International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

Adaptive Load Balancing Using RR and ALB: Resource Provisioning in Cloud

Santosh T. Waghmode[1*], Bankat M. Patil[2]
[1]Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad, Maharashtra,India
stwaghmode@gmail.com
[2] Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad, Maharashtra,India
patilbankat@gmail.com

Abstract- Cloud Computing context, load balancing is an issue. With a rise in the number of cloud-based technology users and their need for a broad range of services utilizing resources successfully or effectively in a cloud environment is referred to as load balancing, has become a significant obstacle. Load balancing is crucial in storage systems to increase network capacity and speed up response times. The main goal is to present a new load-balancing mechanism that can balance incoming requests from users all over globally who are in different regions requesting data from remote data sources. This method will combine effective scheduling and cloud-based techniques. A dynamic load balancing method was developed to ensure that cloud environments have the ability to respond rapidly, in addition to running cloud resources efficiently and speeding up job processing times. Applications' incoming traffic is automatically split up across a number of targets, including Amazon EC2 instances, network addresses, and other entities by elastic load balancing. Elastic load balancing offers three distinct classifications of load balancer, and each one provides high availability, intelligent scalability, and robust security to guarantee the error-free functioning of your applications. Application load balancing and round robin are the two load balancing mechanisms in database cloud that are focus of this research study.

Keywords – AWS, Application Load Balancer, Round Robin, EC2, Amazon RDS.

I. INTRODUCTION

The technologies of distributed computing, server virtualization, and network storage all have worked collectively to become the cloud computing platform. The basic objective of cloud computing is to provide several kinds of IT solutions by scheduling and setting up an aggregate of computing resources, improving the workload for users and helping them to focus on their primary operations [1] [2]. The features of cloud computing include scalability, region independence, adaptability, device independence, elasticity, dependability, resource sharing, cost effectiveness, and on-demand computing, to point out several of them. The usage of cloud computing is growing quickly as a result of these various factors. Load balancing [1] ensures that no node in the cloud computing environment becomes overloaded or inactive for a long amount of time by balancing the workload among different servers in the environment. Each node in the system can perform the exact same amount of work according to a load balancing algorithm. In order to increase both the total appropriate response time and minimal resource utilization, the LB algorithm's goal is to allocate the jobs supplied to the cloud domain between the available resources. Load balancing has been identified as one of the most significant issues when using the cloud because the reason that it is difficult to estimate how many requests will be made per second in a cloud environment. Load balancing in database cloud is designed to dynamically balance the load among the nodes in order to fulfill client requirements and achieve maximum resource efficiency. This is done by sharing the full amount of available load across different nodes.

Load balancing [22] is an algorithm that distributes the workload equally throughout all of the active nodes in the system. The purpose of load balancing is to increase satisfaction for users. Utilizing optimal use of the resources that are available, a reliable or best load balancing algorithm helps to ensure that no node becomes overloaded or under loaded. To distribute the load among multiple machines, different load balancing algorithms [2] have been developed. However, an ideal load balancing algorithm that distributes the load equally throughout the system has been completely developed.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

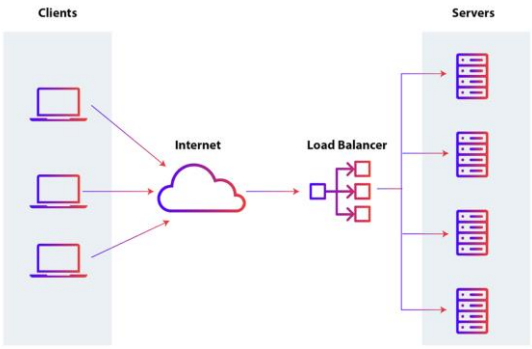

Fig. 1 Load Balancer [22]

An algorithm called load balancing ensures server instances in a cloud computing environment make better use of their resources. Load balancing is a vital method for making sure that task distribution is balanced in the cloud environment while maximizing usage of resources. Applying load balancing in cloud systems minimizes data sending and receiving delays [3]. Three rules compose a mechanism that balances workloads. These are the distribution rule, the selection rule, and the location rule [2, 5]. We can use the selection rule in a proactive or non- proactive approach.

The non-proactive rule always picks up the newly created process, but the proactive rule could pick up the continuing process. Proactive rule transfers are more expensive than non-preemptive transfers, which are the better option. Preemptive transfer, however, can occasionally be superior to non-preemptive transfer. Practically, location and distribution rules work together to decide on load balancing. When a user requests to connect to the server, the load balancer is activated. The test instance in Fig. 1 is presented. Reduce the make span time of upcoming requests and enhance average resource usage are the primary objectives of load-balancing algorithms. The process of load balancing has two steps: first, the workload must be distributed among the nodes (this process is known as task scheduling); second, the virtual machine has to be tracked while the load balancing operation is being carried out using a task scheduling round robin approach [3]. The following section of the article is as follows: chapter 2 discusses the literature review; and chapter 3 and 4 presents the proposed and implemented load balancing algorithm. In the fifth chapter, the modeling tool and experiment results are covered, and Section 6 deals with the conclusion.

II. LITERATURE REVIEW

In [1], the researcher team of a study set up Amazon EC2 instances and made effort to use them in order to outline how to access and use an instance. In addition, the authors of [2] shown that cloud clients don't use attention while choosing EC2 instances. They observed that by transmitting an unauthorized instance, it could be started again and again and that usage data could be gathered. The authors discussed how to use graph techniques to solve security concerns with Amazon EC2 in respect to the deployment of pictures.

In [1], studied a typical setup of the widely used AWS cloud with a security-focused method. The security of the cloud was examined by the installation of honeypots on the given systems for a number of months. Three AWS EC2 instances were running simultaneously in the cities of Singapore, Virginia East, and Paulo Sao in the article experiment comparison based on the number of connections and maximum connections from a single IP address [7].

In [2], this article the many load balancing topologies provided by Amazon Elastic Load Balancing (ELB), including Network load balancing [NLB], application load balancing [ALB], and classic load balancing [CLB]., are covered in this article. The study focuses on their unique features, performance metrics, and factors to take into account while selecting the best load balancer for different applications [9].

In [3], studied the most recent technologies used in and related to the cloud the workplace. According to the authors, it's crucial for maintaining secure firewalls, upgraded operating systems, and current application configurations. The article states that AWS helps clients to independently operate systems. While it could prove functional for the user, it could create safety issues for a larger group. The authors note the need of protecting each component of a cloud-based system because a single security mistake could compromise the infrastructure as a whole [14].

In [6], this article paper discusses various load balancing architectures offered by Amazon Elastic Load Balancing (ELB), such as Network load balancing [NLB], application load balancing [ALB], and classic load balancing [CLB]. The study focuses on their features, performance metrics, and considerations for choosing the right load balancer for different use cases. In [10], the article compares different load balancing techniques used by AWS for distributing workloads and dynamic resource availability. The authors analyze the performance of algorithms like Round Robin, Least Connections, and Weighted Round Robin, focusing on metrics like performance, respond time, and consumption of resources [12].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

In [11],Although this article review covers load balancing in cloud computing more broadly, it provides valuable insights into various load balancing algorithms and techniques utilized in AWS. It compares different algorithms' strengths and weaknesses, highlighting their suitability for different types of workloads. This article explores the potential of leveraging artificial intelligence techniques to enhance load balancing efficiency [16].

