# Enhancement of Auto Scaling and Load Balancing using AWS

**Mohit Gaur, Ankit Kumar, Pankaj Dadheech**
Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan Jaipur-302017, (INDIA)
***Email- mohit_gaur1890@yahoo.com***
Received 23.07.2019 received in revised form 10.08.2019, accepted 13.08.2019

**Abstract : As the use of the internet is increasing the corporate migrating their business from traditional computing to cloud computing and thus the number of the user is increasing on cloud & load is also increasing. Thus to provide congestion-free and reliable on-demand service to client load balancing method is needed. Many algorithms are proposed for load balancing & auto-scaling to handle the load .we can use cloud service to make load efficient model in the cloud environment. This load efficient model will provide the load balancing, scaling capabilities and monitoring of solutions in the cloud environment. To achieve the above mentioned, we use public cloud services such as amazon's EC2, ELB. This research is divided into four parts such as load balancing, auto-scaling, latency based routing, resource monitoring. We will implement the individual service and test while providing load from external software tool Putty and we will produce the result for efficient load balancing.**

**Keywords :** Load Balancing, EC2, Resource Monitoring, Load Optimization, Web Server.

## 1. INTRODUCTION

Cloud is an infrastructure or a platform which enables the computing of applications and services in reliable and elastic mode. Virtual platform means the hardware which is used to create a datacenter (the cloud) such as a server, storage, and network. Same as the software utility in the cloud is referred to as the services and applications provided to the users or clients [1,2]. Cloud computing is one of the most emerging technologies which drew the attention of the entire technocrat in the field of computer science. Cloud computing is the technique which represents both cloud and the application (services). It is basically referred to as accessing computing service (resources and application) over the internet.

The cloud service provider handles the data from the remote location about that the client is unaware but an individual can access his data from anywhere simply by a system with an internet connection.

Cloud computing has changed the classical computing environment in the IT industry. With cloud computing, many corporates are migrating their business from traditional computing to cloud computing in order to meet their business requirements. Cloud computing has been considered as the most revolutionary technology in the IT industry. For example, we can assume the whole internet as a single cloud in which people share space and resources from the pool of virtual space. The most important thing which is provided by cloud computing is the virtualization of resources.The cloud service provider handles [3,4] the data from the remote location about that the client is unaware but an individual can access his data from anywhere simply by a system with an internet connection.

Cloud computing has changed the classical computing environment in the IT industry. With cloud computing, many corporate are migrating their business from traditional computing to cloud computing in order to meet their business requirements. Cloud computing has been considered as the most revolutionary technology in the IT industry. For example, we can assume the whole internet as a single cloud in which people share space and resources from the pool of virtual space. The most important thing, which is provided by cloud computing, is the virtualization of resources[5,6,7].

NIST gives the standard definition of cloud computing as "It is the framework which enables the user or client in the computing environment to access on-demand services and a pool of resources such as servers, network, applications, etc. For example, when we save the image over the internet or send some files using the internet, we use cloud computing. Many websites are running on cloud computing because of its elasticity and auto-scaling feature. The major concern in cloud computing is the security of data as the data of the client is stored on a remote location. One can manipulate the data on the cloud server so the privacy of the data is the biggest factor in this technology. Service providers are working on this issue to resolve it [8,9].


**Figure 1:**Basic cloud model

## 2. LITERATURE REVIEW

### *2.1 Load Balancing*

In general, load balancing algorithms can be categorized as static and dynamic. On the basis of the current status of the node. Static load balancing technique distributes the load among the nodes on the basis of the past state while dynamic load balancing technique does not consider the past state or behaviour for distributing the load, it only depends on the present state of the node. Hence it gives better results than the static one. Dynamic load balancing is of two types distributive and non-distributive [10,11].

The main advantage of dynamic load balancing is, it will not stop the system if a node is down, the following criteria will be used to compare the LB algorithms.

- Throughput: Throughput is the total number of executed job in a fixed time.
- Overhead Associated: The amount of processing overhead to implement the LB i.e. inter-process communication.
- Response Time:It is defined as the total time in which the algorithm responds in the cloud network.
- Resource Utilization:The amount of resource is utilized while implementing the algorithm.
- Scalability: How efficiently it handles the load and scales the system according to its need.

