The demand for computing resources, such as CPU cycles, primary and secondary storage, and network

bandwidth, depends heavily on the volume of data processed by an application. The demand for

resources can be a function of the time of day, can monotonically increase or decrease in time, or can

experience predictable or unpredictable peaks. For example, a new Web service will experience a low

request rate when the service is first introduced and the load will exponentially increase if the service is

successful. A service for income tax processing will experience a peak around the tax filling deadline,

whereas access to a service provided by Federal EmergencyManagement Agency (FEMA)will increase

dramatically after a natural disaster.

The elasticity of a public cloud, the fact that it can supply to an application precisely the amount

of resources it needs and that users pay only for the resources they consume are serious incentives to

migrate to a public cloud. The question we address is: How scaling can actually be implemented in a

cloud when a very large number of applications exhibit this often unpredictable behavior [62,233,357].

To make matters worse, in addition to an unpredictable external load the cloud resource management

has to deal with resource reallocation due to server failures.

We distinguish two scaling modes: vertical and horizontal. Vertical scaling keeps the number of VMs

of an application constant, but increases the amount of resources allocated to each one of them. This

can be done either by migrating the VMs to more powerful servers or by keeping the VMs on the same

servers but increasing their share of the CPU time. The first alternative involves additional overhead;

the VMis stopped, a snapshot of it is taken, the file is transported to a more powerful server, and, finally,

the VM is restated at the new site.

Horizontal scaling is the most common mode of scaling on a cloud; it is done by increasing the

number of VMs as the load increases and reducing the number of VMs when the load decreases. Often,

this leads to an increase in communication bandwidth consumed by the application. Load balancing

among the running VMs is critical to this mode of operation. For a very large application, multiple load

balancers may need to cooperate with one another. In some instances the load balancing is done by a

front-end server that distributes incoming requests of a transaction-oriented system to back-end servers.

An application should be designed to support scaling. As we saw in Section 4.6 in the case of a

modularly divisible application, the workload partitioning is static, it is decided a priori, and cannot be

changed; thus, the only alternative is vertical scaling. In the case of an arbitrarily divisible application

the workload can be partitioned dynamically; as the load increases, the system can allocate additional

VMs to process the additional workload. Most cloud applications belong to this class, which justifies

our statement that horizontal scaling is the most common scaling mode.

Mapping a computation means to assign suitable physical servers to the application. Avery important

first step in application processing is to identify the type of application and map it accordingly. For

example, a communication-intensive application should be mapped to a powerful server to minimize

the network traffic. This may increase the cost per unit of CPU usage, but it will decrease the computing

time and probably reduce the overall cost for the user. At the same time, it will reduce the network

traffic, a highly desirable effect from the perspective of the cloud service provider. To scale up and down a compute-intensive application, a good strategy is to increase or decrease the number of VMs or

instances. Because the load is relatively stable, the overhead of starting up or terminating an instance

does not increase significantly the computing time or the cost.

There are several strategies to support scaling. Automatic VM scaling uses predefined metrics, e.g.,

CPU utilization, to make scaling decisions. Automatic scaling requires sensors to monitor the state of

VMs and servers; controllers make decisions based on the information about the state of the cloud, often

using a state machine model for decision making. Amazon and Rightscale (www.rightscale.com)

offer automatic scaling. In the case of AWS the CloudWatch service supports applications monitoring

and allows a user to set up conditions for automatic migrations.

Nonscalable or single-load balancers are also used for horizontal scaling. The Elastic Load Balancing

service from Amazon automatically distributes incoming application traffic across multiple EC2

instances. Another service, the Elastic Beanstalk, allows dynamic scaling between a low and a high

number of instances specified by the user (see Section 3.1). The cloud user usually has to pay for the

more sophisticated scaling services such as Elastic Beanstalk.