidth and VM cost during load balancing in cloud computing.

REFERENCES

[1] (2016) The ITWORLD website. [Online]. Available: http://www.itworld.com/article/2726701/cloud-computing/where-did--cloud--come-from-.html
[2] Shinde, Vinayak D., Anas Dange, and Muhib Anwar Lambay. "Study of Threats and Security in Cloud Computing Technology", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 10, October 2015
[3] Aslam, S., & Shah, M. A. (2015, December). Load balancing algorithms in cloud computing: A survey of modern techniques. In 2015 National Software Engineering Conference (NSEC) (pp. 30-35). IEEE.
[4] (2016) The TECHTARGET website. [Online]. Available: http://searchnetworking.techtarget.com/definition/
[5] Dr.S. Suguna and R. Barani, Simulation of Dynamic Load Balancing Algorithms, Bonfring International

Journal of Software Engineering and Soft Computing, Vol. 5, No.1, July 2015

[6] Shagufta K., Niresh S, "Ant Colony Optimization for Effective Load
Balancing In Cloud Computing", Volume 2, Issue 6, November – December 2013, pp. 78

[7] Sheeja Y S, Jayalekshmi S Cost Effective Load Balancing Based on honey bee Behaviour in Cloud Environment. IEEE 2014 First International Conference on Computational Systems and Communications (ICCSC), 17-18 December 2014