International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Review on Designing and Implementation of Application Load Balancing using AWS

Prathamesh Sanjay Zingade [1], Akshata Shet [2], Dr. Prakash Biswagar [3], Dr. Sharvani G[4]

[1,2]Department of Electronics and Communication Engineering
RV College of Engineering Banglore, Karnataka, India
[3]Professor, Department of Electronics and Communication Engineering
RV College of Engineering Banglore, Karnataka, India
[4]Associate Professor, Department of Electronics and Communication Engineering
RV College of Engineering Banglore, Karnataka, India

Abstract - *Cloud computing is one of the most growing technologies. The fundamental idea behind cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on the users and helping them focus on their core businesses. These computing resources are hosted on virtual hosts and distributed on-demand to the users by cloud service providers. For efficient resource utilization, systematic load balancing of incoming user traffic across virtual hosts is imperative. In the present work a Distributed Dynamic and Customized Load Balancing (DDCLB) algorithm is proposed to dynamically handle the incoming user requests for the Amazon EC2 instances. Elastic Load Balancing automatically divides incoming application traffic over different targets, like Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can manage the differing load of your application traffic in one Availability Zone or over different Availability Zones. Elastic Load Balancing offers three types of load balancers which all presents the high availability, automatic scaling, and robust security required to make your applications defect liberal.*

Key Words: **AWS, Application Load Balancing, Classic vs Application load balancing, EC2 instance, ELB.**

1. INTRODUCTION TO LOAD BALANCING

Cloud Computing has emerged as a combination of distributed computing technology, server virtualization technology and network storage. The fundamental idea of cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on users and helping them focus on their core businesses [1] [2]. Cloud computing exhibits various characteristics such as scalability, location independence, flexibility, device independence, elasticity, reliability, resource sharing, cost effectiveness and on-demand computing [3]. All these factors have led to the accelerating use of cloud computing. A load balancing operation consists of three rules. These are location rule, distribution rule and selection rule [2, 5] The selection rule works either in pre-emptive or in non- pre-emptive fashion. The newly generated process is always picked up by the non-pre-emptive rule while the running process may be picked up by the pre-emptive rule. Pre-emptive transfer is costly than non-pre-emptive transfer which is more preferable. However pre-emptive transfer is more excellent than non-pre-emptive transfer in some instances. Practically load balancing decisions are taken jointly by location and distribution rules. The balancing domains are of two types: local and global. In local domain, the balancing decision is taken from a group of nearest neighbours by exchanging the local workload information while in global domain the balancing decision is taken by triggering.

Load balancing plays a very important role in the networking technology, Load balancer comes into play when the user tries to connect to the server. This sample example is shown in the Fig 1 Any requests made to the internet will reach the server in various paths. There may be N number of servers which are serving the purpose of the request but the request will go to only one depending upon the traffic. The traffic here refers to how busy server is, in responding the requests made by servers. Requests will be directed towards servers by one of the networks called Load balancer which balances the load of server.

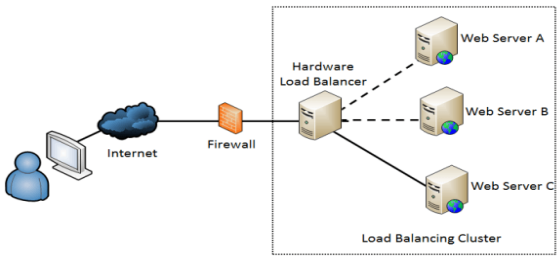

Fig-1: Load balancer

There are different types of load balancer which handles the traffic in different ways. Major types of Load balancer are Static and Dynamic Load balancer. Sub types are shown in Fig.2

In static algorithm the processes are assigned to the processors at the compile time according to the performance of the nodes. Once the processes are assigned, no change or reassignment is possible at the run time.

Number of jobs in each node is fixed in static load balancing algorithm. Static algorithms do not collect any information about the nodes The assignment of jobs is done to the processing nodes on the basis of the following factors: incoming time, extent of resource needed, mean execution time and inter-process communications. Since these factors should be measured before the assignment, this is why static load balance is also called probabilistic algorithm. As there is no migration of job at the runtime no overhead occurs or a little over head may occur. In static load balancing it is observed that as the number of tasks is more than the processors, better will be the load balancing.

