Performance evaluation and optimization of computing systems :

Resource Allocation with Time-varying Capacities

Urtzi AYESTA
CNRS-IRIT (and LAAS)
https://www.irit.fr/~Urtzi.Ayesta

STORE workshop

23/02/2017

Outline

- Toy examples:
  Models are useful to obtain insight and design protocols

- Optimality of index policies (priorities)

- And otherwise?

- Future/ongoing research

Scheduling example

K tasks need to be processed on one machine

- Task ***k*** has processing time t_k
- Holding cost c_k is incurred per unit of time (until task ***k*** is completed)

In which order to schedule the tasks?

If scheduled in order 1,2,..., *K*, then total cost is

$$c_1 t_1 + c_2(t_1 + t_2) + \cdots + c_K(t_1 + \cdots + t_K)$$

$$V(S) = \max_{i \in S} \left(\sum_{j \in S} c_j t_i + V(S - \{i\}) \right)$$

However, there is a simpler way...

Consider the order

$$i_1, \ldots, i_k, i, j, i_{k+3}, \ldots, i_K$$

and consider interchanging tasks i and j

$$i_1, \ldots, i_k, j, i, i_{k+3}, \ldots, i_K$$

Costs under the two schedules are respectively,

$$C_1 + c_i (T + t_i) + c_j (T + t_i + t_j) + C_2$$

$$C_1 + c_j (T + t_j) + c_i (T + t_j + t_i) + C_2$$

Simple algebra, the first schedule is better if and only if

$$c_i / t_i > c_j / t_j$$

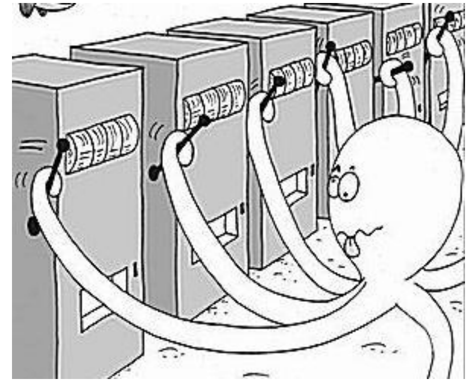
Conclusion

Total cost is minimized by the **index rule** that schedules the tasks in order of highest index c_i / t_i

→**Preference to short tasks!**

→Optimal policy is of **index type!**

When index policy is optimal!



Given a **population** of *K* "bandits"

$N_1(t+1) = n_1$	$N_2(t+1) = n_2$	$N_3(t+1) = n_3 + r_3$	$N_4(t+1) = n_4$	$N_5(t+1) = n_5$

A **bandit** is a **controllable stochastic process**

- **Active bandit:** state of bandit evolves
- **Passive bandit:** state is frozen

Dynamic control π: activates **one** bandit at each time

Reward of active bandit depends on its state → $\max_{\pi} \mathbb{E}\left(\int_{t=0}^{\infty} \sum_{k=1}^{K} \beta^t R_k(N_k^{\pi}(t))\mathrm{d}t\right)$

Gittins index policy maximizes discounted reward [Gittins, 1979]

