In Section 6.2 we saw that we can assimilate a server with a closed-loop control system and we can

apply control theory principles to resource allocation. In this section we discuss a two-level resource allocation architecture based on control theory concepts for the entire cloud. The automatic resource

management is based on two levels of controllers, one for the service provider and one for the application,

see Figure 6.2.

The main components of a control system are the inputs, the control system components, and the

outputs. The inputs in such models are the offered workload and the policies for admission control, the

capacity allocation, the load balancing, the energy optimization, and the QoS guarantees in the cloud.

The system components are sensors used to estimate relevant measures of performance and controllers

that implement various policies; the output is the resource allocations to the individual applications .

The controllers use the feedback provided by sensors to stabilize the system; stability is related to

the change of the output. If the change is too large, the system may become unstable. In our context the

system could experience thrashing, the amount of useful time dedicated to the execution of applications

becomes increasingly small and most of the system resources are occupied by management functions.

There are three main sources of instability in any control system:

1. The delay in getting the system reaction after a control action.

2. The granularity of the control, the fact that a small change enacted by the controllers leads to very

large changes of the output.

3. Oscillations, which occur when the changes of the input are too large and the control is too weak,

such that the changes of the input propagate directly to the output.

Two types of policies are used in autonomic systems: (i) threshold-based policies and (ii) sequential

decision policies based on Markovian decision models. In the first case, upper and lower bounds on

performance trigger adaptation through resource reallocation. Such policies are simple and intuitive but

require setting per-application thresholds.

Lessons learned from the experiments with two levels of controllers and the two types of policies are

discussed in [109]. A first observation is that the actions of the control system should be carried out in a rhythm that does not lead to instability. Adjustments should be carried out only after the performance

of the system has stabilized. The controller should measure the time for an application to stabilize and

adapt to the manner in which the controlled system reacts.

If upper and lower thresholds are set, instability occurs when they are too close to one another if the

variations of the workload are large enough and the time required to adapt does not allow the system

to stabilize. The actions consist of allocation/deallocation of one or more virtual machines; sometimes

allocation/deallocation of a single VM required by one of the thresholds may cause crossing of the other

threshold and this may represent, another source of instability.