Server Consolidation Based Dynamic Load Balancing Approach in Cloud Computing

Shailee Majmudar[1], Krunal Panchal[2]
[1, 2] Department of Computer Engineering
[1, 2] LJIET, Gujarat, India

Abstract- *Power efficiency is one of the main issues that will drive the design of data centers, especially of those devoted to provide Cloud computing services. In virtualized data centers, consolidation of Virtual Machines (VMs) on the minimum number of physical servers has been recognized as a very efficient approach, as this allows unloaded servers to be switched off or used to accommodate more load, which is clearly a cheaper alternative to buy more resources. The consolidation problem must be solved on multiple dimensions, since in modern data centers CPU is not the only critical resource: depending on the characteristics of the workload other resources.*

The problem is so complex that centralized and deterministic solutions are practically useless in large data centers with hundreds or thousands of servers.

Keywords- Cloud computing, VM consolidation, data center, energy saving

I. INTRODUCTION

All main trends in information technology, for example, Cloud Computing and Big Data, are based on largeand powerful computing infrastructures. The ever increasing demand for computing resources has led companies and resource providers to build large warehouse-sized data centers, which require a significant amount of power to be operated and hence consume a lot of energy. The virtualization paradigm can be exploited to alleviate the problem, as many Virtual Machine (VM) instances can be executed on the same physical server. This enables the consolidation of the workload, which consists in allocating the maximum number of VMs in the minimum number of physical machines. Consolidation allows unneeded servers to be put into a low-power state or switched off (leading to energy saving), or devoted to the execution of incremental workload (leading to savings, thanks to the reduced need for additional servers). Unfortunately, efficient VM consolidation is hindered by the inherent complexity of the problem. The optimal assignment of VMs to the servers of a data center is analogous to the NP-hard “Bin Packing Problem,” the problem of assigning a given set of items of variable size to the minimum number of bins taken from a given set [6,3].

II. LITERATURE SURVEY

1. Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines [1].

In this paper, it is briefly summarized the load balancing algorithms used in the cloud computing environment. The main focus is on the efficient utilization of the virtual machines and balancing the virtual machines with the incoming request. It presents a novel VM-assign algorithm which allocates incoming jobs to available virtual machines. Here the virtual machine assigned depending on its load i.e. VM with least request is found and then new request is allotted. With this algorithm underutilization of the virtual machine improved significantly and later it is compared with existing Active-VM algorithm.

VM-assign load balancer algorithm maintains an index/ assign table of virtual machines and also the load of VMs. There has been attempt made to efficient usage of available virtual machines depending on its load. Proposed algorithm employs a method for selecting a VM for processing client's request. It checks for least loaded VM. Initially all VM are free so it follows Round Robin. Then if next request comes then it checks for VM table, if the VM is available and it is not used in the previous assignment then, it is assigned with the request and id of VM is returned to Data Center, else we find the next least loaded VM and it continues and follows the above step, unlikely of the Active load balancer, where the least loaded VM is chosen but it will not check for the previous assignments.

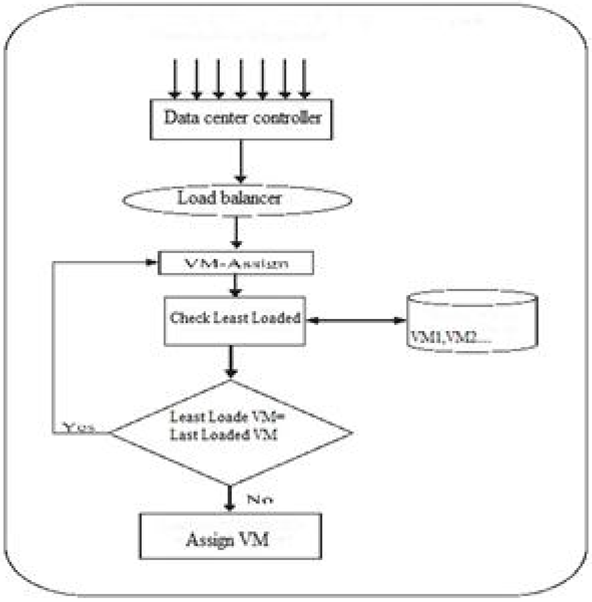

Fig. 1 VM assigning for load balancing [1]

