

PACORA: Performance-Aware Convex Optimization for Resource Allocation

Sarah L. Bird and Burton J. Smith

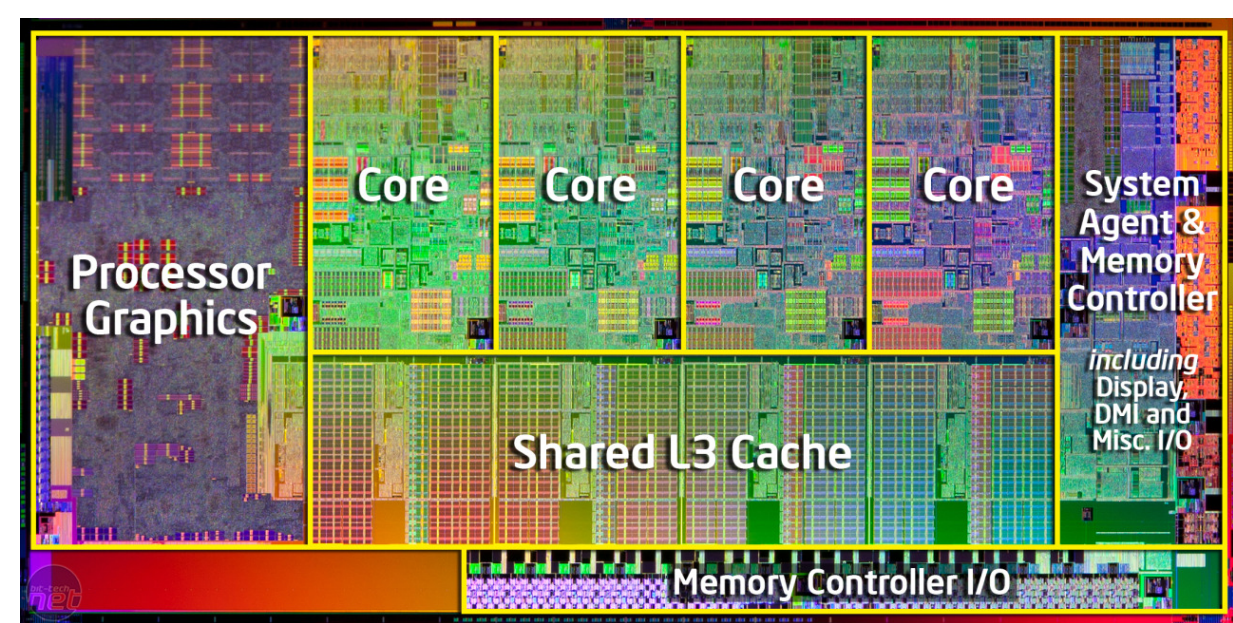
Resource Allocation

Given a set of applications, how many of each resource should the OS give to each application?

