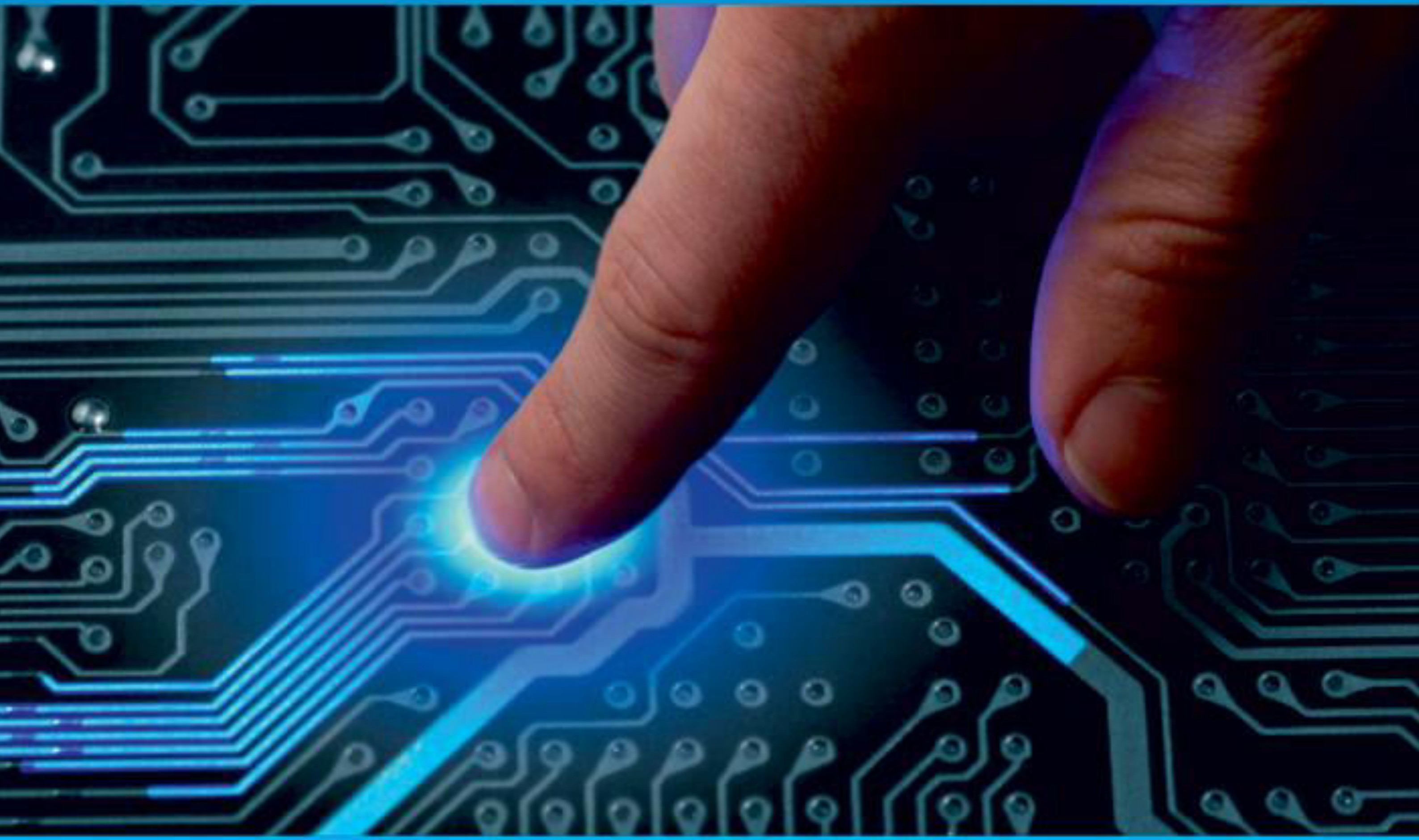
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

☎ 9940 572 462 6381 907 438 ijircce@gmail.com www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2023.1205057 |

A Cloud Secure Storage Mechanism Based On Fault And Load Balancing

Kasturi Harbade[1], Prof. Pushpa Tandekar[2], Prof. Vijay Rakhade[3]

Student, Department of Computer Science and Engineering, Shri Sai College of Engineering and Technology, Chandrapur, India[1]

Assistant Professor, Department of Computer Science and Engineering, Shri Sai College of Engineering and Technology, Chandrapur, India[2,3]

ABSTRACT: In this project we propose a Elastic load balancer. ELB (Elastic Load Balancer) is a service provided by AWS (Amazon Web Services) which is used to balance the incoming load (traffic) on multiple EC2 instances which can be in multiple availability zones. Load balancing distributes the network traffic in such a manner that increases speed and the performance of web applications. Modern ELB uses “Round robin algorithm”. The real time applications of ELB are high traffic websites, E- commerce applications, Media streaming, Gaming applications, Mobile applications. ELB improves scalability, availability and fault tolerance of web applications and service.

