iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Improved Honey Bee inspired load balancing of tasks with position updation

Ms. Anna Baby
PG Scholar, Karunya University, Coimbatore, India

Abstract*— Load balancing is an important aspect of cloud computing. To achieve optimal machine utilization, tasks from overloaded virtual machines should be transferred to under loaded virtual machines. In this paper, the honey bee foraging technique is improved by updating the position of bees. This algorithm mainly looks for better solutions by applying the knowledge of previous solutions. The search process is changed by introducing inertia weight and acceleration coefficients. The evaluation of performance is done by using clouds. Simulation results shows that the modified artificial bee colony algorithm with position updation reduces the make span as well as response time of the Virtual machine and hence it is significantly better than the original honey bee foraging technique.
***Keywords*— Bee foraging, Predict, Load Balancing, Position updation**

I. INTRODUCTION

Cloud computing is the delivery of computing as a service where all applications are hosted on a cloud which provides shared resources. It relies on sharing of resources over a network. The customer need not buy the software or computation platforms .With cloud computing, different services such as servers, storage, software and applications are delivered to an organization's computers and devices through the internet. The customer is interested in reducing the overall execution time of tasks on the machines.

Cloud computing is a type of computing in which shared information, software and other resources are provided according to the clients requirement at specific time. It is one of the distributed computing paradigms which mainly focus on providing everything as a service to the customer and it provides computational or storage resources and database to users. Services in a cloud are of three types as given below:

Software as a Service (SaaS)

In SaaS, the user uses different software applications from different servers through the Internet.

Platform as a Service (PaaS)

PaaS provides all the resources that are required for building services completely from the Internet, without downloading or installing software [3].

Infrastructure as a Service (IaaS)

It offers the hardware as a service to an organization so that it can put anything into the hardware according to its will.

Virtual machine is the processing unit of cloud computing. Scheduling of the customer tasks within the available resources is a challenging task. More than one task is assigned to one or more VMs that run the tasks simultaneously [6]. This kind of environments should make sure that the loads are well balanced in all VMs. It is the responsibility for the scheduler to balance the loads across the machines.

Load Balancing is a technique to divide the amount of work that a computer has to do between many computers, processors or any other resources to utilize them effectively [1] and to minimize the response time, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded.

The load balancing algorithms which is dynamic in nature does not consider the previous state of the system. It depends only on the present behavior of the system. A load balancing algorithm attempts to maximize the throughput, reduce the response time of user's submitted applications by effectively utilizing the available resources. Load balancing ensures that all the processors in the system as well as in the network do approximately the equal amount of work at any instant of time.

Goals of load balancing are:

To minimize the makespan.

To minimize the response time of a virtual machine.

Load balancing is the task of distribution of application tasks to different processors in an efficient manner to minimize program execution time. Effective implementation of load balancing can make cloud computing more effective and it also improves user satisfaction. Load balancing aims for optimal resource use, maximize throughput, reducing the response time of a virtual machine and avoid overload of any one of the resources [21].

Cloud computing provides much more effective computing by centralized memory, processing, storage and bandwidth. With load balancing, the tasks are not loaded heavily on one VM and some VMs do not remains idle and/or under loaded. The main goal of load balancing methods is to maximize the speed of execution of applications on resources whose workload varies at run time in unpredictable way.

Efficient load balancing is a challenging problem in the cloud environment. The existing method is good in exploration only, not in exploitation [22]. Hence, global optimal solution cannot be found out within a short span of time.

The objective of this work is to efficiently balance the load in virtual machine by achieving a fast convergence speed and to discover a better solution with minimum execution time.

II. RELATED WORKS

Load balancing mechanism distributes the workload across multiple computing resources to utilize them effectively and to reduce the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded.

In particle Swarm Optimization (PSO) proposed by Ayed salman, Imtiaz Ahmed [2] combines local search methods with global search methods (through neighborhood experience), attempting to balance exploration and exploitation. The algorithm is proposed for the task assignment problem for homogeneous distributed computing systems. The result shows that this runs faster with less complexity.

