Honey Bee Behavior Load Balancing of Tasks in Cloud Computing

Khushbu Zalavadiya
Department of Computer Science and Engineering,
Parul Institute of Technology,
Vadodara, India

Dinesh Vaghela
Department of Computer Science and Engineering,
Parul Institute of Technology,
Vadodara, India

ABSTRACT
Cloud Computing is usage of computing resources that provided services over the Internet. In cloud computing several resources are available which process incoming request. Because of random appearance of requests for task execution several virtual machines are overloaded and several virtual machines are under loaded or idle for task processing. Therefore, an Enhanced honey bee algorithm for load balancing in cloud computing is proposed.

In proposed Technique priority tasks are removed from overloaded virtual machine and they are allocated to under loaded virtual machine by considering least numbers of same priorities to those tasks, cost effective virtual machine and, least expected completion time of those tasks on that virtual machine also balance the loads of dependent tasks in pre-emptive manner. By considering least expected completion time, cost and priority at submission time of that task, it helps to produce minimum completion time, amount of waiting time of the tasks in the queue is minimal and achieve better resource utilization.

Keywords
Cloud Computing, honey bee behaviour, Load balancing, virtual machine, CloudAnalyst.

1. INTRODUCTION
Cloud computing is a vast concept. Now a day's every organization is the use of cloud computing. Without any doubt we can put it as within few years, there will be lots of user for cloud computing. During that period of time, cloud provider will need to maintain more effective load balancing and effective resource management. And it is of no worth to buy new hardware just to balance the load instead of fully utilizing current resources. Thus, an efficient load balancing system is needed to handle much load and to maximize utilization of current resource. There is several static and dynamic type of cloud computing have been proposed. Honey bee algorithm has been overviewed in this thesis. So the Honey bee algorithm for load balancing described in can also be applied for clouds. The performance of the algorithms have been studied and compared.

load balancing algorithms on which various researches have been made. Many of the algorithms for load balancing in Cloud computing is an entirely internet-based approach where all the applications and files are hosted on a cloud which consists of thousands of computers interlinked together in a complex manner. Cloud computing incorporates concepts of parallel and distributed computing to provide shared resources; hardware, software and information to computers or other devices on demand. These are emerging distributed systems which follows a "pay as you use" model [1].

The customer is interested in reducing the overall execution time of tasks on the machines. The processing units in cloud environments are called as virtual machines (VMs) [4]. In business point of view, the virtual machines should execute the tasks as early as possible and these VMs run in parallel. This leads to problems in scheduling of the customer tasks within the available resources.

The scheduler should do the scheduling process efficiently in order to utilize the available resources fully. More than one task is assigned to one or more VMs that run the tasks simultaneously. This kind of environments should make sure that the loads are well balanced in all VMs i.e., it should make sure that the tasks are not loaded heavily on one VM and some VMs do not remains idle and/or under loaded[6].

Fig 1: Cloud Computing

1.1 Load Balancing

Load balancing is use to balance load between multiple resources to get minimum makespan, improve performance, reduce response time and optimal resource utilization. figure 2 . and 3 illustrates scenario before load balancing and after load balancing. So, load balancing is the most important to balance the load in cloud computing

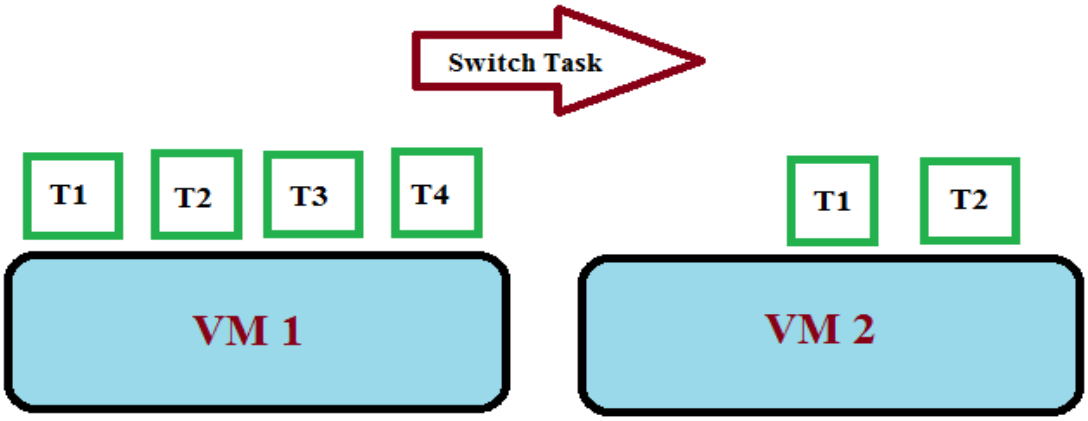

Fig 2: Load balanced before switch task

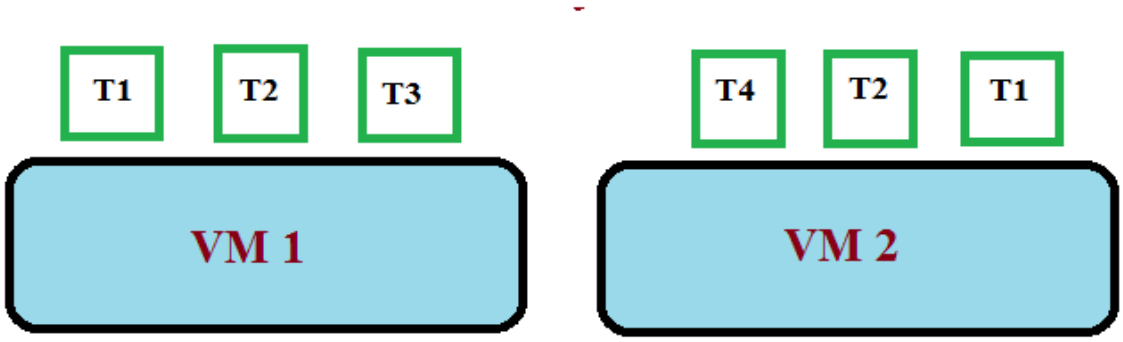

Fig 3: Load balanced after switch task

Load balancing is done at two levels

1. Resource allocation
2. Task scheduling

1 Resource Allocation

Resource provision is procedure of mapping diverse entity of cloud to resources. Resources are assigned in manner, which gives guarantee of no resources are heavily loaded and lightly loaded, and also gives guarantee that they does not suffer from some kind of waste.

2 Task Scheduling

When resources are assigned to task, then task scheduling are done. Scheduling is well-defined as plan for assigning the tasks to resources in way to obtain minimum completion and response time.

2. LITERATURE SURVEY

2.1 Response Time Based Load Balancing

Load Balancing technique that can serve the purpose of not only properly utilizing the servers but also reducing the negative impact on the user services. The existing Load Balancing techniques suffer from various issues like (i) balancing the load after a server has been overloaded, (ii) constant querying the server about availability of its resources, hence, increasing computation costs and bandwidth consumption.

This paper proposes [3] an algorithm that takes a preventive approach of Load Balancing by considering only the response time of the each request. Based on the response time, the proposed method decides the allocation of next incoming request. The approach is not only dynamic in nature, but eliminates the need of unnecessary communication of the Load Balancer with the VMs by not querying about their current resource availability. This model only takes into account the response time which is easily available with the Load Balancer as each request and response passes through the Load Balancer, hence eliminates the need of collecting further data from any other source hence wasting the communication bandwidth.

2.2 Cost-Conscious scheduling

In this paper [4], author focus on the use of cloud resources for dispatching large graph processing tasks. The key component of this framework is a cost-conscious scheduling

heuristic, called CCSH, which is an extension of Heterogeneous Earliest Finish Time (HEFT).

CCSH first constructs a priority list of tasks and then assigns the task with the highest priority value to the cost-efficient virtual machine in a cloud setting. The comparison study, based on randomly generated large graphs and a real-life astronomy application model, demonstrates that CCSH outperforms HEFT by exhibiting significant monetary cost savings at a reasonable increase in overall execution time.

2.3 Cost-efficient task scheduling

Cost- efficient task-scheduling algorithm using two heuristic strategies [5].

2.3.1 Pareto optimal scheduling heuristic (POSH):

HEFT assigns a priority to each task in the DAG and then maps the task with the highest priority to the VM that minimizes the earliest finish time. POSH uses both the running time and the monetary cost to modify the last step to map the task with the highest priority to the most cost-efficient VM based on Pareto dominance.

POSH involves the following three phases:

2.3.1.1 Weighting Phase:

The weights assigned to nodes are calculated based on the predicted execution time of the tasks and the weights assigned to edges are calculated based on predicted time of the data transferred between the VMs.

2.3.1.2 Prioritizing Phase:

Create a sorted list of tasks organized in the order how they should be executed.

2.3.1.3 Mapping Phase:

Assign the tasks to the resources based on Pareto dominance.

2.3.2 Slack time scheduling heuristic (STSH):

To reschedule non-critical tasks for reducing monetary costs, we need to compute the slack time for the non-critical tasks.

2.4 Ant colony optimization algorithm

A novel ant colony based algorithm [6] to balance the load by searching under loaded node. To solve the problem of the load balancing author chooses the behavior of the ants during search of food. Since ants possesses with a very intelligent way for finding the food by the method of shortest distance author took that into the consideration. Ants uses the principle of trail lying by dropping the pheromones on the ground by stopping on some points on -the way through there gesture which secreted by pheromone gland. That put a trail for the ants to come back to their colony after they found food.

2.5 Modified Throttled Algoritham

This algorithm [7] focuses mainly on how incoming jobs are assigned to the available virtual machines intelligently. Modified throttled algorithm maintains an index table of virtual machines and also the state of VMs similar to the Throttled algorithm. There has been an attempt made to improve the response time and achieve efficient usage of available virtual machines.

2.6 Cost Effective Load Balancing

In the proposed method [7], take the cost of VMs and expected running time of tasks for selecting the most optimal VM. Using the concept of Pareto dominance relation, selects optimal VM by comparing the cost of executing a task on one VM with that of all other VMs and expected running time of that task in one VM with that in all other VMs. Finally select a VM which has minimum value of minimization function and assign the task to it. The minimization function is computed based on running time and monitory cost. Thus the method not only reduces the execution time but also reduces the cost of using VM instances.

3. PROPOSED WORK

In proposed algorithm high priority task is removed from machine and allocated to those virtual machines which have less number of similar (high) priorities of jobs (tasks) as well as least expected completion time of that task on that machine and also balance the loads of dependent tasks in pre-emptive manner. By considering minimum expected completion time and priority at submission time of that task, it produced minimum completion time and better resource utilization.

The following are the steps and flow diagram for the proposed work.

3.1 Steps of Proposed Work

1. Find capacity and load of all VMs. Then check the value of & determine whether the system is balanced or not. If balanced then exit.
2. Take load balancing decision based on load. If load > max. capacity then exit.
3. Group VMs based on loads.
4. Compute priority of tasks by considering QoS parameters like deadline, task length, MIPS value of VM.
5. Perform Load balancing
   - Find supply of each VM in
   - Find demand of each VM in OVM
   - Sort VMs in OVM
   - Sort VMs in UVM
   - If there are more than one VM in UVM
   - VMd=Call Pareto optimal Scheduling
6. Update the no. of tasks assigned to VMd
7. Update sets OVM, BVM and LVM

4. CONCLUSION AND FUTURE ENHANCEMENT

In existing algorithm, take the cost of VMs and expected running time of tasks for selecting the most optimal VM. Therefore, sometime it does not cost efficient and not better resource utilization.

In proposed algorithm, priority tasks are removed from machine and allocated to those machines that have less number of same prioritize tasks as well as least expected completion time of that task on that machine. By considering minimum expected completion time and priority at submission time of that task, and also balance the loads of dependent tasks in pre-emptive manner. It helps to produce minimum completion time and better resource utilization. So in future try to consider as many as possible QoS factors.

5. REFERENCES

[1] Mayanka Katyal, Atul Mishra" A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment" International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013J.

[2] Dinesh Babu L D, P.Venkata krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments" Applied Soft Computing Volume 13 Issue 5, Elsevier, May 2013 ,Pages 2292-2303.

[3] Agraj Sharma, Sateesh K. Peddoju "Response Time Based Load Balancing in Cloud Computing" 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT).

[4] J. Li, S. Su, X. Cheng, Q. Huang Z. Zhang "Cost-conscious scheduling for large graph processing in the cloud" IEEE 2011, 13th International Conference on High Performance Computing and Communications (HPCC), pp. 808–813.

[5] Sen Su, Jian Li,Kai Shuang,Jie WangFröhlich, B."Cost-efficient task scheduling for executing large programs in the cloud".Parallel Computing, Volume 39, March 2013, pages 177-188

[6] Santanu Dam, Gopa Mandal, Kousik Dasgupta, and Paramartha Dutta "An Ant Colony Based Load Balancing Strategy in Cloud Computing" 2014 Advanced Computing, Networking and Informatics-Volume 2, Springer International Publishing Switzerland.

[7] Shridhar G.Domanal, G.Ram Mohana Reddy "Load Balancing in Cloud Computing using Modified Throttled Algoritham" **:** Department of Information Technology National Institute of Technology Karnataka Surathkal, Mangalore, India

[8] Sheeja Y S, Jayalekshmi S "Cost Effective Load Balancing Based on honey bee Behaviour in Cloud Environment"2014 First International Conference on Computational Systems and Communications (ICCSC) | 17-18 December 2014 | Trivandrum.

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and Rajkumar Buyya1, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Published online in Wiley Online Library (wileyonlinelibrary.com), 24 August 2010, DOI: 10.1002/spe.995.

[10] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and Rajkumar Buyya1, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Published online in Wiley Online Library (wileyonlinelibrary.com), 24 August 2010, DOI: 10.1002/spe.995.

[11] Sen Su, Jian Li,Kai Shuang,Jie WangFröhlich, B."Cost-efficient task scheduling for executing large programs in the cloud".Parallel Computing, Volume 39, March 2013, pages 177-188

[12] Mayanka Katyal, Atul Mishra" A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment" International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013.