LoaderBalancer, Autocaling & Serverless Computing

COMPARISION

Application 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.

Notwork Load Balancer

Network Load Balancer is best suited for load balancing of TCP 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

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.

Benefits of Load Balancer

- Highly Available
- Secure
- Elastic
- Flexible
- Robust Monitoring and Auditing
- Hybrid Load Balancing

Auto Scaling

Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it’s easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECStasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you’re already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

SETUP SCALING QUICKLY

AWS Auto Scaling lets you set target utilization levels for multiple resources in a single, intuitive interface. You can quickly see the average utilization of all of your scalable resources without having to navigate to other consoles. For example, if your application uses Amazon EC2 and Amazon DynamoDB, you can use AWS Auto Scaling to manage resource provisioning for all of the EC2 Auto Scaling groups and database tables in your application.

MAKE SMART SCALING DECISIONS

AWS Auto Scaling lets you build scaling plans that automate how groups of different resources respond to changes in demand. You can optimize availability, costs, or a balance of both. AWS Auto Scaling automatically creates all of the scaling policies and sets targets for you based on your preference. AWS Auto Scaling monitors your application and automatically adds or removes capacity from your resource groups in real-time as demands change.

AUTOMATICALLY MAINTAIN PERFORMANCE

Using AWS Auto Scaling, you maintain optimal application performance and availability, even when workloads are periodic, unpredictable, or continuously changing. AWS Auto Scaling continually monitors your applications to make sure that they are operating at your desired performance levels. When demand spikes, AWS Auto Scaling automatically increases the capacity of constrained resources so you maintain a high quality of service.

PAY ONLY FOR WHAT YOU NEED

AWS Auto Scaling can help you optimize your utilization and cost efficiencies when consuming AWS services so you only pay for the resources you actually need. When demand drops, AWS Auto Scaling will automatically remove any excess resource capacity so you avoid overspending. AWS Auto Scaling is free to use, and allows you to optimize the costs of your AWS environment.

Auto scaling in commercial cloud

.1 AMAZON Amazon Web Service (AWS) provides compute and storage servers with high speed networks for accessing any type of resources. Amazon provides auto scaling service as IaaS EC2 (Elastic compute cloud) public cloud. EC2 provides an elastic IP address with every user account to reduce the instance failures. Auto scaling in AWS allows increasing or decreasing the number of EC2 instances within the application‟s architecture. With Auto scaling, one can create collections of EC2 instances called as Auto scaling groups. We can also specify minimum and maximum number of instances in each Auto Scaling group. Each auto scaling group contains one or more scaling policies which define when auto scaling launches or terminates EC2 instances within the group. Auto scaling in AWS uses load balancers to distribute traffic across the instances within auto scaling technique along with the elastic load balancing technique

2 MICROSOFT AZURE Platform-as-a-Service (PaaS) clouds offer a runtime environment system where users‟ components can be deployed and executed in a straightforward manner which offers an additional abstraction level when compared to IaaS clouds [11]. The users need not have to handle virtual resources such as machines or networks to start running their systems. Microsoft Windows Azure does not implement any embedded auto scaling solution to its users rather it supports Paraleap software which automatically scales resources in Azure to respond to changes on demand [12]. Data storage for application scheduling and rules based on customer performance counters is an added advantage in Windows Azure platform which is not available in other cloud providers