In [11], this paper evaluates the performance of Amazon's Elastic Load Balancing service under varying workloads and scenarios. It analyzes the system's overall performance in terms of scaling ability, response times, and the effects from different configurations. The comprehensive analysis of Amazon's Elastic Load Balancing features, compares the performance of its different types, and discusses real-world scenarios where they can be applied effectively by authors [14].

In [19], highlighted a few main security concerns that cloud providers in 2014 needed to address. The authors focused into different aspects of cloud security, such as Information security, issues with the cloud, Security of accounts and cloud architecture.

In [12], Intruders executed an XSS stealing attack against AWS. The Amazon Relational Database Service (RDS) was also hacked, giving the attackers an entry point into the Amazon framework even if they lost their special access. Two solutions to identical assaults against AWS are presented by the authors as their conclusion. Initially AWS should never permit customers and services to share login information for accounts. A two-factor authentication system is also recommended. Twenty security standards, including ISO 27001, HIPAA, DoD CSM, and PCI DSS, were examined and certified by AWS, according to the authors' analysis of the security situation with AWS [22].

1. LOAD BALANCER

The technique of equally distributing workloads across computing resources in a cloud computing environment while appropriately allocating network traffic to those resources is known as load balancing. Another advantage of load balancing is the ability to allocate workloads among two or more different sites.

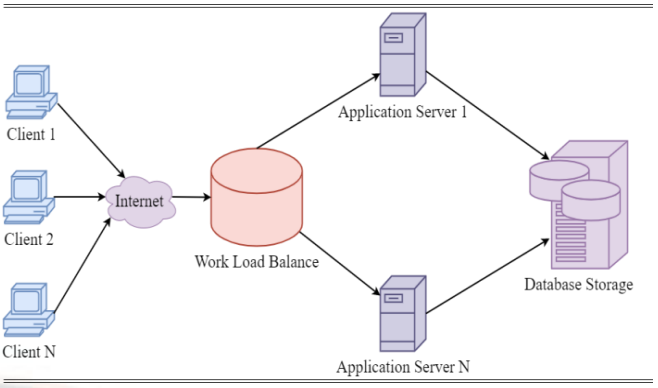

Fig. 2 Load Balancing System Model [3]

There are various load balancing algorithms that handle traffic in various ways. The two primary kinds of load balancers are static and dynamic load balancers as shown in Fig 2. Static load balancing uses algorithms that already know about the existing servers in the dispersed network to divide the incoming demand on a server. But more flexible method of load balancing is the use of a dynamic load balancer, which can determine on-the-fly how much weight, needs to be shed and which system should take it. Job assignments are made in run time. Depending on the situation, jobs are distributed more evenly at runtime such that the load is moved from strongly loaded nodes to lighter loaded nodes.

1.1 ELASTIC LOAD BALANCER

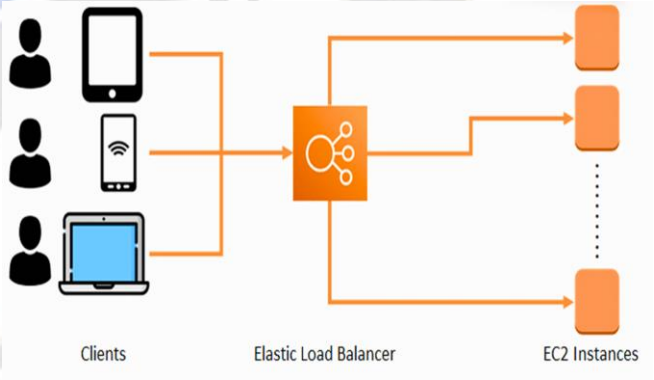

Fig. 3. Elastic Load Balancer [22]

Over the last few years, load balancing methods for cloud systems have been developed for static as well as dynamic environments. There are many different paradigm dynamic load balancing algorithms available, but each one has drawbacks that demonstrate how no single algorithm can accomplish all the objectives simultaneously because certain parameters, like time and expenses, cannot be increased or decreased. Elastic load balancing is used to automatically allocate all incoming traffic among several targets, have EC2 instances, packages, and IP addresses in multiple Availability Zones [Fig.3].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

Elastic Load Balancer provides three distinct types of load balancers:

- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

Load Balance type	Description
Elastic Load Balancer [ELB]	AWS LB is another name for it. It splits up the incoming work load among several Amazon EC2 instances. ALB, NLB, and CLB are the three main load balancer types provided through Amazon ELB.
Network Load Balancer [NLB]	In OSI model, NLB operates at Layer 4, sometimes known as the network layer. It is ideal for distributing TCP traffic's load. NLB assigns a static IP address to each subnet, which applications can utilize as the load balancer's interface IP address. In order to keep from overloading, it is used to distribute network traffic across many systems in a cluster.
Application Load Balancer [ALB]	At OSI model layer 7, ALB is implemented. Advanced load balancing of HTTP and HTTPS connections requires it. ALB ensures that latest and most recent ciphers and protocols are always used to interpret and enhance the application's security.
Classic Load Balancer [CLB]	Simple and straightforward load balancing across several Elastic Cloud Compute (EC2) instances is made possible by CLB. Both the request point level and the connection level have controls. Applications running on the EC2-Classic network are able to utilize CLB.

Table I. Details of load balancers [22]

1.2 APPLICATION LOAD BALANCER

The application layer, the seventh stage in the OSI paradigm, is where an application load balancer operates. The LB ranks the listener rules before deciding which rule should be applied when it receives a request. The target that is most suitable for the kind of load will be selected from the target group and sent there with the highest level of performance. The round-robin routing algorithm is the default. Without rearranging the usual sequence of requests to your application, can be add and remove objects from your load balancer as your requirements change. The load balancer can be modified through ELB as the volume of traffic improvements to your application through time. A significant number of workloads can be expanded automatically using elastic load balancing. Fig 4 Application Load Balancer is done by creating EC2 instances.

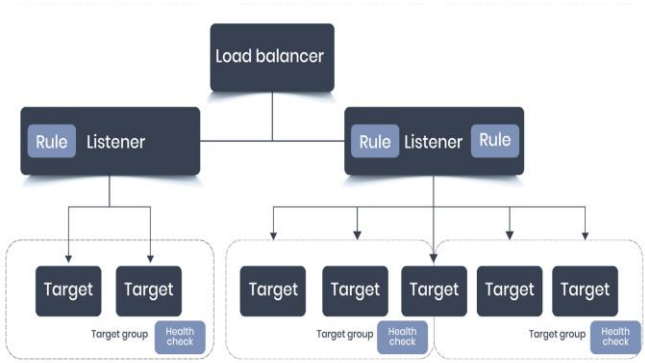

Fig. 4. Application Load Balancer [3] [22]

Various tools can be used to simulate and illustrate how load balancing functions as well as to offer the best advice for employing load balancers. With load balancing, problems in the real world are solved in real time. Nowadays Internet has become so common that everyone started using it for various reasons. It's necessary that servers should handle all the traffic using load balancing techniques.