During the static load balancing too much information about the system and jobs must be known before the execution. These information may not be available in advance. A thorough study on the system state and the jobs quite tedious approach in advance. So, dynamic load balancing algorithm came into existence. The assignment of jobs is done at the runtime. In DLB jobs are reassigned at the runtime depending upon the situation that is the load will be transferred from heavily loaded nodes to the lightly loaded nodes. In this case communication over heads occur and becomes more when number of processors increase. dynamic load balancing no decision is taken until the process gets execution. This strategy collects the information about the system state and about the job information. As more information is collected by an algorithm in a short time, potentially the algorithm can make better decision. Dynamic load balancing is mostly considered in heterogeneous system because it consists of nodes with different speeds, different communication link speeds, different memory sizes, and variable external loads due to the multiple. The numbers of load balancing strategies have been developed and classified so far for getting the high performance of a system.

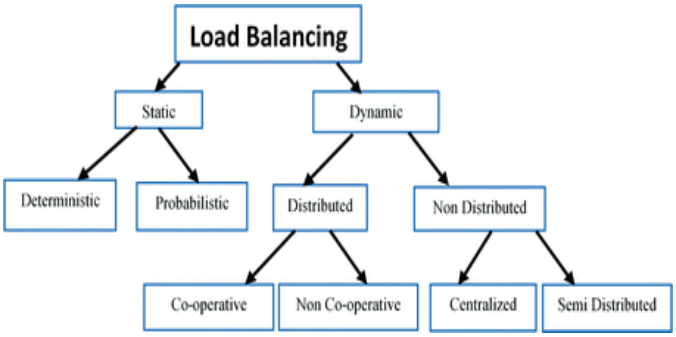

Fig-2: Load balancer types

2. LOAD BALANCING DEMO TOOLS

To Simulate and demonstrate the working of Load balancing and to provide the best solution for using Load balancer various tools can be utilised. Load balancing is a real time approach where the real world problems are resolved. Nowadays Internet has become so common that everyone started using it for various reasons. Its necessary that servers should handle all the traffic using load balancing techniques.

To provide improvised solution for efficient load balancing Tools like following can be used:

- Cloud Analyst
- Cloud Project
- Amazon Web Service

Cloud analyst and Cloud Project seems to be easy as it has a graphical user interface with which it seems like easy to use tool and also seems to have a level of visualisation capability which is even better than just a tool-kit. It separates simulation set up environment exercise and supports the modeller to focus on the parameters used for simulation purposes rather than the programming technicalities only. the accuracy and performance of simulation. It provides user interface to create the server and simulate the algorithm. But to simulate with real time virtual instances AWS is preferred.

AWS provides Elastic Load balancing platform which involves various features. Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

3. TYPES OF ELB

Elastic Load Balancing supports three types of load balancers:

- Application Load Balancers
- Network Load Balancers
- Classic Load Balancers

There is a key difference in how the load balancer types are configured. With Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

To provide simulated results it is essential to create the virtual instance. As Application Load balancing maintains the load right from the 7th layer that is application layer it provides more efficient way to handle the traffic. The data traffics are handled depending upon the application, hence the paper provides detailed view of simulating the application load balancing. The various character comparisons of types of ELB are shown in Fig.3

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Health Checks	✓	✓	✓
CloudWatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal fail-over	✓	✓	✓
Connection draining	✓	✓	✓
Load balancing to multiple ports on an instance	✓	✓	
WebSockets	✓	✓	
IP addresses as targets	✓	✓	
Lambda functions as targets	✓		
Load balancer deletion protection	✓	✓	
Path-based routing	✓		
Host-based routing	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Source IP address CIDR-based routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓

Fig -3: Comparison figure

Application Load balancing provides all the features and services compared to other types; hence virtual instances are created to demonstrate which provides load balancing depending on the application.

