I J C T A, 9(17) 2016, pp. 8593-8598

Artificial Bee Colony Based Load Balancing in Cloud Computing

Jay Ghiya*, Mayur Date* and N. Jeyanthi*

ABSTRACT

Planning of jobs in cloud computing is a NP-hard enhancement issue. Load adjusting of non-pre-emptive autonomous jobs on virtual machines (VMs) is an imperative part of job planning in clouds. At whatever point certain VMs are over-burdened and remaining VMs are lightly loaded with jobs for processing. The workload must be adjusted to accomplish ideal machine usage. In this paper, we propose an efficient artificial bee colony based load balancing which intends to accomplish well-adjusted workload across all virtual machines thereby improving response time and throughput.

Keywords: Job planning, load balancing, cloud computing, artificial bee colony, virtualization

I. INTRODUCTION

Cloud Computing has emerged as a big evolution in context of computing, in which software, infrastructures and platforms are provided as a service to end user [1]. It uses a network of remote servers rather than a local server or a personal computer, which are hosted to manage, process and store data on the internet. In this modern era, most of the companies all around the world are moving towards Cloud computing to cut the company's costs and to maximize the use of all the services provided by cloud computing.

Virtualization [5] is a vital technology in Cloud computing, which enhances the utilization of resources. Virtualization uses a single hardware to host or run multiple virtual machines which can host multiple OS. In a nutshell, virtualization is a software that separates physical infrastructure to create various dedicated resources. Essentially, virtualization differs from cloud computing because virtualization is a software that manipulates hardware, while cloud computing prefers to a service that results from the manipulation. So it can be said that Virtualization is a foundational element of cloud computing and helps deliver on the value of cloud computing."

Job planning is one of the vital elements in Cloud computing, which does the task allocations of the virtual machines by Job planning is one of the vital elements in Cloud computing, which does the task allocations of the virtual machines by assuring proper load-balancing of the job. It works on some conditions like stability and adaptability of the system etc.

Hence it is required to make job planning work efficiently to have a fair distribution of load on all VM's. The Cloud computing is pulling in an expanded number of applications to keep running in the data centers remotely. Numerous tricky applications require parallel execution abilities. Some parallel applications [1] demonstrate a reduction in usage of CPU resources at a point when there is an expansion in parallelism. If the jobs are not planned effectively then it decreases the CPU execution.

In view of the closeness between those conditions and the practices of Artificial Bee Colony (ABC) algorithm [1] which is propelled by the conduct of a honey bee colony [1] to gather nectar, our work is derived from this algorithm to planned a set of jobs on VM's and keep up the load balance of VM's leased by clients.

* School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India-632014.

II. LITERATURE SURVEY

Artificial bee mechanism [2], is a technique for bionics, proposes an enhancement technique taking into account the social occasion conduct of honeybees. The tiniest searching model of insight produced in bees contains a food source, three types of bees i.e. Scout bees, employed bees and onlooker bees [1]. Two fundamental behavioural models are: Scavenging nectar in the food source and surrendering a food source. A bee settles on a choice on scavenging taking into account different elements. For example, nectar's distance, amount and quality [2].

The algorithm [1] derives the general idea from artificial bee colony algorithm to boost the execution of cloud jobs and scheduling performance. The working can be better explained as follows. Let VM= {vm1, vm2, vm3….vm_m} be the set of virtual machines rented by user on which a set of jobs be T= {t1, t2, t3….t_n} be planned. Also it is considered that the planning of jobs in non-pre-emptive and the execution of jobs on VM's cannot be interrupted [1].

According to [1], the algorithm does not shorten the execution time of job scheduling. On the contrary the problem of load balancing occurs when many clients want to access the same VM simultaneously whilst other VM's are free. It expresses VM's load in this work by the execution time of the ongoing jobs in VM. The work is based on context dependent systems to manage the difference in the environment in which the VM's occurrence is set up by exploring and installing the lost libraries such that their execution time and attributes offer the feasible cost.

