Automatic Instance Creation and Load Balancing over Cloud

Suhas B. Shinde
Computer Department
G. V. A. I. E. T.
Mumbai, India

***Abstract*-Automatic Instance Creation allows the user to increase and decrease the number of cloud instances within the application's architecture. Load Balancing automatically distributes incoming application traffic across multiple cloud instances.**

Keywords - Automatic Instance Creation, Load Balancing, Cloud Computing.

I. INTRODUCTION

Automatic Instance Creation allows creation and termination of cloud instances. The instances can be grouped together to form cloud instance groups, termed as Instance Groups. The policies define when Instance Creation launches or terminates cloud instances within a Instance Group.

Load Balancing enables the user to achieve greater level of fault tolerance in user's application, thereby seamlessly providing the required amount of load balancing capacity needed to distribute application traffic.

II. AUTOMATIC INSTANCE CREATION

Automatic Instance Creation allows dynamically increasing or decreasing the number of cloud instances. Each Instance Group may contain one or more scaling policies- these policies define when the creation and termination of cloud instances, within a Instance Group, take place.

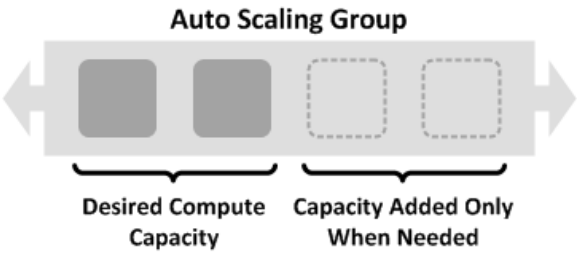


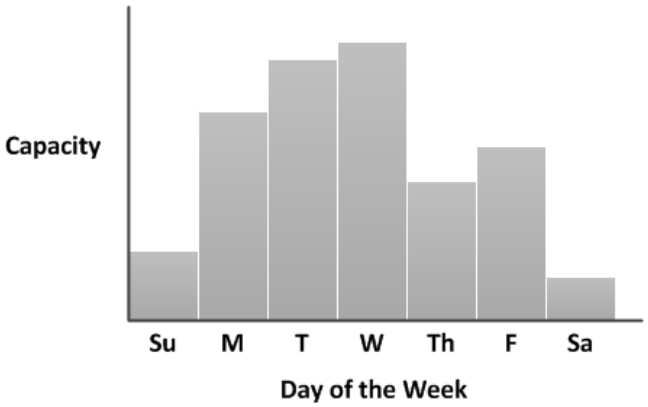
Fig. 1. Automatic Instance/Scaling Group.

User can specify the minimum and maximum number of instances in each Instance Group. Each Instance Group may contain one or more scaling/instance creation policies. User can create as many Instance Groups as needed.Consider an application consists of a web tier and an application tier, user can create two different Groups, one for each tier.

A. An Example

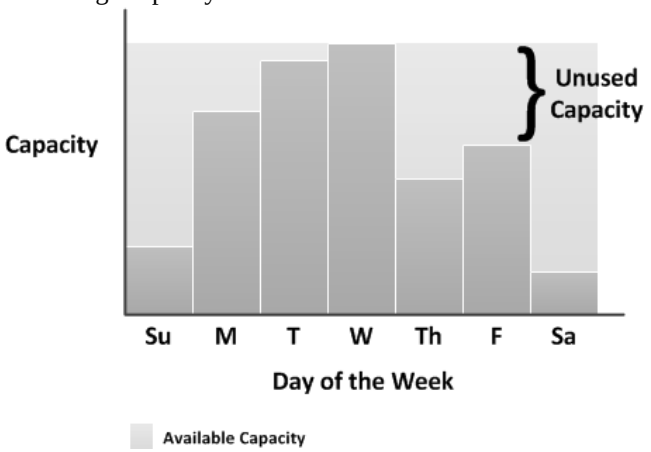
Consider a basic web application running on a web service. The application provides employees with the facility to search for conference rooms that they might want to use for their meetings. The usage of the application is low for the beginning and end part of the week. Along the middle period of the week, more employees are interested in scheduling their meetings, resulting into significant rise in demand for the application usage.

The following graph reflects the application's capacity used over the course of a week.



Traditionally, two options are available for tackling with these changes in the capacity.