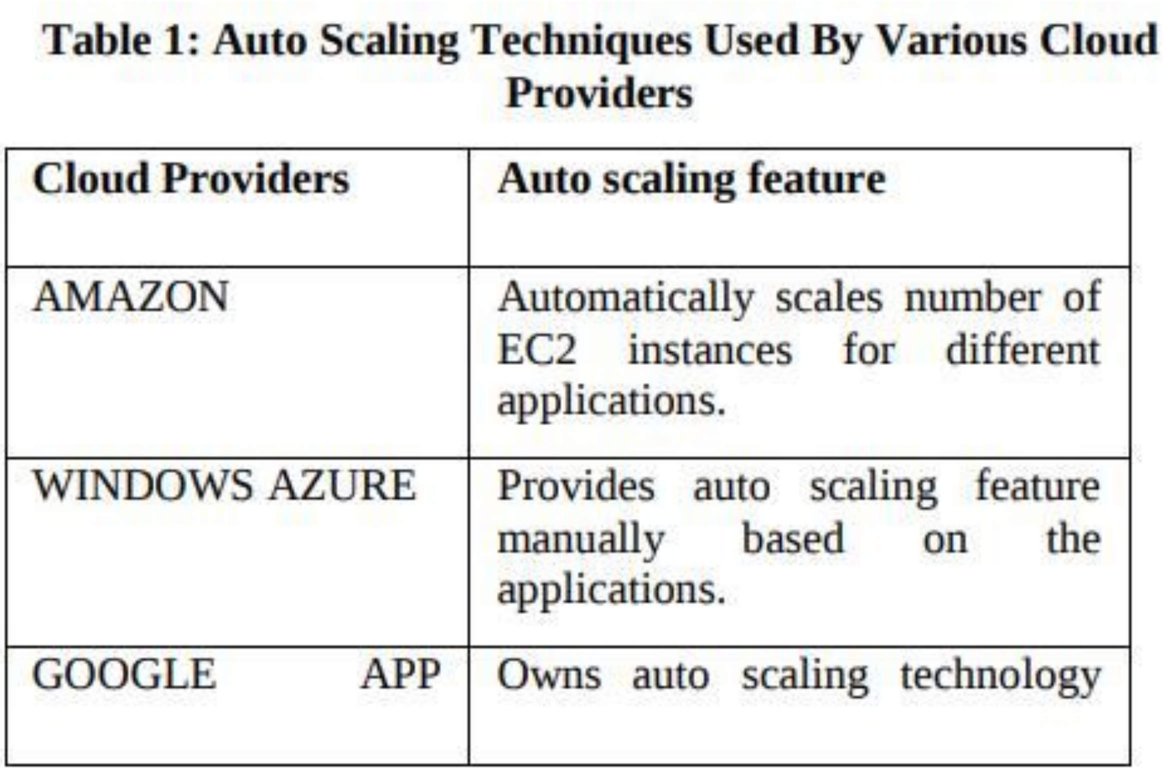

Table 1: Auto Scaling Techniques Used By Various Cloud Providers

Cloud Providers	**Auto scaling feature**
AMAZON	Automatically scales number of EC2 instances for different applications.
WINDOWS AZURE	Provides auto scaling feature manually based on the applications.
GOOGLE APP	Owns auto scaling technology

Load balancing in commercial cloud:

1. AMAZON: Amazon EC2 offers load balancing through Amazon Elastic Load Balancing service (ELB). ELB technique provides high availability of EC2 instances and enhances EC2 applications availability by distributing incoming application traffic across multiple instances [18]. EC2 includes OS such as Linux, Windows, Suse Linux, Fedora, Open Solaris, Red Hat, Open Suse, Ubuntu etc. Any user can interact with EC2 using set of SOAP messages. The elastic load balancer provides high availability of EC2 instances and also enhances EC2 application availability by distributing incoming application traffic across multiple instances. Elastic load balancing also detects unhealthy instances and automatically routes the traffic as necessary. Various metrics evaluation can be done through transactions/second, number of simultaneous users, request latency, performance evaluation, QoS evaluation,energy efficiency, power saving and cost estimation strategy. Amazon EC2 automatically distributes incoming application traffic among multiple instances using ELB feature and monitoring method using Cloud Watch techniques with high scaling policies.

2 MICROSOFT AZURE:

In Azure, the load is automatically distributed among available work resources by using a round robin algorithm transparent to the cloud users. Load balancing for applications running under the AppFabric service is achieved by using hardware load balancers [19]. The load balancers have redundant copies to reduce failure. Windows Azure gives PaaS cloud platforms to its users where SQL is a cloud based version of SQL servers and Azure AppFabric is a collection of services for cloud applications. Windows Azure has three components namely compute, storage and fabric controller. The Fabric Controller ensures scaling, load balancing and memory management and reliability features

CONCLUSION & FUTURE WORK

Auto scaling and load balancing features are the two methods which assure service level objectives in cloud computing era. Various factors affect the cloud services from different cloud providers‟ point of view. This paper has aimed the best to compare both the feature with respect to leading cloud platforms. The next work includes implementation of load balancing and auto scaling features in real time cloud environments.

Serverless Computing

Serverless computing also known as function as a service (FaaS) refers to a model where the existence of servers is simply hidden from developers. I.e. that even though servers still exist developers are relieved from the need to care about their operation. •Developers are relieved from the need to worry about low-level infrastructural and operational details such as scalability, high-availability, infrastructure-security, and so forth.
•Serverless computing is essentially about reducing maintenance efforts to allow developers to quickly focus on developing value-adding code.
•Serverless computing encourages and simplifies developing microservices oriented solutions in order to decompose complex applications into small and independent modules that can be easily exchanged.

Units of Scale

Virtual Machines

- Machine as the unit of scale
- Abstracts the hardware

- Containers
  - Application as the unit of scale
  - Abstracts the OS
- Serverless
  - Functions as the unit of scale
  - Abstracts the language runtime

How Lambda Works

AWS Lambda takes care of provisioning and managing resources needed to run your Lambda function.

- When a Lambda function is invoked, AWS Lambda launches a container (that is, an execution environment) based on the configuration information, such as the amount of memory and maximum execution time that you want to allow for your Lambda function.

- After a Lambda function is executed, AWS Lambda maintains the container for some time in anticipation of another Lambda function invocation.

- When you write your Lambda function code, do not assume that AWS Lambda always reuses the container because AWS Lambda may choose not to reuse the container. Depending on various other factors, AWS Lambda may simply create a new container instead of reusing an existing container.CSYE 6225, Spring 2018, Tejas Parikh, Northeastern University13