IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. V (Mar – Apr. 2015), PP 97-104
www.iosrjournals.org

An efficient load balancing using Bee foraging technique with Random stealing

Ms. Anna Baby [1], Dr. Joshua Samuel Raj [2]
[1](*Department of computer Science and Engineering, Karunya university, India)*
[2](*Department of computer Science and Engineering, Karunya university, India)*

Abstract: *Load balancing is an integrated aspect of cloud computing. Inorder to achieve optimal machine utilization, tasks from overloaded virtual machines have to be transferred to underloaded virtual machines. In this paper, the honey bee foraging mechanism for load balancing is improved by random stealing technique. For finding the state of a virtual machine, the deviation of virtual machine load is calculated and checked for confinement within a threshold condition set. With random stealing method, tasks are stolen from a random Virtual machine when a VM is idle. It thus saves the idle time of the processing elements in the Virtual machine. The performance evaluation is done by using cloudSim. Simulation results show that the modified honey bee foraging technique with random stealing reduces the makespan of algorithm execution together with balancing system load and reduces the idle time of Virtual machine.*
Keywords*: Bee foraging, Load balancing, Random stealing, Task scheduling, Thresholds*

I. Introduction

Cloud computing is one of the distributed computing paradigm which mainly focuses on providing everything as a service to the customer and it provides computational as well as storage resources to users. The processing units in cloud environments are called as virtual machines (VMs). 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 [1]. This kind of environments should make sure that the loads are well balanced in all virtual machines.

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 distributes workloads across multiple computing resources such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize the resource use, maximize the makespan as well as to minimize the response time. A load balancing algorithm attempts to improve the response time of user's submitted applications by ensuring maximal utilization of 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 [2]. To achieve optimal machine utilization, tasks from overloaded virtual machines are transferred to the neighbour Virtual machine whose load value is below threshold. The proposed algorithm improves the honey bee foraging technique by ensuring that no virtual machines remain idle by employing random stealing mechanism. When the virtual machine is idle, [3] the algorithm will make multiple attempts to steal jobs from a random Virtual machine.

Cloud computing provides much more effective computing by centralized memory, processing, storage and bandwidth. It should make sure that the tasks are not loaded heavily on one VM and also ensure that some VMs do not remains idle and/or under loaded [4]. The main objective of load balancing methods is to speed up the execution of applications on resources whose workload varies at run time in an unpredictable manner.

This paper is divided into the following sections: section 2 describes the related works, Section 3 gives the description of enhanced algorithm, Section 4 provides experimental results and finally Section 5 gives the conclusion and future enhancement possibilities.

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 eliminating a condition in which certain nodes are over loaded while others are under loaded.

Dhinesh Babu and P.Venkata Krishna [1] 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. In particle Swarm Optimization (PSO) proposed by Ayed salman, Imtiaz Ahmed [5] 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 [6] proposed 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.

Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros, Humberto Sossa, Valentín Osuna [7] 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 [8] proposed 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 [9] 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 [10] proposed a dynamic tree based model in load balancing strategies for Grid computing. 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 [11] 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. Yun-Han Lee, Seiven Leu, Ruay-Shiung Chang proposed an improved hierarchical load balancing algorithm, iHLBA [12] to schedule jobs in the Grid environment wherein it assigns the fittest resource to the job and compares the clusters’ load with the adaptive balance threshold to balance the system load. The iHLBA is capable of balancing the overall load on the system, and offers an improvement to the makespan due to its selection of the most suitable resource for a given job based on the most current system status.

In the existing load balancing algorithms the idle condition of virtual machine is ignored and thus leads to the wastage of processing time. Hence the global optimal solution with a balanced load cannot be found out within a short span of time.

III. Honey bee behaviour inspired load balancing of tasks

Scheduling of the customer tasks within the available resources is a challenging task. The scheduler should do the scheduling process efficiently in order to utilize the available resources completely. 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 and to minimize the response time, simultaneously eliminating a condition in which some of the nodes are over loaded while others are under loaded [13]. Effective implementation of load balancing can make cloud computing more effective and it also improves user satisfaction. In the proposed method, a honey bee foraging technique with random stealing is used for task allocation and load balancing. When tasks are allocated to the VMs, current load is calculated. If the VM becomes overloaded the task is transferred to the neighborhood VM whose load value is below threshold [14]. Honey Bee foraging technique employs decentralized load balancing methodology and task transfer will be carried out on the fly. The algorithm ensures performance of the system and avoid system imbalance.

