e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in

SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 12, December 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 6381 907 438 ijirset@gmail.com www.ijirset.com

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 12, December 2021 ||

| DOI:10.15680/IJIRSET.2021.1012061|

Analysis of AWS Auto Scaling Strategy in Cloud Computing

V Manikanta Sanjay[1], Ankith I[2], Sameer A Kyalkond[3]

U.G. Scholar, Department of Computer Science and Engineering, JSSATE, Bangalore, Karnataka, India[1]

U.G. Scholar, Department of Computer Science and Engineering, BNMIT, Bangalore, Karnataka, India[2]

U.G. Scholar, Department of Computer Science and Engineering, JSSATE, Bangalore, Karnataka, India[3]

ABSTRACT: In cloud computing settings, auto scaling methods have become a standard paradigm. Such techniques can grow or decrease the number of virtual computers based on user demand, hence accomplishing pay-per-use goals. Auto scaling methods supplied by infrastructure-as-a-service providers, on the other hand, must rigidly adhere to user-defined criteria; the disadvantage of such mechanisms is that they cannot adapt to real-time Internet traffic demands by adhering to user-defined thresholds. As a result, we suggest a dynamic threshold adjustment technique to speed up the generation of virtual machines based on workload needs. When the system is under tremendous stress, the proposed method can minimize web application response time and error rate. Furthermore, when the system is under a light demand, it can accelerate the release of virtual machines to decrease virtual machine operating time. According to our findings, CPU-intensive web applications need an excellent threshold control method. As a result, the suggested technique may meet this goal by effectively lowering application response time, virtual machine operating time, and error rate.

KEYWORDS: Auto scaling,Cloud Computing,EC2 instance.

I. INTRODUCTION

Nowadays, cloud computing is the most extensively utilized platform. It enables users to dynamically allocate and deallocate resources on a per-process basis. In cloud infrastructure, this capability of automatically scaling a resource is referred to as auto-scaling. The majority of cloud service providers charge their customers on an hourly basis for the resources they utilize. As a result, cloud users must prioritize effective resource usage. These resources may comprise the Virtual Machine (VM), network bandwidth, and storage space assigned to the virtual machine. These virtual computers are created using hypervisors. This study focuses only on virtual machine auto-scaling techniques. In an ideal world, the demand on the virtual machine determines whether to scale up or down a resource. If the CPU usage hits a predefined upper limit, the cloud infrastructure scales up the VM resource, which means that a new VM is added to the cluster of VMs and the load balancer begins routing new requests to the newly generated VM, and vice versa. However, just evaluating a threshold number for virtual machine usage is a naive approach that cannot be utilized in the actual world. Another element to consider is the time required for the VM to start up and shut down. Once a VM is started, it does not immediately begin operating; the booting process takes time. The same is true for shutting down a virtual machine. Additionally, it is dependent on the resource's setup. Cost is also a concern in the cloud environment, as every virtual machine produced is charged on an hourly basis, and if the scaling mechanism is not configured properly, the user's cost will be significantly increased. I provide a comprehensive review of the literature on such auto-scaling techniques in this work. The paper discusses numerous methods for scaling virtual machine resources in a cloud environment.

1.Auto Scaling Strategy:

Auto scaling, often spelled autoscaling, auto-scaling, and occasionally automated scaling, is a cloud computing approach for dynamically assigning computational resources. The number of active servers in a server farm or pool often varies automatically in response to changing user requirements. Because an application often scales depending on load balancing serving capacity, auto scaling and load balancing are connected. In other words, the load balancer's serving capacity is one of many variables that influence the auto scaling strategy (along with cloud monitoring metrics and CPU use).

