The elements involved in a control system are sensors, monitors, and actuators. The sensors measure the

parameter(s) of interest, then transmit the measured values to a monitor, which determines whether the

system behavior must be changed, and, if so, it requests that the actuators carry out the necessary actions.

Often the parameter used for admission control policy is the current system load; when a threshold,

e.g., 80%, is reached, the cloud stops accepting additional load.

In practice, the implementation of such a policy is challenging or outright infeasible. First, due to

the very large number of servers and to the fact that the load changes rapidly in time, the estimation

of the current system load is likely to be inaccurate. Second, the ratio of average to maximal resource

requirements of individual users specified in a service-level agreement is typically very high. Once an

agreement is in place, user demands must be satisfied; user requests for additional resources within the

SLA limits cannot be denied.

Thresholds. A threshold is the value of a parameter related to the state of a system that triggers a

change in the system behavior. Thresholds are used in control theory to keep critical parameters of a

system in a predefined range. The threshold could be static, defined once and for all, or it could be

dynamic. A dynamic threshold could be based on an average of measurements carried out over a time

interval, a so-called integral control. The dynamic threshold could also be a function of the values of

multiple parameters at a given time or a mix of the two.

To maintain the system parameters in a given range, a high and a low threshold are often defined.

The two thresholds determine different actions; for example, a high threshold could force the system to

limit its activities and a low threshold could encourage additional activities. Control granularity refers

to the level of detail of the information used to control the system. Fine control means that very detailed

information about the parameters controlling the system state is used, whereas coarse control means

that the accuracy of these parameters is traded for the efficiency of implementation.

Proportional Thresholding. Application of these ideas to cloud computing, in particular to the IaaS

delivery model, and a strategy for resource management called proportional thresholding are discussed

in [217]. The questions addressed are:

• Is it beneficial to have two types of controllers, (1) application controllers that determine whether

additional resources are needed and (2) cloud controllers that arbitrate requests for resources and

allocate the physical resources?

• Is it feasible to consider fine control? Is course control more adequate in a cloud computing

environment?

• Are dynamic thresholds based on time averages better than static ones?

• Is it better to have a high and a low threshold, or it is sufficient to define only a high threshold?

The first two questions are related to one another. It seems more appropriate to have two controllers,

one with knowledge of the application and one that’s aware of the state of the cloud. In this case a coarse

control is more adequate for many reasons. As mentioned earlier, the cloud controller can only have a

very rough approximation of the cloud state. Moreover, to simplify its resource management policies,

the service provider may want to hide some of the information it has. For example, it may not allow a

VM to access information available to VMM-level sensors and actuators.

To answer the last two questions, we have to define a measure of “goodness.” In the experiments

reported in [217], the parameter measured is the average CPU utilization, and one strategy is better than

another if it reduces the number of requests made by the application controllers to add or remove virtual

machines to the pool of those available to the application.

Devising a control theoretical approach to address these questions is challenging. The authors of

[217] adopt a pragmatic approach and provide qualitative arguments; they also report simulation results

using a synthetic workload for a transaction-oriented application, a Web server.

The essence of the proportional thresholding is captured by the following algorithm:

1. Compute the integral value of the high and the low thresholds as averages of the maximum and,

respectively, the minimum of the processor utilization over the process history.

2. Request additional VMs when the average value of the CPU utilization over the current time slice

exceeds the high threshold.

3. Release a VM when the average value of the CPU utilization over the current time slice falls below

the low threshold.

The conclusions reached based on experiments with threeVMs are as follows: (a) dynamic thresholds

perform better than static ones and (b) two thresholds are better than one. Though conforming to

our intuition, such results have to be justified by experiments in a realistic environment. Moreover,

convincing results cannot be based on empirical values for some of the parameters required by integral

control equations.