### *2.2 Load balancing Algorithm*

Lots of algorithms [12,13,14] are proposed and implemented to handle the issue of load balancing in the cloud network. Some of them are:

- Load Balancing Method of Fast Adjustment: This algorithm is proposedby D. Zhang et al. [6]. It is based on the binary tree which was used for dividing the big region into subdomains. This algorithm adjusts the load between the nodes from the local region to the global region. This algorithm portioned the region according to the binary tree. It must contain parent, child and leaf node. Partition is done between binary tree and cell indexes [15]. The fast adaptive algorithm took less time with the speed of rebalancing of the load is faster. Benefits of this algorithm are low overhead, good speed of balancing with higher efficiency. The drawback of this algorithm is it doesn't maintain the topology of nodes.

| Load Balancing Methods | Parameters | Merits | Demerits |
|---|---|---|---|
| Fast adaptive Load Balancing Method [6] | Efficiency and Communication Cost | Fast Balancing Speed<br>High Efficiency<br>Low Communication overhead | Cannot maintain the topology of the cells |
| Honeybee Inspired Load Balancing Method [7] | Makespace Task Migration Execution Time | Maximizing throughput<br>wait time<br>Minimum Overhead | Low priority load has to stay continuously in queue |
| Dynamic and Adaptive Load Balancing for parallel Files System [8] | Throughput<br>Response Time | High Scalability<br>Reduce the decision delay<br>Resource utilization | Degradation of the whole system due to migration effect |
| Equally Distributing Current Processing [9] | No. of Migrated User and Overload Servers | A very little amount of calculation needed<br>High Speed | Wastage of time<br>Network delay is high |
| Load Balancing in Multiuser Virtual Environment [10] | Clustering Coefficient and No. of Links<br>Shortest Path Length | Network becomes reliable<br>Efficient routing<br>Fault-tolerant | More time in balancing the load |
| Load Balancing in Dynamic Structured P2P Systems [11] | Node Utilization and Load Movement Factor | Increase scalability<br>High node utilization | Assignment of the virtual server is difficult |

**Figure 2:** Comparison of Load balancing technique

- Honeybee Load Balancing Algorithm: The basic idea of this algorithm is the behavioral pattern of a honey bee. Finders and reapers are two kinds of honeybee pattern are found. In the process of collecting honey, finder bees first search the honey source outside and after returning they indicate the availability of honey by doing waggle dance. Now to reap the honey from the source the reapers go outside their honeycomb and if honey is left there they indicate it by the woggle dance. Same as this M.S. George et al. [7,16] proposed a self-organized algorithm which was based on the decentralized honey bee theory. In this, a group made up of virtual server's act as a honey bee. In this algorithm, each VMs maintain the priority of tasks and if an overloaded VMs want to assign priority task to under-loaded VMs, it checks which VMs has a minimum number of high priority task so that task is completed in less time. They formed a load queue sorted in ascending fashion.

Information of available VMs is collected from the datacenter. High execution time, lower overhead and maximum throughput are advantages of this algorithm. The drawback of this algorithm is that a lower priority task always is in a queue if a higher priority task is available.

- Dynamic and Adaptive Load Balancing: This algorithm is made for transferring the files dynamically in a distributed architecture. B. Dong et al [8,17] presented this SALB algorithm for a large file system to handle the issue of file migration. In a parallel file transfer system, various issues like availability and scalability are solved by this algorithm. The central node is responsible for the decision making and if the central node is down the whole system will stop working thus reliability decreases. This issue is resolved as each virtual machine can decide to handle the load because the workload varies randomly. This algorithm addressed the issue of load balancing in the distributive file system but the whole system is degraded because of the side effect of migration.
- Equally Distributing Current Processing: It is a dynamic algorithm [9,18] which handles the load on the basis of priority and priority is decided on the basis of the size of the task. It is a spread spectrum technique which first checks the priority of load and then distributes the load randomly on the under-loaded node.
- Load Balancing Algorithm for Multiple User Virtual Environments: This proposed architecture [10] is a hyper verse system which is responsible for hosting of the virtual world. In which load balancing algorithm was self-organized because of it. The whole network is divided into smaller cells. This network is handled by the public server. The basic idea behind this algorithm is to create a smaller hotspot to calculate the exact load of the object. This algorithm creates lots of overhead is higher initially thus required a large amount of time.

