A Survey Paper on Load Balancing Techniques in Cloud Computing

The long-dreamed vision of “computing as a utility” has finally taken shape in the form of cloud computing. This paradigm shift is the biggest buzz in today’s computer world. The pay-as-you-go model of cloud attracts more and more customers towards it. As a result the workload of the data center is increasing enormously. So Load balancing is the major issue in cloud data centre. The main goal of load balancing in cloud computing is to reduce energy consumption and SLA violation by distributing the load from overloaded host to the underloaded hosts in cloud data centre. There exists many load balancing algorithms in cloud. In this paper we have analysed some of these load balancing algorithms and also proposed a new method for load balancing.

INTRODUCTION

As defined by NIST [1] cloud computing is a model which provides convenient, ubiquitous and on-demand network access to a shared pool of configurable computing resources (e.g. networks, storage, servers, applications, and services) that can be immediately allocated and released with very little management effort or service provider interaction. Elasticity of resource provisioning, lack of capital investment and the pay-as-you-use pricing model attract people towards cloud computing. Cloud computing allows users to use computing recourses without installing them on their local computer.

As cloud can be accessed anywhere and anytime through commodity hardware, its demand is increasing day by day. So it must fulfill the Quality of Service (QoS) requirements of the user and at the same time must be advantageous for the Cloud Service Provider (CSP). The key technology behind cloud computing is virtualization which allows simultaneous execution of diverse tasks over a shared hardware platform. It provides on-demand and on-the-fly provision of physical machines to run diverse tasks, hence avoiding resource waste [2]. Cloud provides computing resources in the form of virtual machine (VM), which is an abstract machine that runs on physical machine (PM) [3].

VM live migration technique changes the mapping between PMs and VMs without interrupting the applications for a long time [4]. It transfers state of a VM from one PM to another with minimum downtime. Suspend-and-copy, pre-copy and post-copy are the three main live migration techniques. There is a delay associated with each migration, comprising of the time required for the VM [2] to stop execution at the current server, move the accompanying data to the new one and initialize a new VM there. Irrespective of this burden, VM migration is essential

for load balancing and uninterrupted maintenance activities.

Due to inefficient distribution of load some of the physical machines become overloaded. These overloaded servers produce more heat. As a result cost of the cooling system increases. It causes substantial emission of CO2 contributing to greenhouse effect [6][7]. So to minimize the environmental impact and fulfill the QoS requirements of users, some of the VMs have to be migrated to balance the load. Load balancing algorithms are classified as static and dynamic algorithms where Static algorithms are mostly suitable for homogeneous and stable environments and it can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes that take place during execution. Dynamic algorithms are more flexible and take into consideration different types of attributes in the system both prior to and during run-time [8]. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments. However, as the distribution attributes become more complex and dynamic, some of these algorithms could become inefficient and cause more overhead than necessary resulting in an overall degradation of performance. In this paper we present a survey of the current load balancing algorithms developed specifically to suit the Cloud Computing environments. We provide an overview of these algorithms and discuss their properties.

In addition, we compare these algorithms based on the following parameters: Response Time, Throughput, Energy Efficient, Resource Utilization, Scalability, Support, Heterogeneous, Resources, Data Processing Capacity, and Static/Dynamic. The rest of this paper is organized as follows. In section II we have described load balancing and its goals. In section III, various load balancing algorithms in cloud computing are discussed. Section IV compared multiple algorithms in terms of different parameters. After that, we discuss and compare (table-1) the relevant approaches in Section IV. We then conclude the paper and show possible areas of enhancement and our future plan of improving load balancing algorithms in Section V. II.

LOAD BALANCING

There are a number of challenges in cloud computing that need to be solved, which include infrastructure, load balancing, security and privacy in cloud computing, etc. Among them load-balancing is one of the necessary mechanisms to maintain the service level agreement (SLA) and for better use of resources. Load Balancing [9] is a mechanism which distributes the workload on the resources of a node to respective resources on the other node in a network without eliminating any of the running tasks [10]. So balancing the load between various nodes of the cloud system became a main challenge in cloud computing environment. The load can be any type like network load, memory load, CPU load and delay load etc. Thus it is very important to share work load across multiple nodes of system for better performance and increasing resources utilization. Major goals of load balancing [11] are - Establish fault tolerance system -

Maintain system stability Improve the performance and efficiency - Minimizing the job execution time and waiting time in queue. - Increase user satisfaction - Improve resource utilization ratio.

LITERRAL REVIEW OF VARIOUS LOAD BALANCING ALGORITHM

Following load balancing algorithms are currently prevalent in cloud computing. Round Robin – [12] It is one of the simplest scheduling techniques which uses the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or interval. Initially, loads are equally distributed to all VMs. As the name suggests, round robin works in a circular pattern. It is easy to implement and understand and hence less complex. Since the current load of the system is not considered, at any moment some node may possess heavy load and others may have no request. However, this problem is solved by weighted round robin algorithm. Weighted Round Robin – It is the modified version of Round Robin in which a weight is assigned to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2.

