# *Nature Inspired Preemptive Task Scheduling for Load Balancing in Cloud Datacenter*

G.Shobana[1]
[1]PG Scholar
Department of CSE
Kongu EngineeringCollege,Perundurai,
Erode-638052, Tamilnadu, India.
[1]shobana.csc11@gmail.com

M.Geetha[2]
[2]Assistant Professor
Department of CSE
KonguEngineering College, Perundurai,
Erode-638052, Tamilnadu, India.
[2]shri.aug28@gmail.com

Dr.R.C.Suganthe[3]
[3]Professor
Department of CSE
KonguEngineering College, Perundurai,
Erode-638052, Tamilnadu, India.
[3]suranthe_rc@kongu.ac.in

***Abstract*- Scheduling of the tasks in the cloud computing is assigning tasks to the particular machine. Load balancing of preemptive independent tasks on the virtual machine (VMs) is an important aspect of the task scheduling in clouds. Load balancing aims to optimize resource, minimize response time, maximize throughput and avoid overload of any of the resources. If the load balancing is not properly done, then it may lead to the condition where some virtual machines may get overloaded while others may become underloaded or even idle. Due to unbalanced load, the tasks are not completed in time and as a result the user satisfaction is not achieved. The proposed algorithm almost reduces makespan throughpreemptive task scheduling which follows the foraging behavior of honey bees. This algorithm considers the priority of the tasks and their expected remaining completion time with an aim to maximize throughput and minimize latency. Our approach improves the response time of users by effective utilization of available resources.**

***Keywords- Load balancing; preemptive task scheduling; behavior of honey bee; cloud computing***

## I.INTRODUCTION

Cloud Computing is a model for delivering information technology services in which resources are retrieved from the internet through web-based tools and applications. Software packages and information are hosted in cloud in which thousands of computers are interconnected. It allows more efficient computing by centralized storage, memory, processing and bandwidth. Resources providers' main concern is to reduce the cost of investment and maximize the fulfillment of the user demands. However task scheduling is one of the activities performed in order to gain maximum profit and reducing overall completion time of the tasks.

Virtual Machine (VM) has to be run in parallel to reduce response time of the user. Whenever more than one task is assigned to one or more VMs that run the tasks simultaneously.This situation leads VMs in unbalanced Condition. The unbalanced load in the cloud environment leads to the condition where some resources get overloaded, while others get underloaded or even become idle. So, the improper load balancing may significantly affect response time and throughput of the cloud system. There are basically two kinds of load balancing algorithm 1.Static load balancing 2.Dynamic load balancing

Static algorithms work properly only when node have low variation in the load and do not take into account the previous state or behavior of node. Therefore these algorithms are not suitable for cloud environment where load will be varying at varying times. Dynamic techniques are highly successful for load balancing of tasks among heterogeneous resources.

In cloud environment whenever VMs are heavily loaded the task has to be removed from overloaded VMs and gets placed in one of the underloaded VMs for balancing load in same datacenter. Our approach uses nature inspired algorithm as modeling behavior of honey bee how they find and reap food source. Balancing load in the datacenter will increase throughput of the system.

Rest of this paper is organized as follows: Section 2 discusses about the related works. Section 3 describes the nature inspired load balancing techniques and how it relates to the proposed technique. Section 4 presents performance evaluation of the algorithms. Finally we conclude this paper highlighting the contributions and future enhancements in Section 5.

## II. RELATED WORK

Workload of the servers in the data centers are changing dynamically which in turn affects the workload of the Virtual Machines (VMs) in the servers. Due to this dynamic change in

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

the workload of the VMs, a situation arises where some VMs get overloaded while others may be underloaded or may even remain idle. This unbalanced condition will degrade the performance of the data centers which will reduce the economy of the IT industry. Hence load balancing is remains a challenging issue in cloud computing environment. Various load balancing techniques exists to improve the system performance. Those techniques are studied and discussed in this chapter.

