Efficient Load Balancing Algorithm in Cloud Environment

Akshay Daryapurkar[#], Mrs. V.M. Deshmukh[*]

[#]*PRMIT&R Anjangoan Bari Road*
Badnera, Amravat-444701i

[a]akshaydaryapurkar321@gmail.com

[3]msvmdeshmukh@redifffmail.com

Abstract— **Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. This paper presents an approach for scheduling algorithms that can maintain the load balancing and provides better improved strategies through efficient job scheduling and modified resource allocation techniques. We present benchmarks for the algorithms to show their capabilities. The metric used is throughput and response time. Benchmark workload is synthetic to highlight many aspects of balancing load correctly. From the benchmarks and by analysing the different algorithms we will show a simple algorithm that is platform independent, easy to implement and has the best performance of any of the algorithms implemented.**

Keywords— **Cloud Computing, Load Balancing, Round Robin, Virtualization.**

I. Introduction

A. Load Balancing

Load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time [2]. Dividing the traffic between servers, data can be sent and received without major delay. Different kinds of algorithms are available that helps traffic loaded between available servers [1]. A basic example of load balancing in our daily life can be related to websites. Without load balancing, users could experience delays, timeouts and possible long system responses. Load balancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website availability is conclusively settled .There are many different kinds of load balancing algorithms available, which can be categorized mainly into two groups. The following section will discuss these two main categories of load balancing algorithms [1], [2].

B. Static Algorithms

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [2].

C. Dynamic Algorithms

Dynamic algorithms designated proper weights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time communication with the networks, which will lead to extra traffic added on system [2]. In comparison between these two algorithms, although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result. However; dynamic algorithm predicated on query that can be made frequently on servers, but sometimes prevailed traffic will prevent these queries to be answered, and correspondingly more added overhead can be distinguished on network [1].

D. Load Balancing in Cloud Computing

Cloud vendors are based on automatic load balancing services, which allowed entities to increase the number of CPUs or memories for their resources to scale with the increased demands. This service is optional and depends on the entity's business needs. Therefore load balancers served two important needs, primarily to promote availability of cloud resources and secondarily to promote performance. According to the previous section Cloud computing will use the dynamic algorithm, which allows cloud entities to advertise their existence to presence servers and also provides a means of communication between interested parties [1].

E. Load Balancing in Distributed Systems

Today more modern software development methodologies are being used to enhance the usability of software embedded on compatible hardware in distributed networks [1] [2]. Attaining this objective and improve software infrastructure, middleware have been applied to foster portability and distributed application component interpretability. Middleware characterized as network services and software components that permit scaling of application and networks. Providing the simple and integrated distributed programming environment, middleware eased the task of designing and programming and managing the distributed applications.

II. Cloud Computing & Virtualization

A Cloud system consists of 3 major components such as clients, data center, and distributed servers. Each element has a definite purpose and plays a specific role.

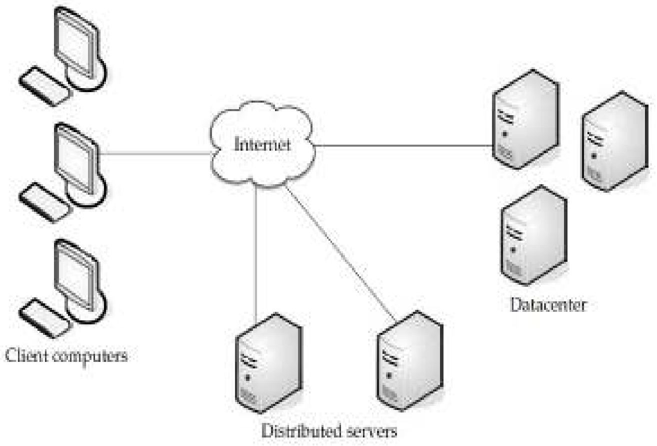

Fig 1: Three components make up a cloud computing solution [3]

A. Type of Clouds

1) Clients:

End users interact with the clients to manage information related to the cloud. Clients generally fall into three categories as given in [3]-[5]:

- **Mobile**: Windows Mobile Smartphone, smart phones, like a Blackberry, or an iPhone.
- **Thin**: They don’t do any computation work. They only display the information. Servers do all the works for them. Thin clients don’t have any internal memory.
- **Thick**: These use different browsers like IE or Mozilla Firefox or Google Chrome to connect to the Internet cloud.

2) Datacenter:

Datacenter is nothing but a collection of servers hosting different applications. A end user connects to the datacenter to subscribe different applications. A datacenter may exist at a large distance from the clients [5]. Now-a-days a concept called virtualisation is used to install software that allows multiple instances of virtual server applications.

3) Distributed Servers:

Distributed servers are the parts of a cloud which are present throughout the Internet hosting different applications. But while using the application from the cloud, the user will feel that he is using this application from its own machine [4].

B. Type of Clouds

Based on the domain or environment in which clouds are used, clouds can be divided into 3 catagories:

_ Public Clouds
_ Private Clouds
_ Hybrid Clouds (combination of both private and public clouds)

C. Virtualization

It is a very useful concept in context of cloud systems. Virtualization means “something which isn’t real”, but gives all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine. Virtualization is related to cloud, because using virtualization an end user can use different services of a cloud. The remote datacenter will provide different services in a full or partial virtualized manner.

Two types of virtualization are found in case of clouds as given in [5]:

_ Full virtualization
_ Para virtualization

1) Full Virtualization:

In case of full virtualization a complete installation of one machine is done on another machine. It will result in a virtual machine which will have all the software’s that are present in the actual server.

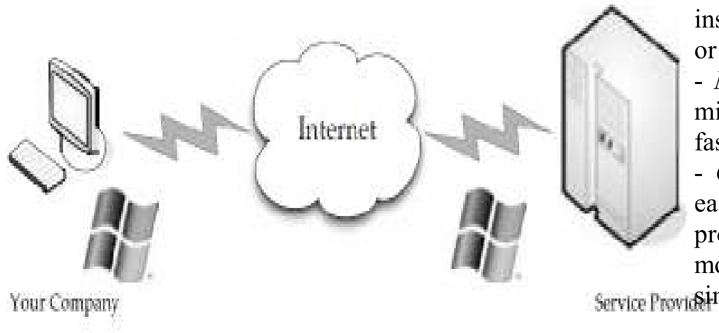

Fig 2: Full Virtualization [5]

Here the remote datacenter delivers the services in a fully virtualized manner. Full Virtualization has been successful for several purposes as pointed out in [5]:
- sharing a computer system among multiple users
- Isolating users from each other and from the control program
- Emulating hardware on another machine

2) Para virtualization:
In paravitualisation, the hardware allows multiple operating systems to run on single machine by efficient use of system resources such as memory and processor e.g. VMware software. Here all the services are not fully available, rather the services are provided partially [4].

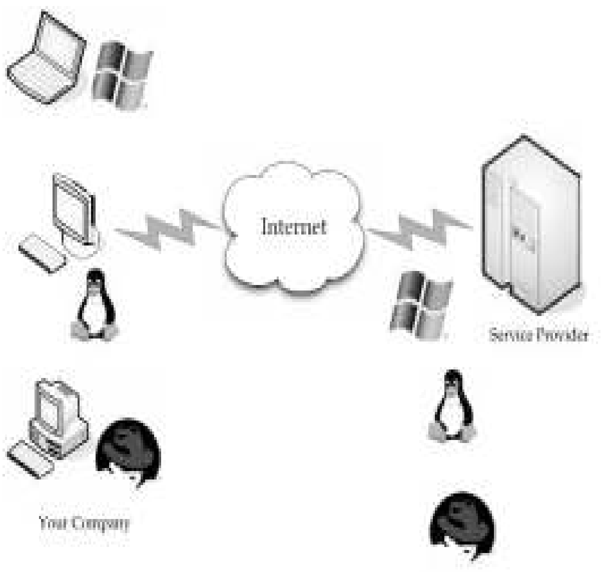

Fig 3: Para virtualization

Para virtualization has the following advantages as given in:

- *Disaster recovery*: In the event of a system failure, guest instances are moved to hardware until the machine is repaired or replaced.
- *Migration*: As the hardware can be replaced easily, hence migrating or moving the different parts of a new machine is faster and easier.
- *Capacity management*: In a virtualized environment, it is easier and faster to add more hard drive capacity and processing power. As the system parts or hardware's can be moved or replaced or repaired easily, capacity management is simple and easier.

III. Analysis of Load Balancing Algorithms

A. Round Robin Algorithm

Round-robin is by far the simplest algorithm available to distribute load among nodes. It is therefore often the first choice when implementing a simple scheduler. One of the reasons for it being so simple is that the only information needed is a list of nodes [8].
However this is only when several key assumptions are true:
1. The nodes must be identical in capacity. Otherwise performance will degrade to the speed of slowest node in the cluster.
2. Two or more client connections must not start at the same time. Should they, the node chosen will be the same, because the order of nodes retrieved from the cluster is the same every time.
3. The jobs must be similar to achieve optimum load distribution among the nodes. If a single node is more loaded than others it will become a bottleneck in the system [8].

B. Weighted Round Robin

In a weighted round-robin algorithm, each destination (server) is assigned a value that signifies, relative to the other servers in the list, how that server performs. This "weight" determines how many more (or fewer) requests are sent that server's way; compared to the other servers on the list [6].

C. Random

This selection is more random, meaning that it will keep on selecting the nodes in a different order each time. The reasons for this added complexity is that some workloads may perform the worst possible way with any of the round robin schedulers. Consider a cluster with four nodes; if unlucky every fourth job is a job that requires much work. A round-robin algorithm would then assign these jobs on a single node and slow everything down. Here it matters little that the list of nodes was randomized to begin with [6]-[8].

D. Load Informed

Input: A priority heap of nodes sorted by load (nodes)

```
2: Input: A list of nodes in the system (orig_nodes)
3: ONLOADUPDATE ()
4: for (iterator it = orig_nodes.Begin(); it != orig_nodes.end() ; ++it)
    {
5: nodes.delete (it);
6: nodes.insert (it.load, node);
    }
7: GETNEXTNODE ()
8: Return nodes.first ();
```

E. *Concluding Remarks*

1) *Round-robin:*

+ Fast.
+ Requires only information about the names of the nodes.
+ Does not require any global decisions.
+ Easy to implement.
+ Superb balancing, when the workload is not skewed, nor are the nodes.
− Breaks down completely when the above item does not hold [8].

2) Weighted Round-robin:

+ Requires only information about the names of the nodes and their weights.
+ Does not require any global decisions, but when the number of jobs are low it greatly improves when available.
+ Superb balancing, when the workload is not skewed.
− Breaks down completely when the above item does not hold.
− Requires more work than the Round-Robin scheduler.

3) Weighted Random:

+ Requires only information about the names of the nodes and their weights.
+ Does not require any global decisions.
− Unpredictable balancing [8].

4) Load informed:

+ Can prevent worst-case scenarios.
− The information required is not platform independent.
− Require global decisions to work properly.
− Requires information about the names of the nodes, their weights and a constantly updated specification of the load of each node.
− The optimal load adjustment when choosing a node is not clear.
− Very complex to implement and even more complex to get right.
− Requires a large machinery to calculate the load.

F. *Equally Spread Current Execution*

The random arrival of load in such an environment can cause some server to be heavily loaded while other server is idle or only lightly loaded. Equally load distributing improves performance by transferring load from heavily loaded server. Efficient scheduling and resource allocation is a critical characteristic of cloud computing based on which the performance of the system is estimated. The considered characteristics have an impact on cost optimization, which can be obtained by improved response time and processing time.

A scheduling algorithm is compared with the existing round robin scheduling to estimate response time, processing time, which is having an impact on cost .A Comparison of Dynamic Load Balancing Algorithms [6].

Here the jobs are submitted by the clients to the computing system. As the submitted jobs arrive to the cloud they are queued in the stack. The cloud manager estimates the job size and checks for the availability of the virtual machine and also the capacity of the virtual machine. Once the job size and the available resource (virtual machine) size match, the job scheduler immediately allocates the identified resource to the job in queue. Unlike the round robin scheduling algorithm, there is no overhead of fixing the time slots to schedule the jobs in a periodic way [6]. The impact of the ESCE algorithm is that there is an improvement in response time and the processing time. The jobs are equally spread, the complete computing system is load balanced and no virtual machines are underutilized. Due to this advantage, there is a reduce in the virtual machine cost and the data transfer cost.

```
ESAE LOAD ALGORITHM
ACTIVE VM LOAD BALANCER [6]
[START] Step1:- find the next available VM
Step2:-check for all current allocation count is less than max length of VM list allocate the VM
Step3:- if available VM is not allocated create a new one in
Step 4:- count the active load on each VM
Step5:- return the id of those VM which is having least load
[END]
```

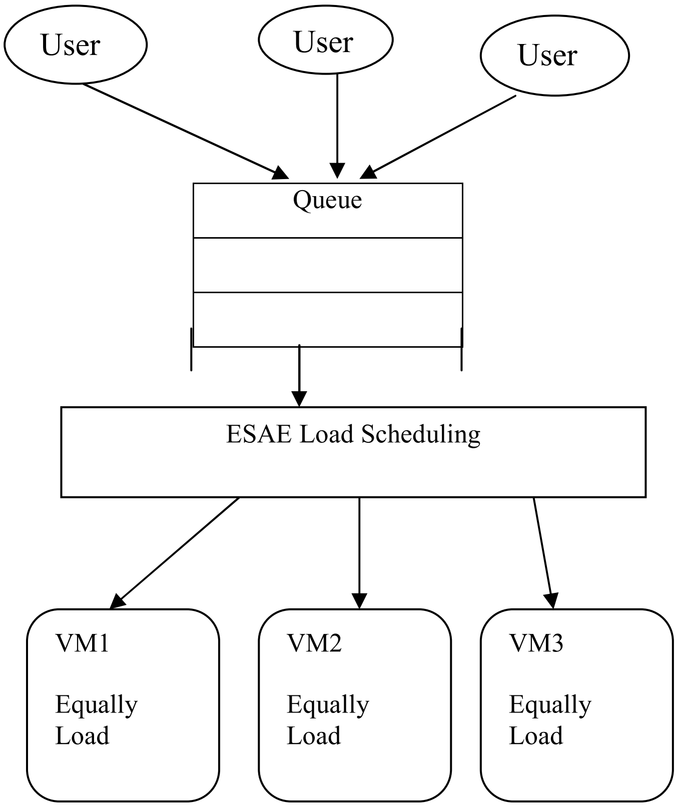

Figure 4: Equally spread Active execution load to the cloud system

IV. Conclusions

Cost and time are the key challenge of every IT engineer to develop products that can enhance the business performance in the cloud based IT sectors. Current strategies lack efficient scheduling and resource allocation techniques leading to increased operational cost and time. This paper aims towards the development of enhanced strategies through improved job scheduling and resource allocation techniques for overcoming the above-stated issues [6]. Here, Equal Spread Current Execution Load algorithm dynamically allocates the resources to the job in queue leading reduced cost in data transfer and virtual machine formation. Comparisons of various load balancing algorithms are made with respect to overall time and cost [6].

References

[1] Zenon Chaczko, Venkatesh Mahadevan , Shahrzad Aslanzadeh Availability and Load Balancing in Cloud Computing University of Technology Sydney, Australia

[2] R. Shimonski. Windows 2000 & Windows Server 2003 Clustering and Load Balancing. Emeryville. McGraw-Hill

[3] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, Tata McGraw-HILL Edition 2010.

[4] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[5] Ram Prasad Padhy, P Goutam Prasad Rao Load Balancing In Cloud Computing system Rourkela-769 008, Orissa, India May, 2011.

[6] Jaspreet kaur Comparison of load balancing algorithms in a Cloud International Journal of Engineering Research and Applications (IJERA) May-Jun 2012

[7] Jiyin Li, Meikang Qiu, Jain-Wei Niu, YuChen, Zhong Ming "Adaptive Resource Allocation for Preeemptable Jobs in Cloud Systems". IEEEInternational Conference on Intelligent Systems Design and Applications, pp. 31-36, 2010.

[8] Dennis Haney & Klaus S. Madsen Load-balancing for MySQL Datalogisk Institut, Københavns Universitet Fall 2003

[9] David Karger and Matthias Ruhl, "New Algorithms for Load Balancing in Peer-to-Peer Systems," Tech. Rep. MIT-LCS-TR-911, MIT LCS, July 2000.

[10] Vinod Pisharody, Michael Wu, Tanya Zhiltsova Active-active operation for a cluster of SSL virtual private network (VPN) devices with load distribution.