UNIT 5 SCALING

Structure:-

5.1 INTRODUCTION

The scalability in cloud computing refers to the flexibility of allocating IT resources as per the demand. Various applications running on cloud instances experience variable traffic loads and hence the need of scaling arises. The need of such applications can be of different types such as CPU allocation, Memory expansion, storage and networking requirements etc. To address these different requirements, virtual machines are one of the best ways to achieve scaling. Each of the virtual machines is equipped with a minimum set of configurations for CPU, Memory and storage. As and when required, the machines can be configured to meet the traffic load. This is achieved by reconfiguring the virtual machine for better performance for the target load. Sometimes it is quite difficult to manage such ondemand configurations by the persons, hence auto scaling techniques plays a good role.

In this unit we will focus on the various methods and algorithms used in the process of scaling. We will discuss various types of scaling, their usage and a few examples. We will also discuss the importance of various techniques in saving cost and man efforts by using the concepts of cloud scaling in highly dynamic situations. The suitability of scaling techniques in different scenarios is also discussed in detail.

For scaling, to understand elastic property of Cloud is important. I would recommend to brief about the Cloud Elasticity here?

5.2 OBJECTIVES

After going through this unit you should be able to:

- ➔ describe scaling and its advantage;
- ➔ understand the different scaling techniques;
- ➔ learn about the scaling up and down approaches;
- ➔ understand the basics of auto scaling
- ➔ compare among Proactive and Reactive scaling;

5.3 SCALING PRIMITIVES

The basic purpose of scaling is to enable one to use cloud computing infrastructure as much as required by the application. Here, the cloud resources are added or removed according to the current need of the applications. The property to enhance or to reduce the resources in the cloud is referred to as cloud elasticity, ~~the process is known as scaling.~~ Scaling exploits the elastic property of the Cloud. The scalability of cloud architecture is achieved using virtualization (see Unit 3: Resource Virtualization). Virtualization uses virtual machines (VM's) for enhancing (up scaling) and reducing (down scaling) computing power. The scaling provides opportunities to grow businesses to a more secure, available and need based computing/ storage facility on the cloud. Scaling also helps in optimizing the financial involved for highly resource bound applications for small to medium enterprises.

Better to include one picture to explain cloud elasticity?

The key advantages of cloud scaling are: -

1. Minimum cost: The user has to pay a minimum cost for access usage of hardware after upscaling. The hardware cost for the same scale can be much greater than the cost paid by the user. Also, the maintenance and other overheads are also not included here. Further, as and when the resources are not required, they may be returned to the Service provider resulting in the cost saving.

2. Ease of use: The cloud upscaling and downscaling can be done in just a few minutes (sometime dynamically) by using service providers application interface.

3. Flexibility: The users have the flexibility to enable/ disable certain VM's for upscaling and downscaling by them self and thus saving configuration/ installation time for new hardware if purchased separately.

4. Recovery: The cloud environment itself reduces the chance of disaster and amplifies the recovery of information stored in the cloud.

The scalability of the clouds aims to optimize the utilization of various resources under varying workload conditions such as under provisioning and over provisioning of resources. In non-cloud environments resource utilization can be seen as a major concern as one has no control on scaling. Various methods exist in literature which may be used in traditional environment scaling. In general, a peak is forecasted and accordingly infrastructure is set up in advance. This scaling experience high latency and require manual monitoring. The associated drawbacks of this type of setup is quite crucial in nature as estimation of maximum load may exist at both ends making either high end or poorly configured systems.