Pham et al [9] suggested, a new population based search algorithm called the Bees Algorithm (BA). The algorithm imitates the food foraging behavior of swarms of the honey bees. In the basic version, the algorithm performs a kind of neighborhood search combined with random search and can be used for both combinatorial optimization and functional optimization. In Bees colony, scout bees are used for searching food sources by moving randomly from one source to other sources. After finding food resource with quality threshold the scout bees return to hive and dance in the dancing floor. Dancing is essential for colony communication and it contains information regarding food sources like direction from the hive, quality and quantity. Bees that have the highest fitness's are chosen as selected bees and sites visited by them are chosen for neighborhood search. This is helpful for choosing the optimal solution. One of the drawbacks of the algorithm is the number of tunable parameters used. However, it is possible to set the parameter values by conducting trial values.

Load balancing algorithms can be defined based on the implementation of the following policies [12]:

- Transfer Policy / Transfer Strategy**:** The part of the dynamic load balancing algorithm which selects a job fortransferring from a local node to a remote node.
- Selection Policy / Selection Strategy**:** It specifies the processes involved in the load exchange (process or matching)
- Location Policy / Location Strategy**:** The part of the load balancing algorithm which selects a destination node for a transferred task.
- Information Policy / Information Strategy: The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is referred to as Information policy or Information strategy.

In [6], suggested Ant Colony Optimization (ACO) algorithm for load balancing. In this approach, incoming ants update the entries in the pheromone table of a node. For instance, an ant traveling from source to destination will update the corresponding entry in the pheromone table. If an ant is at a choice point when there is no pheromone, it makes a random decision. Here tasks are assigned to server with non-preemption scheduling so waiting time of high priority tasks are increased.

In [4], proposed Genetic algorithm (GA) for task scheduling in cloud computing. By considering User satisfaction and provider's profits task scheduler determines where to allocate each task in every scheduling cycle. The tasks to be given as input are sorted before scheduling them. Tasks are classified by time and budgetary constraints. Fitness function generates a fitness value for each task by using roulette wheel selection. The function iterates genetic operations to have the best task scheduling. In [11], proposed dynamic load balancing with main characteristics such as it uses task level load balancing, privileges local tasks transfer to reduce communication costs. This system transforms the grid to tree structure independent of grid topological structure complexity.

In [3], proposed adaptive load sharing in homogeneous distributed systems. Adaptive policies collect very small amounts of system state information for adaptive load sharing. Transfer policy is combined with location policy such as threshold and random policy. In random policy destination node is selected at random and no exchange of the state information in deciding where to transfer a task which improves system response time. Threshold policy uses small amount of information about the destination node as result it provides substantial improved performance

In [10], proposed honey bees behavior, in which nectar collection is partitioned between two groups of bees are nectar foragers and receivers. Foragers collect nectar in the field and then transfer it near the nest entrance to receivers, who work solely within the nest on processing the nectar into honey and storing it. Nectar receivers do not store information however, foragers can store. Nectar receivers move at random on the dance floor until they encounter a forager requiring to unload the nectar. They then receive nectar and become unavailable for further unloading until they unload the contents of their crop. In [7], Collective decision making in the social insects often proceeds via feedback cycles based on positive signaling. Negative signals have been found in a few contexts in which costs exist for paying attention to no longer useful information. It suggests that the stop signal, by acting as a balance to the waggle dance, allows colonies to rapidly shut down attacks on other colonies. The key adaptations are the costs of attacking the colony are becoming very strong enough to defend itself are significant.

## III. NATURE INSPIRED LOAD BALANCING TECHNIQUES

The system performs load balancing with preemptive scheduling. Load balancing mechanism helps to distribute the

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

load among the machines to avoid unbalanced load and idleness of the machines. In our technique load balancing is done by foraging behavior of honey bees.

In bee hives there are scout bees and onlooker bees. Scout bees search for food source and collects information about the nectar and check whether nectar source is saturated or not. Scout bees come back to beehives to advertise this using a dance called waggle / tremble / vibration dance. This dance gives an idea of the quality and quantity of food and also its distance from the beehive. Onlooker bees follow the scout bees to the location of the food sources and begin to reap it. They then return to the beehive and do a dance to the other bees in the hive giving an idea of how much food is left in the food sources.

