International Journal of Pure and Applied Mathematics
Volume 114 No. 11 2017, 127-136
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

An Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing

[1]V.V. Bhavya, [2]K.P. Rejina and [3]A.S. Mahesh

[1]Department of Computer Science and IT,
School of Arts and Sciences, Amrita University, Kochi.

[2]Department of Computer Science and IT,
School of Arts and Sciences, Amrita University, Kochi.

[3]Department of Computer Science and IT,
School of Arts and Sciences, Amrita University, Kochi.

Abstract

Cloud computing strives to provide computational resources according to user's demand through internet on pay per use. The required resources may be software, physical or hardware infrastructure. The chief focus of cloud computing is load balancing-the process of assigning load over multiple virtual servers to achieve user satisfaction as well as optimal resource utilization. Several static and dynamic algorithms have been formulated that efficiently manage the load balancing in a cloud environment. This paper throws light on the performance of the honeybee foraging algorithm based on throughput calculated for each virtual machine. We calculate and store the throughput of each virtual machine while creating each VM in the cloud environment and then execute the modified Honey bee foraging algorithm; the goal is to find the appropriate VM with the help of the throughput calculated earlier and to allocate the load–this leads to more efficient load balancing.

Key Words: Cloud computing, honey bee foraging load balancing algorithms, load balancing and throughput.

1. Introduction

Cloud computing enhances distributed computing, adding provision to run applications across multiple systems having simultaneous network connections. Cloud computing is a utility oriented approach where services are made available to the user on pay-per-use basis.

Service categories in cloud computing are divided into three types: Infrastructure as a Service (IaaS), Platform as a service (PaaS), and Software as a services (SaaS). These are also known as cloud computing stack. IaaS provides computing resources such as virtual server, API's, storage, networking on a pay-per-use basis. PaaS provides cloud based environment for developing, delivering and managing the software applications. SaaS helps in delivering software applications (web services) via internet. Three types of deployment models are: public, private, hybrid. Public clouds are held, maintained and made to work by third party service providers; They make available servers and storage via internet. Private clouds are used by single organization or business or by a third party. Hybrid clouds is a union of public and private clouds which provide flexibility and deployment options for businesses.

Load balancing is a challenge in cloud computing where total loads are distributed among various virtual machine to acquire user satisfaction and resource utilization there by diminishing resource draining. Load balancing algorithms ensure proper traffic among servers and can be broadly categorized as Static and Dynamic load balancing algorithms. Static load balancing algorithm need advance knowledge of system resources rather than current state where as dynamic load balancing algorithm use present state of the system resources. Dynamic load balancing algorithm is divided into distributed and non-distributed approach.

2. Different Load Balancing Techniques

A. Static Load Balancing Algorithm

Round Robin Load Balancing

This algorithm does not verify the previous load state of a node at the time of assigning the job. It selects the first node randomly and allot to all others nodes in round robin manner. It might not be suitable for cloud computing because some nodes may have the chances to get heavily loaded and some may not. We can overcome this by using weighted round robin algorithm.

Min-Max

Data associated with this jobs are available in advance. In this algorithm minimum completion time (MCT) for all jobs is computed, and minimum is selected. The node which has the MCT for all jobs are selected and then map chosen node. The ready time of the node is updated. This process is reworked for quite a times**.** The job with the smallest execution time is executed. One of the drawbacks is that some jobs may experience starvation.

Max-Min

This algorithm is same as that of min max algorithm with exception that maximum value is selected.

B. Dynamic Load Balancing Algorithm

Equally Spread Current Execution

Here it handles the process with priority, which is determined by analyzing the process size and then the load is transferred to a lightly loaded VM. The load is spread on different nodes by the load balancer.

Throttled Load Balancing Algorithm

This algorithm, to perform required operation to the client first appeals to the load balancer to find a suitable VM. From the multiple instances of VM these VM can be grouped on the basis of requests they handle. Immediately after getting request from client, the group that can handle the client request is checked by the load balancer and the request is allocated to lightly loaded instance of that group.

Biased Random Sampling

This is an algorithm that represents a load on the server using a virtual graph. The graph is connected by connecting the nodes that form the vertices in the graph. The in-degree of a vertex/node represents the resources that are free at that node. A task is assigned only to a node having at least one free resource available. Once a node is given a task, the in-degree value is decremented by one and once the task gets completed, the in-degree value gets incremented by one.

Active Clustering

The basic principle of Active clustering is working on nodes that are grouped based on their similarity. Here, first node pick-up a neighbor node called match-maker node and it makes connection with neighbor name as that of initial node. Conclusively, the match-maker gets detached. Here throughput is increased (efficient utilization of process).