## 3 PROPOSED SYSTEM

### *3.1 Cloud Division and Creating Instances*

The first thing to implement load balancing we must ensure availability. This can be done by dividing the cloud in a different service region. We will use two regions in this system 1) US region 2) Asia region. In this service regions under the different availability zone, we will create the General purpose small instances using the Elastic cloud computing services. An Apache webserver running a web application will be deployed on the instances for analysing the performances.

### *3.2 Load Balancing*

Load balancing will be handled in four-parts:

1. latency based load balancing
2. local regional load balancing
3. auto-scaling to handle the excess load
4. Resource monitoring

- Load Generation: Virtual load will be generated through putty, the terminal emulation software on the different instances from the different PC in the lab.
- Main Load Balancer (Latency based Load Balancing: The main load balancer which is a software load balancer based on the latency in which load is distributed among the different service region based upon the location of request, this is done by DNS resolver and create hosted zone. Latency based routing choose latent region instead of choosing any random service region to forward load for the process. This main load balancer will forward the traffic to the regional load balancer. In this proposed work we have created one hosted zone “www.cloudefy.in” and alias as us.cloudefy.in and asia.cloudefy.in and is created as a name and CNAME for each service region. The request coming for “www.cloudefy.in” will be first resolved by DNS resolver then will be sent to the other. We can also implement the weighted rule in latency based routing to improve load handling.
- Local Load Balancing or Regional Load Balancing: A number of secondary load balancer can be created. This load balancer balanced the load between the instances in a different availability zone. These load balancers can also balance the load in the same availability in the same service region. It can’t balance the load between multiple service regions. Load balancing is done according to the Round Robin algorithm. A load balancer balances the load only for the protocol for which it is configured. The instances health check is performed on the HTTP protocol. In the proposed work we are balancing the load on HTTP /TCP protocol.

### *3.3 Load Handling*

To handle the excess load, the system must have scaling properties. In this proposed work, we are providing the auto-scaling feature by using the Auto-scaling group of EC2. In this, we have defined some scaling policy to scale up and scale down the system. Conditions for auto-scaling can be defined by using different parameter available according to the application and incoming load pattern such as disk read and write up, CPU utilization, network utilization, we will use the maximum CPU

utilization because of our service is web service and we are generating virtual load. Whenever the threshold is breached, the auto-scaling performs an action to accommodate the change in the system. User can also fix the size of the group to secure from application intensive attack.

### *3.4 Amazon Web Services Environment*

Cloud computing has a great impact on the market and most exciting service provider is Amazon web services. In this research, we used amazon's services to develop a load efficient model. AWS is one of the top Infrastructure as service cloud service provider or public cloud service provider, according to information available on internet AWS is a group of remote computing facilities forms a cloud platform which is accessible over the internet. Most known service of AWS is EC2 & S3. Amazon web service divided in service regions and located in 10 different location in the world these regions are US West (Northern California), East Asia(Tokyo),US East (N. Virginia),US West (Oregon), China (Beijing), Brazil (Sau Paulo), Europe (Ireland), Australia (Sydney). Service region means that it containsthe whole country and the cloud services and data will be handled & stored in the same region. Multiple availability zonesare created in each service region to avoided outages between the users.

### *3.5 Load generation*

The virtual load will be generated though putty, the terminal emulation software on the different instances from the different PC in the lab. To generate the load in putty we must install stress component in an apache server. This can be done by the following command.

"Sudo apt-get install stress" and the required load is generated by the "stress –c 80 –m 50 -1= 1000". As shown in figure 3.