3.1 Bee foraging behaviour

The artificial bee colony algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm and was proposed by Karaboga in 2005 [15]. The algorithm is completely inspired by natural foraging behavior of honey bees.

3.1.1 Initialization process

During initialization process, Artificial Bee Colony (ABC) algorithm starts by correlating all the bees with arbitrarily produced food sources. Certain food sources are randomly selected by bees and their nectar amount is determined. These bees come onto the hive and share the information with bees waiting in dance area [16]. Each food source is generated as follows:

$$X_{ij} = X_{\min j} + rand[0,1](X_{\max j} - X_{\min j}) \qquad (1)$$

Here $x_{\min j}$ and $x_{\max j}$ are bounds of x_i in jth direction and $rand[0,1]$ is a uniformly distributed random number in the range [0, 1].

3.1.2 Employed bee phase

Employed bees stay on the food source and provide the neighborhood of the source in its memory. After sharing the information in the dance area, employed bees go to food source visited by its previous cycle and choose new food source by using the information in the neighborhood. Then onlooker prefers a food source depending on nectar information provided by employed bees.

3.1.3 Onlooker bee phase

Onlooker bees get the information about food sources from employed bees in hive and select one of the sources. Onlooker bee is waiting for a dance to choose a food source. Waggle/tremble/Vibration dances are performed by the bees to give an idea about quality and quantity of food and its distance from bee hive.

3.1.4 Scout bee phase

Scout bee carried out random search. When the nectar source is abandoned by the bees, a new food source is randomly determined by a scout bee. Probability of selecting food source [17] is defined by the equation (2) as follows:

$$P_i = Fit_i \Big/ \sum_{n=1}^{S_n} Fit_i \tag{2}$$

Where Fit_i will be the Fitness value of food source i and s_n will be the total number of food sources. If the solution is represented by a food source does not improve for a particular number of iterations, then the employed bee becomes scout.

3.2 Random stealing

With Random stealing [3], tasks are stolen from a random Virtual machine when a VM is idle. It thus saves the idle time of the processing elements in the Virtual machine. The proposed algorithm with random stealing makes multiple attempts to steal job from other Virtual machines. If there are N Virtual machines, the algorithm will make N-1/N attempts on an average to steal a job. Job stealing is done only when a VM is idle.

3.3 Enhanced honey bee inspired load balancing algorithm

In Honey bee behavior inspired load balancing (HBB_LB), a method of task allocation and load balancing is put forth. It ignores the idle condition of virtual machine and thus leads to the wastage of processing time. Hence the global optimal solution cannot be found out within a short span of time. In the proposed method, an enhanced honey bee foraging technique with random stealing is used for task allocation and load balancing. When tasks are allocated to the VMs, current load is calculated. If the VM becomes overloaded the task is transferred to the neighborhood Virtual machine whose load value is below threshold [14]. Honey Bee foraging technique employs decentralized load balancing methodology and task transfer will be carried out on the fly. The algorithm thus ensures performance of the system and avoid system imbalance.

Random stealing (RS) is an efficient load balancing technique which steals jobs from a random Virtual machine when it is idle. It thus saves the idle time of the processing elements in the VM. The RS algorithm makes multiple attempts to steal job from other Virtual machines. If there are N Virtual machines, the algorithm will make N-1/N attempts on an average to steal a job. Job stealing is done only when a VM is idle.

Fig.1 illustrates the flow diagram of honey bee inspired load balancing of tasks with random stealing. After allocating tasks to fittest virtual machine, load balancing is done among the available virtual machines using random stealing technique. When a job request comes, the scheduler initializes job parameters and finds the deviation of load. The probability value of deviation is checked for confinement within the range of 0 to 1. All Virtual machines with probability of deviation within the given range are considered as underloaded and outside the given range is marked as overloaded VM. After grouping the VMs, ACAP value of underloaded virtual machines are compared with the ECAP value of tasks. Virtual machines with ACAP less than or equal to ECAP are marked as fittest and jobs are allocated to it [18].

After job allocation to clusters, some clusters may remain underutilized. This leads to the wastage of processor time. To avoid this, the proposed honey bee algorithm with random stealing sets two thresholds. When a VM's load is less than or equal to threshold value $T_{RS\_LOW,}$ then it steal tasks from any overloaded VMs. Jobs are stolen from VMs with system load > T_{RS_HIGH} and are allocated to free Virtual machines. This type of load balancing techniques with random stealing can reduce the idle time of virtual machines.

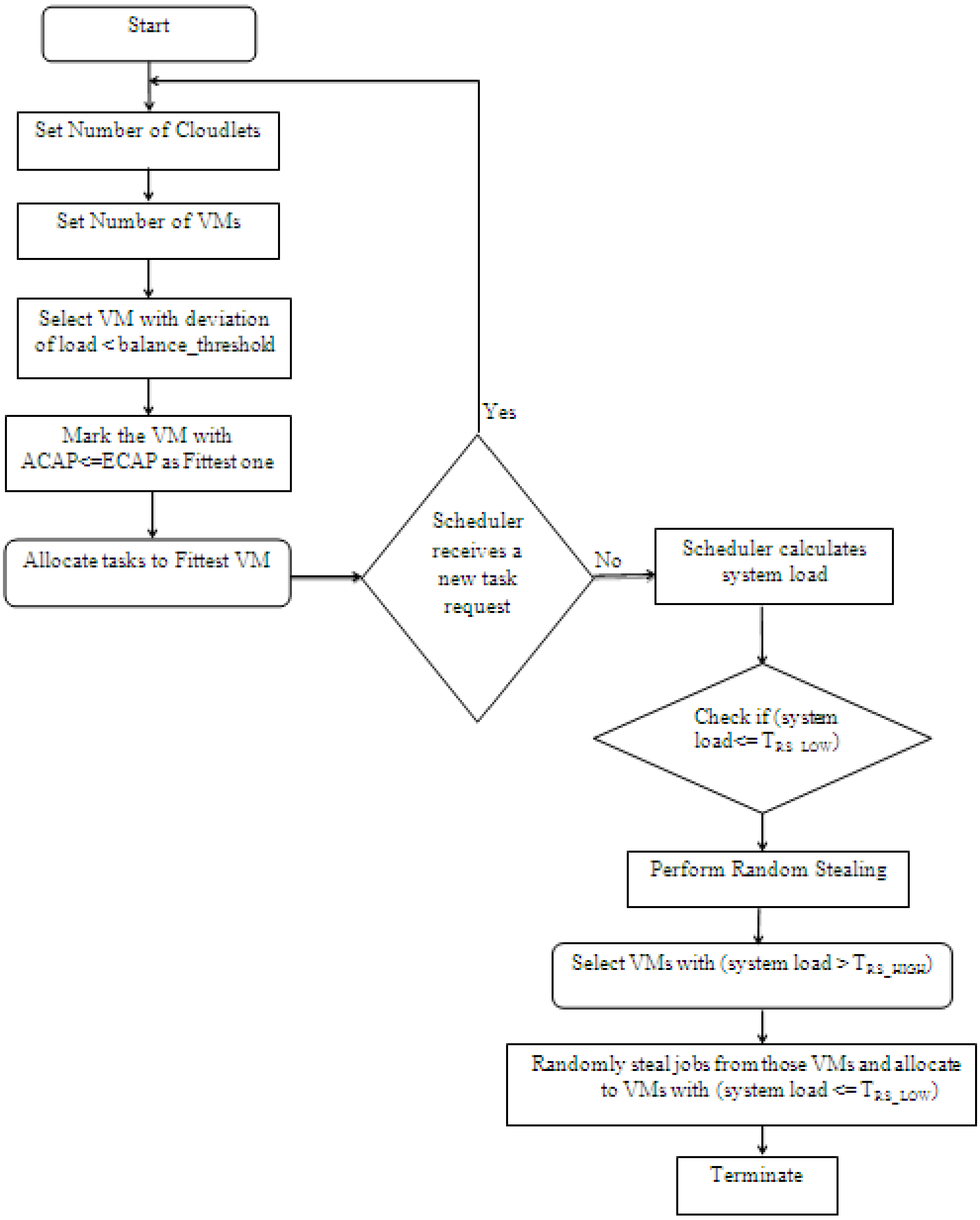

Fig.1 Flow Diagram of Honey bee inspired load balancing of tasks with random stealing

3.4 Algorithm

Step 1 Get the available virtual resources from data center.ie, VM1, VM2… VMm

Step 2 Submit the list of tasks T=T1, T2…Tn by the user.

Step 3 When a request comes, the scheduler finds the Expected computing capacity for tasks

$$ECAP = \sum_{i=1}^{n} mips/n$$

Where mips is million instructions per second and n is the total number of tasks.

Step 4 Compute the average computing capacity for each task using the equation,

$$ACAP = 1/m\sum_{i=1}^{m} ECAP \qquad \text{m: Number of VMs}$$

Step 5 Find the load on a VM

$$LVM_i = Tasklength_i / Servicerate_i$$

Step 6 Compute the average system load

$$ASL = 1/m\sum_{i=1}^{m} LVM_i$$

Step 7 The deviation of Load, σ is found out as,

$$\sigma = ASL - LVM_i$$

Step 7.1 The probability value is checked for confinement within the range 0 to 1 as,
If (0< P(σ)<1)
Underloaded_list[]= VM_i
else
Overloaded_list[]= VM_i

Step 8 Select Underloaded VMs and compare its Average computing capacity with Expected computing power of tasks.
Step 8.1 Check if (ACAP< =ECAP), then
VMs are marked as Fittest and tasks are allocated to it
Step 9 After task allocation to VMs, some VMs remains underutilized. This leads to wastage of processor time
Step 9.1 Check if (system load < = T_{RS_LOW})
Perform Random Stealing
Step 9.1.1 Select VMs with (system load > T_{RS_HIGH})
Step 9.1.2 Randomly steal jobs from those VMs and allocate to VMs with system load < = T_{RS_LOW}
Step 9.2 If there are 'N' VMs, the algorithm will make N-1 / N attempts on an average to steal a job.
Step 10 Stop

IV. Experimental results

In cloud computing, experimenting in real time is practically impossible as it needs excessive cost, security etc. Hence a good simulator can be used for experimenting purpose. CloudSim is a generalized simulation framework that allows modelling, simulation and experimenting the cloud computing infrastructure and application services [19]. The main advantages of CloudSim are time effectiveness, it requires very less time and effort to implement Cloud based application provisioning test environment [20].

Performance of our algorithm for load balancing with random stealing is evaluated by comparing it with the performance of existing load balancing algorithm.

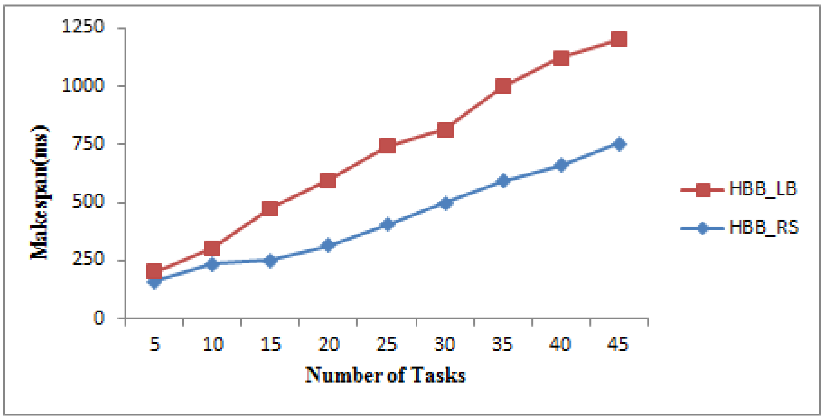

Fig.2 Comparison between algorithms based on Makespan

The comparison of Makespan for load balancing using honey bee inspired load balancing algorithm (HBB_LB) and enhanced honey bee algorithm with random stealing (HBB_RS) is illustrated in Fig.2 Makespan can be defined as the overall task completion time. We denote completion time of task T_i on VM_j as CT_{ij}.