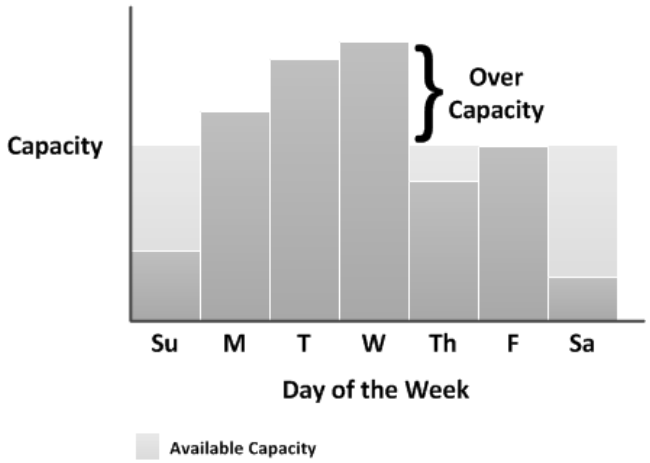
1) To add enough servers so that the application always has enough capacity to meet demand.



The main drawback of this option, however, is that there are days in which the application does not need this capacity.

The extra capacity remains unused and thereby increasing the cost of keeping the application running.

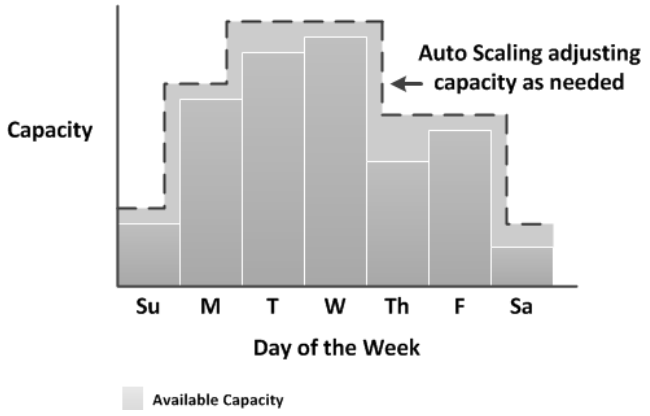
2) To have enough capacity to handle the average demands on the application.



This option is less expensive, because the user is not purchasing equipment that the user might only use occasionally.

However, here the user risks creating a poor customer experience when the demand on the application exceeds its capacity.

By introducing Automatic Instance Creation to this application, the user can make way for a third option.



Here, the user can add new instances to the application only when necessary, and terminate them when they are no longer required.

Also, since Automatic Instance Creation only uses cloud instances, the user is required to pay only for those instances that are utilized by the users in the application.

By involving Automatic Instance Creation in the application's architecture, the resulting architecture becomes cost effective, providing best customer experience while minimizing the expenses.

B. Benefits

Introducing Automatic Instance Creation to a user's network architecture is one way to maximize the benefits provided by a web service. By incorporating Automatic Instance Creation to an application, the user can make the application:

- More fault tolerant. Automatic Instance Creation can detect an unhealthy instance, terminate it, and launch an instance automatically to replace it.

- More highly available. User can configure Automatic Instance Creation to use multiple subnets or Availability Zones. If one subnet or Availability Zone goes down, Automatic Instance Creation can launch instances in another subnet or Availability Zone, thereby compensating the failure of the earlier subnet or Availability Zone.

- Increase and decrease in capacity only when required. Automatic Instance Creation allows dynamically scaling the network.

III. LOAD BALANCING

Elastic Load Balancing automatically distributes incoming web traffic across multiple cloud instances. By involving Load Balancing, user can add and remove cloud instances as per the need or preference of the user, without disrupting the overall flow of information. If one cloud instance fails, Load Balancing automatically reroutes the traffic to the remaining running cloud instances. If over a period of time, the failed cloud instance is restored, the Load Balancing technique automatically restores the traffic to the newly restored instance.

Load Balancer can be set to load balance incoming traffic across cloud instances in a single Availability Zone or multiple Availability Zones. Load Balancing enables even greater fault tolerance in the applications, also it seamlessly provides the amount of load balancing capacity that is needed in response to the incoming application traffic.

Fault tolerant applications can be build by placing cloud instances over multiple Availability Zones. By placing the instances behind an load balancer, the fault tolerance increases drastically, since the load balancer automatically balances the traffic across multiple instances and multiple Availability Zones. This ensures that only healthy cloud instances receive traffic.

Integrating Load Balancing with Automatic Instance Creation ensures back-end capacity available to meet varying traffic levels. Let us consider a user want to make sure that the number of healthy cloud instances behind a Load Balancer is never fewer than two. Then Automatic Instance Creation can be used to set these conditions. So when Automatic Instance Creation detects that a specific condition has been encountered, it thereby automatically incorporates the specified amount of cloud instances to the Instance Group.

Let us consider another example: If a user want to make sure to add cloud instances when the latency of any of the instances exceeds 4 seconds over any 15 minutes period, user can detect that condition. Automatic Instance Creation will take appropriate actions on the user's cloud instances, even when running behind Load Balancer. Automatic Instance Creation works equally well for scaling cloud instances, irrespective of Load Balancer being used or not.

A. Architecture of Load Balancing Service

There are two logical components in the Load Balancing service architecture: Load Balancers and a Controller service. The Load Balancers is the mechanism that monitors traffic and handles requests which penetrate in through the Internet. The controller service monitors the Load Balancer, adds and removes capacity as needed, and verifies that Load Balancers are behaving properly.

B. Features of Load Balancing

1) High Availability: Distributes incoming traffic across cloud instances in a single Availability Zone or multiple Availability Zones. Load Balancing automatically scales its request handling capacity in response to incoming application traffic.

2) Health Checks: Load Balancing can detect the health of cloud instances. When it detects unhealthy cloud instances, it no longer routes traffic to those instances and spreads the load across the remaining healthy instances.

Load Balancing will perform health checks on back-end instances, using the supplied configuration. When an instance is registered with Load Balancer, it won't be considered healthy enough until the number of associated successful health checks which specify a healthy state are completed. The particular instance may also be discarded from the instance pool (but will still be registered and known by the Load Balancer) when the limit on unsuccessful health checks is reached. As long as an instance is registered with Load Balancing, the Load Balancer will continue to perform health checks on an instance. If long intervals of health checks and/or a high healthy threshold are set, it will take more time for instances to start receiving traffic from Load Balancer. This is especially important in case of automated processes and systems to make sure capacity is added to your application pool.

3) Security Features: While using Virtual Private Cloud (VPC), security groups can be created and managed associated with Load Balancing to provide additional networking and security options.

C. Benefits

- Distribution of requests to cloud instances (servers) in multiple Availability Zones so that the risk of overloading one single instance is reduced. Also, if an entire Availability Zone goes down, Load Balancing routes traffic to instances in other Availability Zones.
- Continuous monitoring of the health of cloud instances registered with the Load Balancer so that requests are sent only to the healthy instances.
- Support to end-to-end traffic encryption on those networks that use secure (HTTPS/SSL) connections.
- The ability to centrally handle and manage the encryption and decryption process by the Load Balancer itself, rather than by the cloud instances themselves.

IV. CONCLUSION

Automatic Instance Creation allows automatic, dynamic creation, managing and termination of cloud instances. The cloud instances are dynamically created as per the need and requirement, they are continuously monitored and terminated or discarded after completion of usage or if the instance turns out to be unhealthy.

Load Balancing distributes the incoming network or application traffic across multiple cloud instances or available Availability Zones, which may be either single or multiple.

The proposed technology can enhance the user's cloud experience by improving performance parameters and making optimum utilization of available resources.

The future scope or improvements associated with the technology is to improve the load balancing activities associated with the multiple cloud instances related to a single Availability Zone.

ACKNOWLEDGEMENT

I would like to thank all those individuals who have either directly or indirectly encouraged, supported and/or guided me in the development process of this project of mine.

REFERENCES

[1] Thomas Erl, *Cloud Computing: Concepts, Technology & Architecture*, 2013.
[2] John Rhoton, *Cloud Computing Protected: Security Assessment Handbook*, 2013.
[3] Michael J. Kavis, *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (Saas, PaaS & IaaS)*, 2014.
[4] Raghuram Yeluri, *Building the Infrastructure for Cloud Security*, 2014.
[5] Sanjay Mohapatra and Laxmikant Lokhande, *Cloud Computing and ROI: A New Framework for IT Strategy (Management for Professionals)*, 2014.
[6] Raj Samani, Jim Reavis and Brian Honan, *CSA Guide to Cloud Computing: Implementing Cloud Privacy and Security*, 2014.