ISSN:0975-9646

Load Balancing in Cloud Environment Using Task Transfer: A Review

Syam Sankar[1], Devi Dath[2]

[1]*PG scholar, Department of Computer Science, College of Engineering Perumon, Kerala, India*

[2]*Assistant Professor, Department Of Computer Science, College Of Engineering Perumon, Kerala, India*

***Abstract*-Cloud computing allows the end users to access a set of distributed hosted resources like storage, CPU, applications (both software and hardware) and services over the internet in an on-demand basis. The interlinked servers of cloud network are responsible to deliver these services to the users. Load balancing is an inevitable challenging procedure associated with the computing environment of cloud. The dynamic load (or tasks) has to be distributed across multiple virtual machines or computers in an efficient manner so as to achieve maximum resource utilization and improved response time of submitted tasks. Load balancing algorithms attempt to transfer tasks from overloaded machine to underloaded machine thereby avoiding a situation of creating idle or low loaded virtual machines and thus minimizing the waiting time of tasks in the queue. This paper presents the concepts and techniques of load balancing in cloud environment. The objective of this paper is to analyze the factors that are affecting the load balancing procedure and also to make a study on existing load balancing algorithms.**

***Keywords*- cloud computing, load balancing, virtual machine, overloaded, underloaded**

I. INTRODUCTION

Cloud computing is a subscription-based service. The pool of software and hardware resources hosted in a remote server owned by a company can be made accessible easily to an end user for a period of time depending on the agreement the user has made with the cloud provider and it follows a ‘pay as you use’ model. Amazon, Citrix, Google, Microsoft etc. are the major cloud providers. A user must have cloud interface software so that it connects him to the network of computers that make up the cloud. A web browser can be an interface to the world of cloud. Email clients like Gmail, Yahoo, etc. are the major cloud computing scenarios. Through them, we log into our account remotely. The data of our account is not stored in our computer; it is hosted in the cloud server.

There are mainly three types of cloud: Public, Private and Hybrid [8]. A *public cloud* is commonly accessible to all end users with an internet connection. Public cloud services may be free or paid. Google App Engine and Amazon Elastic Compute Cloud (EC2) are well known examples of public cloud. A *private cloud* infrastructure is built for a specific organization or a group. An access to a private cloud is limited to that organization. A *hybrid cloud* allows to have a combination of both private and public cloud approach.

Cloud can be viewed as layered structure [9] where each layer provides dedicated services to the end users. *SAAS (Software as a service), PAAS (Platform as a service), IAAS (Infrastructure as a service)* are the basic layers of cloud architecture. The major resources managed at the *SAAS* layer are subscription based (on-demand) business applications, web services, multimedia, PPM applications etc. The giant providers of these resources are Google Apps, Gmail, Salesforce.com etc. The resources managed at the *PAAS* layer are development tools, web servers, middleware, application servers, operating systems, execution runtimes and databases. Microsoft Azure, Google App Engine, force.com are *PAAS* providers. Storage systems, physical servers, virtual machines, networks are managed by *IAAS* layer. Amazon, GoGrid, Rackspace etc. are *IASS* providers.

Virtualization is the fundamental component that powers cloud environment. It simply allows to separate applications and operating system from the underlying hardware and also permits to run multiple operating systems on the same node at the same time by sharing the hardware resources. Running a Linux operating system (called guest OS) within a Windows platform is a virtualization technique. Virtualization makes one physical computer to act and perform like many computers (virtual machines are either OS or application environment). By virtualizing a computer (that is, creating multiple virtual machines in a system), resource utilization of the system gets improved. *Hypervisor* [10] *(or Virtual machine monitor)* is a software with the responsibility of creating and running multiple virtual machines in a system and also allocating the available physical resources (CPU, memory, disk etc.) to them. Hypervisor makes an *abstraction layer* on the top of physical resources and this layer allocates the resources to virtual machines (VMs).

Virtual machines are the processing units in the cloud environment. A host can have multiple virtual machines. Tasks are scheduled to run on virtual machines. A task is assigned to a VM by the hypervisor so that resource requirement of the task is satisfied with the resources available at the virtual machine. Sometimes certain number of virtual machines in a host would get overloaded as scheduling proceeds. At the same time, other virtual machines in the host remain idle or underloaded. Load balancing comes here with the intention of distributing load (number of tasks that are active at a particular time) in an effective manner across all virtual machines by transferring load from overloaded virtual machines to underloaded virtual machines and making the resource utilization maximum. Load balancing algorithms balance the load thereby eliminating long waiting time of the tasks to get executed. This paper is an inspired work

II. LOAD BALANCING PROCEDURE

Load balancing procedure is implemented to increase the level of user satisfaction, to avoid a node overwhelmed with tasks and to enhance the overall performance. We consider an entity named *load balancer* to perform load balancing and its function is to ensure that all virtual machines encompass approximately equal amount of work load in their queue at any instant of time. Improved response time of tasks results after load balancing. Consider a simple load balancing situation in the figure (Fig.1) below.

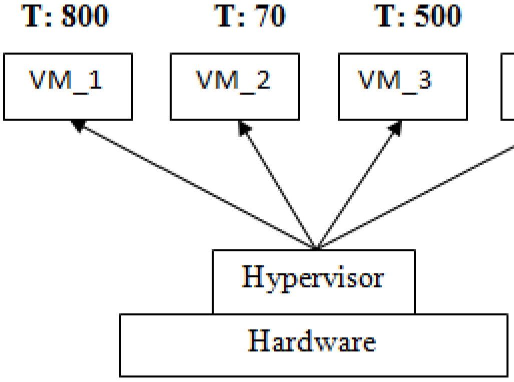

Fig. 1

Suppose, we are having a host with 4 virtual machines*It* (VM_1, VM_2, VM_3, VM_4), generated by a type 1 hypervisor (Example: Xen) running on the hardware. The letter '**T**' denotes the number of tasks (load) waiting in the queue at a virtual machine.VM_1 is having 100 tasks waiting in the queue, VM_2 with 10 tasks, and so on. From the given scenario, it is clear that a load balancing is required here as there exist both overloaded (VM_1 and VM_3) and underloaded virtual machines (VM_2 and VM_4) at the same instant.

Online traffic is managed by the cloud load balancers by distributing workloads across multiple servers and resources- automatically or on- demand. Load balancing is the most straightforward method of scaling out an application server infrastructure. As application demand increases, new servers can be easily added to the resource pool and the load balancer will immediately begin sending traffic to the new server.

Here we consider load balancing of tasks among virtual machines within one host.

A. Steps in load balancing

a. Load balancing decision.
b. Differentiate overloaded and underloaded VMs.
c. Transfer tasks from overloaded VMs to underloaded VMs by considering a suitable QoS factor
d. If system is not yet balanced, go to step c

Task transferring from overloaded VMs to underloaded VMs is done until all VMs are having approximately equal amount of load (Balanced system). Each host is having a maximum capacity (C) to handle. If the sum of load of all VMs (Total load, L) is greater than C, then the host is saturated, load balancing is not at all possible across all VMs. For each task to be transferred from overloaded virtual machine, a proper low loaded VM matching the resource requirement of task has to be identified. HBB_LB algorithm considers priority as the QoS factor in selecting a suitable low loaded VM. Selection of underloaded VM is done with criteria that the assigned task should be executed as soon as possible.

We can find whether a system is balanced or not by simply comparing the standard deviation (SD) of load of all VMs with a predefined threshold value. If the SD value is greater than the threshold, then the system is unbalanced. Large value of SD indicates that there lies a significant difference among the loads on VMs of the host, that is, not having approximately equal amount of load. Unbalanced system requires load balancing.

After taking load balancing decision (say load balancing is needed), we must partition the VMs of the host into three: overloaded, underloaded and balanced. Processing time (PT) of a VM is the ratio of its load and its capacity. If PT of a VM is greater than the sum of total processing time (ratio of total load and total capacity of all VMs) and the SD, then the VM is overloaded. The concept is clearly stated in [2].

III. LOAD BALANCING ALGORITHMS

A Round Robin Algorithm

It is the simplest static load balancing algorithm [3]. Each task is assigned a time quantum, during which it is allocated to the processor and does its operations. Selections of tasks are done in FCFS order.

B. Throttled Load Balancing Algorithm

Here [4] the request of a client (a task) is checked by a load balancer to find a suitable VM from the pool of available VMs (indexed list) which matches the resource requirement of the task. If a suitable VM is found, task is assigned to it. Indexing is used here to find a suitable VM easily that matches the size requirement of task.

C. Biased Random Sampling

It is a dynamic and distributed load balancing algorithm [5]. In this method a virtual graph is constructed. A node in the graph indicates a server. In-degree of a node represents the free resources of the corresponding server. When a task is being executed by the server, in-degree is reduced which specifies the decrease in the number of free resources.

D. Equally Spread Current Execution Algorithm

It [6] spreads or distributes the load equally on all VMs. Here the load balancer maintains a job queue and a list of VMs. If there exists a VM that can handle a job in the queue, then job is allocated to it. Continuous scan of job queue and VM's list are done so that proper assignment of VM to job is done. It follows spread spectrum technique.

E. Honey bee behaviour inspired load balancing algorithm

This load balancing algorithm [1] uses the ideas of natural foraging behaviour of honey bees. The tasks that are removed from overloaded VMs are considered as honey bees and the low loaded VMs are the food source. When a task is submitted to an underloaded VM, it would update the load and number of various priority tasks of that VM. This algorithm considers priority of the tasks as the main VM selection factor. It achieves maximum throughput and reduces response time of the tasks.

IV. SIMULATION TOOL-CLOUDSIM

We can simulate or mimic a cloud environment towards the research benefit using a tool called CloudSim. It supports for the simulation of cloud datacenters, hosts, virtual machines etc. It also supports to create user defined allocation policies. CloudSim [7] allows the host to manage multiple virtual machines created within it. Hosts are created within datacenters. The datacenters must be registered with an entity named CIS (Cloud Information Service). User can create the required number of tasks (cloudlets), which are then assigned to an entity named broker. Broker assigns the cloudlets to the VMs running on the host. We have a provision to define three important policies: VM allocation policy, VM scheduler policy, Cloudlet scheduler policy.

V. CONCLUSION

This paper presents various concepts associated with cloud and the load balancing procedure. It also states the ideas of existing load balancing algorithms especially the honey bee behaviour inspired load balancing algorithm. Cloud computing is a very vast technology. Load balancing is the most challenging issue associated with it. There is a large scope to develop much more efficient algorithms in balancing the cloud load and thereby achieving speedy processing of client requests.

REFERENCES

[1] Honey bee behavior inspired load balancing of tasks in cloud computing environments: Dhinesh Babu L.D , P. Venkata Krishna, Applied Soft Computing 13(2013) 2292–2303,journalhomepage: www.elsevier.com/locate/asoc

[2] B. Yagoubi, Y. Slimani, Task load balancing strategy for grid computing, Journal of Computer Science 3 (3) (2007) 186–194.

[3] Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment, Stuti Dave, Prashant Maheta, International Journal of Computer Applications (0975 – 8887) Volume 94 – No 4, May 2014

[4] Analytic Study of Load Balancing Techniques Using Tool Cloud Analyst. Tanveer Ahmed, Yogendra Singh/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 2, Mar-Apr 2012, pp.1027-1030

[5] A Dynamic Biased Random Sampling Scheme for Scalable and Reliable Grid Networks, O. A. Rahmeh1 P. Johnson, A. Taleb-bendiab.

[6] Efficient Load Balancing Algorithm in Cloud Environment, Akshay Daryapurkar, Mrs. V.M. Deshmukh, International Journal of Computer Science and Applications Vol. 6, No.2, Apr 2013 ISSN: 0974-1011

[7] Efficient Load Balancing Algorithm in Cloud Environment, Akshay Daryapurkar, Mrs. V.M. Deshmukh, International Journal of Computer Science and Applications Vol. 6, No.2, Apr 2013 ISSN: 0974-1011

[8] CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya1, Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995

[8] http://www.thinkgrid.com/business-scenarios/public-vs-private-vs-hybrid-cloud/

[9] http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/

[10] http://en.wikipedia.org/wiki/Hypervisor