$$Makespan = \max\{CT_{ij} \mid i \in T, i = 1,2,....n \ and \ \ j \in 1,2,...m\} \quad (3)$$

The X-axis represents the number of tasks and Y-axis represents Makespan in milliseconds. With load balancing using honey bee inspired load balancing (HBB_LB), the makespan is reduced considerably. When number of tasks increases, the difference in makespan is more and enhanced honey bee algorithm with random stealing is more efficient.

The comparison is also made in terms of response time. Fig.3 shows the response time in milliseconds for HBB_LB and HBB_RS. The X-axis represents number of tasks and Y-axis represents the response time in milliseconds. 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. From this graph, it is clear that enhanced honey bee foraging technique with random stealing is more effective and gives better performance.

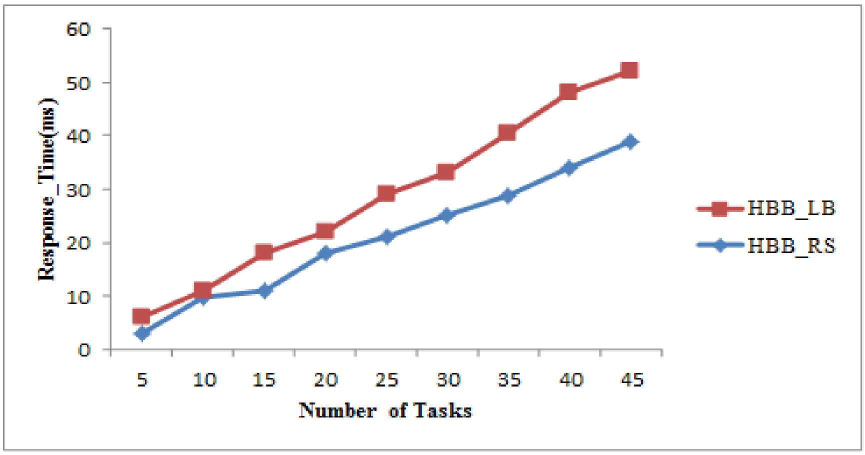

Fig.3 Comparison on Response time

Fig.4 illustrates the degree of imbalance in load in terms of number of tasks. The x-axis represents the number of tasks and Y-axis represents the imbalance degree. Imbalance degree is defined in equation (4)

$$Degree\ of\ \ imbalance = T_{HIGH} - T_{LOW} / T_{AVG} \tag{4}$$

Where T_{HIGH} is the highest task, T_{LOW} is the lowest task among all the virtual machines and T_{AVG} is the average task of virtual machines. From Fig. 4, it is clear that imbalance degree is less for enhanced honey bee inspired load balancing of tasks with random stealing and hence its performance is more when compared to normal honey bee foraging load balancing technique.

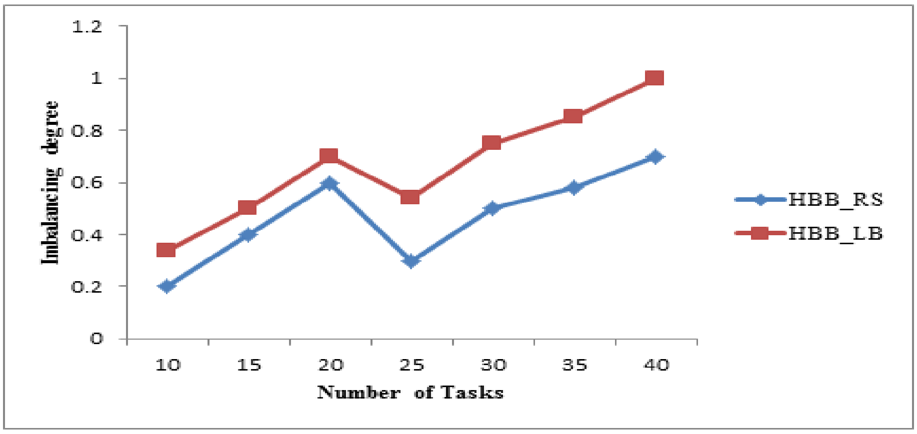

Fig.4 Comparison on Imbalancing degree

An important parameter used in this work to analyze the load balancing strategy of the proposed algorithm is the average resource utilization and is expressed in percentage.

$$\mathrm{Re}\,source\ utilizatio\,n = VM\ \ demand\,/\,Number\ of\ \ tasks \tag{5}$$

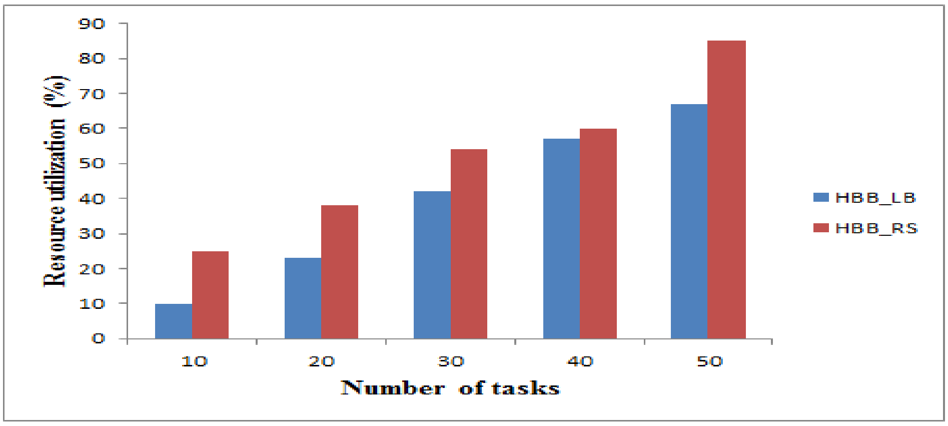

Fig.5 Comparison based on Resource utilization

Fig.5 shows the resource utilization rate of proposed method with random stealing which is relatively high when compared to existing honey bee load balancing method. With random stealing method, tasks are stolen from a random Virtual machine when a VM is idle. It thus saves the idle time of the processing elements in the Virtual machine.

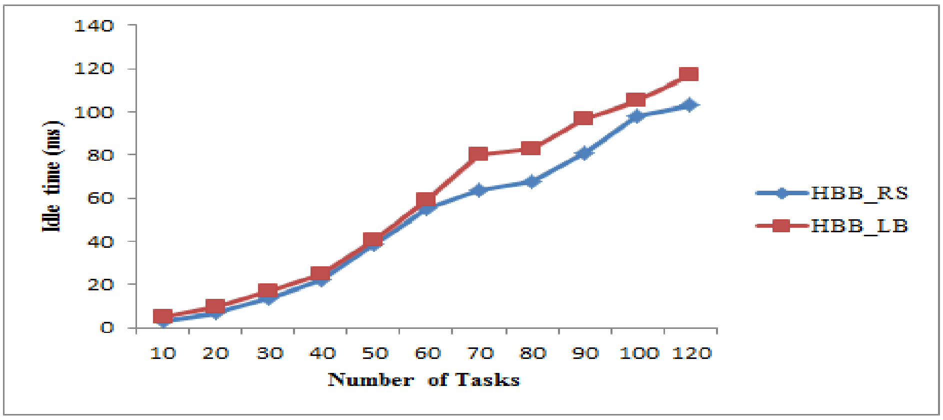

Fig.6 Comparison of algorithms based on idle time

Idle time is the time between the times at which task is arrived on a virtual machine and time of task to be assigned to one certain virtual machine. The comparison is made in terms of Number of tasks and the idle time and the results are shown in Fig. 6.

All these results verify that the proposed honey bee foraging technique with random stealing performs better than the existing honey bee inspired load balancing algorithm.

V. Conclusion

Load balancing of tasks in virtual machine plays an important role in cloud environments. In this paper, an enhanced honey bee behavior inspired load balancing with random stealing is proposed. The existing load balancing algorithms with bee foraging technique ignores the idle condition of a Virtual machine even after the completion of tasks assigned to it. The modified bee foraging technique with random stealing method ensures that load in the overall system is balanced and reduces the idle time of a processor. When tasks are allocated to a VM, current load in that VM is calculated. If the VM becomes overloaded, the task is transferred to the neighborhood Virtual machine whose load value is below threshold. With Random stealing, Jobs are stolen from a random VM when a Virtual machine is idle. It thus saves the idle time of the processing element in a Virtual machine. In the future we plan to extend this research for energy aware scheduling.

References

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

[2]. Tejinder Sharma, Vijay Kumar Banga, “Efficient and Enhanced Algorithm in Cloud Computing”,IJSCE,ISSN:2231-2301,2013

[3]. Joshua Samuel Raj, Hridya K.S,V. Vasudevan, “Augmenting Hierarchical Load Balancing with Intelligence in Grid Environment”, International Journal of Grid and Distributed Computing Vol. 5,No. 2, pp. 9- 8, 2012

[4]. Guopu.Zha, Sam Kwong , “Gbest-guided Artificial bee Colony algorithm for numerical function optimization”, Elsevier 3166-3173,2010

[5]. Ayed salman, Imtiaz Ahmad, “Particle swarm optimization for task assignment problem”, Microprocessors and Microsystems 26,363-371,2002

[6]. Belabbas Yabougi, Meriem Meddeber, “Distributed Load balancing model”, ARIMA journal, Vol 12,pp 43-60,2010

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

[8]. Guopu.Zha, Sam Kwong ,“Gbest-guided Artificial bee Colony algorithm for numerical function optimization”, Elsevier 3166-3173,2010

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

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

[11]. Shagufta khan, Niresh Sharma, “Ant Colony Optimization for Effective Load Balancing In Cloud Computing”,IJARCSSE,Vol.4, Issue 2,2014

[12]. Yun-Han Lee, Seiven Leu, Ruay-Shiung Chang, “Improving job scheduling algorithms in a grid environment”, Elsevier, 991–998,2011

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

[14]. Thilo Kielmann, Henri E. Bal, Jason ,"Programming environments for high-performance Grid computing: the Albatross project", Elsevier 1113–1125,2010
[15]. D. Karaboga, " An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey,2005
[16]. A.M. Bernardino, E.M. Bernardino, J.M. Sánchez-Pérez, M.A. Vega-Rodríguez, J.A. Gómez-Pulido, "Efficient Load Balancing Using the Bees Algorithm", Springer-Verlag Berlin, Heidelberg, ISBN: 978-3-642-21826-2,2011
[17]. Vivek Kumar Sharma, Rajani Kumari," Improved Onlooker Bee Phase in Artificial Bee Colony Algorithm", International Journal of Computer Applications (0975 – 8887),2014
[18]. Ruay-Shiung Chang. Chun-Fu Lin, Jen-Jom Chen, "Selecting the most fitting resource for task execution", Elsevier, 227–231,2011
[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 and evaluation of resource provisioning algorithms, Software: Practice and Experience 41 ,23–50,2011
[20]. R.N. Calheiros, R. Ranjan, C.A.F.D. Rose, R. Buyya, "CloudSim: a novel framework for modeling and simulation of cloud computing infrastructures and services", Computing Research Repository, vol. abs/0903.2525, 2009

AUTHORS

Ms. Anna Baby received B.Tech degree in Computer science and Engineering from Sree Narayana Gurukulam College of Engineering under Mahatma Gandhi University, Kottayam in 2013. She is currently pursuing M.Tech in software Engineering in Karunya University. Her research interests include Cloud computing, Software Engineering and so forth.

Dr. Joshua Samuel Raj received his B.E degree in Computer Science and Engineering from PETEC under Anna University and M.E degree in Computer Science and Engineering from Jaya College of Engineering under Anna University in 2005 and 2007, respectively. He received Ph.D degree in the area of Grid scheduling under Kalasalingam University in 2015. He is currently an Assistant Professor in Computer Science and Engineering at Karunya University, Coimbatore, India. His research interests include Grid computing, Mobile Adhoc Networking, Multicasting, Cloud Computing and so forth.