4. APPLICATION LOAD BALANCING

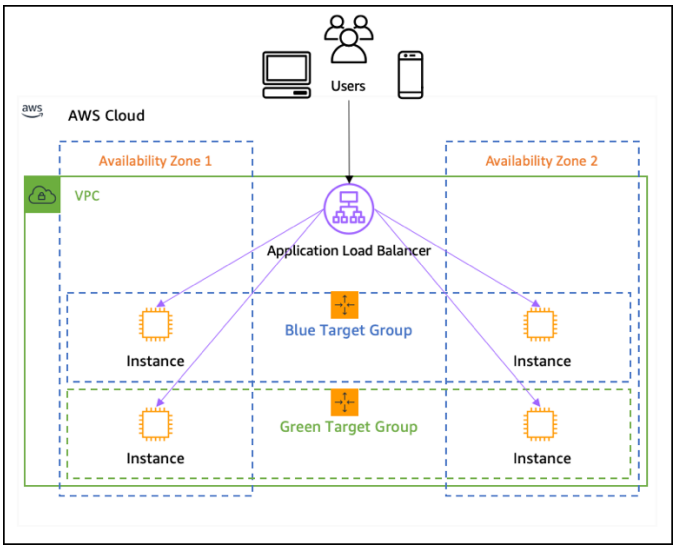

Fig -4: Application load balancing

An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups. You can configure the routing algorithm used at the target group level. The default routing algorithm is round robin; alternatively, you can specify the least outstanding requests routing algorithm.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing can scale to the vast majority of workloads automatically.

You can configure health checks, which are used to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- Support for path-based routing. You can configure rules for your listener that forward requests based on the URL in the request. This enables you to structure your application as smaller services, and route requests to the correct service based on the content of the URL.
- Support for host-based routing. You can configure rules for your listener that forward requests based on the host field in the HTTP header. This enables you to route requests to multiple domains using a single load balancer.

- Support for routing based on fields in the request, such as standard and custom HTTP headers and methods, query parameters, and source IP addresses.
- Support for routing requests to multiple applications on a single EC2 instance. You can register each instance or IP address with the same target group using multiple ports.
- Support for redirecting requests from one URL to another.
- Support for returning a custom HTTP response.
- Support for registering targets by IP address, including targets outside the VPC for the load balancer.
- Support for registering Lambda functions as targets.
- Support for the load balancer to authenticate users of your applications through their corporate or social identities before routing requests.

As shown in Fig. 4 Application load balancing is done by creating EC2 instances. AWS cloud provides the service to create the instances and run the simulation to check for balancing the requests of user depending upon the application.

5. IMPLEMNTATION

Following are the steps followed to implement load balancing:

1. Configure a Load Balancer and a Listener
2. Configure Security Settings for an HTTPS Listener
3. Configure a Security Group
4. Configure a Target Group
5. Configure Targets for the Target Group
6. Create the Load Balancer

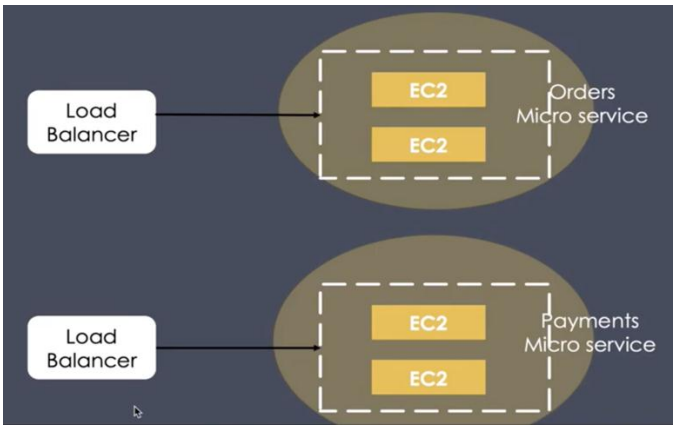

Fig -5: Creating services

Considering a small example of 2 application the implementation is done such that depending upon the load / application the requests are balanced, and path changed to respective servers. Two services as Orders and Paraments are created. As shown in Fig.5. Two servers are created with EC2 instances, each server has implemented with load balancer.

Here order and pavements are considered as two different micro services. These acts as application hence the load balancing has to be done depending upon the application. The requests by users are done through HTTP, the listener and balancer are created such that depending upon the http request the requests are guided to the respective micro services.

As shown in Fig.6 EC2 instances are created with the names order and payment.