To provide improvised solution for efficient load balancing tools as following can be used:

- Cloud Analyst
- Cloud Project
- Amazon Web Service

1.3 AMAZON RDS:

Amazon RDS, or Amazon Relational Database Service, is a managed database service offered by Amazon Web Services (AWS). It is made to make setting up, running, and scaling relational databases on the cloud faster. Amazon RDS is widely used by startups, enterprises, and developers to deploy and manage relational databases in a flexible and cost-effective manner. Its managed nature and integration with other AWS services make it a popular choice for building and scaling applications that rely on relational databases. Key aspects of Amazon RDS include:

Multiple Database Engines: Amazon RDS supports a wide range of popular relational database engines, including MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and MariaDB.

High Availability: Amazon RDS offers high availability features such as Multi-AZ deployments. In Multi-AZ mode, the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

database is automatically replicated to another Availability Zone's ready instance, providing failover support in case of hardware or Availability Zone failures.

Monitoring and Metrics: Amazon RDS provides monitoring and metrics through Amazon CloudWatch. You can track metrics related to CPU utilization, memory, storage, and more, helping you understand the performance of your database.

Round Robin Load Balancing: RR is a load balancing algorithm that distributes incoming traffic equally to a group of backend servers in a circular order. Each server receives its turn in a sequential manner. While RR is easy to implement and doesn't require much computational overhead, it may not be the most efficient load balancing algorithm for all scenarios. For example, if some servers are more powerful or capable of handling more traffic than others, RR won't take this into account.

Combining ALB with RR: utilize a load balancer for applications to implement Round Robin load balancing to Set up ALB: Create an Application Load Balancer on AWS cloud platform. Define target groups and associate them with backend resources such as EC2 instances, containers and Configure Round Robin. The ALB itself inherently distributes traffic across the registered targets using a form of Round Robin. It automatically routes incoming requests in a cyclic order to the registered targets within the associated target group. While ALBs use a form of Round Robin to distribute traffic among registered targets, have more advanced load balancing algorithms also available, such as Least Outstanding Requests, Least Connections, and Lambda Function Weighted[Fig.11].

2. IMPLEMENTATION

The approach to be used to implement load balancing are as follows:

- Set up a listener and a load balancer.
- Set Security Rules for an HTTPS Listener
- Form a Security Group.
- Create a Target Group
- Develop targets for the target group.
- Make a load balancer.

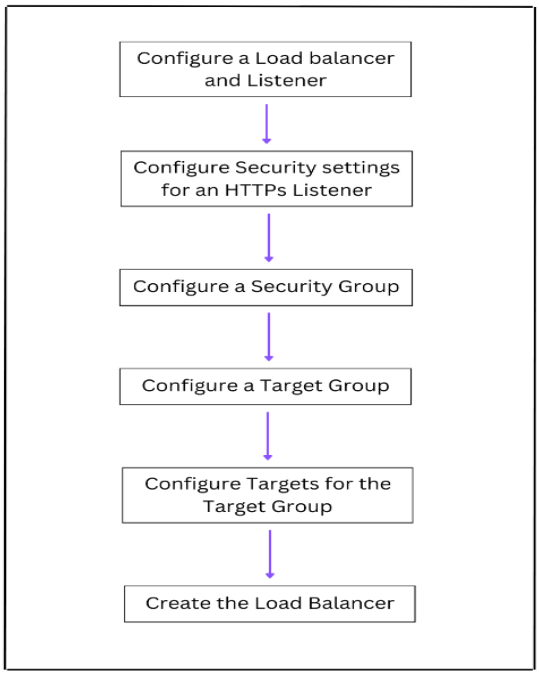

Fig. 5. Implementation Load Balancer [22]

As shown in Fig.5. Using the deployment of three applications as an example, requests are balanced and sent based on the load and application, to the appropriate servers. Three services as Register_Server1, Server2 and Server3 are created. Three servers are built using EC2 instances, and every server having a load balancer configured.

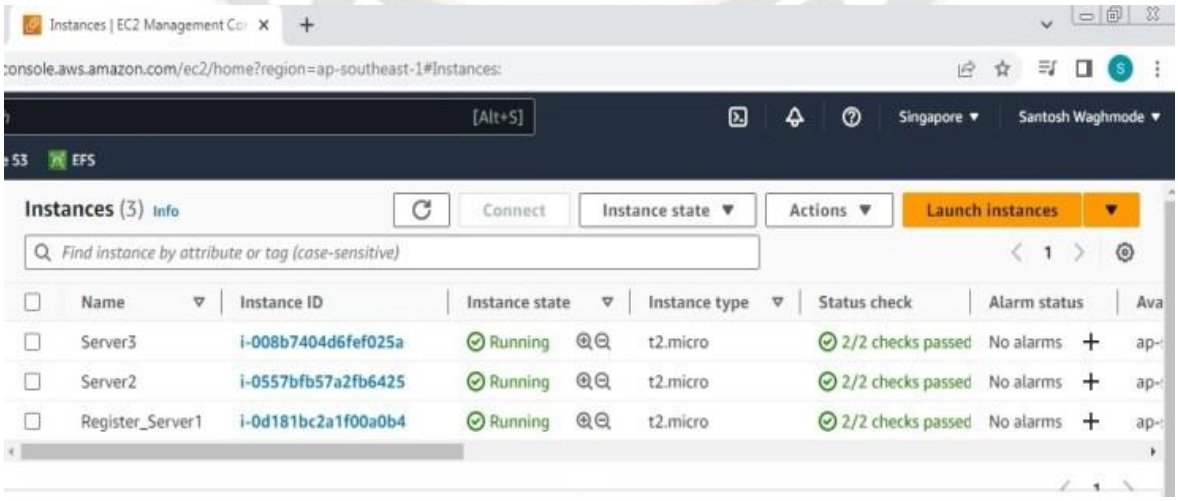

Fig. 6. EC2 instance

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

Shown in Fig. 6. A new EC2 instance is built. with the respective names Register_Server1, Server2 and Server3. These two instances are virtual servers used as micro services. The 'running' status of these three servers continues to be maintained. Once the instances have been created, specific targets must be configured.

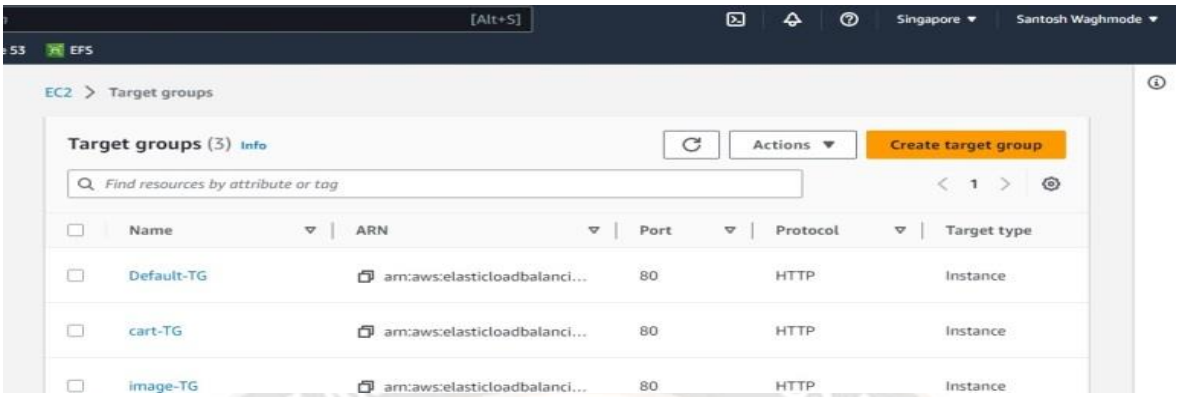

Fig. 7. Target Groups

As shown in Fig.7 targets are created with respective names as Register_Server1, Server2 and Server3. When a target is made, the application load balancer is attached to the targets. The state should be active and the type has been retained to be application. While listeners must be added to the desired targets before load balance may balance the traffic, a first script is built that achieves the main goal of balancing based on the user's request. To route the balancer to an HTTP web page with the specified headers, sample example scripts are shown in table II. The server migrates to the suitable instances depending on the requests received.