Tasks removed are considered as honey bees and UVM as food source to honey bees. VMs are grouped based on load such as UVM, OVM and BVM. When a task is removed from VM under OVM group, it has to finds suitable VM in the UVM group for allocating task

Load balancing techniques are effective in reducing the makespan and response time. Makespan can be defined as the overall completion time .We denotes completion time of task $T_i$ on $VM_j$ as $CT_{ij}$ .

Response time is defined as the amount of time taken between submission of a request and the first response received. The reduction in waiting time is helpful in improving responsiveness of the VMs. Tasks migration is defined as number of tasks migrated from one virtual machine to other virtual machine for balancing load in datacenter.

*A. Load Balancing Decision*

Let VM = {$VM_1$, $VM_2$, $VM_3$…,VMn} be the set of m virtual machines which should process n tasks represented by the set T = {$T_1$, $T_2$, $T_3$…, Tn}

Capacity of a single VM is given in (1)

$$C_j = VM_{mipsj} + \mathrm{VM}_{bwj} \qquad (1)$$

Where $VM_{mipsj}$ is million instructions per second of $VM_j$, $VM_{bwj}$ is the communication bandwidth ability of $VM_j$.

Overall capacity (Capacity of all VMs) is given in (2)

$$C = \sum_{i=1}^{m} C_i \qquad (2)$$

Total length of the tasks that are assigned to a VM is considered as the load ona single VM is given in (3)

$$L_{VM_{i,}t} = \frac{N(T,t)}{S(VM_{i,}t)} \qquad (3)$$

Where N(T,t) is number of tasks at time t on service queue of $VM_i$, $S(VM_{i,}t)$ is Service rate of $VM_i$ at time t.

Load of all VMs in a data center is given in(4)

$$L = \sum_{i=1}^{m} L_{VM_i} \qquad (4)$$

Processing time of a VM is calculated by (5)

$$PT_i = \frac{L_{VM_i}}{c_i} \qquad (5)$$

Processing time of all VMsis given in (6)

$$PT = \frac{L}{C} \qquad (6)$$

Load of all VMs is divided by capacity of all VMs

Based on the values of the above parameters the scout bee checks whether the host is in a balanced state or not. This is done by calculating the standard deviation of load are given in (7)

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(PT_i - PT)^2} \qquad (7)$$

If the $\sigma$ value falls within the threshold condition set (TS=0-1), then the host is said to be in a balanced state, otherwise it is said to be unbalance state.

*B. VM Grouping*

The load balancing algorithm starts by first grouping the VMs based on their loadas Overloaded VMs (OVM), Underloaded VMs (UVM) and Balanced VMs (BVM). Sort VM in OVM set by descending order and UVM set by ascending order. Task which is removed from overloaded VM set has to be

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

placed in one of the underloaded VMs based on the load. In the proposed system, this task is considered as a honey bee and underloaded VMs are considered as the destination of the honey bees. The updates information about the following parameter such as load on a VM, load on all VMs, number of tasks in each VM, the number of VMs in each VM group and task priorities in each VM are spread across the onlooker by the employee bees.

*C. Preemptive Scheduling*

The load balancing can be done with preemptive scheduling of the tasks.Whenever high priority task is removed it has to be submitted to VMs it should consider VM that has minimum number of high priority tasks so that the particular task will be executed earlier and medium priority is removed means it should consider both minimum number of high and medium priority of the tasks.

Preemption takes place only when priority of removed task from overloaded VM is higher than running task and also the remaining expected completion time of the removed task is less than the running task. If the above condition is satisfied then running task is preempted and state of preempted task is stored in process control block. Incoming task is executed in the appropriate VM. After completing the execution of the incoming task, task which is interrupted will restore its status and starts executing from status where it has been stopped. When a task is allocated to VM, it will update information like number of tasks assigned, priority of VM and also update sets OVM, LVB, and BVM. Once when threshold value of VMs is reached then that VM is placed into the balanced set. Load balancing is successful once all the VMs are moved to balanced VM set. Due to preemption of the tasks the higher priority and less remaining expected completion time of the tasks are executed early than other tasks.

For each task T in VMs find machine $VM_{d\,\epsilon}$ UVM such as T is preemptive.

If (Incoming task priority > =priority of task running on VM

If (Remaining expected completion time of incoming task <Task currently running on VM)

Save the state of currently running task and Preempt it

Allocate incoming task to VM and execute it

Return.

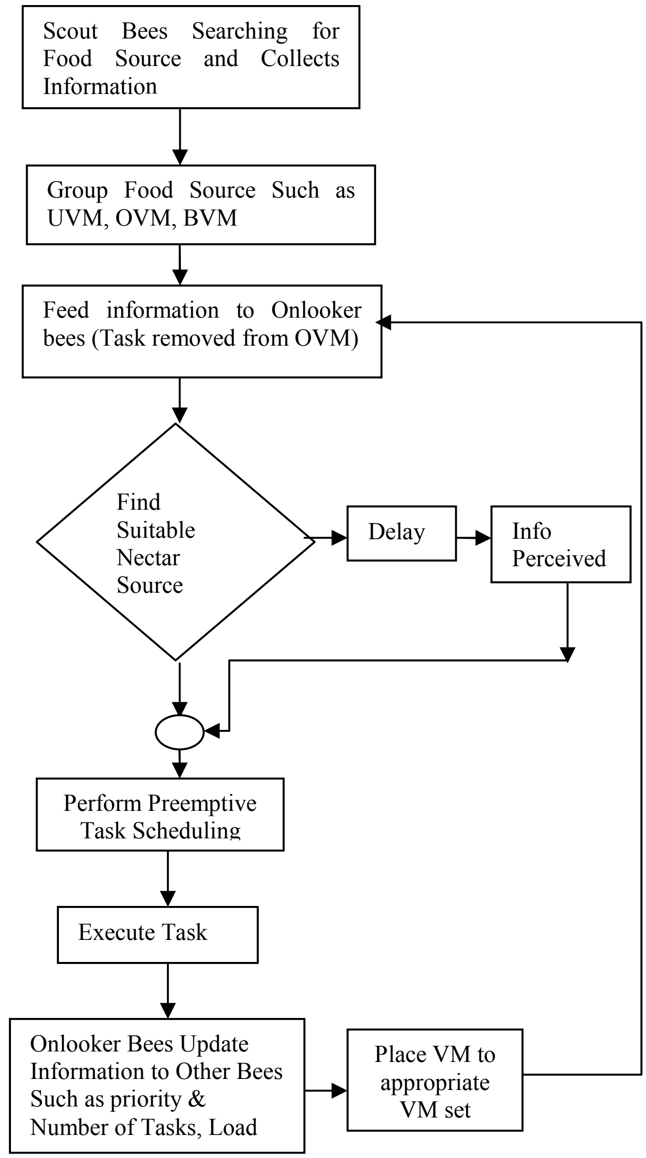


**Figure 1:** Flow diagram gives the behavior of honey bee related to load balancing

*D. Nature Inspired Algorithm*

1. Scout bees searches for food source and collects information about the quality of sources calculating load and capacity of all nectar sources.
2. Check nectar source is balanced or not.
   If (nectar source< = TS)
   System is balanced

Exit.

Else it is not balanced.

a. Scout bees group VMs based on load as UVM, OVM and BVM
b. Scout bees update information to the onlooker bees

Sort VMs in OVM by descending order
Sort VMs in LVM by ascending order
While LVM ≠ ϕ and OVM ≠ ϕ
For s=1 to OVM
Sort tasks in VMs by selection criteria (priority, remaining Expected completion time)

c. Preemptive scheduling of the tasks
d. Onlooker bees update the information such as load, no of tasks on VMs and VMs set to other bees in bee hive
e. Sort OVM, UVM and BVM.

Here we define three sets based on load of the VMs. They are

OVM – Overloaded VM
UVM – Underloaded VM
BVM – Balanced VM

## IV. EXPERIMENTAL RESULT

The tool used for simulation of the algorithm is Cloudsim and Eclipse for running simulations.We have extended the classes of cloudsim simulator to simulate our algorithm.