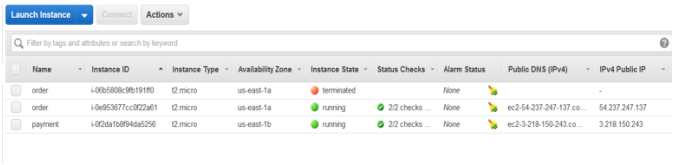

Fig -6: EC2 instance

These two instances are virtual servers used as micro services. These statuses are maintained to be 'running'. Once the instances created there has to be some respective targets.
As shown in Fig.7 Targets are created with respective names as Orders and Payments.

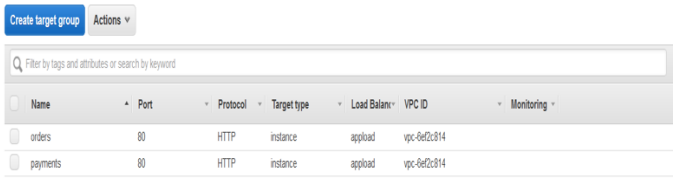

Fig -7 Targets

Once the target is ready, the application load balancer is added to the targets. Type is maintained to be Application and state should be active. This is shown in Fig.8

Fig -8 Load balancing

As load balance is ready to balance the traffic but listeners have to be added to the respective targets hence 1st script is written which performs the main objective of balancing depending upon the request made by user. The sample scripts are written as shown in fig 9 and fig.10 to direct the balancer to HTTP page with headers as written in the script. Depending upon the requests made the server is changed to the respective instances.

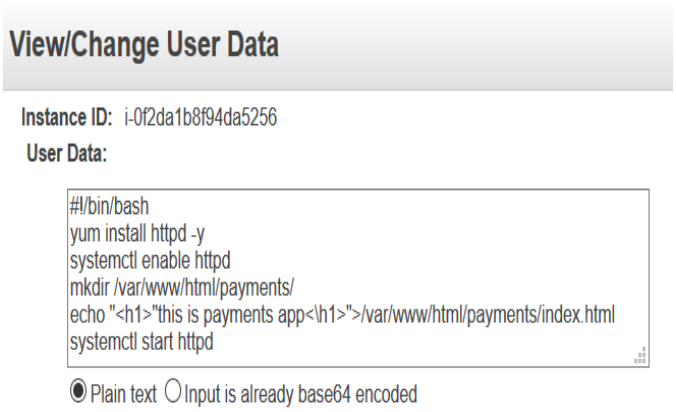

Fig -9: Payment script

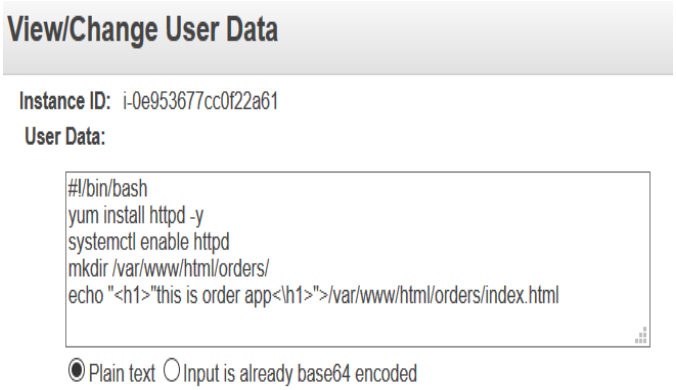

Fig -10: Order script

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 11 The rules are created to direct the paths, If the http contains the payment data then the request is directed to payment service, similarly if request contains orders information then request is directed to Orders micro service.

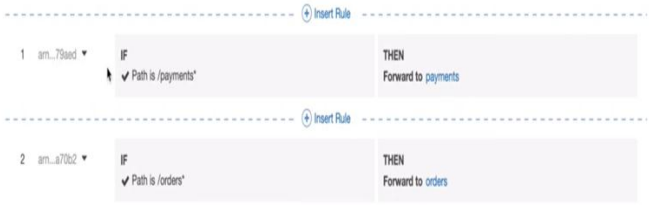

Fig -11: Creating rules

Final model of load balancer looks as shown in Fig 12. This is the final setup for simulation which will demonstrate the load balancing based on the application/ HTTP request.

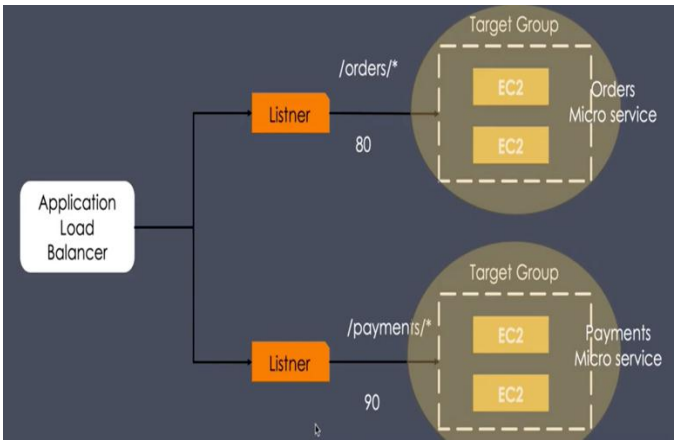

Fig -12: Final implementation

6. 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 Order and Payment is created as shown in fig 13 and 14 if the link or http requests contains the Order or Payment 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 re directed then scripts are read from respective services then details are sent back to the user.

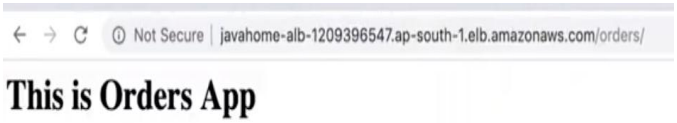

Fig -13: Reading order micro service

Fig -14: Reading payment micro service

6. CONCLUSION

Load balancing plays a very important role as usage of internet is growing day by day. As usage increased the traffic in the network increases hence Application load balancer provides the features which can help to resolve the problems and provide easy solution to redirect the paths of request to the servers based on the request made. Instead of dividing the path when based on freeness of servers it is made based on the application. Consider If College needs a network load balancer for their websites then depending on the request

made (like- Fees, Admissions, Facilities etc.) The requests are redirected to the respective servers hence there will ne no conflicts. Here a sample example of Orders and Payments is explained which will demonstrate the efficient way to direct the path depending on the application that is all the order related request are directed to Order micro service similarly all Payment request are directed to Payment micro service. This is efficient way of solving the traffic problem hence design and simulation of application load balancing is done using EC2 instances available in AWS services.

REFERENCES

[1] "A Comparative Study of Static and Dynamic Load Balancing Algorithms in Cloud Computing", International Conference on Energy, Communication, Data Analytics and Soft Computing, 2017

[2] Mari Marios D. Dikaiakos, George Pallis, Dimitrios Katsaros, Pankaj Mehra, Athena Vakali (2009, Oct.). Cloud Computing: Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing, vol. 13(issue 5), pp. 10-13.

[3] Panagiotis Kalagiakos, Panagiotis Karampelas, "Cloud Computing Learning" in the Proceeding of IEEE International Conference on Application of Informat

[4] A Distributed Dynamic and Customized Load Balancing Algorithm for Virtual Instances International Conference on Engineering (NUiCONE), 2016

[5] Shridhar G.Domanal and G. Ram Mohana Reddy, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines" in the Proceeding of the IEEE International Conference on Communication Systems and Networks, Bangalore, Jan. 2014, pp. 1-4.

[6] "Static load balancing technique for geographically partitioned public cloud", Scalable Computing: Practice and Experience , 2019

[7] "Load balancing algorithm in cloud computing", 5th IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC-2018), 20-22 Dec, 2018

[8] Jitendra Bhatia, Malaram Kumhar, "Perspective Study on Load Balancing Paradigms in Cloud Computing," IJCSC Vol- 6 • Issue-1 Sep - Mar 2015 pp.112-120

[9] Jitendra Bhatia, "A Dynamic Model for Load Balancing in Cloud Infrastructure," NIRMA UNIVERISTY JOURNAL OF ENGINEERING AND TECHNOLOGY VOL. 4, N0. 1, JAN-JUN 2015, pp. 15-19.

[10] Hamid Shoja and Hosseni Nahid, "A comparative survey on load balancing algorithms in CC", IEEE 33044, 5th ICCCNT 2014 July 11-13, Hefei, China