A Survey on Comparative Analysis of Horizontal Scaling and Vertical Scaling of Cloud Computing Resources

Asst Prof. Ushma Tailor[1], Asst Prof. Puja Patel[2]
[1, 2]Department of Computer Engineering
[1, 2] Alpha College of Engineering & Technology, Khatraj-382721, Gujarat, India

Abstract- *Cloud computing basically works on the model of "renting" resources. The cloud data centers provides the facility of purchase or renting resources. It can allocate resources such as disk storage, memory, network bandwidth, or processing time. If the client/user is on large scale organization it requires to run multiple servers. So it will face the problem of managing user traffic. So on demand scaling of resources is needed. In this paper two types of scaling, horizontal scaling and vertical scaling is discussed. This paper also describes the comparative analysis of both the types of Scaling.*

I. INTRODUCTION

One of the often referred benefits of cloud computing services is the resource elasticity: a business customer can scale up and scale down its resource usage as per the need without investing capital investment or long term commitments. The amazon EC2 service, for example, allows users to purchase as many virtual machine (VM) instances as they require and operate them more like physical hardware. Though, the clients still need to decide amount of resources which are necessary and for how much amount of time. Numerous Internet applications can have benefit from an auto scaling property where their usage of resource can be scaled up and scaled down automatically by the cloud service provider. And the users are charged only for what they really use the hence it is sometimes reffered as "pay as you go" model [1].

II. RELATED WORK

Any large-scale application needs to run on numerous servers or VM instances to manage user traffic. Here the question is how many such instances you require.
Existing approaches

- One common and simple approach is to estimate peak load and provision for it, though it is infrequent. The drawback of this approach is that many or most resources may remain idle for long periods. So this approach makes the application much more costly than it has to be, because idle but provisioned instances still cost money. Another drawback is that it is challenging or impossible to predict high traffic, specifically for new applications. For instance, mobile games are tremendously variable in their uptake. Many applications have only limited users, while others have a high number. So resources must be provisioned appropriately for a good user experience [2].

- The second approach is to deliver for average use, which wastes less resource. The drawbacks of this approach are (1) it can be difficult to predict average use, especially for new applications, and (2) above-average load mostly results in a bad user experience, which could mean anything from increased latency to user requests which are dropped entirely [2].

These problems can be solved using Auto-scaling. It is mainly beneficial for applications whose usage is difficult or impossible to predict. Here when load rises, more resources are assigned to fulfill the need. So the result is a good user experience, irrespective of load. And when resources are not required, they are scaled back. This means good resource utilization and provisioning, which minimizes the cost of the application. It is tough to provision resources with tremendously high precision, but even a relatively simple auto-scaling process can lead to enhanced resource utilization without compromising the user experience [2].

III. TYPES OF SCALING

Scaling, from the viewpoint of an IT resource, it is the ability of the IT resource to handle increased or decreased demands of usage.

The types of scaling are:
Horizontal Scaling- This includes scaling out & scaling in
Vertical Scaling - This includes scaling up & scaling down
The further two sections briefly narrate each of above.

Horizontal Scaling

Horizontal scaling or Scaling out refers to resource increment by the addition of units to the system. This means

addition of more units of smaller capacity instead of addition of a single unit of the larger capacity. Then the requests for resources are then spread across multiple units hence reducing the excess load on a single machine [3].

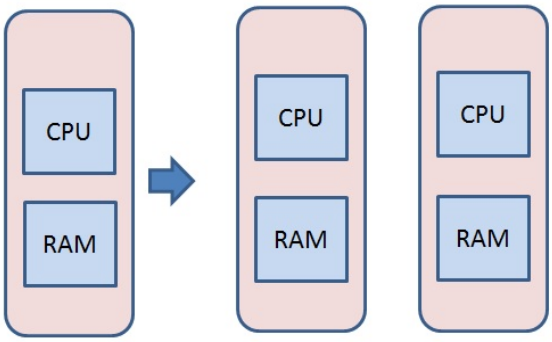

Figure 1: Horizontal Scaling [3]

Having multiple units permits us the probability of keeping the system up even if some of the units go down, thus, avoiding the "single point of failure" issue and also enhancing the availability of the system. Also generally, the aggregate cost incurred by numerous smaller machines is less than the cost of a single larger unit. Hence it can be said that horizontal scaling can be more cost effective in contrast to vertical scaling. Though, there are disadvantages of horizontal scaling as well. Increasing the quantity of units implies that more resources need to be invested in the maintenance. Additionally the code of the application itself should be changed to permit parallelism and distribution of work among various units. In some circumstances this task is not trivial and scaling horizontally can become a tough task [3].