3. Related Works

Load balancing performs the operation of balancing task across the under loaded VM and over loaded VM. Nitin et al.[1] In their work ," Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing" , have done a comparative analysis between dynamic and static load balancing scheme based on different criteria including performance, scalability, throughput, resource utilization, fault tolerance, response time etc. Harshit et al. [2] proposed a flow chart based on the nature exhibited by a foraging honey bee. Studies have proven that load balancing techniques inspired from honey bee nature upgrade throughput of processing.Responce time of a VM is decreased by implementing priority based balancing approach. They have also presented a comparison of their algorithm with other existing ones and have succeeded in proposing an algorithm that perform well without any considerable overhead. Atul et al.[3], in their paper "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", have reviewed different load balancing strategies, discussing their merits and demerits. While static load balancing offers simple and uncomplicated simulation and supervision of the environment as opposed to dynamic load balancing, it fails to offer the best solution when it comes to heterogeneous cloud environments. Dynamic load balancing on the other hand is better suited in heterogeneous cloud environments. If the algorithm implemented and distributed in nature it succeeds in achieving fault tolerance, but with a greater rate of replication. To achieve a more balanced state, hierarchical load balancing algorithms can be employed where the load gets distributed to different hierarchical levels and nodes at hierarchical level requests services from the nodes at lower level in a fair and balanced manner. Although implementation of dynamic load balancing techniques in a distributed or hierarchical cloud environment provides remarkable performance, modeling the dependencies between tasks using workflows can further boost the performance of the cloud. Tushar et al.[4], discuss various load balancing techniques for cloud computing had surveyed. They had also discussed cloud virtualization and required qualitative matrix for load balancing. By referring to this paper we were able to identify the existing challenges in load balancing. In the paper "Honey Bee Behavior Load Balancing of Tasks in Cloud”, Khushbu et al.have focused on achieving minimum task completion time and better resource utilisation.This they achieved by reallocating prioritized task from one machine to another having fewer number of prioritized task with minimum completion time. Besides considering the time required to complete a task, their study also focused on task’s priority at the time of submission of execution, and also adjusting the load of dependent tasks in pre-emptive manner. Priyadarashini et al.[6] compared three dynamic load balancing algorithm: Particle Swarm Optimization, FCFS Algorithm and Modified Throttled Load Balancing Algorithm. They found that the Particle Swarm Optimization has efficient response time and low average cost of datacenters.

4. Proposed Work

Honey bee foraging algorithm works similarly the way in which bees find and gather their food. There are two categories of bees namely, forager bees and scout bees. Food is searched by the Forager bees and they announce it to other bees when they get it. They announce it by doing waggle dance which shows the availability of metadata food. Then the scout bees follows the forager bees and collect honey from the source. After returning, they does a waggle dance to indicate the food left so that more honey can be consumed. In load balancing services are assigned dynamically as the users' demands changes. Here, the servers are made into a group of virtual server and each virtual server maintain a process queue. The profit is calculated after processing the request. High profit makes the server stay, and low profit levels trigger a return to foraging. Each node maintains a separate queue. Profit computation cause additional overhead, which results in deterioration of throughput.

At present, load is send to a randomly selected under-loaded VM after calculating threshold value. By randomly selecting under loaded VM we will not be able to determine the VM having the high performance. First we will be calculating the throughput and based on the highest throughput value we will be selecting that particular VM.

A. Algorithm

We will be calculating throughput at the time of creating the VM.

```
Step 1: Start
Step 2: Set number of tasks
Step 3: Set number of VM
Step 4: Calculate throughput.
Step 5: Initially, set the load on each VM to null
Step 6: Send first task to the VM having high throughput
Step 7: Check current VM load > threshold value
Step 8: If yes, select the VM having high throughput and check the load < threshold value, if yes, assign to that VM
If no, select another VM with next highest throughput and load <threshold value
Step 9: If no, send the task to the current VM
Step 10: If all task are send to the VM go to step 12
Step 11: Else go to step 6
Step 12: Stop
```

B. Flowchart

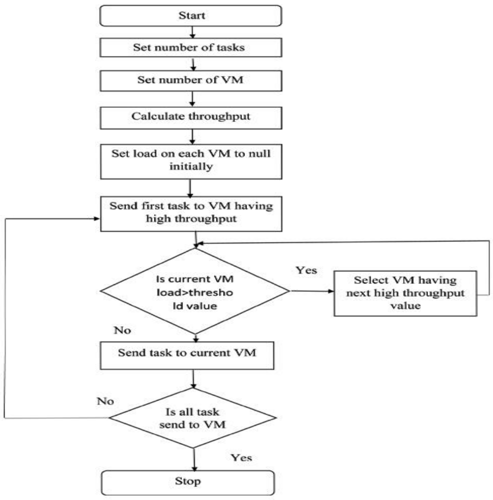

Start
Set number of tasks
Set number of VM
Calculate throughput
Set load on each VM to null initially
Send first task to VM having high throughput
Is current VM load>thresho ld value
Yes
Select VM having next high throughput value
No
Send task to current VM
No
Is all task send to VM
Yes
Stop

Fig 1: Flowchart of modified honey bee foraging algorithm

5. Cloud Analyst

Cloud analyst is a new simulation tool which is used for analyzing better load balancing algorithm. It runs on CloudSim tool kit by extending CloudSim functionality.

A. Components

Region

The entire world is partitioned into 6 regions in Cloud Analyst. The other two components namely, User Bases and Data Centers comes under this region.

Internet

It is an abstraction for the real world internet. It implements important simulation features[12].The transmission latency and existing bandwidth among 6 regions can be configured.

Service Broker

Service Broker manages the traffic between Data Centers and User Bases [13]. There are three categories of Service Brokers - closest data center, optimize response time and dynamically reconfigured.

User Bases

It is a group of users which is seen as an individual unit. Its main task is generating traffic for simulation.

Data Center Controller

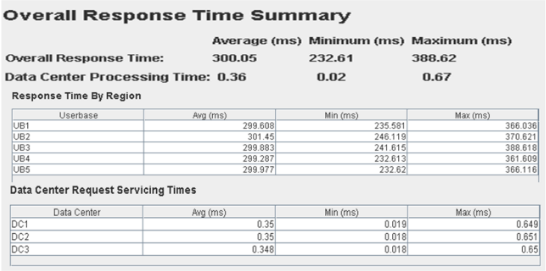
The activities performed by data centers are controlled by data center controller**.**

VM Load Balancer

It use VM load balancer to identify particular allocated for processing of upcoming cloudlet**.**

6. Experimental Result

Here figures showing the experimental result and comparison between the existing honey bee foraging algorithm and the proposed algorithm.



Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	300.05	232.61	388.62
Data Center Processing Time:	0.36	0.02	0.67

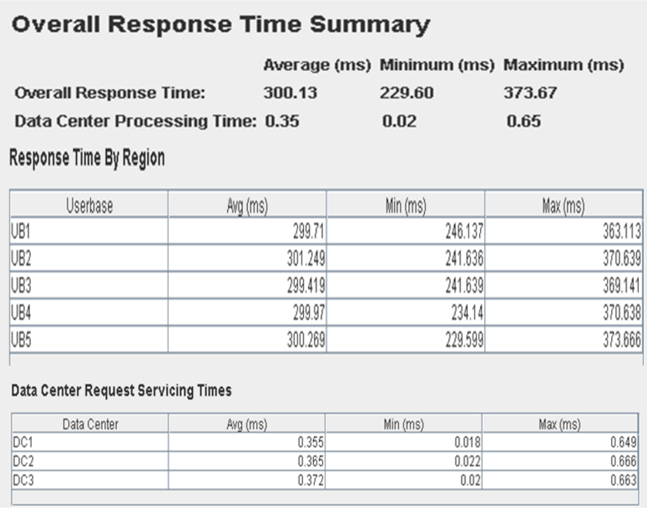
Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	299.608	235.581	366.036
UB2	301.45	246.119	370.621
UB3	299.883	241.615	388.618
UB4	299.287	232.613	361.609
UB5	299.977	232.62	366.116

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.35	0.019	0.649
DC2	0.35	0.018	0.651
DC3	0.348	0.018	0.65

Fig. 2: Overall response time summary, region response time and datacenter request servicing times taken by honey bee foraging algorithm for the completion of simulation



Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	300.13	229.60	373.67
Data Center Processing Time:	0.35	0.02	0.65

Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	299.71	246.137	363.113
UB2	301.249	241.636	370.639
UB3	299.419	241.639	369.141
UB4	299.97	234.14	370.638
UB5	300.269	229.599	373.666

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.355	0.018	0.649
DC2	0.365	0.022	0.666
DC3	0.372	0.02	0.663

Fig. 3: Overall response time summary, region response time and data center request servicing times taken by proposed algorithm for the completion of simulation

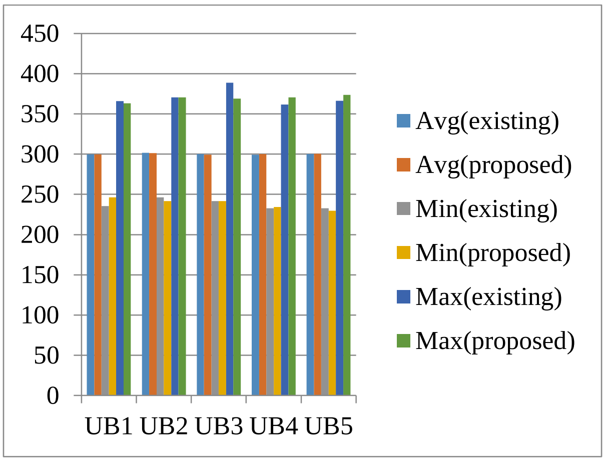


Fig. 4: Max, Min and Avg response time summary of Existing and Proposed Honey bee foraging Algorithm

7. Conclusion

In the existing honey bee foraging algorithm the load balancing is done with the help of the randomly selection of the VM. This paper proposes a modification on the honey bee foraging algorithm which selects the VM on the basis of the high throughput. Hence we can minimize the overall response time of the algorithm by shifting the load to the most efficient virtual machine.

Acknowledgement

We, Bhavya V V and Rejina K P would like to express our profound gratitude to our research guide Mahesh A S, Assistant Professor, School Of Arts And

Sciences, Amrita VishwaVidyapeetham , Kochi for helping us to do this research. We thank our college for giving us this opportunity to do a research work.

References

[1] Nitin Kumar Mishra, Nishchol Mishra, Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing-A Systematic Review, International Journal of Computer Applications 131(18) (2015).

[2] Harshit Gupta, Kalicharan Sahu, Honey Bee Behavior Based Load Balancing of Tasks in Cloud Computing, International Journal of Science and Research (IJSR) (2012).

[3] Mayanka Katyal, Atul Mishra, A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment, arXiv preprint arXiv:1403.6918 (2014).

[4] Tushar Desai, Jignesh Prajapati, A Survey of Various Load Balancing Techniques And Challenges In Cloud Computing, International Journal of Scientific & Technology Research 2(11) (2013).

[5] Khushbu Zalavadiya, Dinesh Vaghela, Honey Bee Behavior Load Balancing of Tasks in Cloud Computing, International Journal of Computer Applications 139(1) (2016).

[6] Priyadarashini Adyasha Pattanaik, Prasant Kumar Pattnaik, Sharmistha Roy, Performance Study of Some Dynamic Load Balancing Algorithms In Cloud Computing Environment, 2nd International Conference on Signal processing and integrated networks (SPIN) (2015), 619-624.

[7] Dhinesh Babu L.D., Venkata Krishna P., Honey Bee Behavior Inspired Load Balancing of Tasks in Cloud Computing Environments, Journal homepage:www.elsevier.com/locate/asoc

[8] Shanti Swaroop Moharana, Rajadeepan D. Ramesh, Digamber Powar, Analysis of Load Balancers In Cloud Computing, International Journal of Computer Science and Engineering 2(2) (2013), 101-108.

[9] Srushti Patel, Hiren Patel, Nimisha Patel, Dynamic Load Balancing Techniques for Improving Performance in Cloud Computing, International Journal of Computer Applications 138(3) (2016).

[10] Ruhi Gupta, Review on Existing Load Balancing Techniques of Cloud Computing, International Journal of Advanced Research in Computer Science and Software Engineering 4(2) (2014).

[11] Obaid Bin Hassan, A Sarfaraz Ahmad, Optimum Load Balancing of Cloudlets Using Honey Bee Behavior Load Balancing Algorithm, International Journal of Advance Research in Computer Science and Management Studies 3(3) (2015).

[12] Bhathiya Wickremasinghe, Cloud Analyst: A Cloud Sim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments, 433-659 Distributed Computing Project, CSSE Dept., University of Melbourne.

[13] Simar Preet Singh, Anju Sharma and Rajesh Kumar, Analysis of Load Balancing Algorithms using Cloud Analyst, International Journal of Grid and Distributed Computing 9(9) (2016), 11-24.

[14] Hetal V. Patel, Ritesh Patel, Cloud Analyst: An Insight of Service Broker Policy, International Journal of Advanced Research in Computer and Communication Engineering 4(1) (2015).

[15] Hemali S. Jinjuwadia, Shyam R. Kotecha, Improving Load Balancing using Dynamic Algorithms in Cloud Environment, International Journal of Advance Research in Engineering, Science & Technology 3(11) (2016).