In the case of the clouds, virtual environments are utilized for resource allocation. These virtual machines enable clouds to be elastic in nature which can be configured according to the workload of the applications in real time. In

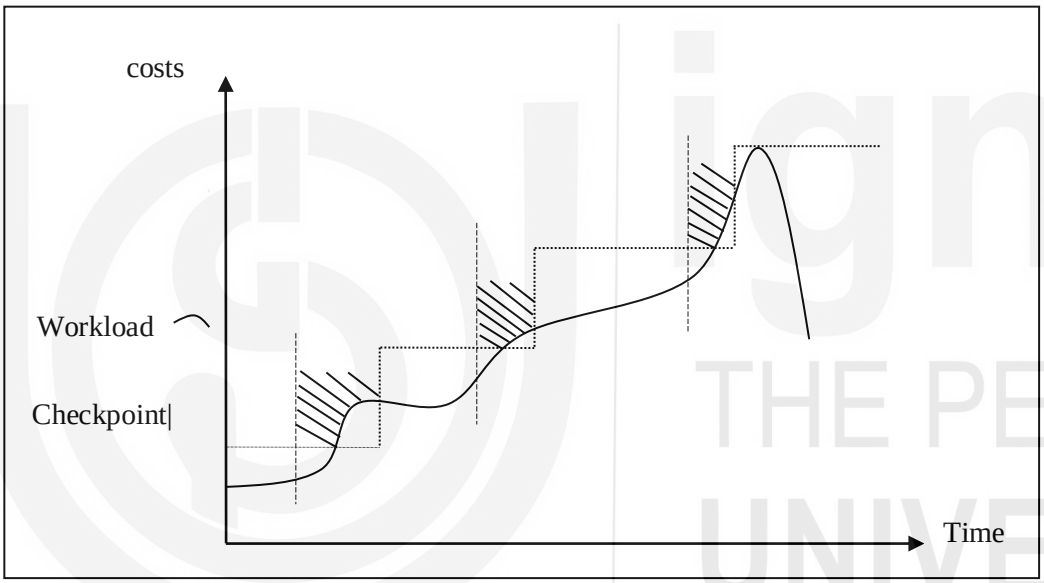


Figure 1. Manual scaling in traditional environments

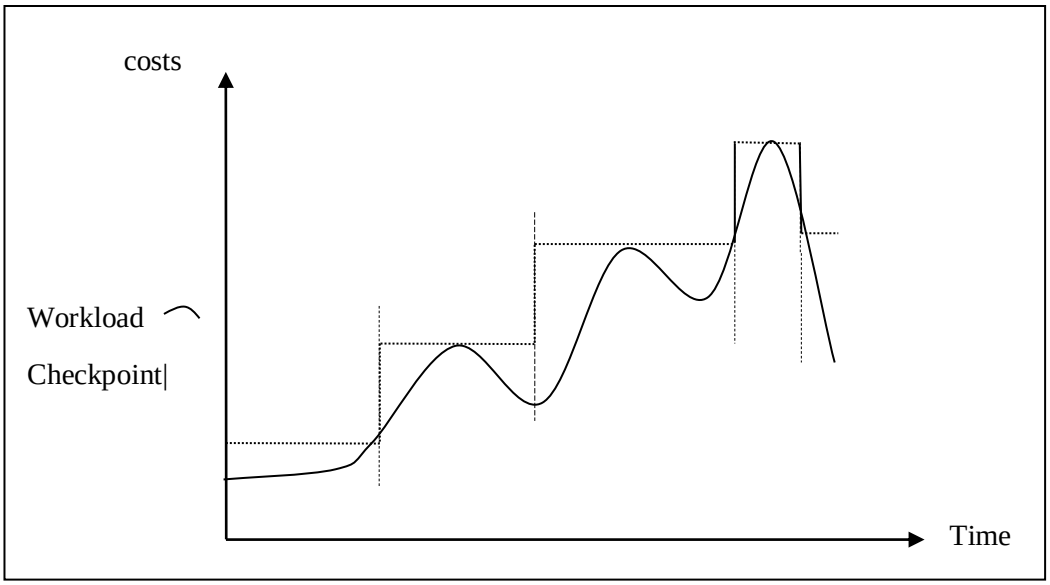


Figure 2. Semi-automatic scaling in cloud environments.

such scenarios, downtime is minimized and scaling is easy to achieve.