→The bandit with highest index value $G_k(j)$ is served, where

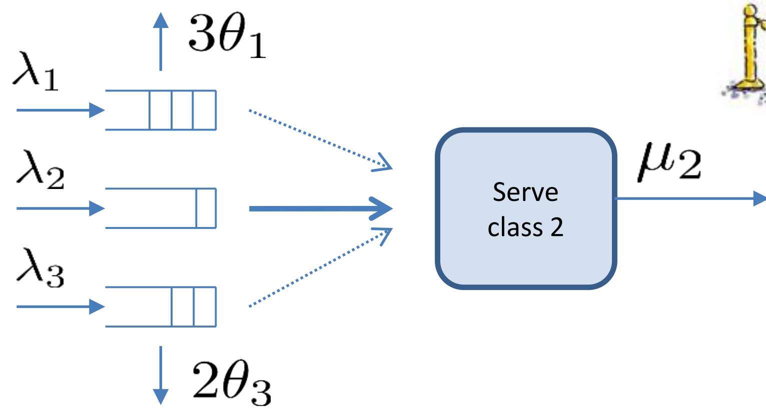
Index type optimality if

- Rested property

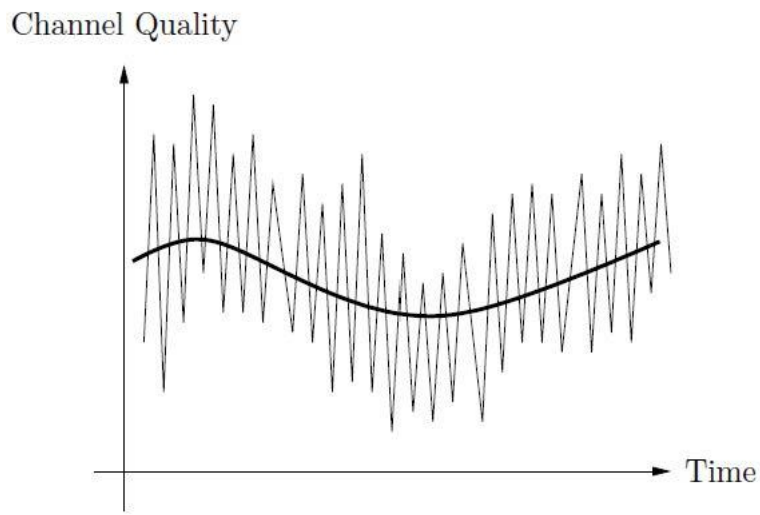
- Infinite horizon:
  Average optimality, total discounted cost etc.

- One server

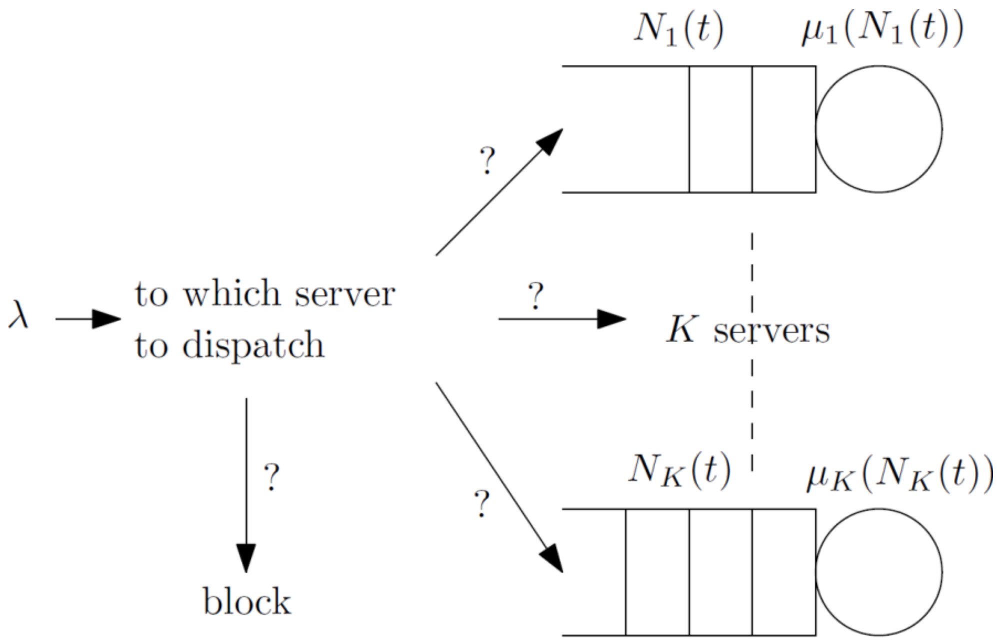
- So what if customers are impatient?