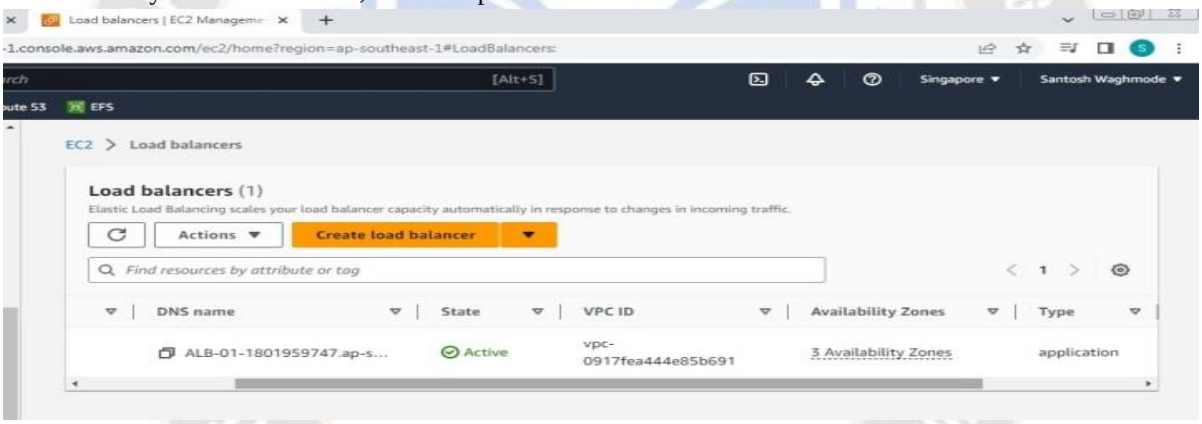

Fig. 8. Load Balancer

The final step is to create the listener and the rule. The requests by the users are made to the ports, the listers will listen the ports. Rules are created which includes the conditions of directing the paths and balancing the path based on the http requests. As shown in Fig.9 the rules are created to direct the paths.

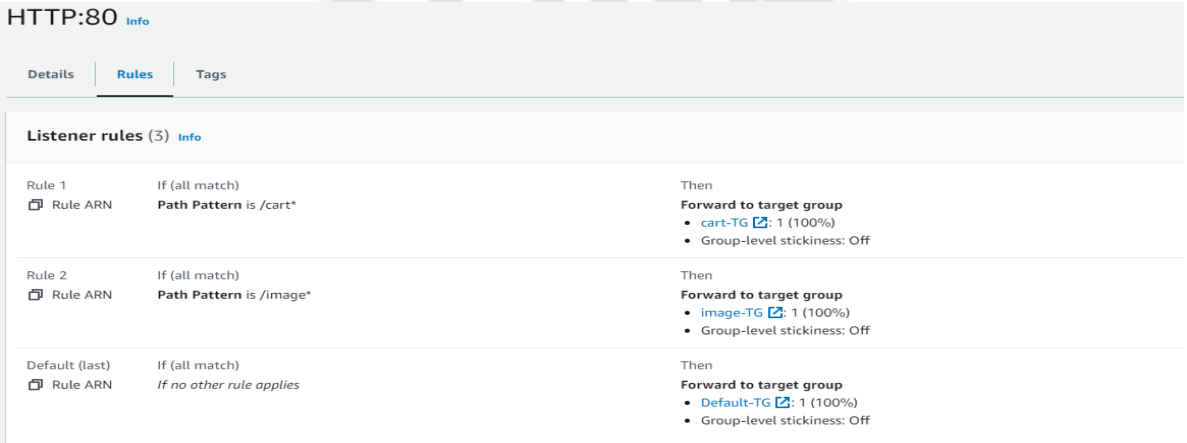

Fig. 9. Creating Listener Rules

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

If the http contains the register data then the request is directed to Register_Server1 service, similarly the Server2 data then the request is directed to Server2 service and if request contains Server3 information then request is directed to Server3 service predefined rule based on algorithm flowchart as shown in Fig.11.

REGISTER_SERVER1

```
#!/bin/bash
 yum install httpd -y
 service httpd start
 chkconfig httpd on
 mkdir / var/ www/ html/register
 echo"<html><h1> Registration Server</h1>
</html>"> /var/ www/ html/register/index.html
```

SERVER2,

```
#!/bin/bash
 yum install httpd -y
 service httpd start
 chkconfig httpd on
 mkdir  /var/ www/ html /server2
 echo"<html><h1>This is the Server2</h1>
 </html>">/ var/www /html/image/index.html
```

SERVER3,

```
#!/bin/bash
 yum install httpd -y
 service httpd start
 chkconfig httpd on
 mkdir / var /www /html/server3
 echo"<html><h1> This is the Server3</h1>
 </html>">/var/www/html/cart/index.html
```

Table II. Server script [22]

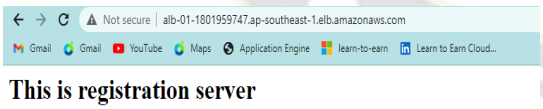

Fig. 10, Register Service [22]

III. SIMULATION RESULTS

According to the application load balancing the balancing of load is done right from the initial part when the requests are read. Thus, the HTTP requests is the starting point where the traffic is balanced. Services and targets with the name Register_Server1, Server2 and Server3 is created as shown in Fig. 06, if the link or http requests contain the Register_Server1, Server2 and Server3 details then automatically the request are directed to respective micro services and hence the message is displayed. Once the load is balanced and the requests are redirected then scripts are read from respective services then details are sent back to the user.

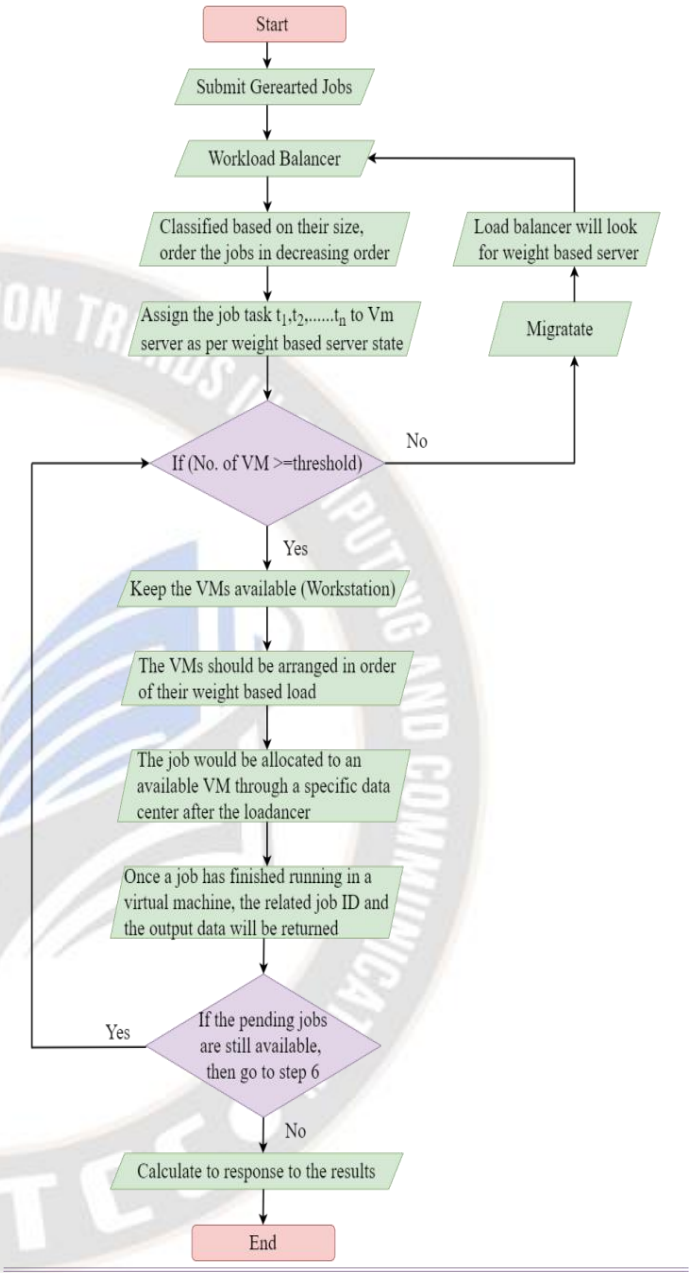

Fig. 11 Adaptive Load Balancer Algorithm [3]

Performance Metrics: define the performance metrics used to evaluate the effectiveness of each load balancing strategy. These could include response time, throughput, server utilization, and request distribution fairness.

Throughput: Throughput refers to the rate at which requests are successfully processed and responded to. To analyze throughput, with following factors:

Request Rate (R): The rate at which requests arrive at the load balancer.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
***Article Received**: 08 May 2023 **Revised**: 26 June 2023 **Accepted**: 12 July 2023*

Server Processing Time (S): The time taken by a server to process a single request.

Number of Servers (N): The total number of backend servers.

Throughput (T): can be estimated using little's law:

$$T = R * (1 - P)$$

Where P: is the probability that all servers are busy

$$P = \frac{R * S}{N}$$

Scalability: Scalability refers to how well the load balancing algorithm handles an increasing number of requests and resources. We could analyze scalability by examining how the system's performance changes as the number of servers or requests increases.

Server Scalability: As you increase the number of servers (N), observe how the throughput and response times change.

Request Scalability: Increase the request rate (R) and study how the system copes with higher loads.

To use a law to find out the transmission load ratio (TLR):

$$TLR = Assign\ load\ /\ observation\ time$$

Use the formula below to determine each server's capacity as well as the combined capacity of all servers:

$$Capacity\ of\ VMs\ = S * N + BWij$$

Where S: server speed,
N: No. of process used,
BW: bandwidth

Resource Utilization (RU): Resource utilization measures how effectively the available resources are being used. It can be calculated for a single resource or across multiple resources. CPU utilization measures the extent to which a server's processing capacity is being used.

For Round Robin, CPU utilization can be modeled as evenly distributed across servers, assuming each request takes approximately the same amount of CPU time.

For ALB, CPU utilization can be more complex, depending on how it intelligently routes requests. The ratio of a resource's busy time to its entire time can be used to calculate it.

Mathematically, RU for a resource 'r' can be calculated as:

$$RU_r = (total\ busy\ time\ of\ resource\ 'r')/(total\ time)$$

$$RU = (\ resource\ used\ /\ total\ capacity) * 100\%$$

Utilization of the server machine is calculated of the process using the outcomes results as stated in Fig. 20.

$$U = \sum_{j=1}^{Mi} (Ui \ldots . . j)$$

Makespan (MT): Makespan refers to the total time taken to complete a set of tasks or jobs. It may stand in for the amount of time needed to complete a specific workload when used in the context of load balancing.

The equation for makespan is straightforward:

$$MT = Finish\ time\ of\ the\ last\ task - start\ time\ of\ the\ first\ task$$

Mathematically, if you have 'n' tasks and each task takes time 't_i' to complete, then the makespan is given by:

$$MT = \max(t_1, t_2, \ldots \ldots \ldots . , t_n)$$

Execution Time (ExT): Execution time is the time taken to complete a specific task or request. It is frequently used to measure how effective load balancing algorithms. The equation for execution time is:

$$ExT\ for\ each\ job = Job\frac{size}{P} * N$$

Total Load on Target Instances: The total load on server machines is influenced by factors like the number of active connections and the processing time of each request. For Round Robin, you can assume that the load is evenly distributed, while for ALB, you might need to consider the distribution logic more explicitly.

To determine the total load on all server machines using this equation:

$$Total\ Load = \sum_{j=1}^{m} Load\ of\ VMs$$

Assign load at the virtual machine is:

$$Load\ = Number\ of\ job * Length\ of\ Job\frac{size}{P*N}$$

Load-balancing (salb), improves load-balancing for web servers. The load balancing solution in mathematical.

$$Ls = fsini$$

$$Lc = No.of\ request\ per\ unit\ time$$

Load considered in salb is:

$$L = f(Lc, Ls)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

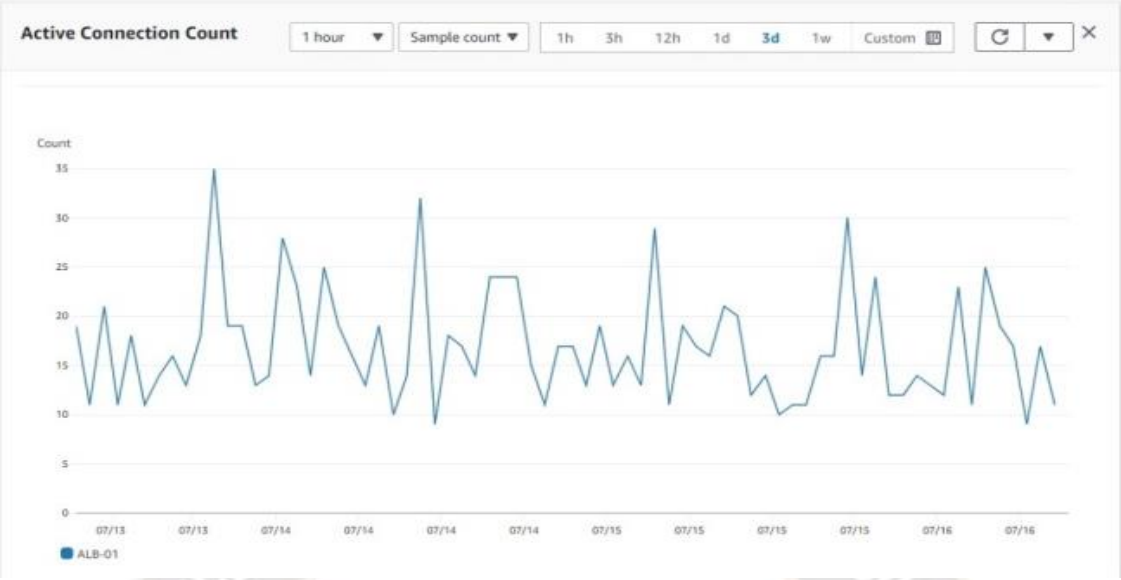

Fig.12, DB Connection

Fig.12, shown an active connection count graph in a load balancer provides the current number of active connections that the load balancer is handling over a specific period. The graph typically shows the following information: Time Period: The x-axis of the graph represents the time period for which the active connection count is being measured. Active Connection Count: The y-axis represents the number of active connections. It displays the total number of connections that the load balancer is presently handling at any given moment.

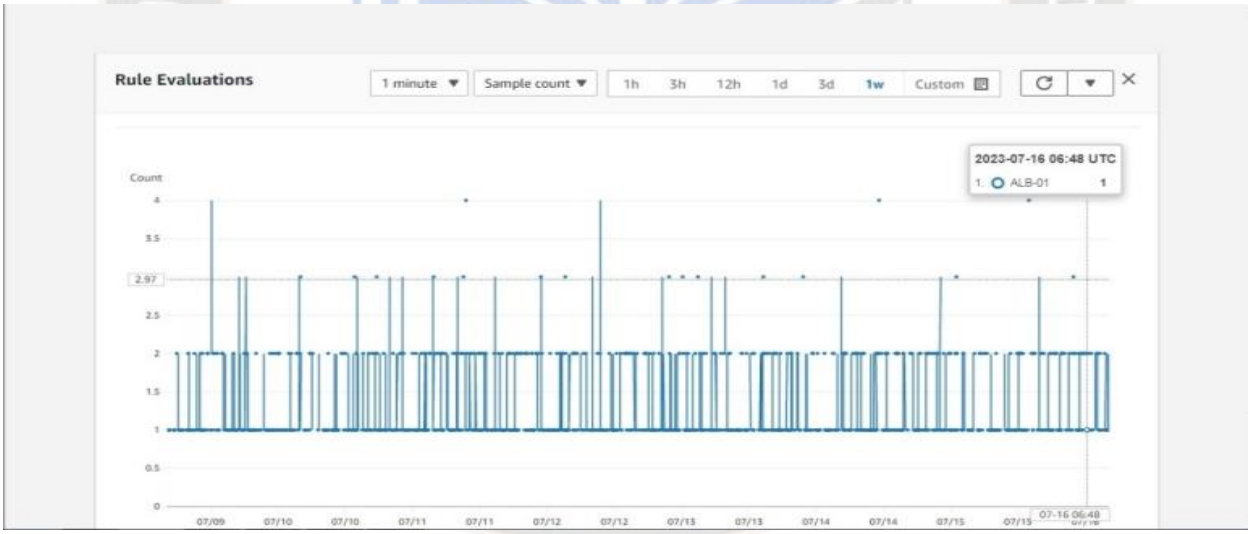

Fig.13, Rule Evaluation

Fig.13 represents the results of Time Period (x axis) and Rule Evaluations Count (y axis). Time Period: The x axis represents the time period for which the rule evaluations are being measured. Rule Evaluations Count: The y-axis shows the total number of times the load balancer evaluates its rules during each time interval.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

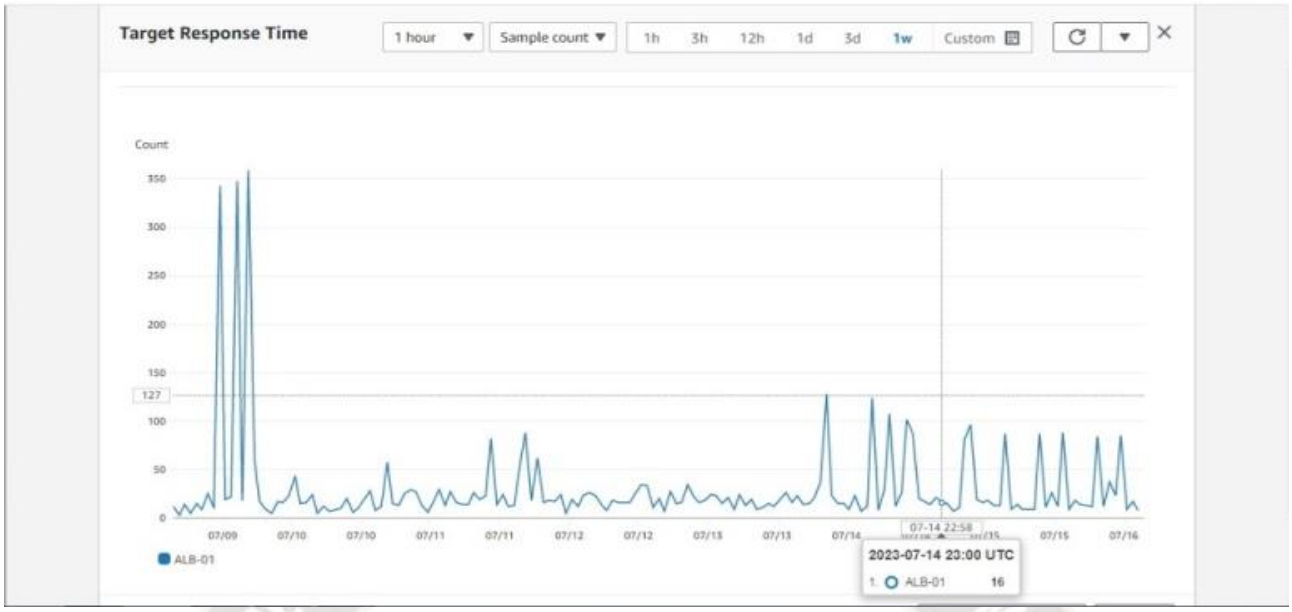

Fig. 14 Response Time by Instances

Fig.14, represent the response time of the backend instances or targets as seen by the load balancer. The response time represents the time it takes for a backend server to process a request and send a response back to the client.

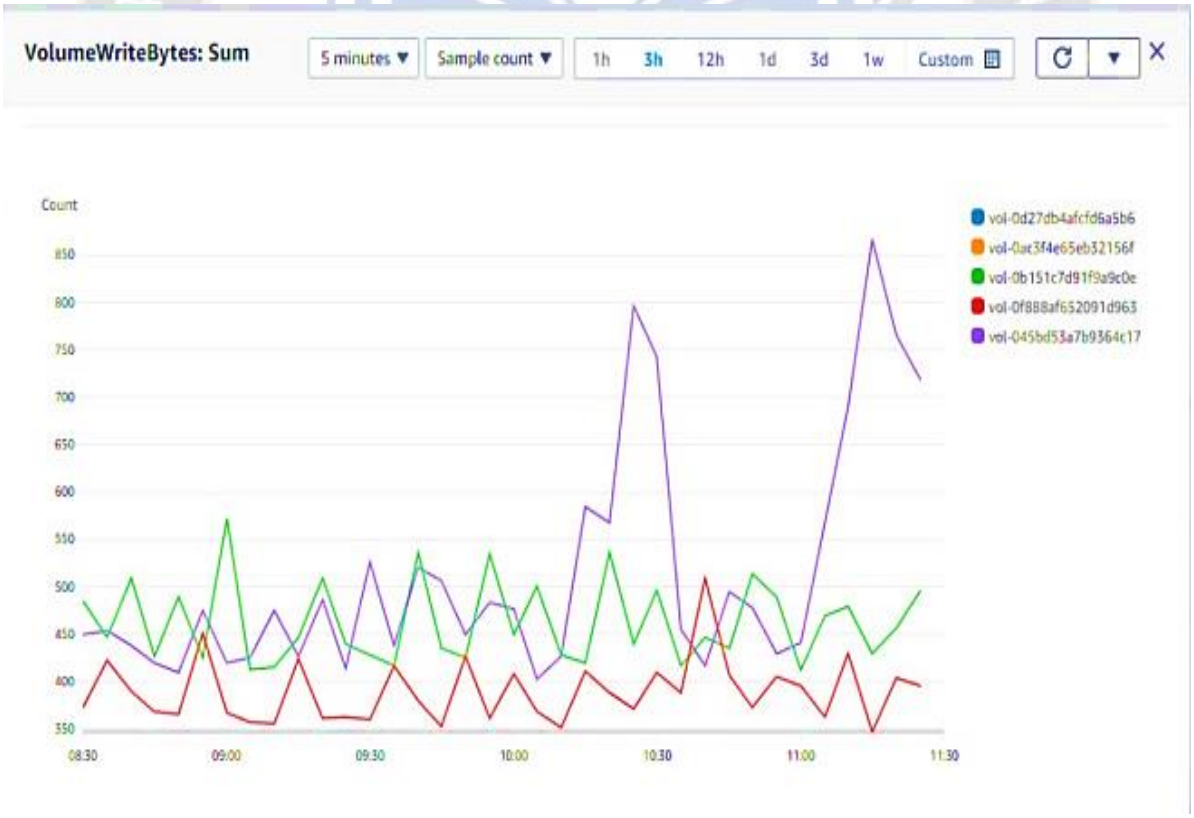

Fig.15, Request and Response by Instances

Fig.15, represent the display the total amount of data written by the load balancer to the backend instances or servers over a specific period. It represents the aggregated data size of requests or responses that the load balancer has written to the backend targets.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

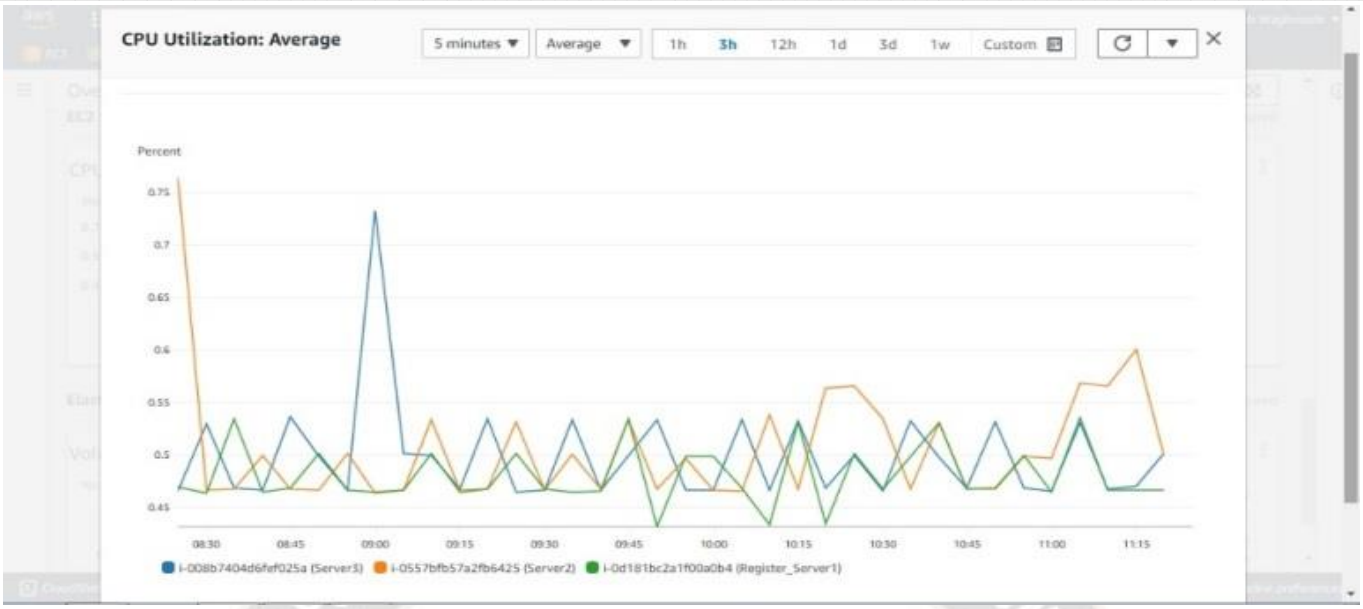

Fig.16, CPU Utilization

Fig.16, represent CPU utilization graphs since their primary role is to distribute traffic and manage connections between clients and backend servers. Load balancers are designed to be lightweight and efficient, with minimal processing requirements. The CPU utilization graph for backend instances helps you understand how much of the server's processing power is being utilized.

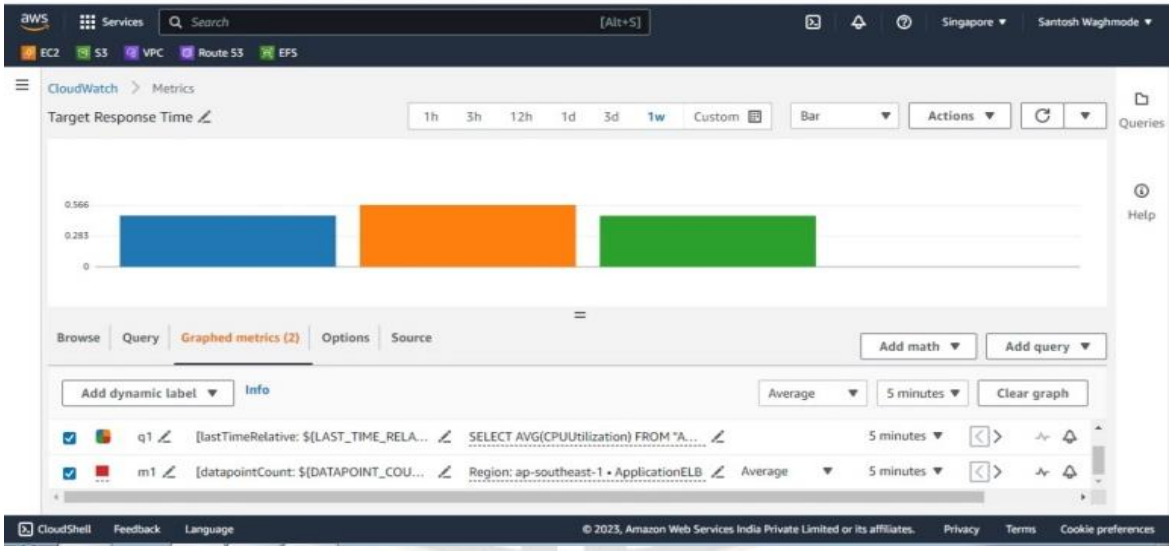

Fig.17, Response Time

Fig.17, represent a target response time graph in a load balancer provides insights into the response times of individual backend instances or servers that are handling incoming requests, measures the CPU utilizations using equation. The x axis shows the time period for which the response times are being measured. The y axis shows the response time of the backend instances. It shows the time taken by each backend to process and respond to incoming requests. The response time is usually measured in milliseconds.

Table III. Summary of result analysis

Load balancing algorithms	Target response time	Performance	Resource utilization	reliability
	In ms	throughput	In percent	Accuracy
Static LBA	max	min	max	min
Dynamic LBA	moderate	avg	avg	avg
ALB [AWS]	min,0.566ms [Fig.17]	high, [Fig.18]	min, 74 % [Fig.16]	high

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7940
Article Received: 08 May 2023 Revised: 26 June 2023 Accepted: 12 July 2023

IV. CONCLUSION

Findings and outcomes from the proposed LB algorithm are presented in this article, which stated that the conclusion of the article. The LB technique that was proposed in this article successfully distributed the incoming jobs among multiple servers located in cloud environments. Task scheduling can enhance the load balancing technique and enhance effective cloud resource use. The goal of this article was to present helped improve resources utilization using load balancing algorithm. Results showed that, when compared to the current Dynamic LBA, our technique minimizes time-span and provides efficient use resource utilization up to 74% [Fig.16 & Fig.17]. It demonstrates that the proposed approach works well in a dynamic cloud environment where user requests arrives in a predetermined order and increase in length gradually. When comparing the proposed technique with the current approach, large-scale requests are capable of being responded to by the algorithm.

REFERENCES

[1] Hemalatha, M. "An approach on semi-distributed load balancing algorithm for cloud computing system." International Journal of Computer Applications 975 (2018): 8887.
[2] Al-Rayis, Ektemal, and Heba Kurdi. "Performance analysis of load balancing architectures in cloud computing." 2013 European Modelling Symposium. IEEE, 2013.
[3] Waghmode, Santosh T., and Bankat M. Patil. "Adaptive Load Balancing in Cloud Computing Environment." International Journal of Intelligent Systems and Applications in Engineering 11.1s (2023): 209-217.
[4] Kaur, Rajwinder, and Pawan Luthra. "Load balancing in cloud system using max min and min min algorithm." International Journal of Computer Applications 975 (2014): 8887.
[5] Ristenpart, Thomas, et al. "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds." Proceedings of the 16th ACM conference on Computer and communications security. 2009.
[6] Bleikertz, Sören, et al. "Security audits of multi-tier virtual infrastructures in public infrastructure clouds." Proceedings of the 2010 ACM workshop on Cloud computing security workshop. 2010.
[7] Balduzzi, Marco, et al. "A Security Analysis of Amazon's Elastic Compute Cloud Service–Long Version–." Publication: ACM (2011): 10.
[8] Ms. Ritika Dhabalia, Ms. Kritika Dhabalia. (2012). An Intelligent Auto-Tracking Vehicle. International Journal of New Practices in Management and Engineering, 1(02), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/5
[9] Al Awadhi, Eman, Khaled Salah, and Thomas Martin. "Assessing the security of the cloud environment." 2013 7th IEEE GCC Conference and Exhibition (GCC). IEEE, 2013.
[10] Randles, Martin, David Lamb, and Azzelarabe 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. IEEE, 2010.
[11] Serrano, Nicolas, Gorka Gallardo, and Josune Hernantes. "Infrastructure as a service and cloud technologies." IEEE Software 32.2 (2015): 30-36.
[12] Kashani, Mostafa Haghi, and Ebrahim Mahdipour. "Load Balancing Algorithms in Fog Computing." IEEE Transactions on Services Computing 16.2 (2022): 1505-1521.
[13] Sharma, Pradeep Kumar, et al. "Issues and challenges of data security in a cloud computing environment." 2017 IEEE.
[14] 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). IEEE, 2017.
[15] Panda, Sanjaya K., and Prasanta K. Jana. "Load balanced task scheduling for cloud computing: A probabilistic approach." Knowledge and Information Systems 61.3 (2019): 1607-1631.
[16] Ghomi, Einollah Jafarnejad, Amir Masoud Rahmani, and Nooruldeen Nasih Qader. "Load-balancing algorithms in cloud computing: A survey." Journal of Network and Computer Applications 88 (2017): 50-71.
[17] Waghmode, Santosh T., and Bankat M. Patil. "Load Balancıng Technique in Distributed Systems: A Review." 2021 2nd Global Conference for Advancement in Technology (GCAT). IEEE, 2021.
[18] Kumar, Mohit, and Subhash Chander Sharma. "Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment." International Journal of Computers and Applications 42.1 (2020): 108-117.
[19] Waghmode, Santosh T., and Bankat M. Patil. "Optimized and adaptive dynamic load balancing in distributed database server." (2022): 145-149.
[20] Naaz, S. ., ShivaKumar, K. B., & B. D., P. . (2023). Aggregation Signature of Multi Scale Features from Super Resolution Images for Bharatanatyam Mudra Classification for Augmented Reality Based Learning. International Journal of Intelligent Systems and Applications in Engineering, 11(3s), 224–234. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2565
[21] Mishra, Sambit Kumar, Bibhudatta Sahoo, and Priti Paramita Parida. "Load balancing in cloud computing: a big picture." Journal of King Saud University-Computer and Information Sciences 32.2 (2020): 149-158.
[22] Shukur, Hanan, et al. "Cloud computing virtualization of resources allocation for distributed systems." Journal of Applied Science and Technology Trends 1.3 (2020): 98-105.
[23] Narhe, Aditya, et al. "SQL Query Formation for Database System using NLP." International Journal of Engineering Research and 8 (2019).
[24] Andrew Hernandez, Stephen Wright, Yosef Ben-David, Rodrigo Costa, David Botha. Risk Assessment and Management with Machine Learning in Decision Science. Kuwait Journal of Machine Learning, 2(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/196
[25] Sharma T, Banga VK. Efficient and enhanced algorithm in cloud computing. International Journal of Soft Computing and Engineering (IJSCE) ISSN. 2013 Mar:-2231-307.

[26] Domanal SG, Reddy GR. Optimal load balancing in cloud computing by efficient utilization of virtual machines. In2014 sixth international conference on communication systems and networks (COMSNETS) 2014 Jan 6 (pp. 1-4). IEEE.

[27] Morzelona, R. (2021). Human Visual System Quality Assessment in The Images Using the IQA Model Integrated with Automated Machine Learning Model . Machine Learning Applications in Engineering Education and Management, 1(1), 13–18. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/5

[28] Tilak S, Patil D. A survey of various scheduling algorithms in cloud environment. International Journal of Engineering Inventions. 2012 Sep-1(2):36-9.

[29] Kumar M, Dubey K, Sharma SC. Job scheduling algorithm in cloud environment considering the priority and cost of job. InProceedings of Sixth International Conference on Soft Computing for Problem Solving: SocProS 2016, Volume 2 2017 (pp. 313-320). Springer Singapore.

[30] https://aws.amazon.com/

[31] Ana Silva, Deep Learning Approaches for Computer Vision in Autonomous Vehicles , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[32] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.

[33] Vyas, A. ., & Sharma, D. A. . (2020). Deep Learning-Based Mango Leaf Detection by Pre-Processing and Segmentation Techniques. Research Journal of Computer Systems and Engineering, 1(1), 11–16. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/18

[34] R. Patil Rashmi, Y. Gandhi, V. Sarmalkar, P. Pund and V. Khetani, "RDPC: Secure Cloud Storage with Deduplication Technique," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 1280-1283, doi: 10.1109/I-SMAC49090.2020.9243442.