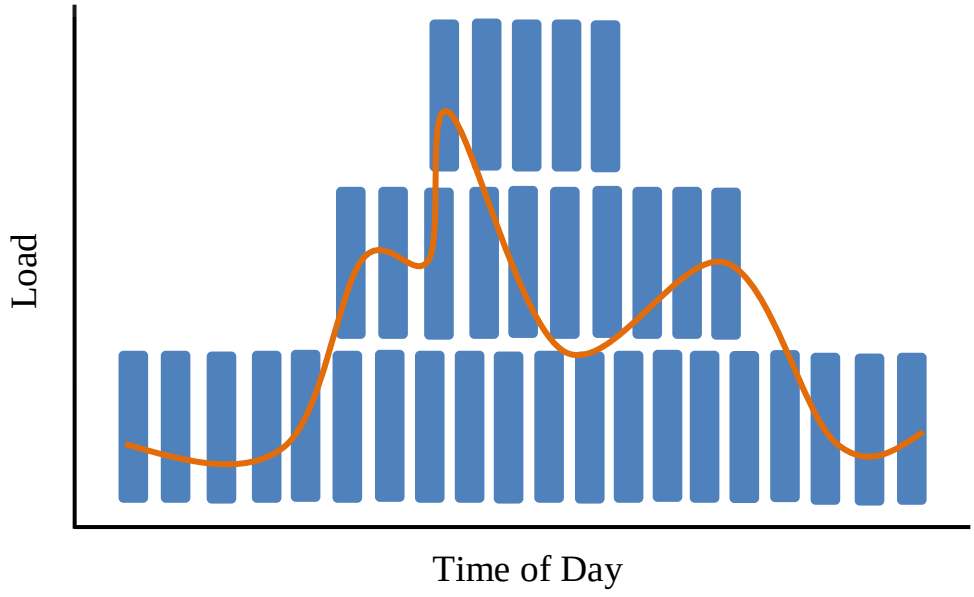
On the other hand, scaling saves cost of hardware setup for some small time peaks or dips in load. In general most cloud service providers provide scaling as a process for free and charge for the additional resource used. Scaling is also a common service provided by almost all cloud platforms. Also need to mention that user saves when usage of the resources declines by using scale down.?

5.4 SCALING SRATEGIES

Let us now see what are the strategies for scaling, how one can achieve scaling in a cloud environment and what are its types. In general, scaling is categorized based on the decision taken for achieving scaling. The three main strategies for scaling are discussed below.

