Dynamic Load Balancing Using Honey Bee Algorithm: Load Balancing

Sudha S. Senthilkumar
Vellore Institute of Technology, India

Brindha K.
Vellore Institute of Technology, India

Nitesh Kumar Agrawal
Vellore Institute of Technology, India

Akshat Vaidya
Vellore Institute of Technology, India

INTRODUCTION

Distributed system include a set of interconnected and virtual computers that are provisioning dynamically. The Distributed system and cloud computing characteristics are basically as on-demand access, broad network access, resource pooling, rapid elasticity, measured service and security.

Providing the virtualization technology in cloud platforms helps enterprises to rent computing power in the form of virtual machines to the users. The users may utilize hundreds or thousands of Virtual Machines (VMs). To handle a very large size of data many techniques to optimize load and simplify operations are needed to obtain desirable performance level for the users. There is a need an effective algorithm in term of load balancing in the cloud computing. The load can be differentiated in various categories, CPU load, delay or network load, memory capacity. Load balancing guarantee that each processor in the system does approximately the same quantity of load at any point of time .The task scheduling is used to determine the suitable resources to execute the tasks that received from users. For increasing the utilization of resources, we use the load balancing algorithm. Efficient utilization is achieved by using the idle resources until finishing the resources of processors that have the high load. The load balancing mechanism divides the loads within all the resources which are available. The dynamic cloud computing used many of load balancing strategies. The Max-Min algorithm for each task, the minimum completion time is calculated, then the task with the maximum of minimum completion time is mapped to the corresponding virtual machine. The Shortest Job-First (SJF) is a various algorithm to CPU scheduling. This way associated with every process length's next CPU burst. The process with shortest length assigned to CPU when it is available. When there are two processes that have same CPU bursts then scheduling based on FCFS. In Round Robin (RR) approach, the processes are separated between all processor. Every process is mapping the processor in a round robin order by using quantum time. The allocation of process is not based on the allocations from remote processors. The processing time of task for various processes is not equal but the distributions of workload are same.

DOI: 10.4018/978-1-7998-3479-3.ch008

With the ever increasing size of the systems, there is a greater need for load balancing. The servers tend to get over worked and are rendered useless in catering to the needs of millions of users, and hence deny the services. So, a challenge in this modern era of smart computing is to make use of the various Load Balancing techniques to get the servers to function efficiently. Various algorithms are used for balancing the overall load of the cloud and a few of them are the honey bee foraging algorithm, a biased random sampling on a random walk procedure and Active clustering. Here, we will focus on the honey-bee foraging algorithm and write a pseudo code for it, as well as develop a generic system model. The honeybee foraging algorithm is inspired from the behavior of honeybees for finding and reaping food. There is a type of bees called the forager bees who continually search for food sources and upon finding the same they return to the hive and advertise their discovery by a dance called as the waggle. The type of the movements in the dance reveal the quality and quantity of the food source as well as the distance of it from the beehive. Scout bees follow the forager bees to the location and reap the food. In case of load balancing in the web servers, whenever the demand sees a spike there is a dynamic allocation of services to regulate the changing demands of the user. The servers are grouped under Virtual servers (VS), each virtual server is assigned a specific queue for itself. Each server while processing a request calculates the reward and this is analogous to the quality of the find. The dance floor in case of the bees can be analogous to the advert board here which advertises the reward to the entire colony. Each server takes the role of either a forager or a scout. If the profit is high then the server stays at the current virtual server; posting an advertisement for its profitability (pr), if it was low then the server returns to the forage or scout behavior.

As mentioned earliest, the users send the requests to the service provider, the service provider has to serve many users to provide the best results and take into account maximizing resources utilization. Distributing the tasks into the resources to avoid overloading where there are under loaded resources. The objectives of this work to achieve users' satisfaction and provider satisfaction by balancing the loads and increase resources utilization, by minimizing the average of both make span time and response time using LBDA algorithm effective performance.

Proposed honey bee behavior inspired load balancing (HBB-LB) to achieve a good balance load through VMs for maximizing the throughput. We have compared the results of our proposed algorithm with the currently used algorithms. The results prove that our proposed algorithm has yielded better and efficient scheduling.

BACKGROUND

In an appropriated framework, dynamic load adjusting should be possible in two distinctive ways: distributed and non-distributed. In the conveyed one, the dynamic load adjusting calculation is executed by all hubs display in the framework and the errand of load adjusting is shared among them. The collaboration among hubs to accomplish stack adjusting can take two structures: cooperative and non-cooperative. Karaboga, D., & Ozturk, C. (2011)

In the first, the hubs work one next to the other to accomplish a typical target, for instance, to enhance the general reaction time, and so on. In the second frame, every hub works freely toward an objective neighborhood to it, for instance, to enhance the reaction time of a nearby local task. Dynamic load adjusting calculations of conveyed nature, normally create a larger number of messages than the non-dispersed ones on the grounds that, each of the hubs in the framework needs to connect with each other hub. An advantage, of this is regardless of the possibility that at least one hubs in the framework come