Vertical Scaling

Scaling up or vertical scaling means resource maximization of a single unit to increase its ability to handle increasing load. By aspect of hardware, this includes adding memory and processing power to the physical machine running the server. By aspect of software or programming, scaling up may include optimizing application code and algorithms. Here the Optimization of hardware resources, such as parallelizing or optimizing number of running processes is also considered methods of scaling up.

Although scaling up may be relatively straightforward, vertical scaling method suffers from several disadvantages.

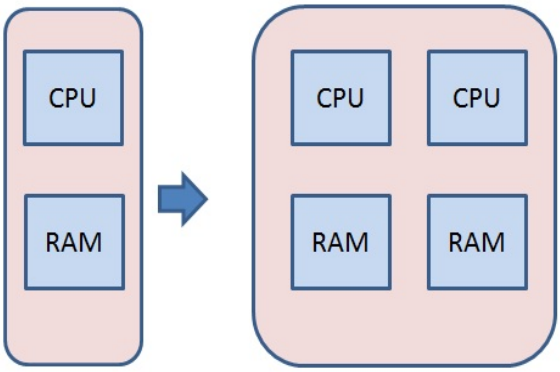

Figure 2: Vertical Scaling [3]

Initially the addition of hardware resources reflects in decreasing returns instead of super-linear scale. The expense for expansion also increases exponentially. The curvature of cost to computational processing is a power law in which the cost starts increasing proportionally to the increase in processing power produced by larger servers. In addition, there is the inevitable downtime need for scaling up. If all of the web application services and data remain on a single unit, vertical scale on this unit does not gives assurance the application's availability [3].

IV. COMPARATIVE ANALYSIS OF HORIZONTAL AND VERTICAL SCALING

TABLE I COMPARATIVE ANALYSIS OF HORIZONTAL AND VERTICAL SCALING [4, 5, 6, 7, 8]

	Sr no.	Vertical Scaling	Horizontal Scaling
Meaning	1	To increase the capacity if we increase the resources in same logical unit or server then it is called vertical scaling.	Horizontal scaling referred as increasing the performance of server or node by adding more instances of server to the pool of servers so load can be spreaded.
Reason to scale	2	It includes increasing IOPS (Input / Ouput Operations), increasing disk capacity and CPU/RAM capacity**.**	It includes increasing I/O concurrency, increasing disk capacity and reducing the load on existing nodes.
Efficiency	3	Vertical scaling is also fairly inefficient regarding resource reallocation. Because servers are inclined be dedicated to	Horizontal scaling, on the other hand, adds more nodes to the system as it scales, rather than it beefs up the

		specific tasks, it can be tough to reallocate the spare resources of a server to other, more processing tasks.	existing nodes. This is relatively the more popular scaling strategy in modern times.
Complexity	4	Less complex	More complex
Throughput	5	Less throughput	Horizontal scaling grants us more throughputs.
Application/database server	6	Application server or database server remains common.	Each node has separate application or database server.
Failure Recovery	7	Failure recovery is very difficult task	Failure Recovery is easy
approach	8	Scale up approach	Scale out approach
Scenario	9	This scenario concentrates on the situation that a hardware platform has enough capacity to host more than one instance of an application.The application is reproduced on the same hardware until the capacity requirements are met.	This scenario means the ability to increase a system's capability or its performance by replicating a system(comprising of hardware or a virtualized platform) until the capacity requirement is satisfied.
Cost	10	expensive	Cost effective

V. CONCLUSION

In this paper a comparison is shown between two of techniques of scaling of cloud computing resources, which are horizontal scaling and vertical scaling. The comparison regarding complexity, throughput, cost and efficiency of both the techniques is described.

REFERENCES

[1] Xiao, Zhen, Qi Chen, and Haipeng Luo. "Automatic scaling of internet applications for cloud computing services." (2012): 1-1.

[2] https://cloud.google.com/developers/articles/auto-scaling-on-the-google-cloud-platform/

[3] www.comp.nus.edu.sg/~seer/book/2e/Ch06.%20Scalability.pdf

[4] http://www.esds.co.in/blog/what-is-the-difference-between -horizontal-vertical-scaling/

[5] http://www.pc-freak.net/blog/vertical-horizontal-server-services-scaling-vertical-horizontal-hardware-scaling/

[6] http://www.techopedia.com/definition/7594/horizontal-scaling

[7] http://blogs.vmware.com/vcloud/2010/11/thinking-differently-about-scalability-with-cloud-computing.html

[8] http://www.dnsmadeeasy.com/blog/vertical-and-horizontal-scaling