Resource management is a core function of any man-made system. It affects the three basic criteria for

the evaluation of a system: performance, functionality, and cost. An inefficient resource management

has a direct negative effect on performance and cost and an indirect effect on the functionality of a

system. Indeed, some functions provided by the system may become too expensive or may be avoided

due to poor performance.

A cloud is a complex system with a very large number of shared resources subject to unpredictable

requests and affected by external events it cannot control. Cloud resource management requires complex

policies and decisions for multi-objective optimization. Cloud resource management is extremely

challenging because of the complexity of the system, which makes it impossible to have accurate global

state information, and because of the unpredictable interactions with the environment.

The strategies for resource management associated with the three cloud delivery models, IaaS, PaaS,

and SaaS, differ from one another. In all cases the cloud service providers are faced with large, fluctuating

loads that challenge the claim of cloud elasticity. In some cases, when a spike can be predicted, the

resources can be provisioned in advance, e.g., for Web services subject to seasonal spikes. For an

unplanned spike, the situation is slightly more complicated. Auto Scaling can be used for unplanned

spike loads, provided that (a) there is a pool of resources that can be released or allocated on demand

and (b) there is a monitoring system that allows a control loop to decide in real time to reallocate

resources. Auto Scaling is supported by PaaS services such as Google App Engine. Auto Scaling for

IaaS, discussed in Section 6.14, is complicated due to the lack of standards.

It has been argued for some time that in a cloud, where changes are frequent and unpredictable,

centralized control is unlikely to provide continuous service and performance guarantees. Indeed, centralized

control cannot provide adequate solutions to the host of cloud management policies that have

to be enforced. Autonomic policies are of great interest due to the scale of the system, the large number

of service requests, the large user population, and the unpredictability of the load. The ratio of the mean

to the peak resource needs can be very large.

We start our discussion with an overview of policies and mechanisms for cloud resource management

in Section 6.1. A control theoretic approach to resource allocation is discussed in Sections 6.2, 6.3, and

6.4. A machine learning algorithm for coordination of specialized autonomic performance managers

is presented in Section 6.5. In Section 6.6 we discuss a utility model for resource allocation for a Web

service. Next we present resource bundling and combinatorial auctions in Section 6.7. The fair queuing,

start-time fair queuing, and borrowed virtual time scheduling algorithms are analyzed in Sections 6.9,

6.10, and 6.11, respectively. Scheduling with deadlines and the impact of application scaling on resource

management are presented in Sections 6.12, 6.13, and 6.14, respectively.