KEYWORDS: Load Balancer, Amazon Web Services, Round Robin.

I. INTRODUCTION

Load balancing is an essential technique used in cloud computing to optimize resource utilization and ensure that no single resource is overburdened with traffic. It is a process of distributing workloads across multiple computing resources, such as servers, virtual machines, or containers, to achieve better performance, availability, and scalability.

1. In cloud computing, load balancing can be implemented at various levels, including the network layer, application layer, and database layer. The most common load balancing techniques used in cloud computing are:
2. Network Load Balancing: This technique is used to balance the network traffic across multiple servers or instances. It is implemented at the network layer and ensures that the incoming traffic is distributed evenly across the available servers.
3. Application Load Balancing: This technique is used to balance the workload across multiple instances of an application. It is implemented at the application layer and ensures that each instance receives an equal share of the incoming requests.
4. Database Load Balancing: This technique is used to balance the workload across multiple database servers. It is implemented at the database layer and ensures that the incoming queries are distributed evenly across the available database servers.

Load balancing helps to improve the overall performance and reliability of cloud-based applications by ensuring that resources are used efficiently and that there is no single point of failure. It also helps to scale applications on demand and provides high availability and fault tolerance to handle spikes in traffic or server failures.

II. MAJOR EXAMPLES

1. **Direct Routing Requesting Dispatching Technique:** This approach of request dispatching is like to the one implemented in IBM’s Net Dispatcher. A real server and load balancer share the virtual IP address. In this, load balancer takes an interface constructed with the virtual IP address that accepts request packets and it directly routes the packet to the selected servers.
2. **Dispatcher-Based Load Balancing Cluster:** A dispatcher does smart load balancing by utilizing server availability, workload, capability and other user-defined criteria to regulate where to send a TCP/IP request. The dispatcher module of a load balancer can split HTTP requests among various nodes in a cluster. The dispatcher splits the load among many servers in a cluster so the services of various nodes seem like a virtual service on an only IP address; consumers interrelate as if it were a solo server, without having an information about the back-end infrastructure

.

3. **Linux Virtual Load Balancer:** It is an opensource enhanced load balancing solution used to build extremely scalable and extremely available network services such as HTTP, POP3, FTP, SMTP, media and caching and Voice Over Internet Protocol (VoIP). It is simple and powerful product made for load balancing and fail-over. The load balancer itself is the primary entry point of server cluster systems and can execute Internet Protocol Virtual Server (IPVS), which implements transport-layer load balancing in the Linux kernel also known as Layer-4 switching.

III. LOAD BALANCING ALGORITHMS

A load balancing algorithm is the set of rules that a load balancer follows to determine the best server for each of the different client requests. Load balancing algorithms fall into two main categories.

1) Static load balancing

Static load balancing algorithms follow fixed rules and are independent of the current server state. The following are examples of static load balancing.

Round-robin method

Servers have IP addresses that tell the client where to send requests. The IP address is a long number that is difficult to remember. To make it easy, a Domain Name System maps website names to servers. When you enter aws.amazon.com into your browser, the request first goes to our name server, which returns our IP address to your browser.

In the round-robin method, an authoritative name server does the load balancing instead of specialized hardware or software. The name server returns the IP addresses of different servers in the server farm turn by turn or in a round-robin fashion.

Weighted round-robin method

In weighted round-robin load balancing, you can assign different weights to each server based on their priority or capacity. Servers with higher weights will receive more incoming application traffic from the name server.

IP hash method

In the IP hash method, the load balancer performs a mathematical computation, called hashing, on the client IP address. It converts the client IP address to a number, which is then mapped to individual servers.

2) Dynamic load balancing

Dynamic load balancing algorithms examine the current state of the servers before distributing traffic. The following are some examples of dynamic load balancing algorithms.

Least connection method

A connection is an open communication channel between a client and a server. When the client sends the first request to the server, they authenticate and establish an active connection between each other. In the least connection method, the load balancer checks which servers have the fewest active connections and sends traffic to those servers. This method assumes that all connections require equal processing power for all servers.

Weighted least connection method

Weighted least connection algorithms assume that some servers can handle more active connections than others. Therefore, you can assign different weights or capacities to each server, and the load balancer sends the new client requests to the server with the least connections by capacity.

Least response time method

The response time is the total time that the server takes to process the incoming requests and send a response. The least response time method combines the server response time and the active connections to determine the best server. Load balancers use this algorithm to ensure faster service for all users.

Resource-based method

In the resource-based method, load balancers distribute traffic by analyzing the current server load. Specialized software called an agent runs on each server and calculates usage of server resources, such as its computing capacity and memory. Then, the load balancer checks the agent for sufficient free resources before distributing traffic to that server.

Companies usually have their application running on multiple servers. Such a server arrangement is called a server farm. User requests to the application first go to the load balancer. The load balancer then routes each request to a single server in the server farm best suited to handle the request.

Load balancing is like the work done by a manager in a restaurant. Consider a restaurant with five waiters. If customers were allowed to choose their waiters, one or two waiters could be overloaded with work while the others are idle. To avoid this scenario, the restaurant manager assigns customers to the specific waiters who are best suited to serve them.

IV. Types Of Load Balancing

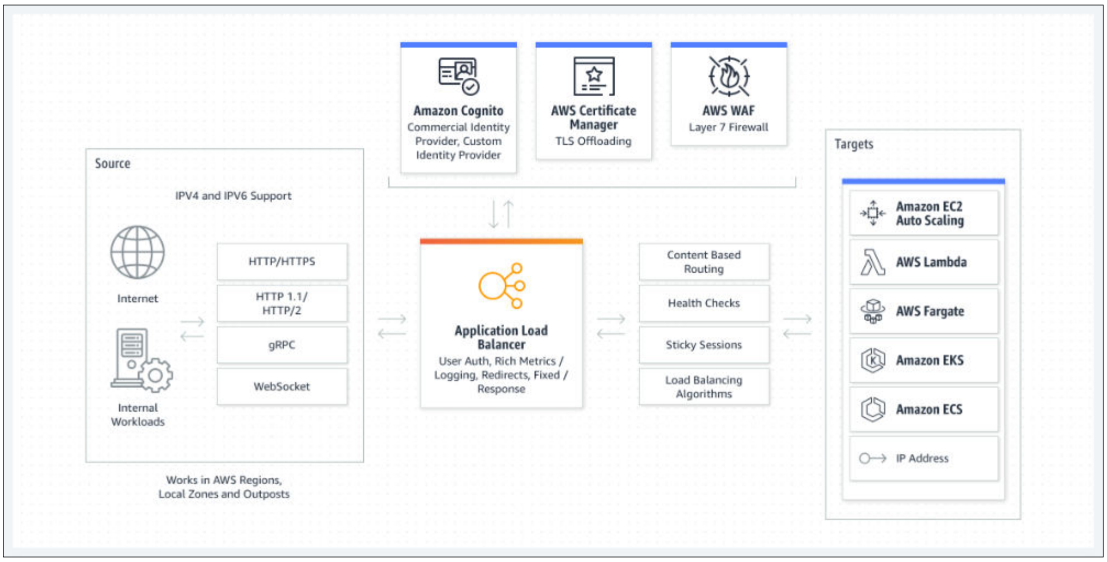


Fig1: Types of Load Balancing

We can classify load balancing into three main categories depending on what the load balancer checks in the client request to redirect the traffic.

1. Application load balancing

Complex modern applications have several server farms with multiple servers dedicated to a single application function. Application load balancers look at the request content, such as HTTP headers or SSL session IDs, to redirect traffic. For example, an ecommerce application has a product directory, shopping cart, and checkout functions. The application load balancer sends requests for browsing products to servers that contain images and videos but do not need to maintain open connections. By comparison, it sends shopping cart requests to servers that can maintain many client connections and save cart data for a long time.

2. Network load balancing

Network load balancers examine IP addresses and other network information to redirect traffic optimally. They track the source of the application traffic and can assign a static IP address to several servers. Network load balancers use the static and dynamic load balancing algorithms described earlier to balance server load.

3. Global server load balancing

Global server load balancing occurs across several geographically distributed servers. For example, companies can have servers in multiple data centers, in different countries, and in third-party cloud providers around the globe. In this case, local load balancers manage the application load within a region or zone. They attempt to redirect traffic to a server destination that is geographically closer to the client. They might redirect traffic to servers outside the client's geographic zone only in case of server failure.

4. DNS load balancing

In DNS load balancing, you configure your domain to route network requests across a pool of resources on your domain. A domain can correspond to a website, a mail system, a print server, or another service that is made accessible through the internet. DNS load balancing is helpful for maintaining application availability and balancing network traffic across a globally distributed pool of resources.

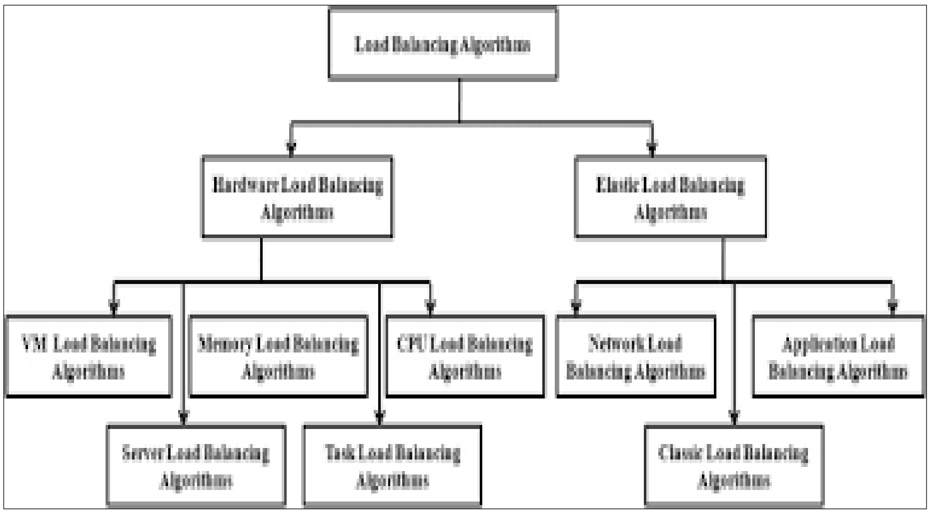

Load Balancing Algorithms
Hardware Load Balancing Algorithms
Elastic Load Balancing Algorithms
VM Load Balancing Algorithms
Memory Load Balancing Algorithms
CPU Load Balancing Algorithms
Network Load Balancing Algorithms
Application Load Balancing Algorithms
Server Load Balancing Algorithms
Task Load Balancing Algorithms
Classic Load Balancing Algorithms

Fig2: Load Balancing

Load balancers are one of two types: hardware load balancer and software load balancer.

1. Hardware load balancers

A hardware-based load balancer is a hardware appliance that can securely process and redirect gigabytes of traffic to hundreds of different servers. You can store it in your data centers and use virtualization to create multiple digital or virtual load balancers that you can centrally manage.

2. Software load balancers

Software-based load balancers are applications that perform all load balancing functions. You can install them on any server or access them as a fully managed third-party service.

Comparison of hardware balancers to software load balancers

- Hardware load balancers require an initial investment, configuration, and ongoing maintenance. You might also not use them to full capacity, especially if you purchase one only to handle peak-time traffic spikes.
- If traffic volume increases suddenly beyond its current capacity, this will affect users until you can purchase and set up another load balancer.
- In contrast, software-based load balancers are much more flexible. They can scale up or down easily and are more compatible with modern cloud computing environments. They also cost less to set up, manage, and use over time.