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 12, December 2021 ||

| DOI:10.15680/IJIRSET.2021.1012061|

- Autoscaling is a cloud computing capability that allows enterprises to dynamically scale cloud services such as server capacity or virtual machines depending on specified conditions such as traffic or usage levels. Autoscaling technologies are available from cloud computing companies such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).
- Core autoscaling technologies provide lower-cost, more dependable performance by automatically adding and subtracting new instances as demand climbs and falls. As such, autoscaling ensures consistency in the face of dynamic and, at times, unforeseen application demand.
- The general advantage of autoscaling is that it avoids the need to manually react in real time to traffic surges that need more resources and instances by dynamically adjusting the active server count. Each of these servers must be configured, monitored, and decommissioned, which is at the heart of autoscaling.
- For example, it might be difficult to discern whether such a spike is caused by a distributed denial of service (DDoS) assault. Improved monitoring of autoscaling metrics and rules may sometimes assist a system in responding promptly to this problem. Similarly, an auto scaling database automatically adjusts capacity, starts up, and shuts down in response to an application's demands.

2. Important Auto Scaling Terminologies

A single server or computer is considered an instance if it is subject to auto scaling rules set for a group of machines. The group is self-scaling, with each instance subject to the group's auto scaling regulations.
For instance, the Elastic Computing Cloud (EC2) is the AWS ecosystem's compute platform. Within the AWS cloud, EC2 instances provide scalable, customized server alternatives. Amazon EC2 instances are virtual, elastically scaleable on demand, and user-friendly.
An auto scaling group is a logical grouping of Amazon EC2 instances for the purpose of automated scaling. Each Amazon EC2 instance in the group will be automatically scaled according to the same rules.
The size of an auto scaling group refers to the number of instances included inside it. The desired capacity or size of an auto scaling group is the optimal number of instances. If the difference between these two values is greater than zero, the auto scaling group may either instantiate (provision and attach) new instances or delete (detach and terminate) existing instances.
Minimum and maximum size threshold values define cutoff points above and below which instance capacity should not increase or decrease depending on the rules and auto scaling algorithms managing a specific auto scaling group. Additionally, an auto scaling policy often describes any modifications to the planned capacity of the auto scaling group in response to metrics exceeding predefined criteria.
Auto scaling rules often include cooldown times to guarantee that the whole system remains capable of managing traffic. After certain scaling activities, auto scaling cooldown periods provide time to enable freshly created instances to begin managing traffic.
Changes to the intended capacity of an auto scaling group may be either fixed or gradual. Simply said, fixed modifications offer the necessary capacity value. Instead of specifying an end value, incremental adjustments drop or increase by a specified number. Scaling up policies, sometimes referred to as scaling out policies, results in an increase in intended capacity. Scaling down policies, also known as scaling up policies, results in a reduction in intended capacity.
An auto scaling group performs a health check to ensure that associated instances are operating correctly. Inappropriate occurrences should be identified for replacement.
Elastic load balancing software has the capability to do health checks. Amazon EC2 status checks and custom health checks are also available. A positive health check indicates that the instance is still registered and operational with the load balancer that it is connected with, or that the instance is still accessible and exists.
The term "launch configuration" refers to the scripts and settings required to start a new instance. This comprises the instance type, machine image, probable launch zones, buying choices (such as spot vs. on-demand), and launch scripts.

3. Advantages of Auto Scaling

Auto scaling has a number of advantages:
Cost. When traffic is low, auto scaling enables both on-premises and cloud-based enterprises to put certain servers to sleep. This minimises power and water expenses in areas where cooling is accomplished via the use of water. Additionally, cloud auto scaling entails paying for overall utilisation rather than maximum capacity.
Security. Additionally, auto scaling guards against application, hardware, and network failures by identifying and replacing unhealthy instances while maintaining application resilience and availability.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 12, December 2021 ||

| DOI:10.15680/IJIRSET.2021.1012061|

Availability. Auto scaling increases availability and uptime, which is particularly beneficial when production workloads are unpredictable.

While many firms adhere to a daily, weekly, or annual cycle for server utilisation, auto scaling is unique in that it eliminates the possibility of having too many or too few servers to handle the real traffic load. This is because auto scaling, in contrast to a static scaling approach, is sensitive to real use patterns.

For instance, a static scaling solution may take advantage of the fact that traffic is often less around 2:00 a.m. and put certain servers to sleep at that time. In actuality, though, there may be increases at that time—perhaps due to a viral news event or other unforeseen circumstances.

II. RELATED WORK

A. A. Bankole and S. A. Ajila have proposed a paper “Cloud client prediction models for cloud resource provisioning in a multitier web application environment,” (2019).

This paper discusses in depth Predictive models for cloud resource provisioning. To ensure that Service Level Agreement (SLA) requirements are met, effective scaling of Virtual Machine (VM) resources must be supplied several minutes in advance owing to the VM boot-up time. Proactive resource provisioning may be accomplished by forecasting future resource needs. A key difficulty in the domain of resource scaling is that it takes between 5 and 12 minutes for VM instantiation (scaling) to complete before requests can be accepted and delivered. Numerous writers have used a variety of methodologies, including control theory, threshold-based provisioning, and machine learning. [1].

H. Wang and D. Hu, have proposed a paper on “Utilization of Artificial intelligence to Monetize the capacity of Auto Scaling”.

This is a proposed paper for the International Conference on Neural Networks and the Brain in 2020. This article discusses Artificial Intelligence Techniques for Monetizing the Auto Scaling Capability. It created and tested cloud client prediction models for the TPC-W benchmark web application in this study by using three machine learning techniques: Support Vector Machine (SVM), Neural Networks (NN), and Linear Regression (LR). We included reaction time and throughput SLA criteria into the prediction model to deliver a more robust scaling decision to the customer.Our findings indicate that Support Vector Machine is the optimal prediction model. In addition to the typical forecast of a single measure based on CPU use, the monitoring metric is expanded to include reaction time and throughput. [2]

Bouterse B., &Perros, H. have proposed a paper entitled “Scheduling cloud capacity for time-varying customer demand”.

This is a 2019 paper from the IEEE Explore and Processing sections of the International Conference on Cloud Networks. They analyse various alternative methods for sizing the capacity of cloud-based systems in the context of changing client demand in this research. As utility computing resources grow more pervasive, service providers are increasingly turning to the cloud for a fully or partially cloud-based infrastructure to provide utility computing consumers on demand. Given the expense of cloud infrastructure, dynamic scheduling of cloud resources has the potential to dramatically reduce costs while maintaining an acceptable service quality. [3]

III. METHODOLOGY

3.1. System Model

Auto Scaling ensures that there are enough Amazon EC2 instances to execute your application. We may construct an auto-scaling group and populate it with EC2 instances. We may define a minimum number of EC2 instances for that group, and auto-scaling will guarantee that the minimum number of EC2 instances is maintained. Additionally, we may define a maximum number of EC2 instances in each auto scaling group, which will guarantee that instances never exceed that limit. Additionally, with Amazon EC2 auto-scaling, we may set target capacity and auto-scaling rules. Auto-scaling may deploy or terminate EC2 instances based on demand by using the scaling policy.

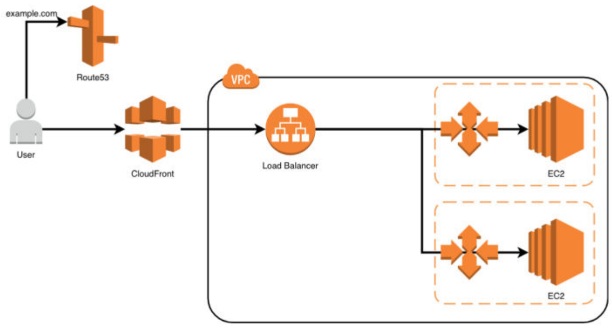

Fig. 1 Auto Scaling System Model

3.2. Auto Scaling Components

i. Groups
They are logical groups that comprise a set of EC2 instances with identical scaling and management features. We may provide the minimum, maximum, and desired number of EC2 instances when we establish a group.

ii. Launch Configuration
It is a template for launching EC2 instances that is utilised by the auto scaling group. While creating the launch configuration, we may enter information about your instances like as the AMI ID, instance type, key pair, and security groups.

iii. Scaling Plan
Auto Scaling uses the scaling strategy to determine how and when to scale. Amazon EC2 auto-scaling enables you to scale the auto scaling group in the following ways:
a>Maintaining Current instance level at all times: In the auto scaling group, we may specify and maintain a specific number of operating instances at all times. Amazon EC2 auto-scaling does this by doing a periodic health check on all running EC2 instances inside an auto scaling group. If an unhealthy instance is discovered, auto-scaling terminates it and creates new ones in its stead.
b>Manual Scaling**:** Manual scaling requires you to declare adjustments to your auto scaling groups' maximum, minimum, or intended capacity. It keeps the instances' capacity up to date.
c>Scaling according to the schedule: Scaling on a timetable entails performing scaling activities at a specified time and date.
d>Demand-driven Scaling: It is the most sophisticated scaling model available. Resources are scaled in this approach by a scaling policy. We may scale up or down your resources based on the specifications. In the policy, we may define parameters such as CPU utilisation and memory use. For instance, we may grow your EC2 instances to reach 80 percent CPU usage. If CPU usage exceeds this threshold value, additional instances will be deployed using the launch configuration by the auto scaling group. Two scaling policies are available: one for scaling in (for instance termination) and one for scaling out (for launching instances).

3.3. Types of Scaling policies

i>Target tracking scaling:We may adjust the existing capacity of the auto scaling group based on the target value of a specific measure.
ii>Step scaling:Using a series of scaling changes, we may raise or reduce the group's present capacity in accordance with the amount of the alarm breach.
iii>Simple scaling:Using a single scaling change, we may raise or reduce the group's existing capacity.

3.4 Auto Scaling Mechanism

The three entities of the end user, AP, and CP communicate with one another through a top-down technique in this architecture. The Web application receives requests from end users through Internet-connected devices. The AP routes

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 12, December 2021 ||

| DOI:10.15680/IJIRSET.2021.1012061|

the incoming request across the load balancer to the VM configured for the application layer. The web application is three-tiered in design, and each tier is typically provided with its own virtual machine. User queries may be routed via many layers of response. Following processing, virtual machines provide the necessary answer to the user. The AP utilises an auto-scaling technique to calculate the needed number of VMs based on the variation in the incoming load. This study contributes by optimising the auto-scaling method during the execution phase. The CP is the lowest layer of the cloud environment architecture and offers computing resources to the AP in the form of virtual machines on a pay-per-use basis.

Monitoring is critical for ensuring Application Auto Scaling's stability, availability, and performance, as well as the performance of your other AWS products. You should gather monitoring data from all components of your AWS system to enable easier debugging in the event of a multi-point failure. AWS offers monitoring tools that keep an eye on Application Auto Scaling, alert you when anything goes wrong, and take relevant action automatically.

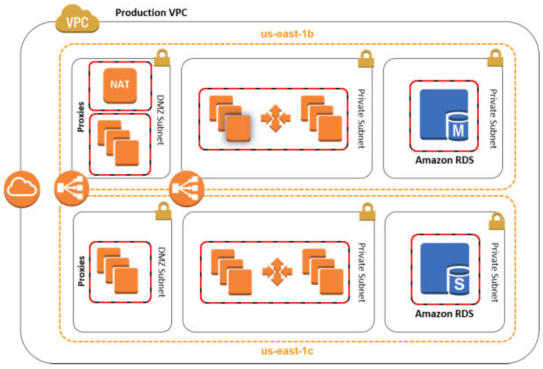

Fig. 2 Model Architecture in VPC

IV. EXPERIMENTAL SETUP

As a pre-requisite,first establish an AMI for your application that is running on your EC2 instance:

• **Launch Configuration:**

1)Motion navigate to the EC2 console and choose Launch Configuration from the Auto Scaling menu.

2) From Choose AMI, go to the My AMIs page and choose the Amazon Machine Image that was used to construct the image for your web application.

3) Then, click Next and choose the instance type that is appropriate for your web application. Configure the specifics.

4) Under Configure details, you may name the launch configuration, specify if a certain IAM role has been allocated to your web application, and enable comprehensive monitoring.

5)Subsequently, add the storage and security groups and do a review.

Note: Allow your programme to execute by opening the relevant ports.