Goal: Guarantee Quality-of-Service to applications while maximizing efficiency

i.e., Good user experience maximizing battery life

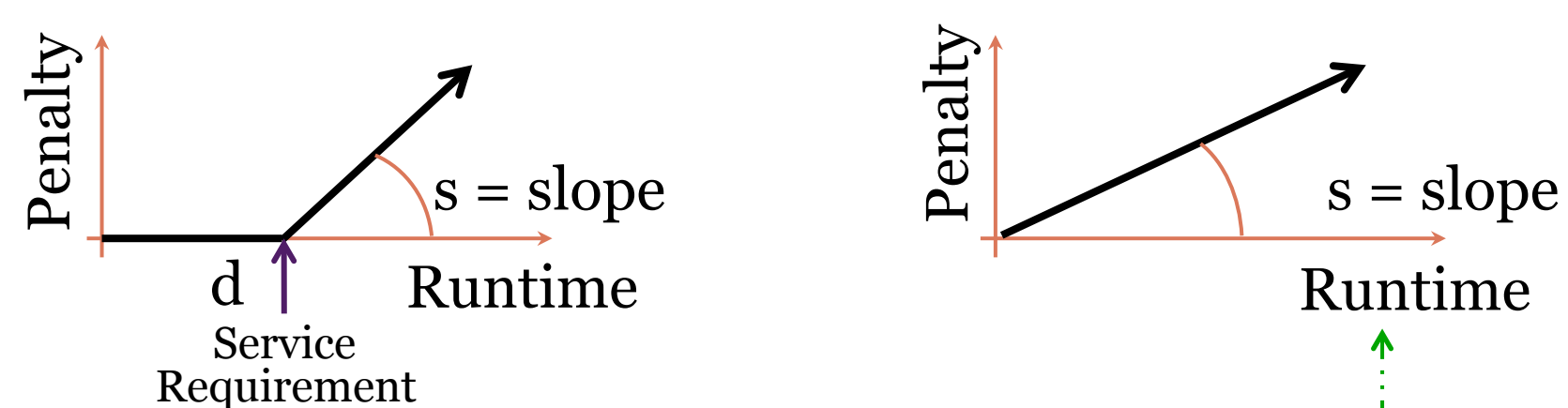
Resources: processing elements, cache slices, memory pages, bandwidth to memory, etc.



Intel Sandy-Bridge die photo showing the growing diversity of manageable resources on modern architectures

Penalty Functions

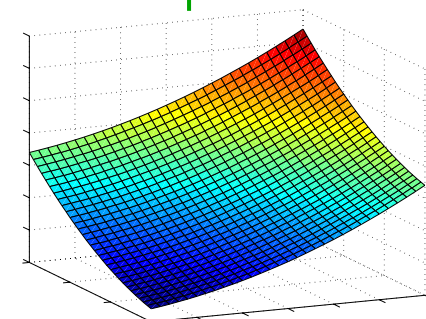
Represent the importance of an app to the system as a function of the app's performance



$$\text{Penalty}(\text{Runtime}) = \text{MAX}(s \cdot (\text{Runtime} - d), 0)$$

Penalty functions are set by the system to represent the relative cost of missing a deadline for an app

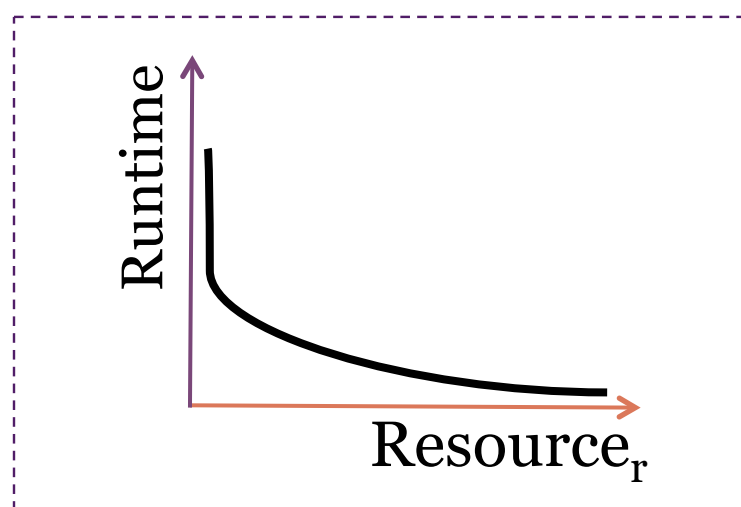
Application deadlines are represented by the service requirement e.g., frame time, responsiveness deadline



PACORA Framework

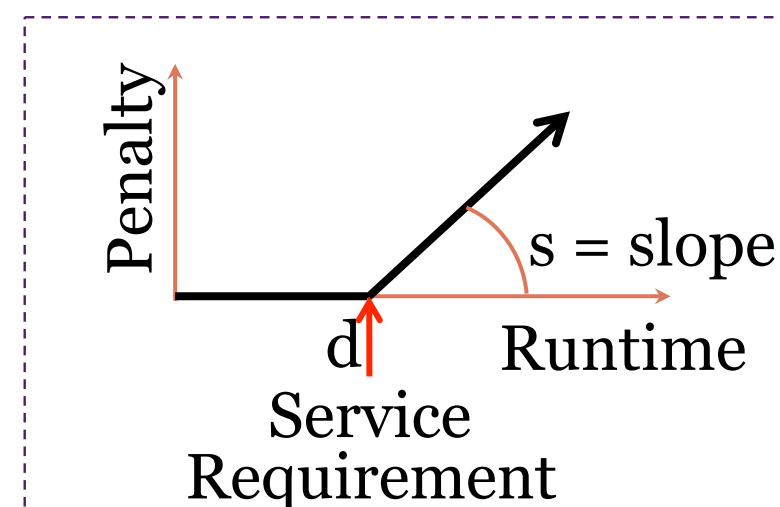
Construction of the resource allocation problem as a convex optimization

Runtime Functions



Value of Resources to Applications

Penalty Functions



Value of Application to the System

Two functions represent each application penalty function, runtime function

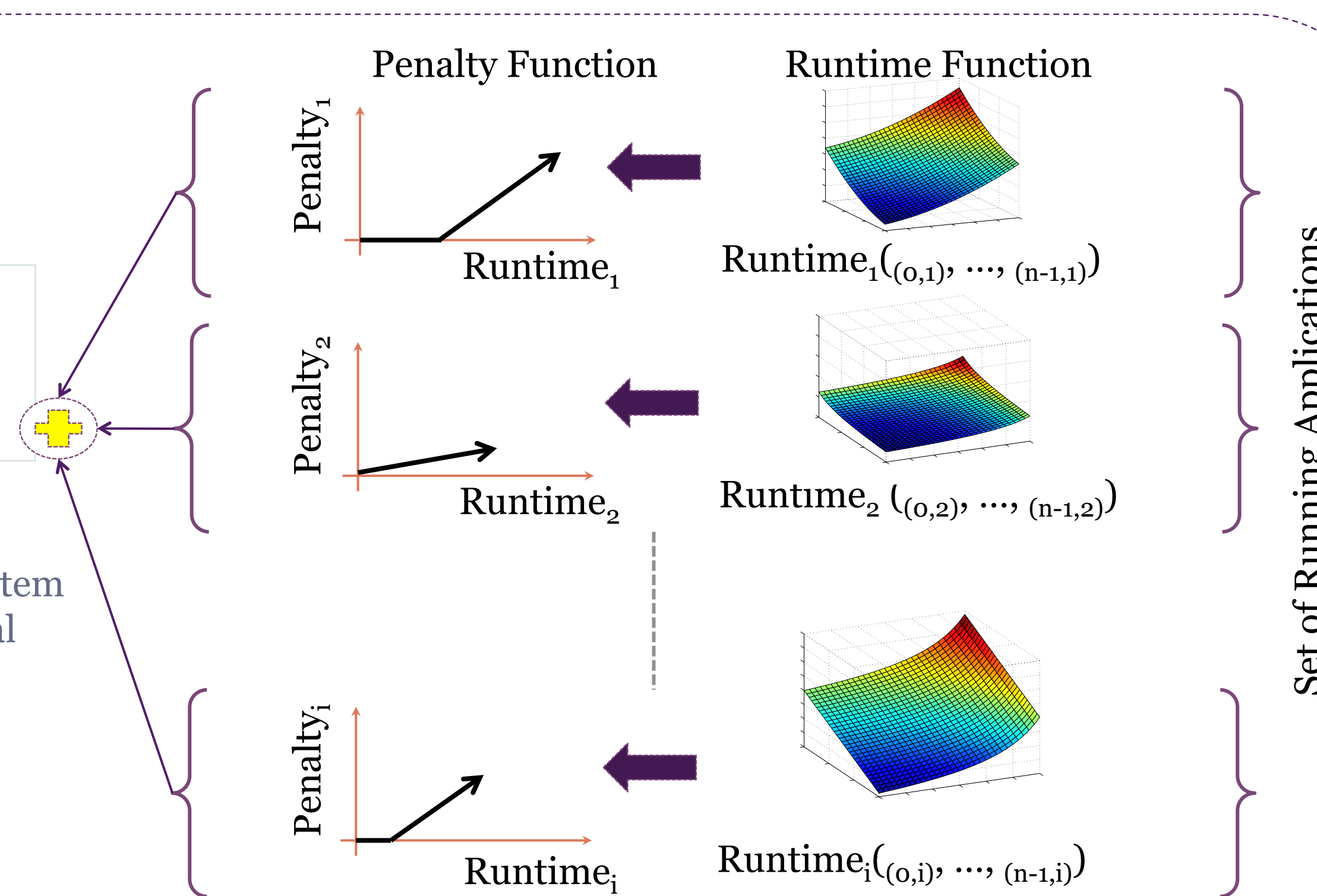
A user-level scheduler is responsible for managing threads on the resources

The convex optimization tries to minimize the total penalty of the system

PACORA Convex Construction

Continuously minimize using the penalty of the system (subject to restrictions on the total amount of resources)

Partially solving the convex optimization still enables to system to move towards a more optimal allocation

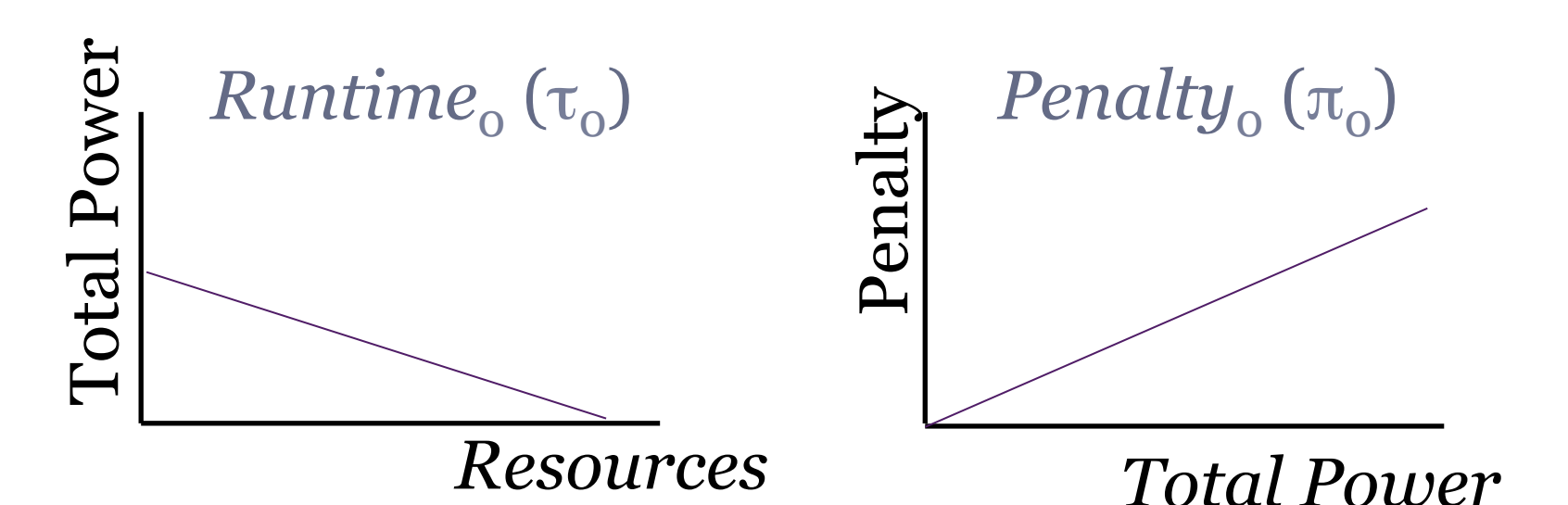


Managing Power and Energy

Application 0 can be used to represent the idle resources in the system Assume all idle resources are powered off

Energy can be managed using $\text{Penalty}_0(\pi_0)$ and $\text{Runtime}_0(\tau_0)$

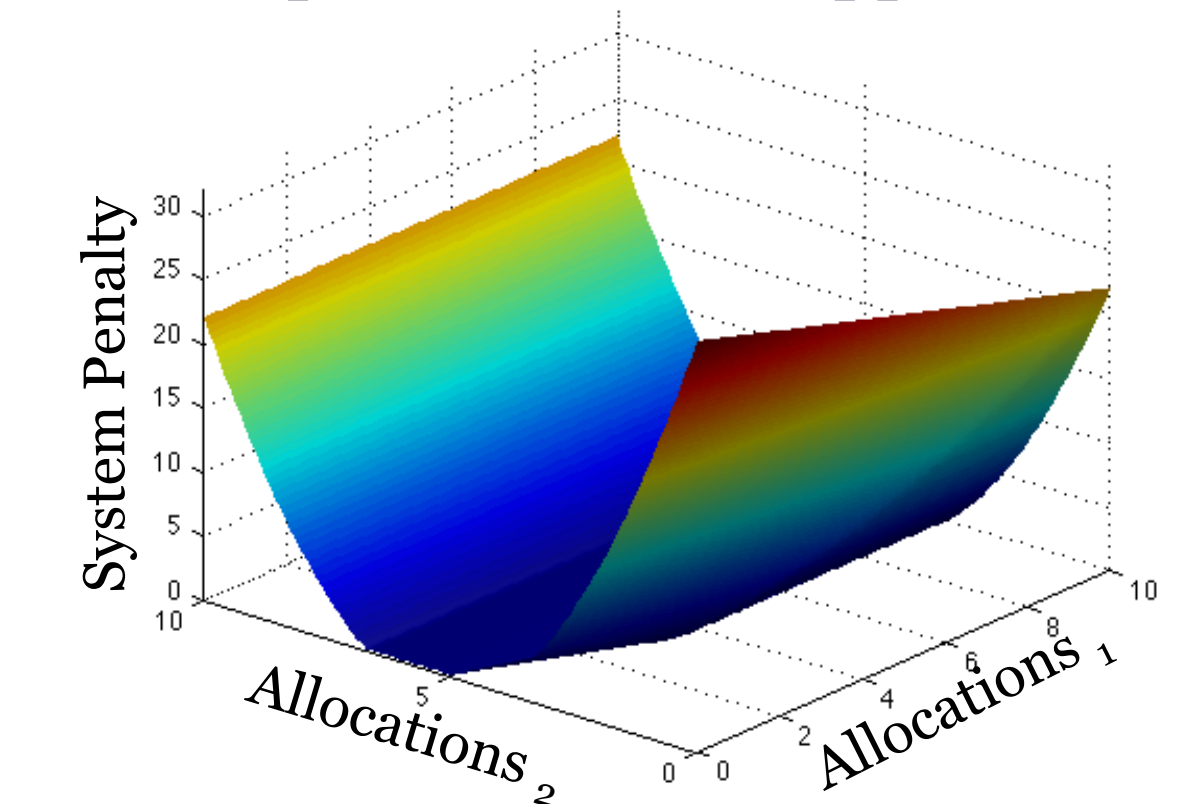
τ_0 is defined to be the total system power π_0 has a slope that depends on the battery charge



As battery depletes the OS may choose to increase the slope of π_0 to reflect the increased value of saving energy

Advantages

Enables the OS to efficiently use resources and provide QoS to applications



- Convex optimization is **relatively inexpensive** optimization problem with a **single extreme point**
- Fast, incremental** solutions are feasible
- Penalty Function Slopes allow the system to express **relative priorities** of application
- Priorities change as a function of performance (criticality)
- Penalty Function Intercept **encapsulates QoS requirements (responsiveness)**
- And additional process can be used to **represent system energy**

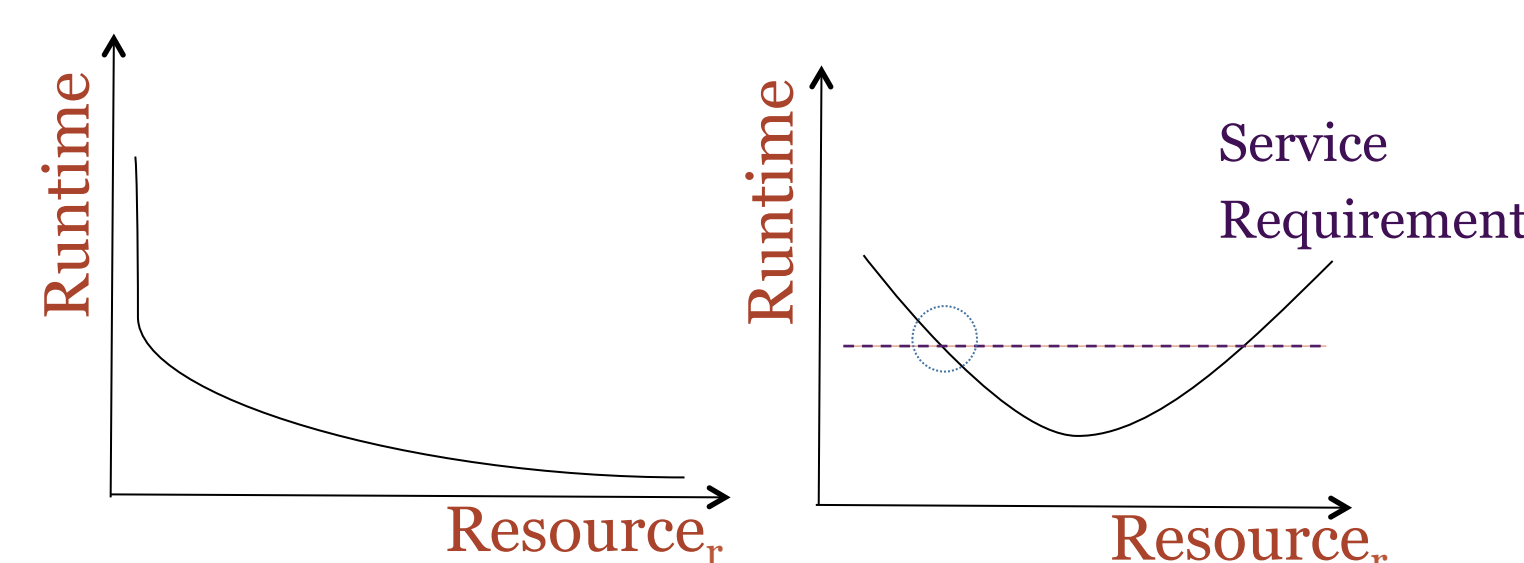
Runtime Functions

Convex approximation of the measured performance functions

Performance Model of the Applications

$\text{Runtime}_{(0,p), (1,p), \dots, (n-1,p)}$

(i,p) : Allocation of resource of type 1 to App P



Runtime functions represent responsiveness e.g., time from a mouse click to its result

App Runtime Model

$$\tau(w, b, \alpha, m) = \sqrt[p]{\sum_j \left(\frac{w_j}{b_j \cdot a_j(m_j)} \right)^p}$$

w quantities of work (learned)
 b allocations of bandwidth resources
 a bandwidth amplification functions (learned)
 m allocations of cache or memory resources