Belabbas Yabougi and Meriem Meddeber [4] proposes a distributed load balancing model for grid computing which can represent any grid topology into forest structure. Distributed approach gains are always better than those achieved by the hierarchical approach.

Dhinesh Babu and P.Venkata Krishna [6] proposed an algorithm for Honey Bee inspired load balancing(HBB-LB) of tasks in cloud computing environments which aims to achieve well balanced load across Virtual machines for maximizing the throughput.

Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros, Humberto Sossa, Valentín Osuna [8] introduces a Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC). Here the numbers of search locations are drastically reduced by considering a fitness calculation. In this algorithm, the computation of search locations is drastically reduced by considering a fitness calculation strategy which indicates when it is feasible to calculate or only estimate new search locations. It reduces the computational complexity.

Guopu.Zha, Sam Kwong [9] proposes Gbest guided artificial bee Colony algorithm for numerical function optimization incorporating the information of global best (gbest) solution into the solution search equation to improve the exploitation. GABC algorithm consists of the three different stages that are the employed bee stage, onlooker stage and the scout stage. The onlooker stage tends to select the good solution to further update, while both the employed bee stage and update every individual in the population.

Pei-Wei Tsai, Jeng-Shyang Pan, Bin-Yih Liao, and Shu-Chuan Chu [14] introduce an Enhanced Artificial Bee Colony Optimization. The onlooker bee is designed to move straightly to the picked co- ordinate indicated by the employed bees and evaluate the fitness value near it in the original ABC algorithm in order to reduce computational complexity.

M.V Panduranga Rao, S.Basavaraj Patil [12] from load balancing strategies for Grid computing proposes a dynamic tree based model. It defines a hierarchical load balancing strategy that estimate the current workload of a grid based on the workload information received from its elements.

Ant Colony Optimization (ACO) for effective load balancing in Cloud Computing proposed by Shagufta khan, Niresh Sharma [19] aims at the load balancing of nodes. ACO is inspired from the ant colonies that work together in foraging behavior. It works efficiently and achieves better utilization of resources, but give more overhead during runtime.

III.LOAD BALANCING MECHANISM USING BEE FORAGING TECHNIQUE

Scheduling of the customer tasks within the available resources is a challenging task. Scheduler can assign each tasks to virtual machine based on the availability and load of virtual machines.

In the proposed system, an improved honey bee foraging technique with position prediction is used for load balancing. However, ABC is good at exploration but poor at exploitation; its convergence speed is also an issue in some cases. Both the exploration and exploitation are important mechanisms in ABC algorithm.

In ABC algorithm, the exploration process refers to the ability of seeking for global optimum in the solution space of various unknown optimization problems, while the exploitation process refers to the ability of applying the knowledge of previous

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

solutions to look for better solutions.

To change the search process, in Reference [15] it is suggested that to introduce two parameters called inertia weight and acceleration coefficients. Consequently, to improve the exploitation, modification forms of the employed bees and the onlooker ones are different in the second acceleration coefficient. The search process can be modified in the following form:

$$V_{ij} = X_{ij}W_{ij} + 2(\phi_{ij} - 0.5)(X_{ij} - X_{kj})\Phi_1 + \varphi_{ij}(X_j - X_{kj})\Phi_2 \quad (1)$$

Where V_{ij} is the new possible solution that is a modified possible solution which depends on its previous solution X_{ij}. W_{ij} is a parameter called inertia weight which controls impacts of the previous solution X_{ij}. X_j is defined as the jth parameter of the best so far solution Φ_{ij} and φ_{ij} are random numbers between the interval [0, 1], Φ_1 and Φ_2 are positive parameters that could control the maximum step size.

On the other hand, if the global fitness is very huge, bees are far away from the best values. So a big correction is needed to search the global optimum solution and then w, $\Phi_1$and Φ_2 should be bigger values. Conversely, only a small modification needed, then W, $\Phi_1$and Φ_2 must be smaller values.

In this work, inertia weight and acceleration coefficients are defined as functions of the fitness in the search process of ABC. They are proposed in Reference [15] as follows:

$$W_{ij} = \Phi_1 = {}^{1}\!\big/_{1 + \exp\left(-\frac{\text{Fitness(i)}}{\text{ap}}\right)} \quad (2)$$

$\Phi_2 = 1$, if a bee is employed one.

(3)

$$\Phi_2 = {}^{1}\!\big/_{1 + \exp\left(-\frac{\text{Fitness(i)}}{\text{ap}}\right)}$$, if a bee is onlooker one (4)

Where ap is the Fitness in the first iteration [15]. In order to further balance the process of the exploration and the exploitation, the modification forms of the employed bees and the onlooker ones are different in the acceleration coefficient Φ_2. The main objectives of this method are to achieve a fast convergence speed and to discover a best solution.

In initialization process, the proposed algorithm like ABC starts by correlating all employed bees with arbitrarily produced food sources. After initialization, the population of the food sources is subject to repetitive cycles of the search processes of the employed bees, the onlooker bees and the scout bees.

In this algorithm, an employed bee firstly works out three new solutions by three different solution search equations, and then selects and decides the best one as the candidate solution. Because of the computation of the candidate solution before the employed bee is selected where they should go to explore, the procedure of computing new food position is known as 'predict'. After the bees 'predict' new candidate solution by three dissimilar solution search equations, [15] they decide the best one from the three solutions as the candidate solution.

If the fitness values of the candidate solution are better than the best fitness value accomplishes so far, then the employed bee's moves to this new food source and synchronously abandons the old one, otherwise it leftovers the previous food source in its mind. When all employed bees have completed this process, they divide the fitness information with the onlookers, each of which chooses a food source based on the probability.

We have to set priority for each task as high, middle and low priority [6]. If the employed bees are of overloaded VM, then it will replace with a low priority task in onlooker bees.

In every iteration, the employed bees will search for a good position for assigning the task. The final candidate solution will be the best solution so far.

As in the case of the employed bee, an onlooker 'predicts' three modification on the position in their memory, and then chooses the best one as the candidate source and verifies the fitness value of the candidate source. Providing that the fitness value of the candidate source is better than that of the previous one, the bee would memorize the new position and forget the old one. In the proposed honey bee foraging technique, the three solution search equations are separately computed, but influence each other by the chosen best solution.

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

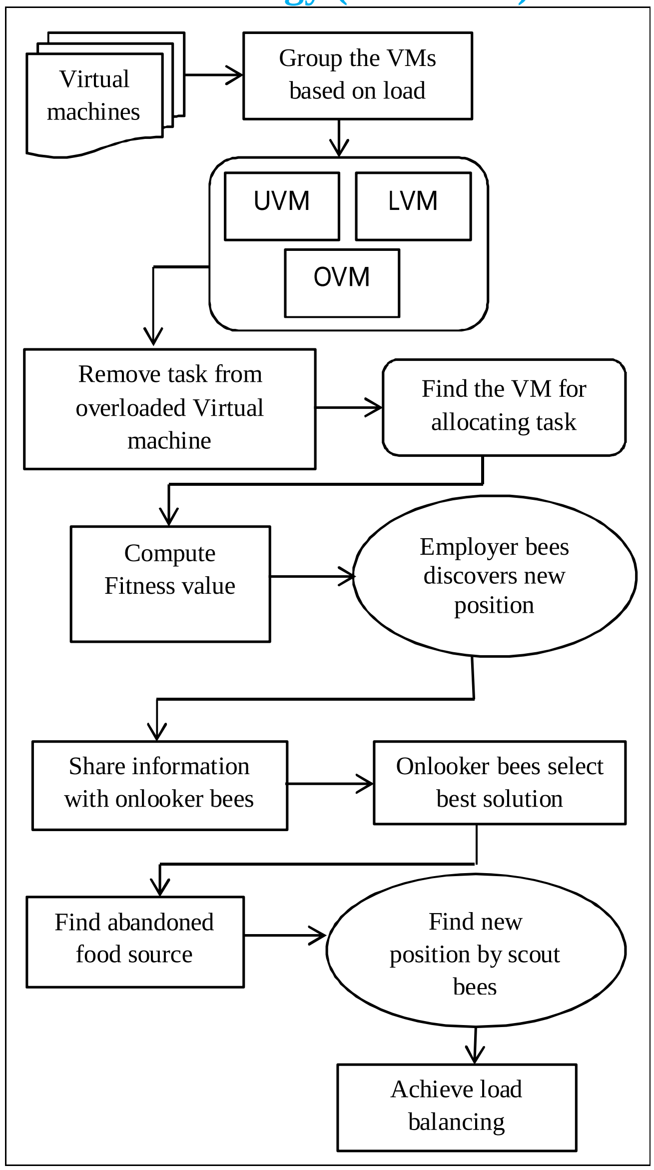

Fig 3. Flowchart of load balancing mechanism using Bee foraging technique with position updation

IV. ALGORITHM

Step 1 Submit the list of tasks T=T1, T2…Tn by the user.

Step 2 Get the available virtual resources from data center. i.e., VM_1, VM_2… VM_m.

Step 3 For each VM, find the capacity and load.

C= mips * no of cpu + bw

L=N (T,t) / S(VM,t)

Where N (T, t) is the total number of tasks given at time t and S(VM,t) is the Number of tasks which is finished until time t(Dhinesh Babu et.al.,2013).

Step 4 Assign the priority to the tasks by considering the length of task. Task with Minimum length has higher priority.

Step 5 Check if (Standard deviation< Threshold time)

System is balanced

Exit

where $\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(PT_i - PT)^2}$

m: No of Virtual machine

σ: Standard deviation

PT=Load / Capacity

Else

Go to Step 6

Step 6 Take each task one by one and allocate it into the VMs by considering the priority and load

Step 6.1 Compute the fitness value, $\sigma \leq T_s$, where threshold value T_s is in between 0 and 1.

Step 6.2 Based on Fitness value, employed bees update the source position.

$$V_{ij} = X_{ij}W_{ij} + 2(\phi_{ij} - 0.5)(X_{ij} - X_{kj})\Phi_1 + \varphi_{ij}(X_{ij} - X_{kj})\Phi_2$$

$W_{ij}, = \Phi_1 = 1/(1 + \exp\left(-\frac{Fitness(i)}{ap}\right))$

$\Phi_2 = 1$, if a bee is Onlooker one.

Φ_1, Φ_2: Fixed numbers between [0, 1].

$\Phi_2 = 1/(1 + \exp\left(-\frac{Fitness(i)}{ap}\right))$, if a bee is Employed one.

W_{ij}: Inertia weight

X_{ij} : Nearest neighborhood search solution of employed bees.

X_{kj} : Nearest search solution of onlooker bees.

ap: Fitness in first iteration.

ϕ_{ij}: Random numbers between [0,1] for employed bees.

φ_{ij}: Random numbers between [0, 1] for onlooker bees.

Step 6.3 Employed bee share the neighborhood position information with onlookers and scout bees.

Step 6.4 Repeat step 6.1 to 6.3 for each iterations until a best solution is found.

Step 7 After allocating all tasks, check the load of the Vms.

Step 7.1 If the VM is overloaded, it goes for next underloaded VM and assign the task.

Step7.2 After completing the task, continue the process for all the task till the system become balanced

Step 8 Stop

V. PERFORMANCE METRICS AND RESULT ANALYSIS

Performance metric is a standard definition of a measurable quantity that indicates some aspect of performance. A performance metric should be measurable, have a clear definition, including boundaries of the measurements, should indicate progress toward a performance goal, and should answer specific questions about the performance. Performance metrics should be consistent with the performance objectives and performance goals of the projects. The performance metrics must be directly related to the performance objectives.

Makespan: Makespan can be defined as the overall task completion time. We denote completion time of task Ti on VMj as CTij [6].

$$\text{Makespan}=\max \{CT_{ij} \mid i \in T, i = 1, 2 \ldots n \text{ and } j \in VM, j \in 1, 2 \ldots m\}$$

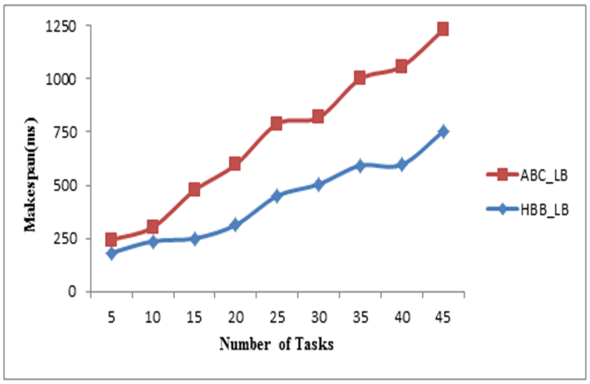

Fig 4.1 Comparison of Makespan

Here, we are comparing honey bee inspired load balancing algorithm (HBB_LB) with position updation with artificial bee colony algorithm (ABC_LB). The comparison is made in terms of Number of tasks and overall completion time.

Response time: It is the amount of time taken between submission of a request and the first response that is produced. The reduction in waiting time is helpful in improving the responsiveness of the VMs.

Response time =min {Finish time – Submission time}

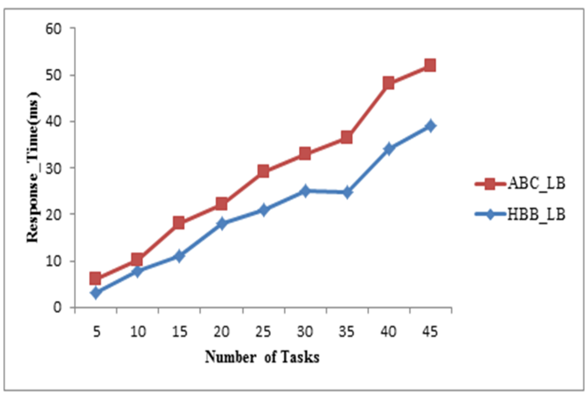

Fig 4.2 Comparison of Response time

The comparison is made in terms of Number of tasks and the response time. Response time is minimum for ABC algorithm with position updation than the HBB-LB algorithm. It can reduce the waiting time of tasks within a virtual machine.

VI. CONCLUSION AND FUTURE WORKS

Load balancing of tasks in virtual machine plays an important role in cloud environments. The existing load balancing algorithms in cloud computing does not consider previous solutions for finding the global optimal solution. The modified bee foraging technique with position updation applies the knowledge of previous solution and mainly looks for a better solution. The search process the bee foraging technique is entirely changed to by introducing best so far solution, inertia weight and acceleration coefficients. The position of virtual machine is updated in an efficient manner for each iteration. If the number of iterations is less, then the convergence rate is high. Hence, a global optimal solution can be found out by achieving fast convergence speed.

The algorithm ignores the idle condition of a Virtual machine even after the completion of jobs assigned to it. In future, we would like to extend our work to ensure that no Virtual machines remain idle by employing random stealing method.

REFERENCES

[1] Arash Ghorbannia Delavar, yalda Aryan (2011)" A synthetic heuristic algorithm for independent task scheduling (SHIS) in cloud systems", IJCSI vol 8, Issue 6, No 2

[2] Ayed salman, Imtiaz Ahmad (2002)" Particle swarm optimization for task assignment problem", Microprocessors and Microsystems 26,363-371

[3] B. Yagoubi and Y. Slimani(2007)" Task Load Balancing Strategy for Grid Computing", Journal of computer science,ISSN 1546-9239

[4] Belabbas Yabougi, Meriem Meddeber (2010)" Distributed Load balancing model", ARIMA journal, Vol 12,pp 43-60

[5] Deepak.C.Vegda,Harshad.b.Prajapati(2014)"Scheduling of dependent tasks application using random search technique",IEEE,978-1-4799-2572

[6] Dinesh Babu L.D, P.Venkata Krishna (2013)"Honey Bee inspired Load Balancing of tasks in cloud computing environments", Applied soft computing 13,2292-2303

[7] E.Iniya Nehru (2012)" Priority based load balancing in cloud for data intensive applications",Journal of Artificial Intelligence 5(4):200-206

[8] Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros, Humberto Sossa, Valentín Osuna(2012)" Block matching algorithm for motion estimation based on ABC", Applied soft computing.

[9] Guopu.Zha, Sam Kwong (2010) "Gbest-guided Artificial bee Colony algorithm for numerical function optimization", Elsevier 3166-3173

[10] Jing Yao, Ju-hou He(2013)" Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm",IJNCM, vol 3,no 1

[11] Kobra etminami, Prof.M.Naghibzadeh (2007)" A Min-Min Max-Max selective algorithm for Grid task scheduling",IEEE,1-4244-1007-X/07

[12] Kavya S.A, M.V Panduranga Rao,S.Basavaraj Patil(2012)"Grid computing for load balancing strategies",International journal of science and research,ISSN:2319-7064

[13] Nuttapong Netinda, Booncharoen Sirinaovakul (2014)"Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization", Springer,68:1579-1603

[14] Pei-Wei Tsai, Jeng-Shyang Pan, Bin-Yih Liao, and Shu-Chuan Chu(2009)"Enhanced Artificial Bee Colony Optimization", International journal of innovative computing,vol 5, no 12

[15] R. Murugan, M.R Mohan (2012) "Modified artificial bee colony algorithm for solving economic dispatch problem", ARPN Journal of Engineering and Applied Sciences, ISSN 1819-6608, VOL. 7, NO. 10

[16] R.SRao,S.V.L.Narasimham,(2008),"Optimization of distribution network configuration for loss reduction using ABC algorithm", International journal of electrical power and electrical systems engineering,116-122

[17] Ratan Mishra and Anant Jaiswal(2012)" Ant colony Optimization: A Solution of Load balancing in Cloud",International Journal of Wen and Semantic Technology,Vol 3,No 2

[18] Sandip kumar Goyal, Manpreet singh(2012)" Adaptive and dynamic load balancing in ant colony optimization",IJET,vol 4, no 4

[19] Shagufta khan, Niresh Sharma(2014)" Ant Colony Optimization for Effective Load Balancing In Cloud Computing",IJARCSSE,Vol.4, Issue 2

[20] Snehal D.Sonawane, R.H. Borhade(2014)"Load Distribution and Balancing over Cloud using Cloud Partitioning", International journal of current engineering and technology,ISSN 2277-4106

[21] Sowmya Suryadevera, Jaishri Chourasia, Sonam Rathore & Abdul Jhummarwala(2012)" Load Balancing in Computational Grids Using Ant Colony Optimization Algorithm" IJCCT,ISSN(ONLINE):2231-0371

[22] Tejinder Sharma, Vijay Kumar Banga(2013)" Efficient and Enhanced Algorithm in Cloud Computing",IJSCE,ISSN:2231-2301

[23] Basturk B and Karaboga D. 2006. An artificial bee colony (ABC) algorithm for numerical function optimization. In: Proceeding of IEEE Swarm Intell. Symp. Indianapolis.