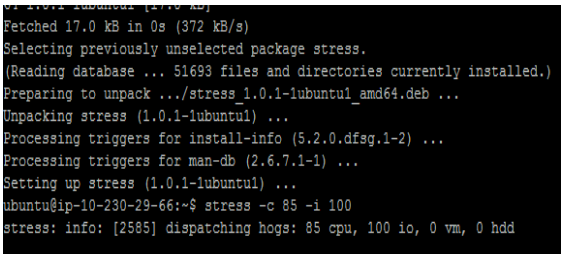


**Figure 3:** Stress Component in Apache Server

### *3.6 Testing*

For testing, we have deployed a simple web application on each instance differentiating the different availability zone. An apache server is installed on these instances by the command "***SUDO APT-GET APACHE2"*** this will install the apache web server on instances. Following image will show the creation of instances. We can connect to our instances by using either DNS address or IP address.

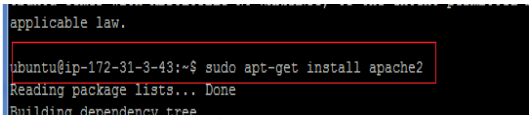


**Figure 4:** Creation of Instances using Apache Server

The main feature of EC2 is its security mechanism which enables user with the inbound and outbound access control. While configuring the security groups and ACLs. While considering the economical point, EC2 provides instance on a very low rate. It has 3 types of instances:

- On-demand instances: On-demand instances are charged on the basis of per hour means this frees the user from the planning complexity.
- Reserved instances: These instances are reserved for frequent use. It needs the only on-time payment for the period of time one can share the reserved instances before the term expires.
- Spot instances: It is based on a bid system for unused instances and uses the instances for the time spot. Price does not exceed their bid system.

## 4. RESULTS & DISCUSSION

With the proxy server and the main load balancer, we search www.cloudefy.in from different geographical regions and move to our nearest hosting service as shown in figure 5, this request leads to our first latency-based routing based in latency to the regional load balancer, second represents us.clodefy.in and asia.cloudefy.in requested by loading process.

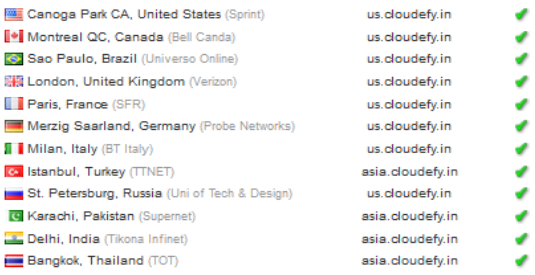


**Figure 5:** Cloudefy on different Servers

The load balancer in regional load balancer distributes the load in round robin fashion, as shown in figure 6. The requested image is randomly generated and thrown on the load balancer.

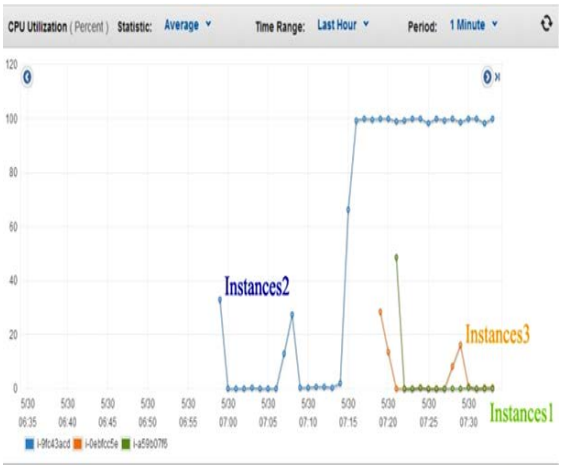


**Figure 6:** The Result on Regional Load Balancer

For example, some virtual loads have been implemented as we can see that the load of the image is distributed in the examples so it is the load graph between the second suggested issues of balancing our EC2 as shown in figure 6.

Auto Scaling Results are collected from the Overview of Auto Scaling Group activity in Cloud Clock. Load pattern and polis are used, this result is in accordance with the load pattern sown in step 5. As we defined at the beginning the size of the primary group was 2 instances. However, we further define that when maximum CPU usage is low, 30% closes an instance that will appear in the fourth line from the bottom of the image. The auto-scaling system automatically launches new instances (5thand 6thlines) when we apply the load. The auto-scaling system closes the newly created instances after removing the litter. (First and second lines) Cloud network system error failed.

## 5. CONCLUSION

This work is focused on the load balancing technique and over the cloud computing and technique used for load balancing. There are lots of technique used like auto-scaling, job-based synchronization, and ant colony technique are used. But when it comes to cloud the performance of load balancing technique is not up to mark. To improve the performance of cloud computing using load balancing we proposed an algorithm. In the proposed algorithm we hosted the web server over the cloud and request is generated from the different client to the server. Cloud web server captures the request and analysis the load on the server based on job nature. It transfers the request of the client to any particular server. We use the Amazon cloud environment for hosted the cloud servers. The result shows that the load balancing technique improves the job execution rate by 20% and when it is compared with the existing load balancing technique then its performance is better than ant colony technique and SJF and FCFS job shedding process.

**REFERENCES**

[1] Eddy Caron, Luis Rodero-Merino, FrédéricDesprez, Adrian Muresan, “Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds”(2012).

[2] Miss.RudraKoteswaramma, “Client-Side Load Balancing and Resource Monitoring in Cloud”, International Journal of Engineering Research and Applications (2012), 2, 6, 167-171.

[3] N. Ajith Singh, M. Hemalatha, “An approach on semi-distributed load balancing algorithm for cloud computing systems”, International Journal of Computer Applications (2012), 56, 12.

[4] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, International Proceedings of Computer Science and Information Technology, Singapore,14, 2011.

[5] Amazon web services cloud watch Web Site,November 2013. (https://aws.amazon.com/about-aws/whats-new/2013/)

[6] Dongliang Zhang, Changjun Jiang,Shu Li, “A fast adaptive load balancing method for parallel particle-based simulations”, Simulation Modelling Practice and Theory (2009), 17, 1032–1042.

[7] DhineshBabu L.D, P. VenkataKrishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Applied Soft Computing, (2013), 2292–2303.

[8] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, “A dynamic and adaptive load balancing strategy for a parallel file system with large-scale I/O servers”, J. Parallel Distribution Computing (2012),72, 1254–1268.

[9] Yunhua Deng, Rynson W.H. Lau, “Heat diffusion based dynamic load balancing for distributed virtual environments, 17th ACM Symposium on Virtual Reality Software and Technology(2010), 203–210.

[10] Markus Esch, Eric Tobias, “Decentralized scale-free network construction and load balancing in Massive Multiuser Virtual Environments”, 6th International Conference onCollaborative Computing: Networking, Applications and Worksharing(2010), 1–10.

[11] Ankit Kumar, Dinesh Goyal, Pankaj Dadheech, “A Novel FrameworkforPerformance Optimization of Routing Protocol in VANET Network”,Journal of Advanced Research in Dynamical & Control Systems (2018),10, 2, 2110-2121.

[12] Pankaj Dadheech, Dinesh Goyal, Sumit Srivastava, Ankit Kumar, “A Scalable Data Processing Using Hadoop &MapReduce for Big Data”, Journal of Advanced Research in Dynamical & Control Systems, (2018), 10, 2 2099-2109.

[13] PankajDadheech, Dinesh Goyal, Sumit Srivastava, C. M. Choudhary,An Efficient Approach for Big Data Processing Using Spatial Boolean Queries”, Journal of Statistics and Management Systems 2018, 21, 4, 583-591.

[14] A. Kumar and M. Sinha, "Overview on vehicular ad hoc network and its security issues," International Conference on Computing for Sustainable Global Development(2014), 792-797.

[15] PankajDadheech et al. “An Enhanced 4-Way Technique Using Cookies for Robust Authentication Process in Wireless Network”, Journal of Statistics and Management Systems (2019), 22, 4, 773-782.

[16] Ankit Kumar et al., “An Enhanced Quantum Key Distribution Protocol for Security Authentication”,

Journal of Discrete Mathematical Sciences and Cryptography(2019), 22, 4, 499-507.

[17] Ankit Kumar, Pankaj Dadheech, Vijander Singh, Ramesh C. Poonia, LineshRaja, "An Improved Quantum Key Distribution Protocol for Verification", Journal of Discrete Mathematical Sciences and Cryptography (2019), 22, 4, 491-498

[18] Ankit Kumar and Madhavi Sinha, "Design and analysis of an improved AODV protocol for black hole and flooding attack in vehicular ad-hoc network (VANET)", Journal of DiscreteMathematical Sciences and Cryptography(2019), 22, 4, 453-463