In [4], it is shown that suppose, a job removed from overloaded VM has to search a proper under loaded VMs it can be allocated to [4]. It has two chances i.e., either it searches the VM (+ve signal) or it won't find the appropriate VM (-ve signal). There could be at least one VM from a set of VMs for allocation which can take this job. Now the job needs to discover best among these VMs taking into account the QoS criteria called job priority. It is called as battle for the VM among jobs. When this battle is over, the triumphant job is given to that particular VM found and the specifics are revised. If a job cannot search an appropriate VM, it goes to postpone the allotment, amid which it experiences and begins listening the data upgrades by different jobs (like honey bees listening various dance moves). When it affirms the data, the procedure begins from searching the VM sets and after effective VM set recognition, it will battle with other contending jobs to search its most appropriate VM and gets allotted to it. Data is revised once it finishes its battle with different jobs despite of the result. As more jobs arrive, the cycle begins until all jobs are allotted to VMs and the planning system is very much adjusted in light of burden and preferences.

III. SOLUTION METHOLOGY

3.1. Artificial Bee Colony

This Algorithm is basically based on the behaviour of the swarm of honey bees". There are two types of bees, Scout bees and onlooker bees. The scout bees go in search of the nectar (food source) and on finding the best available nectar they return to their hive and do a waggle dance with the other bees to share the information they found. Once the sharing is done the onlooker bees go to that nectar to find the suitable and the most feasible food source for themselves. Two types of behaviour models are depicted by honey-bees:

1. It finds suitable food source.
2. It does not find suitable food and tries again.

3.2. Algorithm

1) Initialize population. Set maximum number of iterations as max_it for which the algorithm will run. Let J be number of jobs submitted to system and V be number of virtual machines. According to ABC, the number of jobs are initialized to double the number of virtual machines.

2) Allocate initialized tasks to virtual machines randomly.

3) Examine the status of submitted tasks and calculate load accordingly on each Virtual machine based on given formula:

$$Load_i = \frac{(\Sigma_{k=1}^{j} Length\ of\ task\ assigned\ to\ V_i)}{(NPE_i * MIPS_i)} \quad (1)$$

Where NPE_i = number of processing elements of V_i

$MIPS_i$ =Million instructions per second of

4) Find the virtual machine having the minimum load value. Set a range based on that minimum value and memorize the virtual machines whose load values fall under the range. The range of best values depends on the context of application.

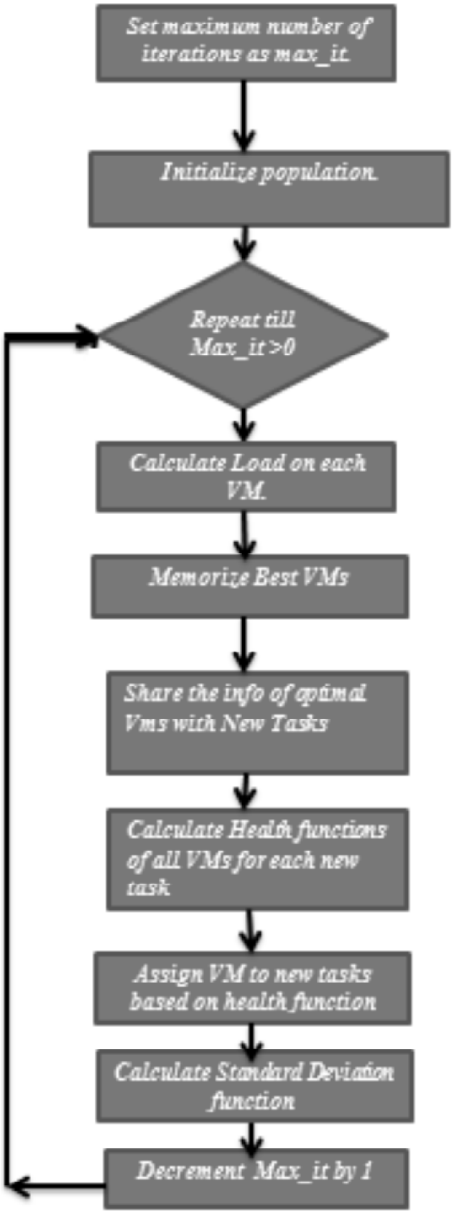

Figure 1: Algorithm flow

5) Share the memorized information with new incoming tasks (onlooker bees).These new tasks select corresponding virtual machines based on health function stated below:

   a) Calculate Health function for all memorized virtual machine for a new onlooker bee:

$$Health\ function_i = [\![Load_i]\!] + [\![L\_Input\ size_i / BW_i]\!] \quad (1)$$

   Where represents length of a job when it is given as an input.

   Represents the amount of bandwidth of a virtual machine.

6) Allocate new jobs to virtual machines based on Health function.

7) If task fails decrement the max_limit value. Submit it again. Max_limit represents the total number of chance a cloudlet is given. If max_limit reaches zero then that particular cloudlet is abandoned.

8) Calculate load of all virtual machines and check whether the system is balanced or not. [3]

9) Decrement max_it value.

10) Repeat steps from 3 to 9 till max_it becomes zero. The Flowchart for the above algorithm is depicted in figure 1.

The System diagram for the artificial bee colony based load balancing is depicted in the figure 2. Here jobs are Cloudlets (CL) submitted by the user. Current Load of each virtual machine is calculated based on load function. Based on Load function, best virtual machines are selected and shared with each incoming cloudlet. For each new incoming cloudlet, health function is calculated on best virtual machines. Virtual machine having the minimum value of health function is assigned new task. This process continues until all cloudlets are served.

IV. RESULTS

The above algorithm has been implemented using CloudSim framework [9]. The characteristics of Datacentres, Virtual machines and cloudlets chosen are listed in table 1, 2 and 3 respectively.

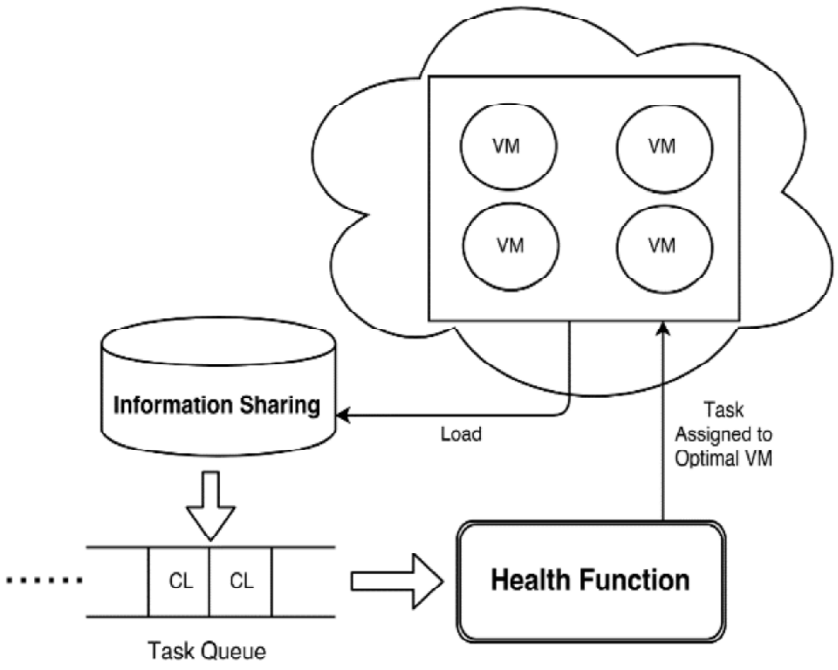

Figure 2: System Diagram

The metrics considered for the result analysis in table 4 and 5 are:

a) Number of Virtual machines (A)

b) Number of Cloudlets (B)

c) Average time taken by the cloudlet to execute(C)

d) Average number of cloudlets per virtual machine (D)

e) Total time taken for all cloudlets to execute (E)

Table 1
Data center characteristics

Data Center Characteristics	
No. of Data Centers	2
No. of Processors in each Datacenters	Quadcore
MIPS	3000
RAM	8 GB
Storage	1 TB
Bandwidth	30 GB

Table 2
Virtual machine characteristics

Virtual Machine Characteristics	
Image size	10,000 MB
RAM	512-1024 MB
MIPS	1000-3000
Bandwidth	1000 MB
Process Elements	1-3

Table 3
Cloudlet characteristics

CloudLet Characteristics	
Length	1000-3000 MB
File size	300-900
Output Size	300
Process Element number	1-3
Cloudlet Scheduler Policy	Space Shared

Table 4
Scenario a

A	B	C	D	E
10	50	0.86 s	6	5.33 s
10	97	0.82 s	9	9.78 s
10	152	0.85 s	15	15.78 s

Results are analysed on the basis of two scenarios:

a) On increasing number of cloudlets while keeping the number of virtual machines constant.

b) On decreasing number of virtual machines while keeping the number of cloudlets constant.

The above scenarios a and b are described in tables 4 and 5 respectively.

The above table depicts that on increasing number of cloudlets, the average number of cloudlets assigned to each virtual machine increases in order to distribute the load in a fair manner. Due to this, total time taken by the cloudlets for their execution increases slightly with respect to the change in their number. The average time taken by the cloudlet to execute depends upon the cloudlet scheduler policy.

In this simulation space shared policy is chosen. Now in this Cloudlet Scheduler, Space Shared is a scheduling policy performed by a virtual machine. It considers that there will be only one cloudlet per VM. Other cloudlets will be in a waiting list. We consider that file transfer from waiting cloudlets happens before cloudlet execution. i.e., even though cloudlets must wait for CPU, data transfer happens as soon as cloudlets are submitted.

Table 5
Scenario b

A	B	C	D	E
10	96	0.82 s	9	9.76 s
8	96	0.81 s	12	9.89 s
6	96	0.81 s	15	12.89 s

The above table depicts that on decreasing the number of virtual machine, the average number of cloudlets assigned to each virtual machine increases in order to distribute the load in a fair manner. Due to this, the total time taken by the cloudlets for their execution shows a change on a very small scale with respect to their number.

V. CONCLUSION

A load Balancing Algorithm on artificial honey bee colony approach has been implemented. Through impersonation of conduct of honey bees, it streamlines the measure of nectar to come to the greatest throughput and enhanced reaction time for submitted jobs.

VI. REFERENCES

[1] A. Benali, B. El Asri and H. Kriouile, A pareto-based Artificial Bee Colony and product line for optimizing scheduling of VM on cloud computing, *2015 IEEE International Conference on Cloud Technologies and Applications (CloudTech)* pp.1-7.

[2] Jing Yao, Ju-hou He. 2012. Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm *In proceedings of 2012 International Conference on Computing Technology and Information Management*, vol. 1 pp185-189.

[3] D. L.D. and P. Ventra Krishna. 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, vol. 13, no. 5, pp. 2292-2303.

[4] S. Ghafari, M. Fazeli, A. Patooghy and L. Rikhtechi.2013. Bee-MMT: A load balancing method for power consumption management in cloud computing. *2013 Sixth International Conference on Contemporary Computing* (IC3).

[5] BM Salih and HA Edreis, 2016, Comparison between Virtualization and Cloud Computing, *IJSR*, vol. 5, no. 6, pp. 195-199.

[6] Malhotra L, Agarwal D and Jaiswal A. 2014. Virtualization in Cloud Computing Journal of *Inform Tech Soft Engg*. vol. 04, No. 02, pp-1 -3.

[7] D. Karaboga. 2010. Artificial bee colony algorithm, *Scholarpedia*, vol. 5, no. 3, p. 6915.

[8] B. Verma and D. Kumar.2013. A review on Artificial Bee Colony algorithm, *International Journal of Engineering & Technology*, vol. 2, no. 3, pp.175-186.

[9] R. Kumar and G. Sahoo.2014. Cloud Computing Simulation Using CloudSim, *IJETT*, vol. 8, no. 2, pp. 82-86.

[10] T. Goyal, A. Singh and A. Agrawal.2012. Cloudsim: simulator for cloud computing infrastructure and modeling *In Procedia Engineering*, vol. 38, pp. 3566-3572.