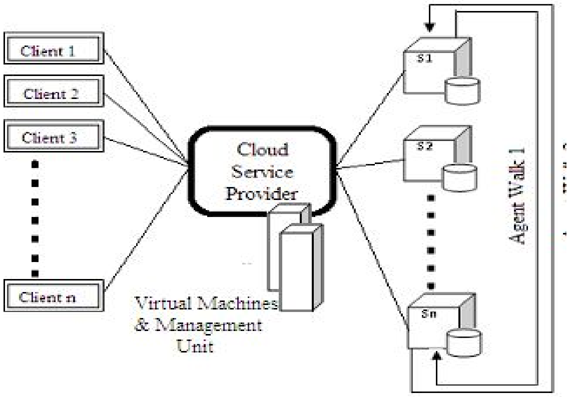

Fig 2: Agent based load balancing

2. Response Time Based Load Balancing in Cloud Computing [2].

The proposed paper aims to overcome the limitations of the previous algorithms and to make the algorithm suitable for real time application in the Cloud scenario. The Load Balancer has list of the VMs and the list of Services provided by Cloud Service Provider.

The algorithm is Preventive in nature, and schedules the incoming requests in such a way that the Load Imbalance does not occur. The algorithm eliminates the need of unnecessary communication of the Load Balancer with the VMs by not querying about their current resource availability. This reduces the computation of VM that is needed to collect the available resources. This eliminates the exchange of query packets and hence bandwidth of communication channel saved since the frequency of these query packets is very high

Algorithm

The Algorithm has been divided into three primary modules:

i. **Threshold Adjustment Module:**

This module increases or decreases the threshold as required

ii. **Average and Predicted Average Response Time Calculation Module:**

In this module we calculate the average response time and the predicted average response time if current request was assigned to this VM. There may or may not be a number of requests already assigned or serviced by this VM. We have to traverse those requests one by one.

iii. **Service allocation module.**

This module allocates the new request to the current VMs by checking the threshold criteria against each VM as described previously.

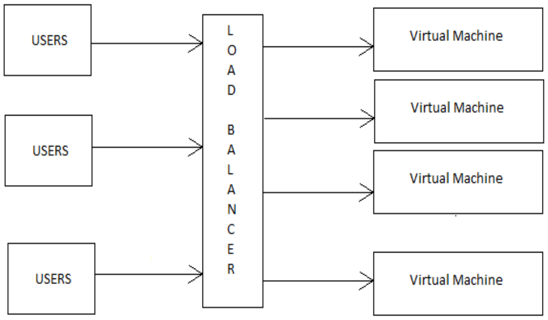

Fig.3Arrangement of load balancer [2]

3. Implementing a Novel Load-aware Auto Scale Scheme for Private Cloud Resource Management Platform [3].

Many open-source IaaS cloud management platforms have been put forward for deploying private clouds. Although all of them have main basic functions and could be used in simple environment, little of them have the ability of automatically scaling instances. This is an important obstacle to deploy private cloud in large data centers. In this paper, we propose a novel load-aware auto scale scheme for IaaS private cloud resources management platform.

In most private cloud environment, the total system has several components: One or more physical machine serve as control node(s), which runs main function modules. Some others only run basic computing components and service as computing nodes. There also are some other related servers with specific functions, including database servers, NFS storage servers. To build a complete auto scale module, here added some necessary components to origin private cloud system architecture as shown in Figure 5.

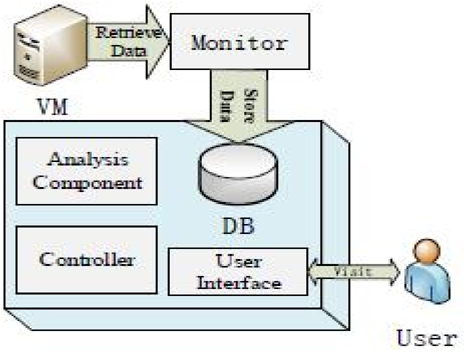

Fig. 4 Auto Scale Module Architecture [3]

Monitoring component: Measures the resources consumption of virtual machines and stores monitoring data into database periodically.

Auto scale analysis component: Apply related algorithms to monitoring data, and make decisions such as scale in or scale out. These decisions should solve the problems such as when and how many new virtual machines should be launched.

Auto scale controller: Call system APIs to doreactions. For example, run a new virtual machine
based on a user-specified image.

User interface: Provide control panel for users to make configurations, define restrictions or submit other requests. It is also responsible for giving user operation feedbacks.

4. Cloud Light Weight: a New Solution for Load Balancing in Cloud Computing [4]

In this paper, we propose a new load balancing solution for cloud computing environments. We present a dynamic load balancing algorithm that not only can balance the incoming workload in the cloud environment and among the VMs, but also it can provide a high level of QoS for users. It can also prevent SLA violation by balancing the system load based on the tasks' requirements. We call our new load balancer algorithm "Cloud Light Weight Load Balancing".

CLW architecture is a multi-level event-driven architecture. The CLW algorithm, which will be presented in the next section, has been developed based on this architecture. Figure 1 shows the block diagram of CLW architecture. As shown, the CLW architecture consists of three main levels:

- Level 1 includes the cloud datacenter and its central components: VMs, hosts and host managers.
- The VM manager is placed in level 2 and its main task is to control and handle the activities of the level 1 components.
- Finally, there is one head VM manager and an ESB (Enterprise Service Bus) in level 3.

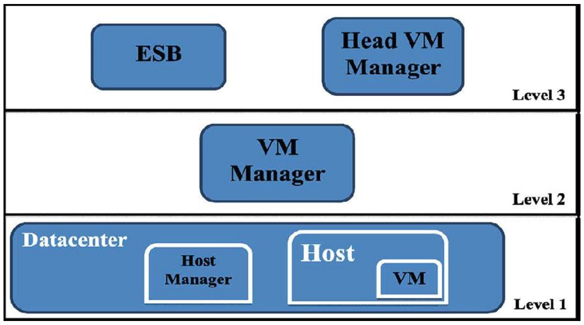

Fig 5 Multilevel CLW architecture [4]

A. Datacenter

A datacenter is one of the most important components in the CLW architecture. It is a centralized repository, either physical or virtual, for the storage, management, and dissemination of data and information organized in cloud resources. A datacenter, in a cloud environment, consists of a network of several physical servers (hosts) with heterogeneous processing attributes. A cloud environment includes several datacenters that are located in different geographical areas and have various processing resources. As Figure 2 shows, the CLW architecture components and their interactions, a cloud may consist of N different datacenters, which are distributed in the entire world. These datacenters are presented in the level 1 of CLW architecture as mentioned earlier.

B. Virtual Machines

Virtualization of an operating system is a technique for running several operating systems simultaneously on one physical server. Similarly, we have virtualization of a computer, storage device, network resources etc. It is mainly driven by the benefit of application isolation, resource sharing, fault tolerance and cost efficiency. A special middleware (hypervisor) abstracts from the physical hardware resources and provides them as the so called VMs, which act like real physical servers with their own hardware resources. The CLW algorithm balances the system workload among these VMs. Each VM in the CLW architecture has an id, which is unique in all cloud data centers. In addition, for assuring the quality of service in the load balancing mechanism, we consider some main attributes for each VM such as: type of services, operating system type, capacity of VM's resources (e.g. memory, storage, and bandwidth) and etc.

C. Host

Hosts are physical servers, which contain several VMs, allocate the hardware resources among them, collect the VM's information and deliver this information to host managers.

D. Host Manager

It is mainly driven by the benefit of application isolation, resource sharing, fault tolerance and cost efficiency. A special middleware (hypervisor) abstracts from the physical hardware resources and provides them as the so called VMs, which act like real physical servers with their own hardware resources.

E. Virtual Machine Manager

As shown in Figure 2, each VM manager is responsible for managing the VMs of one or more datacenters. The VM managers gather information from the VMs, which is distributed in the different datacenters and provide this information to the central head VM manager.

5. Load Balancing Techniques: Major Challenge In Cloud Computing – A Systematic Review. [5]

The different policies in dynamic load balancing are:

Transfer Policy:

The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.

Selection Policy:

It specifies the processors involved in the load exchange (processor matching)

Location Policy:

The part of the load balancing algorithm which selects a destination node for a transferred task is referred to as location policy or Location strategy.

Information Policy:

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 Infonnation strategy.

Load estimation Policy:

The policy which is used for deciding the method for approximating the total work load of a processor or machine is termed as Load estimation policy.

Process Transfer Policy:

The policy which is used for deciding the execution of a task that is it is to be done locally or remotely is termed as Process Transfer policy.

Priority Assignment Policy:

The policy that is used to assign priority for execution of both local and remote processes and tasks is termed as Priority Assignment Policy.

Migration Limiting Policy:

The policy that is used to set a limit on the maximum number of times a task can migrate from one machine to another machine

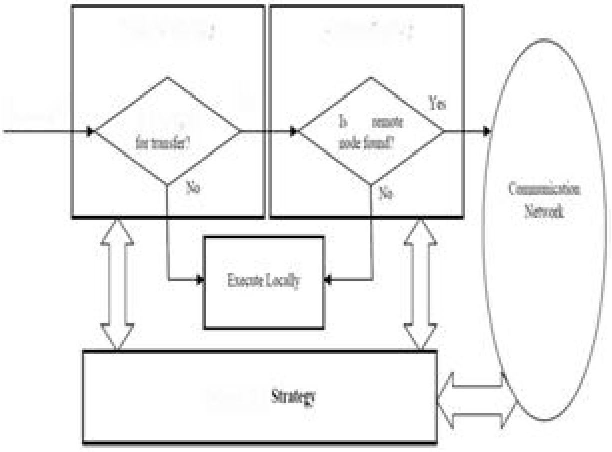

Fig. 6. Interaction among components of a Dynamic Load Balancing Algorithm [5]

III. CONCLUSION

As per the literature review is concerned of referred research paper, there are many issues in consolidation to achieve. By the time the data centers will grow and more and more resources will be required to fulfill the need of users. At that time other problems will be encountered and more efficient algorithm will be needed as per the concern of server consolidation. It is possible by appropriate advance approaches in algorithm during consolidation.

REFERENCES

[1] Shridhar G.Damanal , G. Ram Mahana Reddy, “Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines” IEEE Cloud Computing, 2014

[2] Agraj Sharma, Sateesh K. Peddoju, Response Time Based Load Balancing in Cloud Computing”, IEEE Control, Instrumentation, Communication and Computational Technologies, 2014

[3] JieBao, Zhihui Lu, Jie Wu, Shiyong Zhang, YipingZhong, “Implementing a Novel Load-aware Auto Scale Scheme for Private Cloud Resource Management Platform”, IEEE Cloud Computing, 2014

[4] Mohammadreza Mesbahi, Amir MasoudRahmani, “Cloud Light Weight: a New Solution for Load Balancing in Cloud Computing”, IEEE Data Science and Engineering , 2014

[5] VelagapudiSreenivas ,Prathap.M , Mohammed Kemae, “load balancing techniques: major challenge in cloud computing – a Systematic review” IEEE Electronics and Communication Systems , 2014

[6] http://www.webopedia.com/TERM/C/cloud_computing. html, 24 Nov 2015 22:20