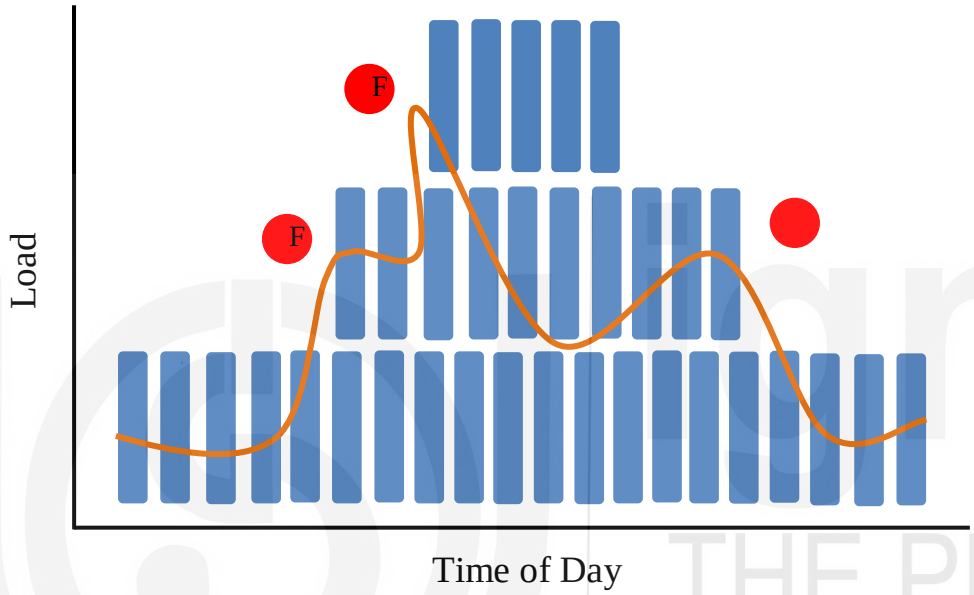
5.4.1 Proactive Scaling

Consider a scenario when a huge surge in traffic is expected on one of the applications in the cloud. In this situation a proactive scaling is used to cater the load. The proactive scaling can also be pre scheduled according to the expected traffic and demand. This also expects the understanding of traffic flow in advance to utilize maximum resources, however wrong estimates generally lead to poor resource management. The prior knowledge of the load helps in better provisioning of the cloud and accordingly minimum lag is experienced by the end users when sudden load arrives. The given below figure shows the resource provision when load increases with time.



5.4.2 Reactive Scaling

The reactive scaling often monitors and enables smooth workload changes to work easily with minimum cost. It empowers users to easily scale up or down computing resources rapidly. In simple words, when the hardwares like CPU or RAM or any other resource touches highest utilization, more of the resources are added to the environment by the service providers. The auto scaling works on the policies defined by the users/ resource managers for traffic and scaling. One major concern with reactive scaling is a quick change in load, i.e. user experiences lags when infrastructure is being scaled.



5.4.3 Combinational Scaling

Till now we have seen need based and forecast based ~~scaling~~ techniques for scaling. However, for better performance and low cool down period we can also combine both of the reactive and proactive scaling strategies where we have some prior knowledge of traffic. This helps us in scheduling timely scaling strategies for expected load. On the other hand, we also have provision of load based scaling apart from the predicted load on the application. This way both the problems of sudden and expected traffic surges are addressed.

Given below is the comparison between proactive and reactive scaling strategies.

Parameters	Proactive Scaling	Reactive Scaling
Suitability	For applications increasing loads in expected/ known manner	For applications increasing loads in unexpected/ unknown manner
Working	User sets the threshold but a	User defined threshold values

	downtime is required.	optimize the resources
Cost Reduction	Medium cost reduction	Medium cost reduction
Implementation	A few steps required	Fixed number of steps required

Check your Progress 1

1) Explain the importance of scaling in cloud computing?

……

……

……

2) How proactive scaling is achieved through virtualization?

……

……

……

3) Write differences between combinational and reactive scaling.

……

……

……

5.5 AUTO SCALING IN CLOUD

One of the potential risks in scaling a cloud infrastructure is its magnitude of scaling. If we scale it down to a very low level, it will adversely affect the throughput and latency. In this case, a high latency will be affecting the user's experience and can cause dissatisfaction of the users. On the other hand, if we scale-up the cloud infrastructure to a large extent then it will not be a resource optimization and also would cost heavily, affecting the host and the whole purpose of cost optimization fails.

In a cloud, auto scaling can be achieved using user defined policies, various machine health checks and schedules. Various parameters such as Request counts, CPU usage and latency are the key parameters for decision making in autoscaling. A policy here refers to the instruction sets for clouds in case of a particular scenario (for scaling -up or scaling -down). The autoscaling in the cloud is done on the basis of following parameters.

1. The number of instances required to scale.
2. Absolute no. or percentage (of the current capacity)

The process of auto scaling also requires some *cooldown* period for resuming the services after a scaling takes place. No two concurrent scaling are triggered so as to maintain integrity. The cooldown period allows the process of autoscaling to get reflected in the system in a specified time interval and saves any integrity issues in cloud environment.