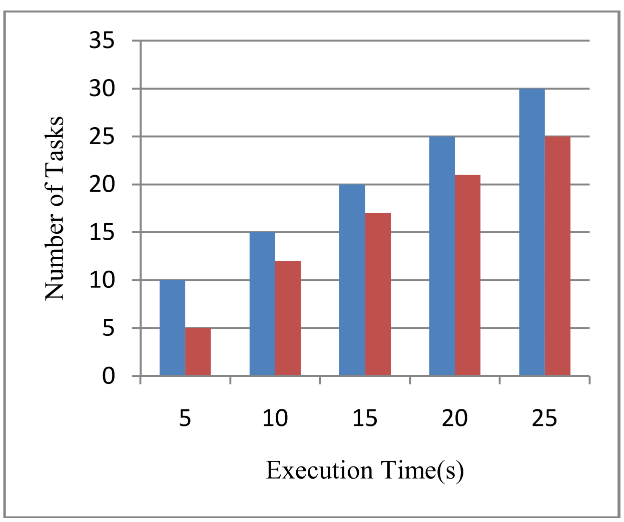


**Figure 2**: Comparison of number of task and execution time of the task

Whenever new task is submitted to VMs, capacity and load of VMs is calculated, VMs are grouped based on load. If we done load balancing of the tasks with preemption scheduling. The result of the algorithm may be obtained as in Figure 2. Execution times of the tasks arereduced. As the result makespan of the tasks are reduced.

## V. CONCLUSION

In cloud environment workload of virtual machines are changing dynamically due to which some VMs get overloaded while others may be underloaded or even become idle. This unbalanced load results in degradation of the system performance. Performance of the system can be improved by bringing the overloaded VMs to balanced state and effectively utilizing the underloaded and idle VMs. This preemptive task scheduling follows the foraging behavior of honey bees for allocating VMs to the tasks. The priority of the independent tasks is considered while preempting the tasks to achieve minimum makespan and maximum throughput. When preempting tasks remaining expected completion time of the tasks and priority is considered so that the waiting time of the tasks can be reduced this will also lead to improve the performance of the datacenters. In future we plan to extend this kind of load balancing for workflows with dependent tasks and to improve this algorithm by considering other parameters.

## REFERENCE

1. Buyya.R, Ranjan.R, CalheirosR.N , “Modeling simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities,”Proc of the 7th High Performance Computing and Simulation Conference, Leipzig, Germany 2009 .
2. DhineshBabu .L.D and Krishan.P.V, “Honey Bee Behavior inspired load balancing of tasks in cloud computing environments” Applied Soft Computing 2013
3. Eager.D.L, Lazowska E.D, Zahorjan.J, “Adaptive load sharing in homogeneous distributed systems,” in The IEEE Transactions on Software Engineering 2012, pp 662–675.
4. Jang .S.H, Kim.T.Y, Lee.J.S, “The study of genetic algorithm based task scheduling for cloud computing” Journal of Control and Automation 2012, vol.5, No.4.
5. Karatza .H.D,“Job scheduling in heterogeneous distributed systems,” Journal of system and software 1994, pp203-212.
6. Mishra.Rand Jaiswal.A, “Ant colony Optimization: A Solution of Load balancing in Cloud” in International Journal of Web & Semantic Technology (IJWesT) 2012.
7. Nakrani.S, Tovey.C , “On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers,” Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 2004, pp223–240.
8. Naimesh D. Naik , “Load Balancing in Cloud Computing: A Survey’’ IJRIT International Journal of Research in Information Technology, 2013 Volume 1, Issue 5, Pg. 188-195

9. Pham.D.T, Kog.E, Ghanbarzadeh.A, Otri.S, Rahim.S, Zaidi.M ,"The bees algorithm-a novel. Tool for complex optimisation problems,"Proc 2nd International Virtual Conference on Intelligent Production Machines and Systems 2006.
10. Vries.H.de, Biesmeijer.J.C, "Modelling collective foraging by means of individual behavior rules in honey-bee," Behavioral Ecology and Sociobiology 1998, pp109–124.
11. Yagoubi.B and Slimani.Y, "Dynamic load balancing strategy for grid computing," transactions on engineering Computing and Technology 2006, pp260–265.
12. Yagoubi.B and Slimani.Y, "Task load balancing strategy for grid computing,"Journal of Computer Science 2007, pp186–194.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE