A Nature Inspired Bee Colony Optimization Model for Improving Load Balancing in Cloud Computing

B. Mallikarjuna, P. Venkata Krishna

Abstract*: Cloud computing is treated as one of the emerging fields in the distributed systems. In the recent years, the IT industry had rely more on the cloud distribution, this ultimately leads to the load over the VMs. In this paper, we are dealing with the load balancing issue in cloud computing. The reduction of load over the VMs will ultimately lead to the increase of performance. To solve the load balancing issue we proposed a model as bee colony optimization model which is inspired by the nature bee colony mechanism. The proposed bee colony mechanism considers the iteration process for efficient scheduling of tasks to the VMs. The performance estimation of the proposed algorithm is tested with other existing algorithms called as FCFS and Dynamic load balancing algorithm. The proposed algorithm is well performed when compared to the other algorithms.

Key words: Cloud computing, VMs, Load balancing, Scheduling, Iteration.

I. INTRODUCTION

In the recent trends cloud computing is providing major services to the IT industries, because the most of the organizations are not willing to spend money on the infrastructures. The key features in the cloud computing technology areautomatic deployment, web services, virtualization technology and high speed internet. The services provided by the cloud computing are infrastructure as a service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS). To attract more users over the cloud, the service providers introduced virtualisation technology; the virtualization technology creates the virtual machines over the limited hardware. The virtualization technology provides the users with own personal hardware that are not interfered with the any other hardware.

The cloud computing had a limited hardware resources; if the utilization of the resources is high it will directly impacts the performance. So, how to balance the load over the VMs is a worth wile concern [1]. Due to this, the load balancing over the cloud is a major problem [2]. Although, there are number of load balancing algorithms were proposed, not all are eligible for solving the load balancing issue in the cloud computing [3-4]. Most of the load balancing algorithm requires the complete details of the task and the resources which are not suitable for the dynamic environments.

In this paper, we are concentrating on the VM load balancing using the Bee Colony optimization (BCO). The proposed algorithm process the VM requests by considering the factors like reducing the scheduling time, minimizing the amount of data by considering the historical data and uses the fixed tasks.

The remainder of the thesis is organized as follows. Section 2 explains about the literature survey. Section 3 deals with the proposed model called as bee colony optimization. Section 4 explains about the experimental evaluation of the proposed model and finally conclusion is drawn in section 5.

II. LITERATURE SURVEY

Load balancing is a mechanism offered by the cloud computing. By the implementation of load balancing, the cloud will achieve throughput, fault tolerance, high performance and response time. Load balancing is defined as the process of handling overloaded VMs, removes their load and assigns to the under loaded VMs. The load balancing process influences the overall performance of the task execution. The major approaches in load balancing process are classified as follows [2].

Static load balancing mechanisms: Static load balancing mechanisms make the decisions regarding the balancing of load at the time resource allocation process itself. The main benefit of this mechanism is it reduces the overload and simple to implement.so, there no requirement of performance evaluation of the VMs under different conditions. In [5] Houle et al. proposed a mechanism for static load balancing based on the tree structure process. Hu et al. [6] proposed an algorithm for load balancing using the lagrange principle by calculating the Euclidian distance among the weights of the VMs.

Dynamic load balancing mechanisms: the dynamic load balancing mechanism makes the decision of the work load distribution at the run time. The distribution process requires a complex mechanism to distribute the load to the under load VMs [7]. In [8], a distributed load balancing mechanism was proposed for load balancing in the cloud computing [17]. This model utilizes the nature inspired approach for load balancing the potential VMs in the large scale cloud. In [3], the honey bee foraging mechanism was proposed for load balancing process. They used the natural phenomenon for distributing the load. The random based sampling method was proposed for maintain the information of the load among the nodes. The artificial bee colony algorithm was proposed by karaboga [9-10]. The algorithm was inspired by new

Revised Manuscript Received on December 08, 2018.

B. Mallikarjuna, Assistant Professor, SCSE, Galgotias University, Greater Noida, Uttar Pradesh, India

P. Venkata Krishna, Professor & Director, Dept of CS, SPMVV, Tirupati, Andhra Pradesh, India.

foraging behaviour called as honey bee swarm. The algorithm concentrated on the numerical function. The artificial bee colony algorithm was proposed in many fields such as inverse analytical problems [11], flow shop scheduling problems [12], minimum spanning tree problems [13], and digital signal processing problems [14].

In [15] a, new mechanism was proposed called as bees algorithm, which will works on the principle of population based search algorithm. The honey bee optimization algorithm was proposed by Dinesh babu et al [16]. They proposed honey bee behavioural load balancing algorithm (HBB-LB) for maximizing the throughput of the model and also they made an attempt to reduce the queue size at the resources.

III. PROPOSED MODEL

3.1. System Design

The load balancing among the cloud computing is done by assigning the VMs to the servers by managing the balance over the resource utilization among all the servers.Let us consider that here is m type of tasks and n type of VMs. The parameters which can be used for the initialization of VMs are given as follows

- The identification of VM is based on the VID and it is unique for every VM.
- α represents the million instructions per second which is used for measuring the speed of the CPU and it is a fundamental parameter for measuring the computational speed.
- β represents the storage size of the disc, i. e, the space required to store the data in the VM.
- The RAM memory is denoted as γ, where the instructions are copied into the VM memory for the execution, after the completion the results will be stored in to the storage disc.
- The bandwidth is represented as δ, which is used to calculate how much data is transferred with in a particular period of time.

The parameters for the tasks are given below

- The unique ID is represented for every task, based on this ID the tasks aare allocated to the VMs.
- The length of task is represented with T_{Length}, the performance of the VM is depend on the T_{Length}.
- The size of the job is denoted with J_{size} and the size of the output file is denoted as O_{size}.

The total number of tasks and the number of VMs decides the load of the virtual machines. Equation 1 shows the capacity of the VM.

$$C = (\alpha \times NCP) + \delta \quad (1)$$

Where NCP represents the number of processors or CPUs are available.

The selection of the optimal VMs is based on the equation 2.

$$X_i = T_i (\sum_{i=1}^{n} T_i)^{-1} \quad (2)$$

Where Xi is the percentage of the i^{th} task length on overall VMs, T_i is the length of the allocated task in the i^{th} VM.

To calculate the load on the VM, the equation 3 is framed as follows.

$$L = (\sqrt{\sum ((\sum_{i=1}^{n} X_i) / VM_{total} - X_i)^2})(VM_{total})^{-1} \quad (3)$$

Here, the VM_{total} is the average rate of the load of all VMs.

3.2. Bee Colony based algorithm for Load Balancing

The main objective of this paper is to propose an optimization approach that is inspired by decision making process for load balancing using artificial bee colony algorithm. Before going to the in detail procedure of the load balancing approach, let us consider the bee in nature. The bee colony optimization makes the decision process by searching the best food resources among various opportunities. For instance, swarm based decision making method.

The Natural bee colony algorithm is shown in algorithm 1:

Algorithm 1: Nature bee colony algorithm

```
Initialize the population for random solutions
Calculate the fitness function for each population
If (stopping criteria is not met) then
   Create new population;
   Select new locations for population
   Recruit the new population for new locations
   Evaluate fitness function for each population
 Select the fitness bee from each location
Allocate remaining bees to search randomly
And evaluate the fitness
Otherwise
  Stop the process
end
```

Algorithm 1 describes about how the natural bee colony optimization mechanism works. As a first step, the bees are initialized randomly for the solution search. Calculate the fitness function for each population to fit in to the solution set. If the bees are not fit for the location set then search for the neighbourhood bees and find the new locations. Again calculate the fitness function for the newly recruited bees and assign the best fit bee to the location. This process is continued until the stopping criterion is met. The flow diagram for the natural bee colony process is shown in figure 1.

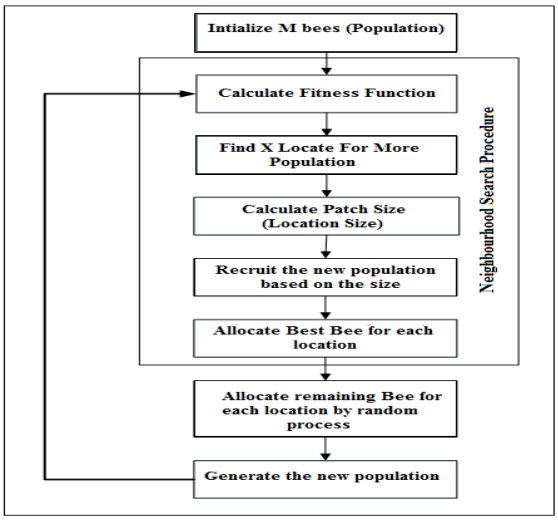

Figure1: Nature Bee colony flow diagram

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

By the inspiration of nature bee colony algorithm, the artificial bee colony algorithm is proposed for the load balancing of task s over the VMs in the cloud computing. The task scheduling in the cloud computing is treated as the NP hard problem. i.e, the proposed algorithm doesn't have the exact solution for the problem, but it provides the approximate solution for global optimization problem.

Algorithm1: Bee Colony Algorithm

Begin

Step 1: Initialize the Tasks and VMs

Step 2: Allocate the VMs to the tasks based on the demand. The optimal selection of VMs is carried by using equation 2.

Step 3: If (selected VM ≠ overload)

{

Allocate the tasks to VM;

}

Otherwise goto step 4:

Step 4: select the most efficient VM from the list using iteration by the equation 4.

$$L_i(k+1) = L_i + \varphi(L_i(k) - L_p(P)) \quad (4)$$

Where i represent the number of times the iteration occurs, φ lies in between 0 and 1

Theφ is calculated by using the equation 5.

$$\varphi = ((T_j \times T_i)(X_j - X_i)^{-1})(\sum(T_m \times T_n)(X_m - X_n)^{-1})^{-1} \quad (5)$$

The φ is acquired by the newton gravity function and through the iteration better scheduling process is identified. The φ is different from the artificial bee colony algorithm, where it lies in between 0 and 1.

Step 5: By the equation 3, it can be identified that the given VM is overloaded or not. If the VM is overloaded go to step 6, otherwise step 7.

Step 6: By the equation 4, it can be identified that the system is overloaded, then, the load of the VM is shifted to the suitable VM based on the equation 6.

$$L_i = L_i + R \times (L_{max} - L_{min}) \quad (6)$$

Where R is the random number ranges in between 0 to 1, L_{max} and L_{min} are the maximum and minimum loads over the VM.

Step 7: Allocate the tasks to the VM

Step 8: if (task ==Null)

Stop the process;

Otherwise

Go to step 2;

end

The algorithm 1 efficiently manages the overloaded VMs by distributing the load to the available VMs. The algorithm manages scheduling of tasks more efficiently and the tasks are allocated to the suitable VMs. The flow diagram for load mechanism is shown in figure 2. As an initial step in the load balancing process, the VMs are selected randomly from the active servers. Find the fitness function for each VM based on the equation 2. If the selected VM is well balanced, then schedule the task to the VM, otherwise implement the equation 4 on selected VMs and finds the suitable VM, if it is also not balanced, then apply the equation 6 and choose the best VM.

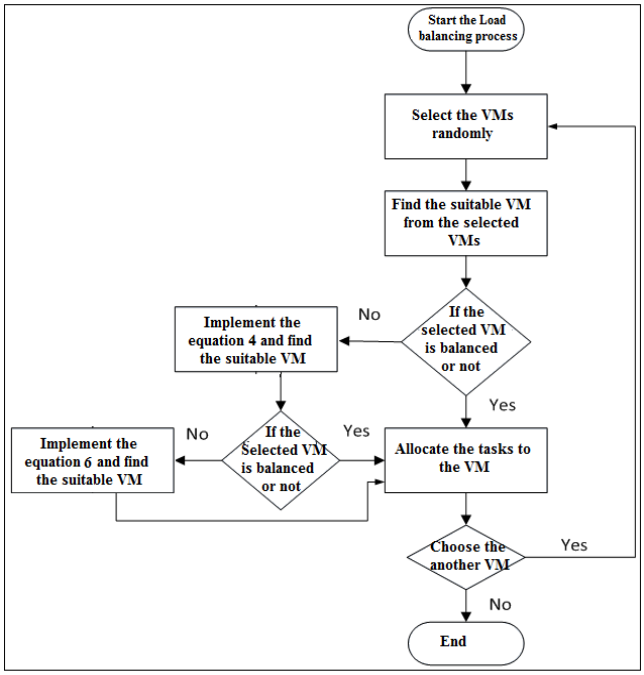

Figure 2: Artificial Bee colony load balancing Flow diagram

IV. EXPERIMENTAL EVALUATION

This section explains about the experimental setup for the proposed artificial bee colony algorithm which handles the load balancing on virtual machines. To implement the proposed algorithm, we need good simulator. The cloudsim [18-20] is one of such simulators which provide flexibility in designing the load balancing algorithms. In this paper, we analysed the performance of the proposed algorithm and compare the results with the existing algorithms called as FCFS and Dynamic Load Balancing algorithm [1].

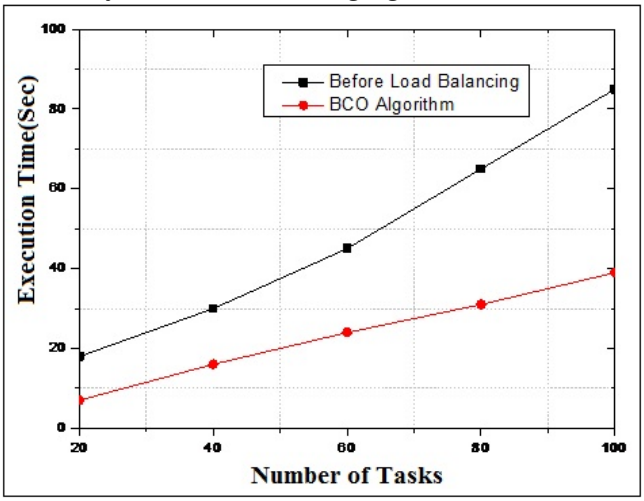

Figure 3: Performance Measurement of the BCO algorithm

Figure 3 shows the performance of the proposed algorithm in terms of execution time. The execution time of the BCO algorithm is less while comparing with the normal approach. The execution time of the normal scheduling process is high. The BCO algorithm finds the suitable VM based on the

length of the task. So, if the load over the VM is high then the task is migrated another suitable VM. So, the proposed algorithm works well in reducing the execution time of the tasks.

Figure 4 explains about the degree of imbalance over the scheduling model. The proposed BCO algorithm presents the better performance when compared to the normal approach. The BCO algorithm evenly distributes the load over the VMs and this will decrease the degree of imbalance over the VMs.

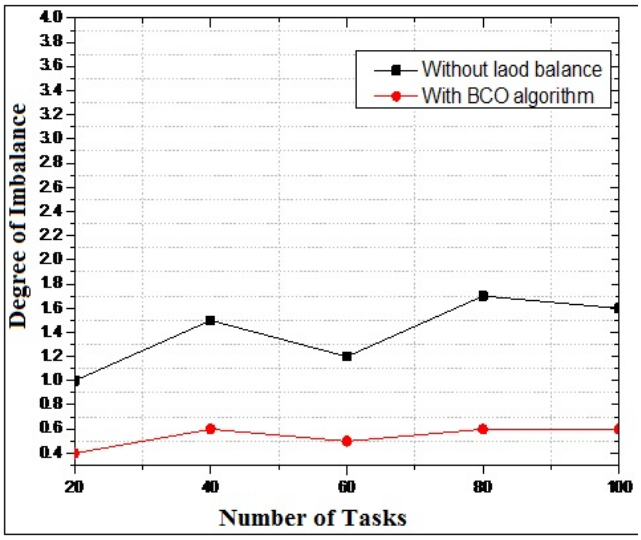

Figure 4: Performance estimation in terms of imbalance degree of VMs

Figure 5 explains about the comparison of proposed BCO algorithm with the well-known existing algorithm called as dynamic load balancing algorithm (DLB) in terms of task migrations. The task migrations are dependent on the capacity of the VMs and the balancing factor of the VMs. The number of task migrations decides the performance of the algorithm. In figure 5, BCO algorithm reduces the number of task migrations when compared to the DLB algorithm. The experiment is conducted with the 100 tasks, only minimum number of tasks has been migrated in BCO algorithm.

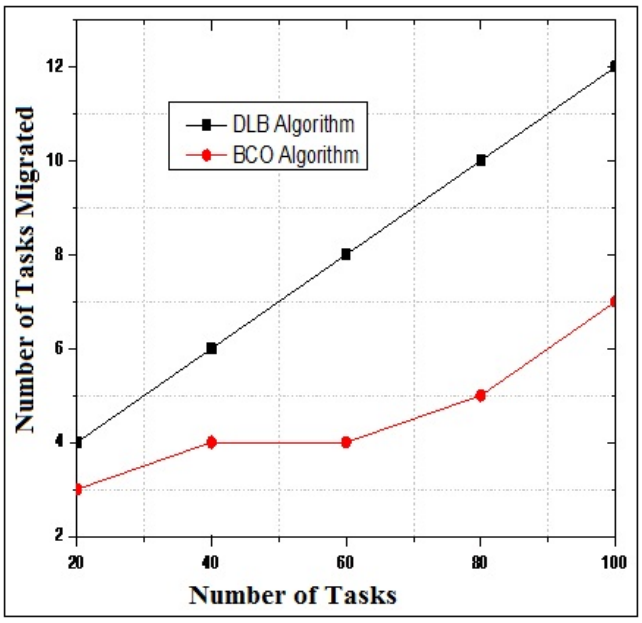

Figure 5: Number of Task migrations when VMs=4

Figure 6 shows the makespan of the three algorithms such as FCFS, DLB and BCO. The makespan is defined as the completion time in between the starting task and ending task. The algorithm is said to be effective when the makespan of the algorithm is minimum. The BCO algorithm recorded the minimum makespan value when compared to the FCFS and DLB. This will justify the effectiveness of the proposed model.

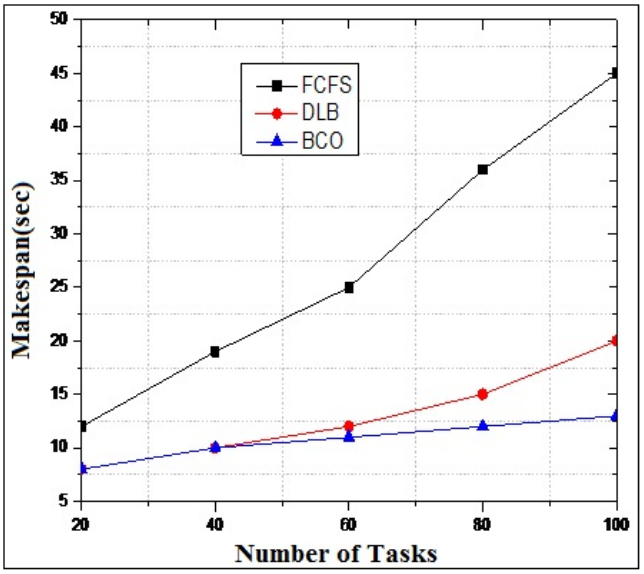

Figure 6: Performance Evaluation of FCFS. DLB, BCO in terms of Makespan

V. CONCLUSION

The paper presents the Bee colony optimization algorithm that deals with load balancing issue in cloud computing. This approach follows the search procedure for finding the better VM for shifting the load. The proposed bee colony mechanism considers the iteration process for efficient scheduling of tasks to the VMs. The iteration process verifies whether the VM is overloaded or not. The experimental evaluation of the proposed model is developed on the cloudsim toolkit. The results proved that the proposed BCO algorithm is superior in terms of reducing makespan when compared with the FCFS and DLB algorithms.

REFERENCES:

1. B. Yagoubi, Y. Slimani, Task load balancing strategy for grid computing, Journal of Computer Science 3 (3) (2007) 186–194.
2. A. Revar, M. Andhariya, D. Sutariya, M. Bhavsar, Load balancing in grid environmentusing machine learning-innovative approach, International Journalof Computer Applications 8 (10 (Oct)) (2010) 975–8887.
3. B. Yagoubi, Y. Slimani, Dynamic load balancing strategy for grid computing,transactions on engineering, Computing and Technology 13 (May) (2006)260–265.
4. B. Yagoubi, M. Medebber, A load balancing model for grid environment,computer and information sciences, 2007. iscis 2007, in: 22nd InternationalSymposium on, 7–9 Nov, 2007, pp. 1–7.
5. M. Houle, A. Symnovis, D. Wood, Dimension-exchange algorithms for load balancingon trees, in: Proc. of 9th Int. Colloquium on Structural In
6. formation andCommunication Complexity, Andros, Greece, June, 2002, pp. 181–196.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

7. Y. Hu, R. Blake, D. Emerson, An optimal migration algorithm for dynamic loadbalancing, Concurrency: Practice and Experience 10 (1998) 467–483.
8. N. Malarvizhi, V. RhymendUthariaraj, Hierarchical load balancing scheme forcomputational intensive jobs in Grid computing environment, in: AdvancedComputing, 2009. ICAC 2009. First International Conference on, 13–15 Dec,2009, pp. 97–104.
9. M. Randles, D. Lamb, A. Taleb-Bendiab, A comparative study into distributedload balancing algorithms for cloud computing, in: Proceedings of 24th IEEEInternational Conference on Advanced Information Networking and ApplicationsWorkshops, Perth, Australia, April, 2010, pp. 551–556.
10. D. Karaboga, An idea based on honey bee swarm for numerical optimization,Technical Report TR06, Computer Engineering Department, Erciyes University,Turkey, 2005.
11. D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm,Applied Soft Computing 8 (1) (2008) 687–697.
12. F. Kang, J. Li, Q. Xu, Structural inverse analysis by hybrid simplex artificialbee colony algorithms, Computers and Structures 87 (13) (2009)861–870.
13. Q.K. Pan, M.F. Tasgetiren, P. Suganthan, T. Chua, A discrete artificial bee colonyalgorithm for the lot-streaming flow shop scheduling problem, InformationSciences 181 (12) (2011) 2455–2468.
14. A. Singh, An artificial bee colony algorithm for the leaf constrained minimumspanning tree problem, Applied Soft Computing Journal 9 (2) (2009)625–631.
15. N. Karaboga, M.B.C. etinkaya, A novel and efficient algorithm for adaptive filtering:artificial bee colony algorithm, Turkish Journal of Electrical Engineering& Computer Sciences Vol. 19 (2011) 175–190.
16. D.T. Pham, E. Kog, A. Ghanbarzadeh, S. Otri, S. Rahim, M. Zaidi, The beesalgorithm—a novel. Tool for complex optimisation problems, in: IPROMS2006 Proceeding 2nd InternationalVirtual Conference on Intelligent ProductionMachines and Systems, Oxford, Elsevier, 2006.
17. LD, DhineshBabu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." Applied Soft Computing 13.5 (2013): 2292-2303.
18. Krishna, P. V., Saritha, V., Vedha, G., Bhiwal, A., &Chawla, A. S. (2012). Quality-of-service-enabled ant colony-based multipath routing for mobile ad hoc networks. Communications, IET, 6(1), 76-83.
19. R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, CloudSim:a toolkit for modeling and simulation of cloud computing environments andevaluation of resource provisioning algorithms, Software: Practice and Experience41 (2011) 23–50, http://dx.doi.org/10.1002/spe.995.
20. R.N. Calheiros, R. Ranjan, C.A.F.D. Rose, R. Buyya, CloudSim: a novel frameworkfor modeling and simulation of cloud computing infrastructures and services,Computing Research Repository, vol. abs/0903.2525, 2009.
21. R. Buyya, R. Ranjan, R.N. Calheiros, Modeling simulation of scalable cloud computingenvironments and the cloudsim toolkit: challenges and opportunities in: Proceedings of the 7th High Performance Computing and Simulation Conference(HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE Press, New York, USA),Leipzig, Germany, June 21–24, 2009.