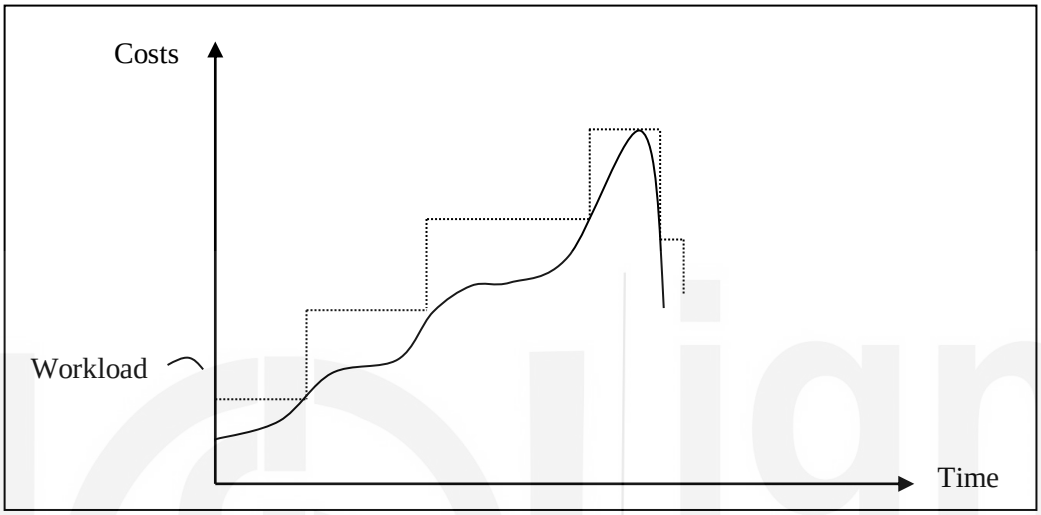


Figure 4. Automatic scaling in cloud environments

Consider a more specific scenario, when the resource requirement is high for some time duration e.g. in holidays, weekends etc., a *Scheduled* scaling can also be performed. Here the time and scale/ magnitude/ threshold of scaling can be defined earlier to meet the specific requirements based on the previous knowledge of traffic. The threshold level is also an important parameter in auto scaling as a low value of threshold results in under utilization of the cloud resources and a high level of threshold results in higher latency in the cloud.

After adding additional nodes in scale-up, the incoming requests per second drops below the threshold. This results in triggering the alternate scale-up-down processes known as a ping-pong effect. To avoid both underscaling and overscaling issues load testing is recommended to meet the service level agreements (SLAs). In addition, the scale-up process is required to satisfy the following properties. Need to brief on SLA also?

1. The number of incoming requests per second per node > threshold of scale down, after scale-up.
2. The number of incoming requests per second per node < threshold of scale up, after scale-down

Here, in both the scenarios one should reduce the chances of ping-pong effect.

Now we know what scaling is and how it affects the applications hosted on the cloud. Let us now discuss how auto scaling can be performed in fixed amounts as well as in percentage of the current capacity.

Fixed amount autoscaling

As discussed earlier, the auto scaling can be achieved by determining the number of instances required to scale by a fixed number. The detailed algorithm for fixed amount autoscaling threshold is given below. The algorithm works for both scaling-up and scaling-down and takes inputs U and D for both respectively.

Algorithm : 1

Input : SLA specific application

Parameters:

N_min minimum number of nodes
D - scale down value.
U scale up value.
T_U scale up threshold
T_D scale down threshold

Let T (SLA) return the maximum incoming request per second (RPS) per node for the specific SLA.

T_D ← 0.50 x T_U
T_U ← 0.90 x T (SLA)

Let N_c and RPS_n represent the current number of nodes and incoming requests per second per node respectively.

L1: /* scale up (if RPS_n> T_U) */
Repeat:
    N_(c_old) ←N_c
    N_c ←N_c + U
    RPS_n ←RPS_n x N_(c_old) / N_c
Until RPS_n> T_U

L2: /* scale down (if RPS_n< T_D) */

Repeat:
    N_(c_old) ←N_c
    N_c ← max(N_min, N_c - D)
    RPS_n ←RPS_n x N_(c_old) / N_c
Until RPS_n< T_D or N_c = N_min

Now, let us discuss how this algorithm works in detail. Let the values of a few parameters are given as U = 2, D = 2, T_U = 120 and T_D = 150. Suppose in the beginning, RPS = 450 and N_c = 4. Now RPS is increased to 1800 and RPS_n almost reached to T_U, in this situation an autoscaling request is generated leading to adding U = 2 nodes. Table - 1 lists all the parameters as per the scale -up requirements.