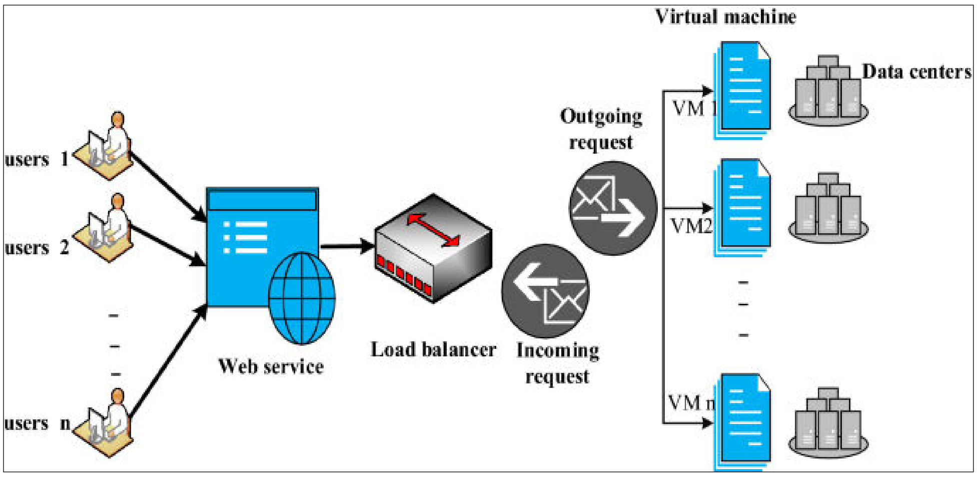


Fig3: Load Balancing

V. METHODOLOGY

A. *Amazon Web Services*

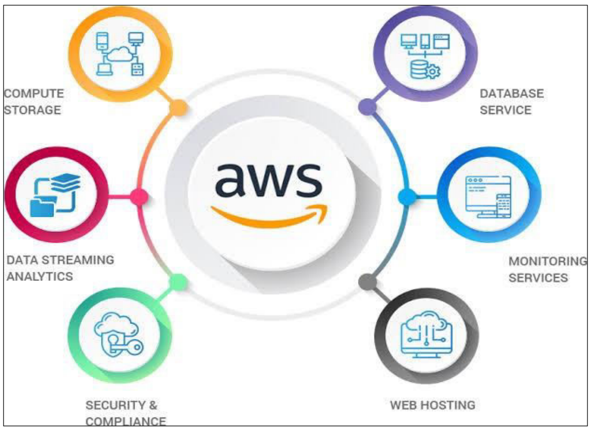


Fig4: AWS

- The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand. It provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).
- Amazon launched AWS, a cloud computing platform to allow the different organizations to take advantage of reliable IT infrastructure.

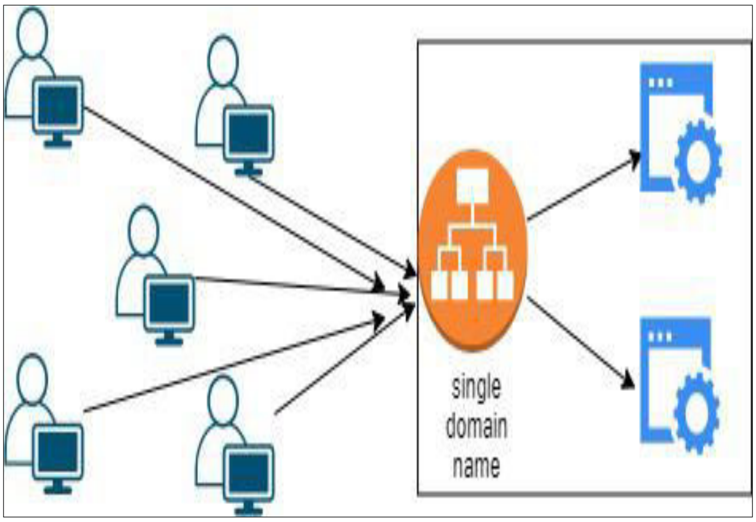


Fig5: AWS

Elastic Load Balancing (ELB) is a fully managed load balancing service that automatically distributes incoming application traffic to multiple targets and virtual appliances across AWS and on-premises resources. You can use it to scale modern applications without complex configurations or API gateways. You can use ELB to set up four different types of software load balancers.

- An Application Load Balancer routes traffic for HTTP-based requests.
- A Network Load Balancer routes traffic based on IP addresses. It is ideal for balancing TCP and User Datagram Protocol (UDP)-based requests.
- A Gateway Load Balancer routes traffic to third-party virtual appliances. It is ideal for incorporating a third-party appliance, such as a network firewall, into your network traffic in a scalable and easy-to-manage way.
- A Classic Load Balancer routes traffic to applications in the Amazon EC2-Classic network—a single, flat network that you share with other customers.

Uses of AWS

- A small manufacturing organization uses their expertise to expand their business by leaving their IT management to the AWS.
- A large enterprise spread across the globe can utilize the AWS to deliver the training to the distributed workforce.
- An architecture consulting company can use AWS to get the high-compute rendering of construction prototype.
- A media company can use the AWS to provide different types of content such as ebox or audio files to the worldwide files.