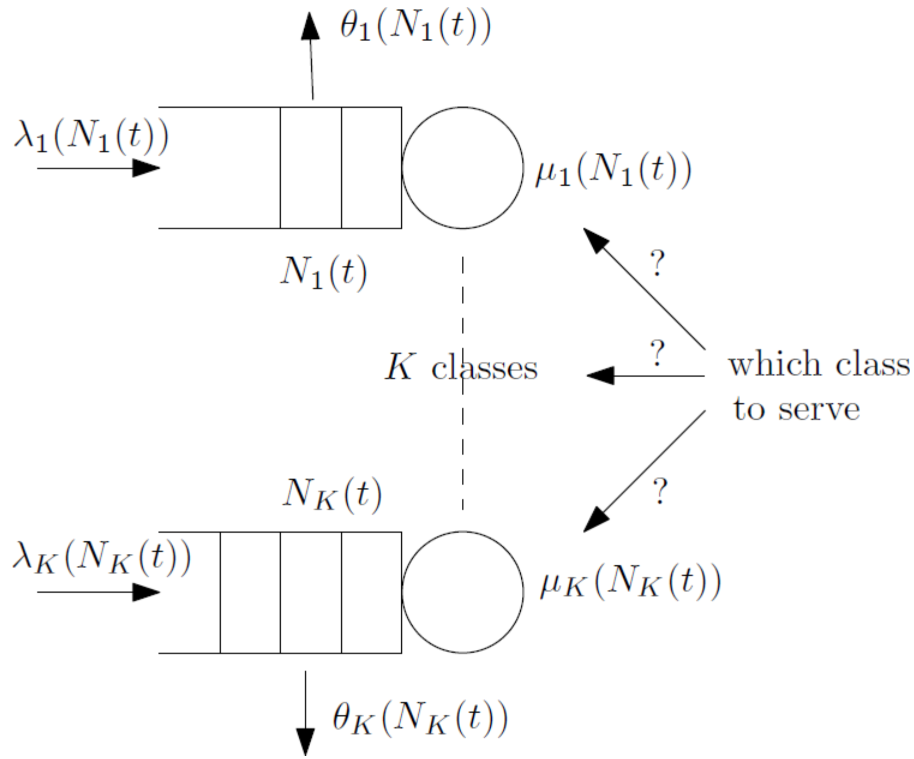


- In wireless downlink channel?

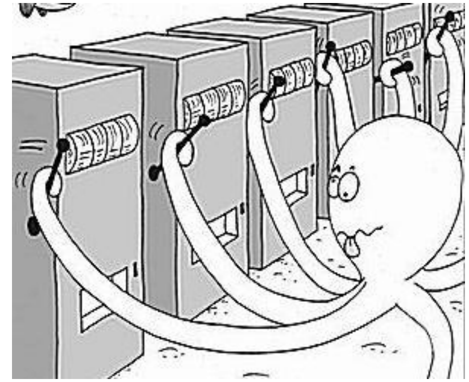


8


$N_1(t)$
$\mu_1(N_1(t))$
?
λ
to which server
to dispatch
?
K servers
?
?
$N_K(t)$
$\mu_K(N_K(t))$
block


$\theta_1(N_1(t))$
$\lambda_1(N_1(t))$
$\mu_1(N_1(t))$
$N_1(t)$
?
K classes
?
which class
to serve
?
$N_K(t)$
$\lambda_K(N_K(t))$
$\mu_K(N_K(t))$
$\theta_K(N_K(t))$

Restless bandit problem



Given a **population** of K bandits

$N_1(t+1) = n_1+r_1$	$N_2(t+1) = n_2+r_2$	$N_3(t+1) = n_3+r_3$	$N_4(t+1) = n_4+r_4$	$N_5(t+1) = n_5+r_5$

$\alpha=2$

A **bandit** is a **controllable stochastic process**

- **Active bandit:** state of bandit evolves
- **Passive bandit**: state of bandit evolves (different law)

Can **activate** $\boldsymbol{\alpha}$ **bandits** at a time

Cost/reward depends on state of bandit

Dynamic optimization problem: Which $\boldsymbol{\alpha}$ **bandits to make active** in order to **minimize discounted or time-average cost**?

Objective: Determine policy that maximizes

$$\max_{\pi} \sum_{t=0}^{\infty} \beta^t \sum_{k=1}^{K} \mathbb{E}(R_{k,X_k(t)}^{a_k(t)}), \qquad \text{subject to} \qquad \sum_{k=1}^{K} a_k^{\pi}(t) = 1$$

Relax the constraint so that it is satisfied in the long-run

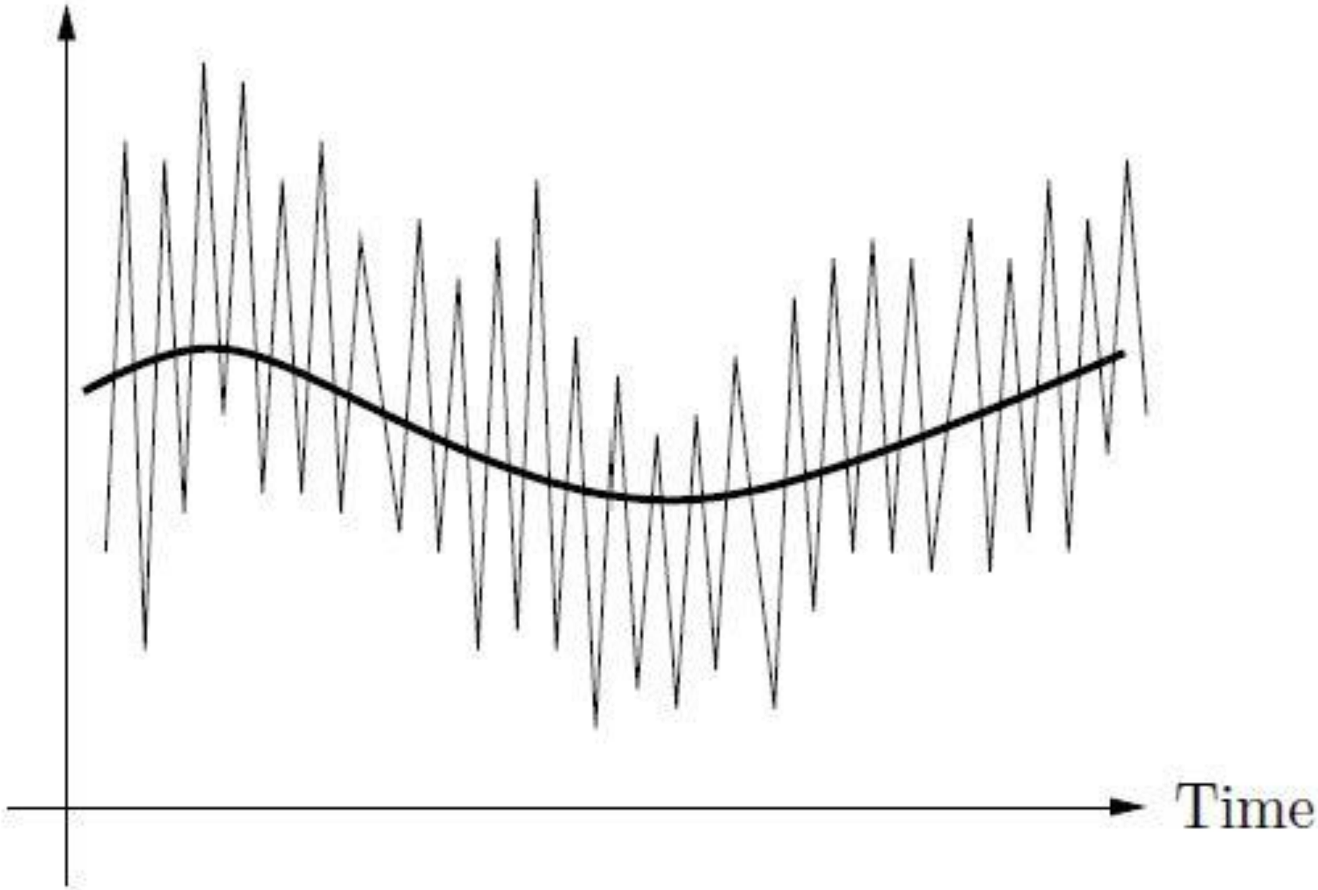
$$\sum_{t=0}^{\infty} \beta^t \sum_{k=1}^{K} a_k(t) = \frac{1}{1-\beta}.$$

multi dimensional problem ➔ multiple unidimensional problem

$$\max_{\pi_k} \sum_{t=0}^{\infty} \beta^t \mathbb{E}(R_{k,X_k(t)}^{a_k(t)} - \nu a_k(t)).$$

Optimal solution to relaxed problem is of index type!

Scheduling in a wireless link

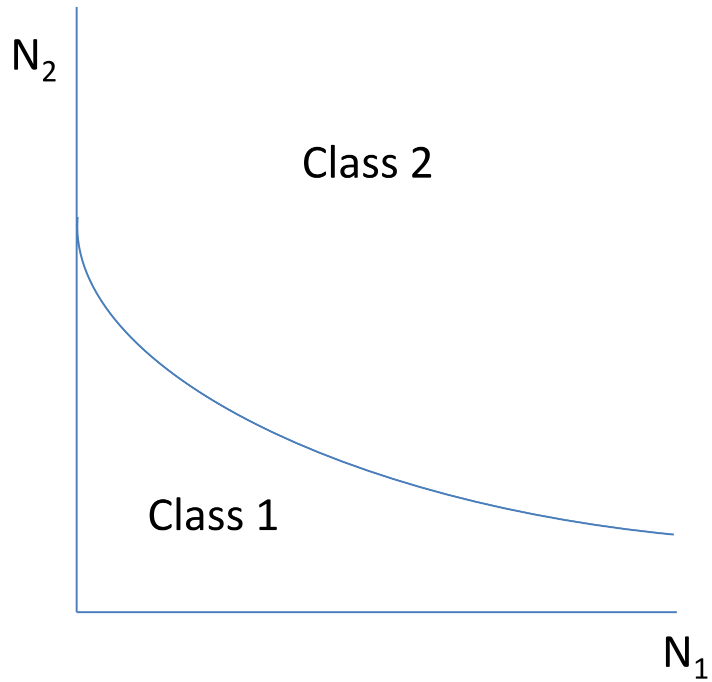


The optimal solution for relaxed is to serve all jobs for which

$$\frac{\text{current rate}}{\text{margin of improvement}} > v$$

- Allows us to build a heuristic for original problem
- First explanation why “opportunistic scheduling” works well
- Using fluid approach to establish **maximum stability** condition and **asymptotic fluid** optimality.

Abandonments



Optimal in high load

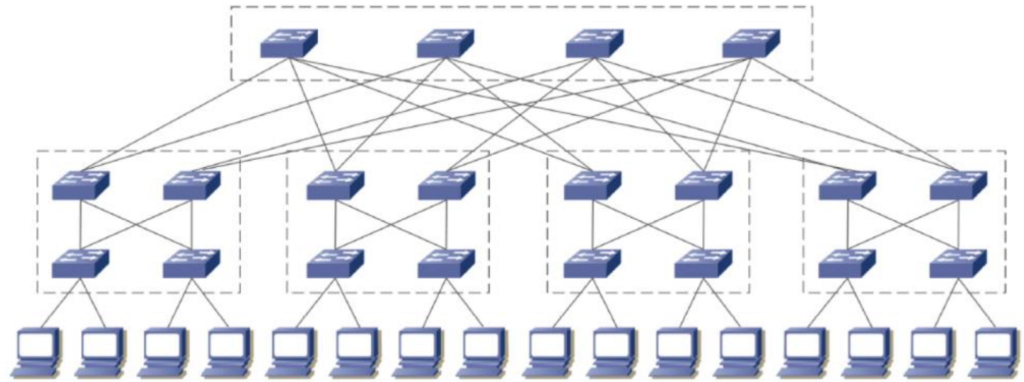
$$\lim_{\lambda \to \infty} \frac{C^{WI} - C^{OPT}}{C^{OPT}} = 0$$

And ongoing ideas…

DATACENTER

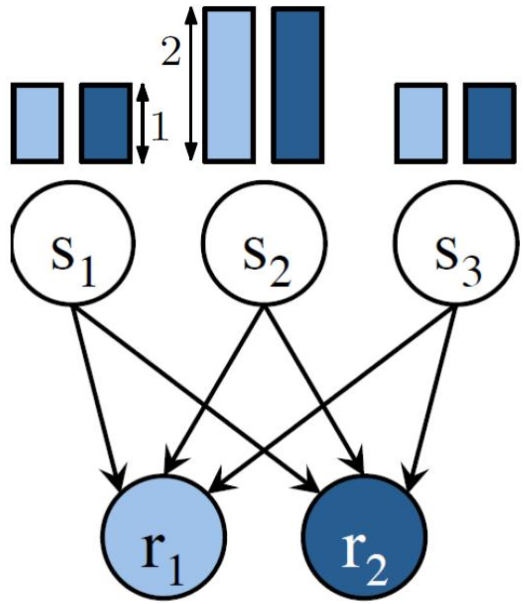
Rethinking data centers: new paradigms for resource allocation

Definition: pool of resources interconnected using a communication network

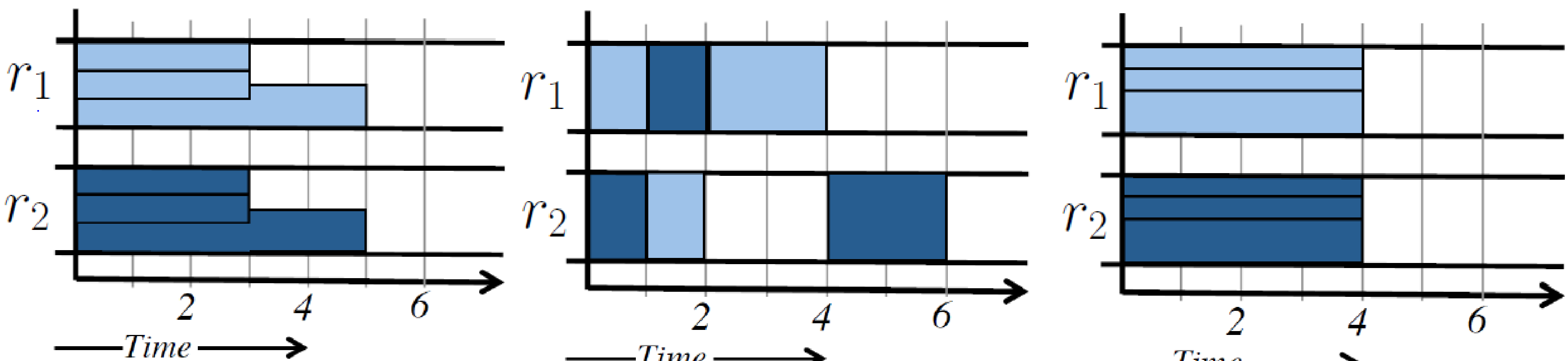


Main infrastructure for Internet applications, enterprise operations, scientific computations

Very difficult to engineer → overprovisioning (10% utilization of network and servers)



3 × 2 shuffle



(a) Per-flow fairness (TCP)

(b) Optimal flow sharing

(c) Optimal coflow sharing

Performance evaluation and optimization of computing systems :
Resource Allocation with Time-varying Capacities

Urtzi AYESTA
CNRS-IRIT (and LAAS)
https://www.irit.fr/~Urtzi.Ayesta

STORE workshop

23/02/2017