Nodes (Current)	Nodes (added)	RPS (required)	RPS_n	Total nodes	New RPS_n
4	0	450	112.5	4	
		1800			
	2			6	300
		2510			
	2			8	313.75
		3300			
	2			10	330.00
		4120			
	2			12	343.33
		5000			
	2			14	357.14

Similarly, in case of scaling down, let initially RPS = 8000 and N_c = 19. Now RPS is reduced to 6200 and following it RPS_n reaches T_D, here an autoscaling request is initiated deleting D = 2 nodes. Table - 2 lists all the parameters as per the scale -down requirements.

Nodes (Current)	Nodes (reduced)	RPS (required)	RPS_n	Total nodes	New RPS_n
18		8000	421.05	19	
		6200			
	2			17	364.7
		4850			
	2			15	323.33
		3500			

	2			13	269.23
		2650			
	2			11	240.90
		1900			
	2			8	211.11

The given table shows the stepwise increase/ decrease in the cloud capacity with respect to the change in load on the application(request per node per second).

Percentage Scaling:

In the previous section we discussed how scaling up or down is carried out by a fixed amount of nodes. Considering the situation when we scale up or down by a percentage of current capacity we change using percentage change in current capacity. This seems a more natural way of scaling up or down as we are already running to some capacity.

The below given algorithm is used to determine the scale up and down thresholds for respective autoscaling.

Algorithm : 2

Input : SLA specific application

Parameters:

N_min - minimum number of nodes
D - scale down value.
U - scale up value.
T_U - scale up threshold
T_D - scale down threshold

Let T (SLA) returns the maximum requests per second (RPS) per node for specific SLA.

$T\_U \leftarrow 0.90 \times T(SLA)$
$T\_D \leftarrow 0.50 \times T\_U$

Let N_c and RPS_n represent the current number of nodes and incoming requests per second per node respectively.

L1: /* scale up (if RPS_n> T_U) */
Repeat:
N_(c_old) ←N_c

```
        N_c ←N_c + max(1, N_c x U/100)
        RPS_n ←RPS_n x N_(c_old) / N_c
Until   RPS_n> T_U

L2: /* scale down (if RPS_n< T_D) */

Repeat:
        N_(c_old) ←N_c
        N_c ← max(N_min, N_c - max(1, N_c x D/ 100))
        RPS_n ←RPS_n x N_(c_old) / N_c
Until   RPS_n< T_D or N_c = N_min
```

Let us now understand the working of this algorithm by an example. Let N_min = 1, at the beginning RPS = 500 and N_c = 6. Now the demand rises and RPS reaches to 1540 while RPS_n reaches T_U. Here an upscaling is requested adding 1 i.e. max(1, 6 x 10/200) nodes.

Similarly in case of scaling down, initial RPS = 5000 and N_c = 19, here RPS reduces to 4140 and RPS_n reaches T_D requesting scale down and hence deleting 1 i.e. max(1, 1.8 x 8/100). The detailed example is explained using Table -3 giving details of upscaling with D = 8, U = 1, N_min = 1, T_D = 230 and T_U = 290 .

Nodes (Current)	Nodes (added)	RPS (required)	RPS_n	Total nodes	New RPS_n
6	0	500	83.33	6	
		1695			
	1			7	242.14
		2190			
	1			8	273.75
		2600			
	1			9	288.88
		3430			
	1			10	343.00
		3940			
	1			11	358.18
		4420			
	1			12	368.33

		4960			
	1			13	381.53
		5500			
	1			14	392.85
		5950			
	1			15	396.6

The scaling down with the same algorithm is detailed in the table below.

Nodes (Current)	Nodes (added)	RPS (required)	RPS_n	Total nodes	New RPS_n
19		5000	263.15	19	
		3920			
	1			18	217.77
		3510			
	1			17	206.47
		3200			
	1			16	200
		2850			
	1			15	190
		2600			
	1			14	185.71
		2360			
	1			13	181.53
		2060			
	1			12	171.66
		1810			
	1			11	164.5
		1500			
					150

Here if we compare both the algorithms 1 and 2, it is clear that the values of the threshold U and D are at the higher side in case of 2. In this scenario the utilization of hardware is more and the cloud experiences low footprints.

Check your Progress 2

1) Explain the concept of fixed amount auto scaling.

……

……

……

2) In Algorithm 1 for fixed amount auto scaling, calculate the values in table if U = 3.

……

……

……

3) What is a cool down period?

……

……

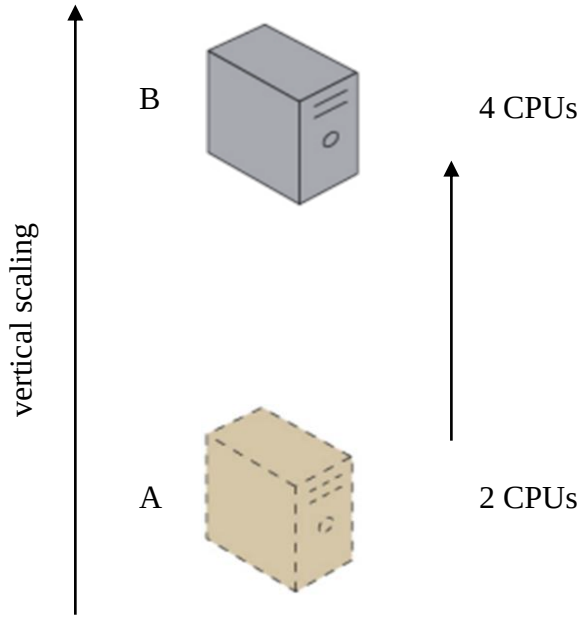
……

5.6 TYPE OF SCALING

Let us now discuss the types of scaling, how we see the cloud infrastructure for capacity enhancing/ reducing. In general we scale the cloud in a vertical or horizontal way by either provisioning more resources or by installing more resources.

5.6.1 Vertical scaling or scaling up

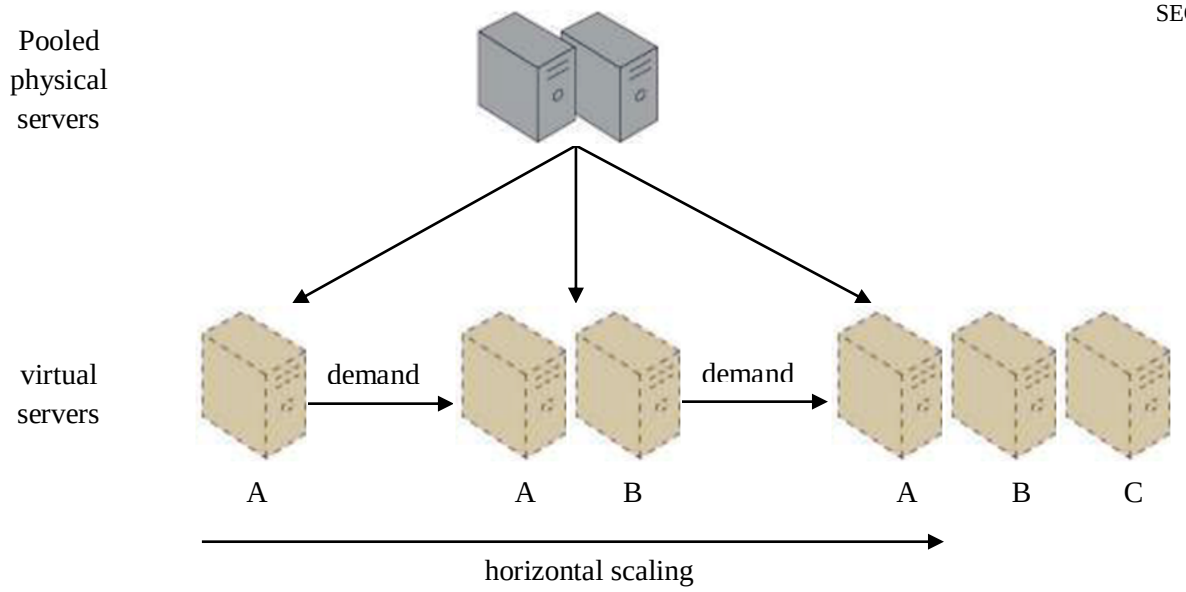
The vertical scaling in the cloud refers to either scaling up i.e. enhancing the computing resources or scaling down i.e. reducing/ cutting down computing resources for an application. In vertical scaling, the actual number of VMs are constant but the quantity of the resource allocated to each of them is increased/ decreased. Here no infrastructure is added and application code is also not changed. The vertical scaling is limited to the capacity of the physical machine or server running in the cloud. If one has to upgrade the hardware requirements of an existing cloud environment, this can be achieved by minimum changes.



An IT resource (a virtual server with two CPUs) is scaled up by replacing it with a more powerful IT resource with increased capacity for data storage (a physical server with four CPUs).

5.6.2 Horizontal scaling or scaling out

In horizontal scaling, to meet the user requirements for high availability, excess resources are added to the cloud environment. Here, the resources are added/ removed as VMs. This includes addition of storage disks, new server for increasing CPUs or installation of additional RAMs and work like a single system. To achieve horizontal scaling, a minimum downtime is required. This type of scaling allows one to run distributed applications in a more efficient manner.



An IT resource (Virtual Server A) is scaled out by adding more of the same IT resources (Virtual Servers B and C).

Another way of maximizing the resource utilization is Diagonal Scaling. This combines the ideas of both vertical and horizontal scaling. Here, the resource is scaled up vertically till one hit the physical resource capacity and afterwards new resources are added like horizontal scaling. The new added resources have further capacity of being scaled like vertical scaling.

SUMMARY

In the end, we are now aware of various types of scaling, scaling strategies and their use in real situations. Various cloud service providers like Amazon AWS, Microsoft Azure and IT giants like Google offer scaling services on their application based on the application requirements. These services offer good help to the entrepreneurs who run small to medium businesses and seek IT infrastructure support. We have also discussed various advantages of cloudscaling for business applications.

SOLUTION/ANSWERS

Answers to CYPs 1.

1. Explain the importance of scaling in cloud computing: Clouds being used extensively in serving applications and in other scenarios where the cost and installation time of infrastructure/ capacity scaling is expectedly high. Scaling helps in achieving optimized infrastructure for the current and expected load for the applications with minimum cost and setup time. Scaling also helps in reducing the disaster recovery time if happens. (for details see section 5.3)

2. How proactive scaling is achieved through virtualization: The proactive scaling is a process of forecasting and then managing the load on the could infrastructure in advance. The precise forecasting of the requirement is key to success here. The preparedness for the estimated traffic/ requirements is done using the virtualization. In virtualization, various resources may be assigned to the required machine in no time and the machine can be scaled to its hardware limits. The virtualization helps in achieving low cool down period and serve instantly. (for details you may refer Resource Utilization Unit.)

3) Write differences between proactive and reactive scaling: The reactive scaling technique only works for the actual variation of load on the application however, the combination works for both expected and real traffic. A good estimate of load increases performance of the combinational scaling.

Answers to CYPs 2.

1) Explain the concept of fixed amount auto scaling: The fixed amount scaling is a simplistic approach for scaling in cloud environment. Here the resources are scaled up/ down by a user defined number of nodes. In fixed amount scaling resource utilization is not optimized. It can also happen that only a small node can solve the resource crunch problem but the used defined numbers are very high leading to underutilized resources. Therefore a percentage amount of scaling is a better technique for optimal resource usage.

2) In Algorithm 1 for fixed amount auto scaling, calculate the values in table if U = 3: For the given U = 3, following calculation are made.

Nodes (Current)	Nodes (added)	RPS (required)	RPS_n	Total nodes	New RPS_n
4	0	450	112.5	4	
		1800			
	3			7	257.14
		2510			
	3			10	251
		3300			
	3			13	253.84
		4120			
	3			16	257.50

		5000			
	3			19	263.15

3) What is a cool down period: When auto scaling takes place in cloud, a small time interval (pause) prevents the triggering next auto scale event. This helps in maintaining the integrity in the cloud environment for applications. Once the cool down period is over, next auto scaling event can be accepted.