In such cases, the Data Center Controller will assign two requests to the powerful VM for each request assigned to a weaker one. Like Round Robin it also does not consider the advanced load balancing requirements such as processing times for each individual requests [13]. Dynamic Round Robin [14]- This algorithm mainly works for reducing the power consumption of physical machine. The two rules used by this algorithm are as follows: i) If a VM has finished its execution and there are other VMs hosted on the same PM, this physical machine will accept no more new virtual machine. Such physical machines are called to be in "retiring" state, i.e. when rest of the VMs finishes their execution, then this physical machine can shut down. ii) ii) The second rule says that if a physical machine is in retiring state for a long time then instead of waiting, all the running VMs are migrated to other physical machines. After the successful migration, we can shut down the physical machine. This waiting time threshold is called “retirement threshold”. The algorithm reduces the power consumption cost but it does not scale up for large data centers. Throttled - The Throttled Load Balancer (TLB) maintains a record of the state of each virtual machine (Busy/idle) [15]. When a request arrives it searches the table and if a match is found on the basis of size and availability of the machine, then the request is accepted otherwise -1 is returned and the request is queued [16].

During allocation of a request the current load on the VM is not considered which can in turn increase the response time of a task. Modified Throttled - Like the Throttled algorithm it also maintains an index table containing list of virtual Machines and their states. The first VM is selected in same way as in Throttled. When the next request arrives, the VM at index next to already assigned VM is chosen depending on the state of VM and the usual steps are followed, unlikely of the Throttled algorithm, where the index table is parsed from the first index every time the Data Center queries Load Balancer for allocation of VM [17]. It gives better response time

compare to the previous one. But in index table the state of some VM may change during the allocation of next request due to de allocation of some tasks. So it is not always beneficial to start searching from the next to already assigned VM. Active Monitoring Load Balancing (AMLB) Algorithm – It maintains information about each VM and the number of requests currently allocated to each VM. When a request to allocate a new VM arrives, it identifies the least loaded VM. If there are more than one, the first identified is selected. Load Balancer returns the VM id to the Data Center Controller. It sends the request to the VM identified by that id and notifies the Active VM Load Balancer of the new allocation [18]. During allocation of VM only importance is given on the current load of VM, its processing power is not taken into consideration. So the waiting time of some jobs may increase violating the QoS requirement.

VM-Assign Load Balancing Algorithm – It is a modified version of Active Monitoring Load Balancing algorithm. The first allocation of VM is similar to the previous algorithm. Then if next request comes it checks the VM table, if the VM is available and it is not used in the previous assignment then, it is assigned and id of VM is returned to Data Center, else it finds the next least loaded VM. Sridhar G. Domanal et. al stated that this algorithm will utilize all the VMs completely and properly unlike the previous one where few VMs will be overloaded with many requests and rest will remain under utilized [19]. But it is not clearly mentioned in the paper that how it happens. This algorithm will not use the VM if it is already allocated in the last round. But there is no logic behind it. Because it may still be the least loaded VM having good processing speed. So more tasks can be assigned to it.

Finding the next least loaded VM will distribute the tasks evenly only when there are multiple VMs which are equally loaded or the next least loaded VM has a high processing speed compare to the previous one. But the algorithm only considers the load and if the VMs are equally loaded then the task can be assigned to any of them irrespective of the fact that whether the VM is used in the last iteration or not. Since allocation of a task change the state of VM so in the previous algorithm least loaded VM will be found automatically and even task distribution will take place. Weighted Active Monitoring Load Balancing Algorithm - Jasmin James et. al proposed this method [15] which is a combination of Weighted Round Robin and Active Monitoring Load Balancing Algorithm. In this algorithm different weights are assigned to VMs depending on the available processing power. Among the least loaded VMs the tasks are assigned to the most powerful one according to their weights. In this way it removes the shortcomings of Active Monitoring Load Balancing Algorithm by not only considering the load but also the processing power of available VMs. IV.

COMPARISON OF VARIOUS LOAD BALANCING ALGORITHM IN CLOUD COMPUTING

Table 1 below compares performance of different load balancing algorithms in terms of different parameters as already mentioned in Section 1. V. PROPOSED WORK In this paper we have proposed a load balancing algorithm called “Dynamic Throttled” which works as follows: Step 1:- Task scheduling will be done like Throttled algorithm. Step 2:- All the PMs will be monitored in a regular interval to check whether it has become overloaded or not (If current load of the CPU is greater than a fixed threshold then the node will be considered as overloaded). Step 3:- If an overloaded PM is found, then some of its VM will be migrated to other PM. VM having minimum RAM content will be migrated. Migration will be continued until the current load of the PM becomes less than the threshold. Destination PM will be selected using simple Throttled algorithm.

CONCLUSION AND FUTURE WORK

In this paper, we have studied various algorithms for load balancing in Cloud Computing. The main purpose of load balancing is to satisfy the customer requirement by distributing load dynamically among all the available nodes and improve the performance and efficiency. So the resource utilization ratio increased. We have also compared different algorithms to describe their overall performance. In future we will simulate the above mentioned “Dynamic Throttled” algorithm using Cloudsim and compare its performance with other existing algorithms.