B. Round Robin Algorithm

- Round Robin CPU Scheduling is the most important CPU Scheduling Algorithm which is ever used in the history of CPU Scheduling Algorithms. Round Robin CPU Scheduling uses Time Quantum (TQ). The Time Quantum is something which is removed from the Burst Time and lets the chunk of process to be completed.
- Time Sharing is the main emphasis of the algorithm. Each step of this algorithm is carried out cyclically. The system defines a specific time slice, known as a time quantum.

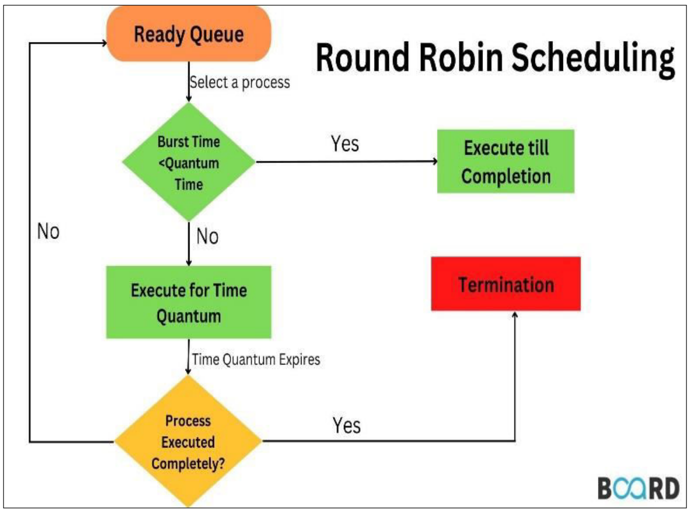


Fig6: Round Robin Scheduling

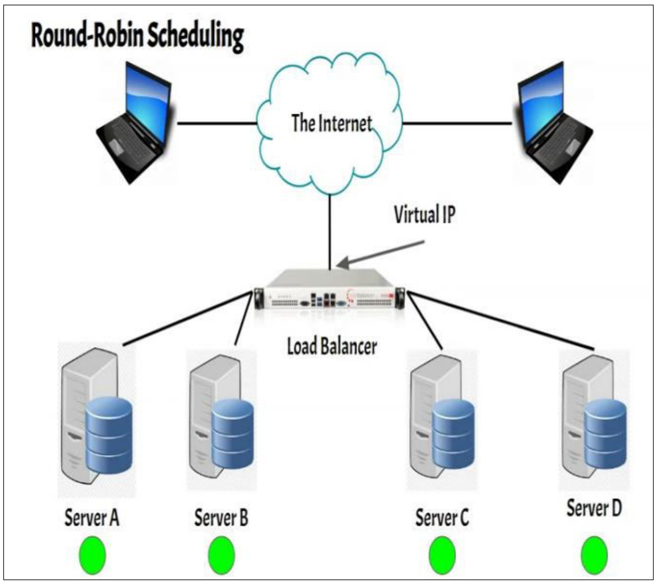


Fig7: Round Robin Scheduling

Advantages

The Advantages of Round Robin CPU Scheduling are:

1. A fair amount of CPU is allocated to each job.
2. Because it doesn't depend on the burst time, it can truly be implemented in the system.
3. It is not affected by the convoy effect or the starvation problem as occurred in First Come First Serve CPU Scheduling Algorithm.

VI. Existing Problem

- Elastic Load Balancers (ELBs) are an integral part of distributing incoming network traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.
- One exciting problem in this context could involve optimizing the ELB configuration to achieve the best performance, high availability, and cost effectiveness for your application. This might include:
- Auto Scaling: Designing the ELB setup to dynamically adjust its capacity based on traffic spikes or drops, ensuring your application can handle varying loads efficiently.effectiveness for your application. This might include:
- Health Checks: Ensuring that the ELB accurately monitors the health of its registered instances and routes traffic only to healthy instances.
- Cross-Zone Load Balancing: Configuring the ELB to distribute traffic evenly across multiple Availability Zones, providing better fault tolerance and reducing latency.
- SSL/TLS Termination: Implementing SSL/TLS termination at the ELB to offload the decryption process from backend instances, improving performance and security.
- Path-Based Routing: Utilizing path-basedrouting to direct different types of requests to specific backend instances or groups, enabling more efficient resource utilization.
- Session Persistence: Handling session persistence effectively, either through sticky sessions or other mechanisms, to maintain user sessions across requests

VII. Advantages

1. Improved Performance: Load balancing helps to distribute the workload across multiple resources, which reduces the load on each resource and improves the overall performance of the system.
2. High Availability: Load balancing ensures that there is no single point of failure in the system, which provides high availability and fault tolerance to handle server failures.
3. Scalability: Load balancing makes it easier to scale resources up or down as needed, which helps to handle spikes in traffic or changes in demand.
4. Efficient Resource Utilization: Load balancing ensures that resources are used efficiently, which reduces wastage and helps to optimize costs.

VIII. Conclusion and Future Work

In conclusion, our paper successfully demonstrated the implementation of an Elastic Load Balancer (ELB) in Amazon Web Services (AWS) using the round- robin algorithm, all achieved through the power of the C programming language. By leveraging AWS services and the round-robin algorithm, we've created an efficient and scalable load balancing system. Throughout this paper, we've gained valuable insights into cloud computing, networking, and load balancing principles. We've learned how ELB can evenly distribute incoming requests to a group of backend servers, ensuring high availability and optimizing resource utilization. The round-robin algorithm, implemented in C, proved to be a simple yet effective means of achieving this load balancing. Implementing Elastic Load Balancing (ELB) in Amazon Web Services (AWS) with the round-robin algorithm offers a straightforward and effective approach to distributing incoming network traffic across multiple EC2 instances. This load balancing methodology ensures even traffic distribution, promoting high availability and scalability for applications. It is particularly suitable when the backend servers are homogeneous and capable of handling requests independently. By systematically routing requests to each instance in a circular fashion, round- robin load balancing optimizes resource utilization and enhances fault tolerance, contributing to a robust and efficient workload management solution within the AWS ecosystem.

Future iterations could involve integrating more advanced load balancing algorithms, adding monitoring and logging features, or extending the project to include other AWS services for a comprehensive cloud-based solution.

REFERENCES

1. Lua et al (2011) "Load Balancing Algorithms in Cloud Computing Environment: A Review",International Journal on Recent and Innovation Trends in Computing and Communication, Vol 2,Aug 2011.
2. Chen (2014)"An Overview of Cloud Computing Adoption Challenges in the Norwegian Context." Utility and Cloud Computing (UCC), 2014.
3. Liu (2014) "A comparative study into distributed load balancing algorithms for cloud computing."Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. IEEE, 2014.
4. Nitish (2015) "Comparative study on load balancing techniques in cloud computing." Open Journal of Mobile Computing and Cloud Computing 1.1 (2014).
5. M. Randle's (2010) "An in-depth analysis and study of Load balancing techniques in the cloud computing environment." Procedia Computer Science 50 (2015).
6. Chunling (2018)" Load balancing and resource monitoring in cloud", Proceedings of the CUBE International Information Technology Conference. ACM, 2018.
7. Brian Chang (2023) "Leading Load Balancing" ICDCN '23: Proceedings of the 24th International Conference on Distributed Computing and Networking January 2023.
8. Steven Hofmeyr (2010) "Load Balancing on Speed" PPoPP '10: Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming January 2010.
9. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, "What is an object?",. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.
10. Lowlesh Yadav and Asha Ambhaikar, "IOHT based Tele-Healthcare Support System for Feasibility and performance analysis," Journal of Electrical Systems, vol. 20, no. 3s, pp. 844–850, Apr. 2024, doi: 10.52783/jes.1382.
11. L. Yadav and A. Ambhaikar, "Feasibility and Deployment Challenges of Data Analysis in Tele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-5, doi: 10.1109/ICAIIHI57871.2023.10489389.
12. L. Yadav and A. Ambhaikar, "Approach Towards Development of Portable Multi-Model Tele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6, doi: 10.1109/ICAIIHI57871.2023.10489468.
13. Lowlesh Yadav and Asha Ambhaikar, Exploring Portable Multi-Modal Telehealth Solutions: A Development Approach. International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol. 11, no. 10, pp. 873–879, Mar. 2024.11(10), 873–879, DOI: 10.13140/RG.2.2.15400.99846.
14. Lowlesh Yadav, Predictive Acknowledgement using TRE System to reduce cost and Bandwidth, March 2019. International Journal of Research in Electronics and Computer Engineering (IJRECE), VOL. 7 ISSUE 1 (JANUARY- MARCH 2019) ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE).