A policy typically refers to the principal guiding decisions, whereas mechanisms represent the means to

implement policies. Separation of policies from mechanisms is a guiding principle in computer science.

Butler Lampson [208] and Per Brinch Hansen [154] offer solid arguments for this separation in the

context of operating system design.

Cloud resource management policies can be loosely grouped into five classes:

1. Admission control.

2. Capacity allocation.

3. Load balancing.

4. Energy optimization.

5. Quality-of-service (QoS) guarantees.

The explicit goal of an admission control policy is to prevent the system from accepting workloads in

violation of high-level system policies; for example, a system may not accept an additional workload that

would prevent it from completing work already in progress or contracted. Limiting the workload requires

some knowledge of the global state of the system. In a dynamic system such knowledge, when available,

is at best obsolete. Capacity allocation means to allocate resources for individual instances; an instance

is an activation of a service. Locating resources subject to multiple global optimization constraints

requires a search of a very large search space when the state of individual systems changes rapidly.

Load balancing and energy optimization can be done locally, but global load-balancing and energy

optimization policies encounter the same difficulties as the one we have already discussed. Load balancing

and energy optimization are correlated and affect the cost of providing the services. Indeed, it

was predicted that by 2012 up to 40% of the budget for IT enterprise infrastructure would be spent on

energy [104].

The common meaning of the term load balancing is that of evenly distributing the load to a set of

servers. For example, consider the case of four identical servers, A, B,C, and D, whose relative loads

are 80%, 60%, 40%, and 20%, respectively, of their capacity. As a result of perfect load balancing,

all servers would end with the same load − 50% of each server’s capacity. In cloud computing a

critical goal is minimizing the cost of providing the service and, in particular, minimizing the energy

consumption. This leads to a different meaning of the term load balancing; instead of having the load

evenly distributed among all servers, we want to concentrate it and use the smallest number of servers

while switching the others to standby mode, a state in which a server uses less energy. In our example,

the load from D will migrate to A and the load from C will migrate to B; thus, A and B will be loaded

at full capacity, whereas C and D will be switched to standby mode. Quality of service is that aspect of

resource management that is probably the most difficult to address and, at the same time, possibly the

most critical to the future of cloud computing.

As we shall see in this section, often resource management strategies jointly target performance

and power consumption. Dynamic voltage and frequency scaling (DVFS)1 techniques such as Intel’s SpeedStep andAMD’s PowerNow lower the voltage and the frequency to decrease power consumption.2

Motivated initially by the need to save power for mobile devices, these techniques have migrated to

virtually all processors, including the ones used for high-performance servers.

As a result of lower voltages and frequencies, the performance of processors decreases, but at a

substantially slower rate [213] than the energy consumption. Table 6.1 shows the dependence of the

normalized performance and the normalized energy consumption of a typical modern processor on

clock rate. As we can see, at 1.8 GHz we save 18% of the energy required for maximum performance,

whereas the performance is only 5%lower than the peak performance, achieved at 2.2 GHz. This seems

a reasonable energy-performance tradeoff!

Virtually all optimal – or near-optimal – mechanisms to address the five classes of policies do not

scale up and typically target a single aspect of resource management, e.g., admission control, but ignore

energy conservation. Many require complex computations that cannot be done effectively in the time

available to respond. The performance models are very complex, analytical solutions are intractable,

and the monitoring systems used to gather state information for these models can be too intrusive

and unable to provide accurate data. Many techniques are concentrated on system performance in

terms of throughput and time in system, but they rarely include energy tradeoffs or QoS guarantees.

Some techniques are based on unrealistic assumptions; for example, capacity allocation is viewed as

an optimization problem, but under the assumption that servers are protected from overload.

Allocation techniques in computer clouds must be based on a disciplined approach rather than ad

hoc methods. The four basic mechanisms for the implementation of resource management policies are:

• Control theory. Control theory uses the feedback to guarantee system stability and predict transient

behavior [185,202], but can be used only to predict local rather than global behavior. Kalman filters

have been used for unrealistically simplified models.

• Machine learning. A major advantage of machine learning techniques is that they do not need a

performance model of the system [353]. This technique could be applied to coordination of several

autonomic system managers, as discussed in [187].

• Utility-based. Utility-based approaches require a performance model and a mechanism to correlate

user-level performance with cost, as discussed in [9].

• Market-oriented/economic mechanisms. Such mechanisms do not require a model of the system,

e.g., combinatorial auctions for bundles of resources discussed in [333].

A distinction should be made between interactive and noninteractive workloads. The management

techniques for interactive workloads, e.g., Web services, involve flow control and dynamic application

placement, whereas those for noninteractive workloads are focused on scheduling. A fair amount of

work reported in the literature is devoted to resource management of interactive workloads, some to

noninteractive, and only a few, e.g., [344], to heterogeneous workloads, a combination of the two.