6) Select an existing key pair or create a new key pair by clicking on Create launch configuration.

• **Auto Scaling Group:**

The goal is to develop machines with perceptual and sensory capacities. It employs an unobtrusive mode of sensing to identify user behaviours using physiological measurements.

1)From the EC2 console, go to the Auto Scaling Groups section, which is located under the launch configuration. Then click on auto scaling group creation.

2) On the Auto scaling Group page, you can create a group using either the Launch Configuration or the Launch Template. Here, I've established a launch configuration using Launch Configuration. Additionally, from this page, you may build a new Launch Configuration. Given that you've previously built a launch configuration, you may build an auto scaling group by selecting "Use an existing launch configuration."

3) After clicking on the next step, you may define the group name, starting size of the group, as well as the VPC and subnets. Additionally, by choosing Advanced Details, you may establish load balancing with an auto scaling group.

V. RESULT ANALYSIS

This paper demonstrates that the result of auto scaling is a decrease in resource capacity wastage, system failure, or customer loss.

The reserve capacity model seems to greatly outperform all other models, as seen by the composite metric graph in Figure 3. Resource should be recalibrated annually to ensure optimal performance year after year. Because the overall

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 12, December 2021 ||

| DOI:10.15680/IJIRSET.2021.1012061|

error of an optimised reserve capacity approach is 118,028 unused seat-hours, compared to 167,476 for a Poisson approximation provisioning schedule assuming pre-known demand, the reserve capacity model surpasses all Poisson-based traffic capacity planning methods. This may imply that treating the VCL arrival process as a Poisson process is not a valid assumption. This, however, requires additional investigation.

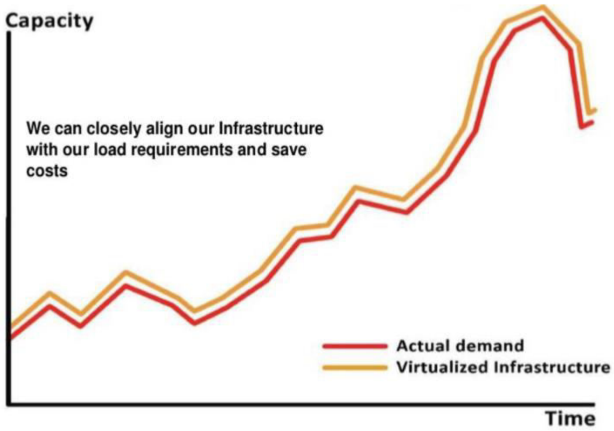

Fig. 3 Auto Scaling Used Capacity Graph

VI. CONCLUSION

In cloud computing systems, auto scaling methods are critical. Typically, such procedures are employed to adapt to workload demands or to terminate inactive virtual machines to save running costs.

It provides an overview of how cloud platforms function and how cloud service providers deliver IaaS architecture to its consumers. With this i can say, utilize the Autoscaling Concept to save costs while also increasing customer satisfaction.

REFERENCES

[1] M. Bertalmio,G.Sapiro,V.Caselles,and C. Ballester, “Image inpainting”, inProc.SIGGRAPH, pp. 417–424, 2000.

[2] Amazon Web Service, Amazon Inc., http://aws.amazon.com/

[3] M. Mao, J. Li, and M. Humphrey, “Cloud Auto-Scaling with Deadline and Budget Constraints,” Proc. 2019 IEEE/ACM Int’l. Conf. Grid Computing, Oct. 2019, pp. 41–48.

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud Computing and Emerging It Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility,” Future Generation Computer Systems, Vol. 25, No. 6, pp. 599-616, 2019

[5] Liao, W.-H., Kuai, S.-C., &Leau, Y.-R. (2019). Auto-scaling Strategy for Amazon Web Services in Cloud Computing. 2019 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). doi:10.1109/smartcity.2019.209

[6] A. A. Bankole and S. A. Ajila, “Cloud client prediction models for cloud resource provisioning in a multitier web application environment,” Proceedings of7thIEEE International Symposium Service System Engineering, pp. 156–161, 2019.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY