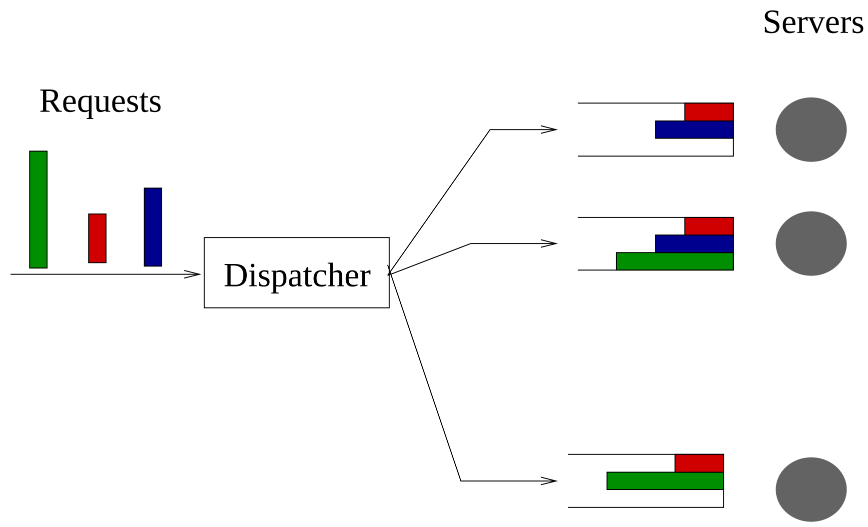
Load Balancing in Processor Sharing Systems

Eitan Altman (INRIA)
Urtzi Ayesta (LAAS-CNRS)
and Balakrishna Prabhu (LAAS-CNRS)

GameComm, 20 October 2008

Server farms

- Diverse applications : e-service industry, database systems, grid computing clusters

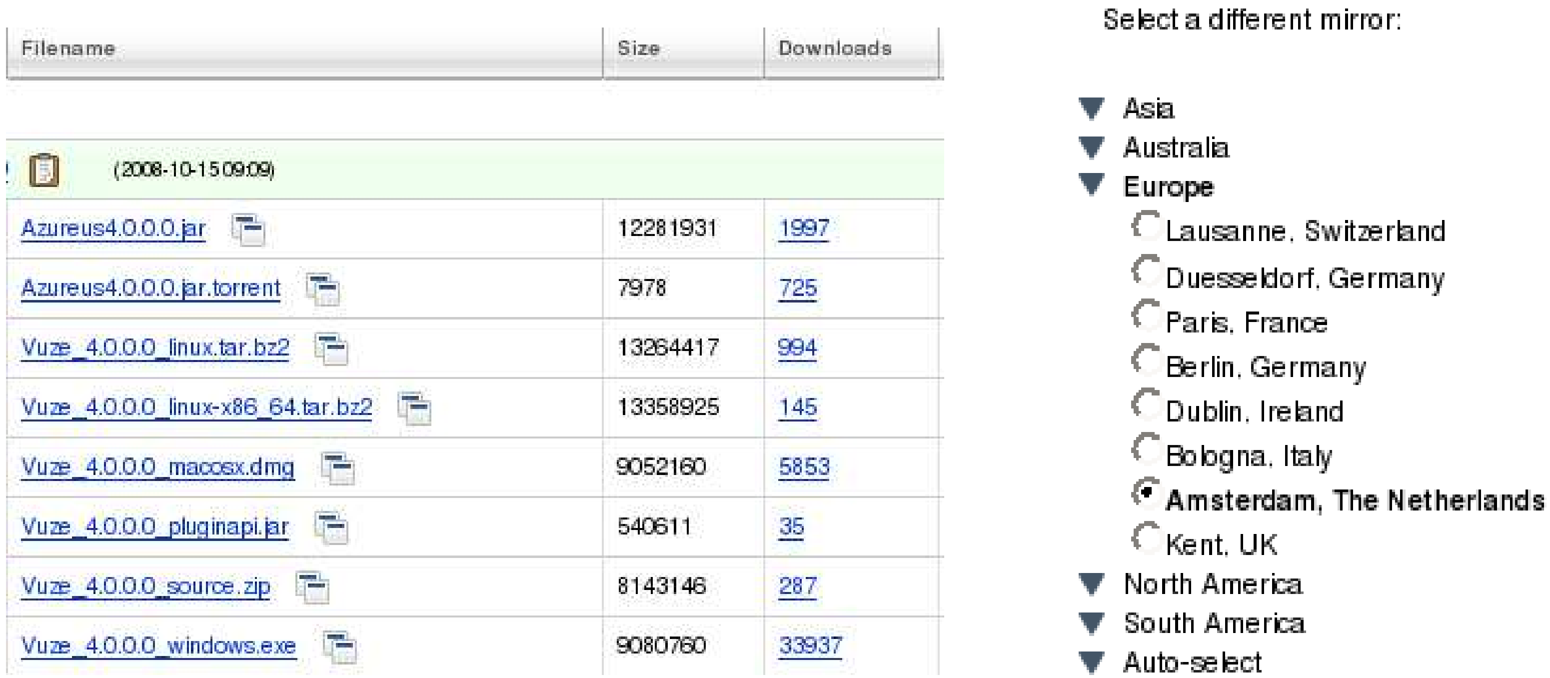


Design problem: What is the optimal routing policy?

- Centralized setting: dispatcher takes decisions
- Decentralized setting: requests take decisions

Example application

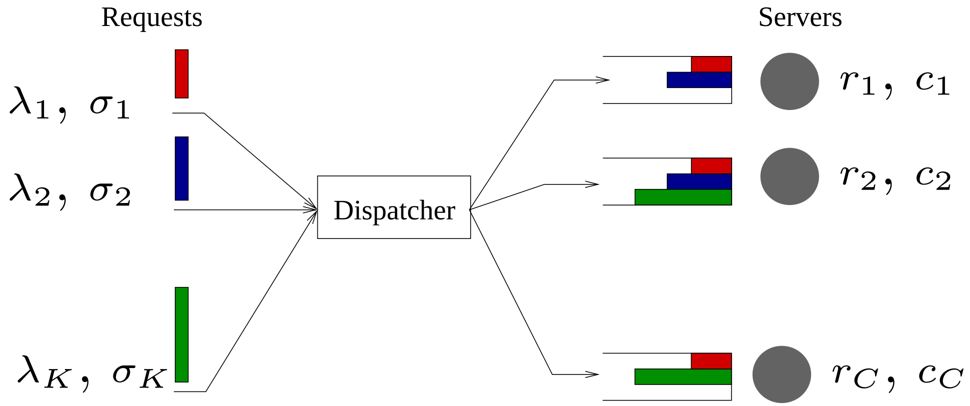
Internet based source code repositories - SourceForge, Google Code: Source files are hosted on several mirror sites



- Decision is taken either by the central unit or by the downloader
- Downloads progress in parallel $\Rightarrow$ Processor Sharing (PS) at each server

Problem Description

- Server farms with K job classes , C servers, PS discipline



- **Objective:**
  - Centralized setting: minimize the weighted mean sojourn time
  - Decentralized setting: each user seeks to minimize its own weighted mean sojourn time

- **Decision variable** : $\mathbf{p} = (p_{ij})$ - probability that class i job is routed to server j.

Outline

- Centralized setting
- Decentralized setting
- Comparing the Centralized and Decentralized solution
  $\longrightarrow$ Price of Anarchy
- Conclusions and future work

Mean sojourn times in M/G/1/PS queues

- By Little's law mean the sojourn time is proportional to the mean number of jobs
- Let $\eta_i = \lambda_i \sigma_i^{-1}$ be the traffic offered by class i
- Let $\overline{\eta} = \sum_i \eta_i$ be the total offered traffic
- The load on server j is

$$\rho_j = \frac{\sum_i \eta_i p_{ij}}{r_j}$$

- The mean number of jobs in server j is

$$\mathrm{E}[N_j] = \frac{\rho_j}{1 - \rho_j}$$

$\Rightarrow$ $\mathrm{E}[N_j]$ is **insensitive** to the **second moment** of the service time distribution

Centralized setting : problem formulation

- Solve the following mathematical program :

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j\in\mathcal{S}} c_j \frac{\rho_j(\mathbf{p})}{1-\rho_j(\mathbf{p})} \\
\text{subject to} \quad & \sum_{j\in\mathcal{S}} p_{ij} = 1, \quad \text{for all } i \in \mathcal{K}; \\
& \mathbf{p} \succeq \mathbf{0}; \\
& \sum_{i\in\mathcal{K}} \eta_i p_{ij} < r_j, \quad \text{for all } j \in \mathcal{S}.
\end{aligned}
$$

- Solution need not be unique
- If $\rho_j(\mathbf{p}^1) = \rho_j(\mathbf{p}^2), \ \forall j \in \mathcal{S}$, then either both $\mathbf{p}^1$ and $\mathbf{p}^2$ are optimal or both are suboptimal

Stability and Size-unaware multi-strategy

Proposition. There exists a stable multi-strategy if and only if $\sum_{j \in \mathcal{S}} r_j > \overline{\eta}$.

Stability and Size-unaware multi-strategy

Proposition. There exists a stable multi-strategy if and only if $\sum_{j\in\mathcal{S}} r_j > \overline{\eta}$.

Proposition: Let $\mathbf{p}$ be a feasible multi-strategy. For all $i \in \mathcal{K}$ and for all $j \in \mathcal{S}$ define the multi-strategy $\hat{\mathbf{p}}$ by

$$\hat{p}_{ij} = \frac{\rho_j(\mathbf{p}) r_j}{\overline{\eta}}.$$

The multi-strategy $\hat{\mathbf{p}}$ is also feasible and $\rho_j(\hat{\mathbf{p}}) = \rho_j(\mathbf{p}) \ \ \forall j \in \mathcal{S}$.

Stability and Size-unaware multi-strategy

Proposition. There exists a stable multi-strategy if and only if $\sum_{j \in \mathcal{S}} r_j > \overline{\eta}$.

Proposition: Let $\mathbf{p}$ be a feasible multi-strategy. For all $i \in \mathcal{K}$ and for all $j \in \mathcal{S}$ define the multi-strategy $\hat{\mathbf{p}}$ by

$$\hat{p}_{ij} = \frac{\rho_j(\mathbf{p}) r_j}{\overline{\eta}}.$$

The multi-strategy $\hat{\mathbf{p}}$ is also feasible and $\rho_j(\hat{\mathbf{p}}) = \rho_j(\mathbf{p}) \ \ \forall j \in \mathcal{S}$.

Corollary: If $\mathbf{p}$ is an optimal multi-strategy then $\hat{\mathbf{p}}$ is a **size-unaware** optimal multi-strategy

Implication: Reduces the dimensionality of the program, we can optimize directly over $\boldsymbol{\rho}$.

Centralized setting : Reduced Mathematical Program

- Solve the following convex mathematical program :

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in \mathcal{S}} c_j \frac{\rho_j}{1-\rho_j} \\
\text{subject to} \quad & 0 < \rho_j < 1, \quad \text{for all } j \in \mathcal{S}; \\
& \sum_{i \in \mathcal{K}} r_j \rho_j = \overline{\eta}.
\end{aligned}
$$

- Assume servers are indexed such that $\frac{c_1}{r_1} \leq \frac{c_2}{r_2} \leq ... \leq \frac{c_C}{r_C}$.
- c/r is the cost per unit workload

Centralized setting: solution structure

Theorem. The subset of servers that are used in the optimal load balancing is $\mathcal{S}_G = \{1, \ldots, j^*\}$, where $j^* = \sup\left\{ j \leq C : \sum_{k=1}^{j} \sqrt{c_j r_j} > \left(\sum_{k=1}^{j} r_k - \overline{\eta} \right) \sqrt{\frac{c_j}{r_j}} \right\}$. Under the optimal multi-strategy, the load on server $j \in \mathcal{S}_G$ is

$$\rho_j^* = 1 - \sqrt{\frac{c_j}{r_j}} \frac{\sum_{k \in \mathcal{S}_G} r_k - \overline{\eta}}{\sum_{k \in \mathcal{S}_G} \sqrt{c_k r_k}}.$$

Centralized setting: solution structure

Theorem. The subset of servers that are used in the optimal load balancing is $\mathcal{S}_G = \{1, \ldots, j^*\}$, where $j^* = \sup\left\{ j \leq C : \sum_{k=1}^{j} \sqrt{c_j r_j} > \left(\sum_{k=1}^{j} r_k - \overline{\eta}\right) \sqrt{\frac{c_j}{r_j}} \right\}$. Under the optimal multi-strategy, the load on server $j \in \mathcal{S}_G$ is

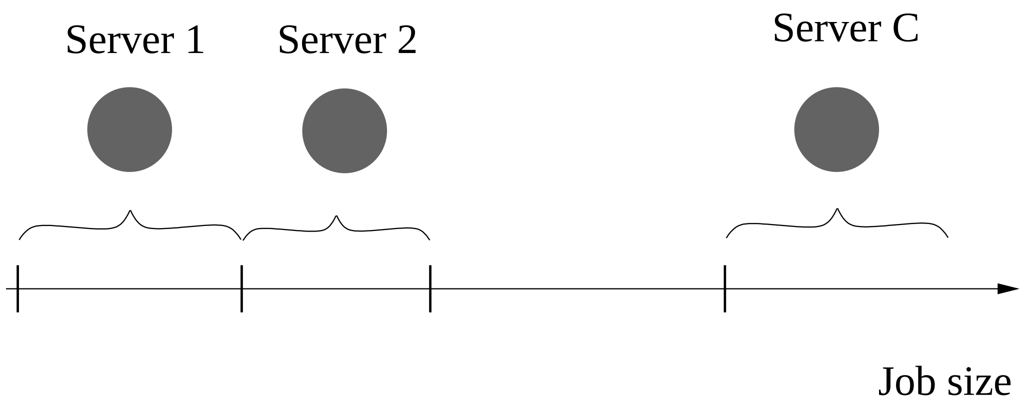
$$\rho_j^* = 1 - \sqrt{\frac{c_j}{r_j}} \frac{\sum_{k \in \mathcal{S}_G} r_k - \overline{\eta}}{\sum_{k \in \mathcal{S}_G} \sqrt{c_k r_k}}.$$

Corollary The size-unaware multi-strategy, $\hat{\mathbf{p}}^*$, is given by $\hat{p}_{ij}^* = \frac{\rho_j^* r_j}{\overline{\eta}}$, for all $i \in \mathcal{K}$ and for all $j \in \mathcal{S}$.

Remarks: The solution structure is known as water-filling and server with a larger c/r ratio receives lesser traffic.

Related results

- FCFS as back-end scheduling [Feng *et al.*, 2005]:



  - **intuition:** reduce the variability of service-time distribution

- For PS [Starobinski and Wu 2005, Haviv and Roughgarden 2007]
  - Homogenous cost rates and one type of requests.

- We allow for multi-class requests and heterogenous cost rates

Decentralized setting

Equilibrium: A strategy $\mathbf{p}$ is **an equilibrium** for the individual selfish setting if for each class $i = 1, ..., K$, each server $j = 1, ..., C$ and each queue k used by class i,

$$E[c_k T_k(\mathbf{p})|i] = \min_{j=1,\ldots,K} E[c_j T_j(\mathbf{p})|i]$$

Proposition. The distributed non-cooperative game can be transformed into the standard convex optimization problem

$$\min_{\mathbf{p}} \sum_{k=1}^{C} c_k \log\left(\frac{1}{1-\rho_k(\mathbf{p})}\right)$$

$\Rightarrow$ The game belongs to a particular type of games known as "Potential Game".

Characterizing the Individual Optimal Solution

Theorem. The subset of servers that are used in the optimal routing strategy in the non-cooperative setting is of type $\mathcal{S}_I = \{1, \ldots, j^*\}$, where

$$j^* = \sup\left\{ j \le C : \sum_{k=1}^{j} c_j > \left(\sum_{k=1}^{j} r_k - \overline{\eta} \right) \frac{c_j}{r_j} \right\}$$

For every $j \in \mathcal{S}_I$, the load is

$$\rho_j = 1 - \frac{c_j}{r_j} \frac{\sum_{k \in \mathcal{S}_I} r_k - \overline{\eta}}{\sum_{k \in \mathcal{S}_I} c_k}.$$

Characterizing the Individual Optimal Solution

Theorem. The subset of servers that are used in the optimal routing strategy in the non-cooperative setting is of type $\mathcal{S}_I = \{1, \ldots, j^*\}$, where

$$j^* = \sup\left\{ j \leq C : \sum_{k=1}^{j} c_j > \left(\sum_{k=1}^{j} r_k - \overline{\eta} \right) \frac{c_j}{r_j} \right\}$$

For every $j \in \mathcal{S}_I$, the load is

$$\rho_j = 1 - \frac{c_j}{r_j} \frac{\sum_{k \in \mathcal{S}_I} r_k - \overline{\eta}}{\sum_{k \in \mathcal{S}_I} c_k}.$$

Water-filling structure: As the arrival rate λ increases, server 2 will start being used when:

$$\frac{c_1}{r_1} \times \frac{1}{1 - \rho_1(\mathbf{p})} = \frac{c_2}{r_2}.$$

Comparing the Global and Individual

Proposition. For any arrival rate and service time distribution it holds

$$\mathcal{S}_I \subseteq \mathcal{S}_G$$

Price of Anarchy: is defined as the ratio between the performance (mean delay) obtained by the Wardrop equilibrium and the global optimal solution.

Theorem. For every θ, there exist c_j and r_j, $j \in \mathcal{S}$, such that $PoA > \theta$.

$\Rightarrow$ The PoA is unbounded.

When $c_k = 1$, then $PoA \leq C$ [Haviv and Roughgarden, 2007].

Sketch of proof: $PoA = \frac{\sum_{j=1}^{C} c_j E[N_j^I]}{\min_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j^G]}$

Assume $c_j = r_j = 1$, for $j = 2, \ldots, C$.

We take $r_1 \downarrow \overline{\eta}$ and $c_1 \to 0$.

Sketch of proof: $PoA = \frac{\sum_{j=1}^{C} c_j E[N_j^I]}{\min_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j^G]}$

Assume $c_j = r_j = 1$, for $j = 2, \ldots, C$.

We take $r_1 \downarrow \overline{\eta}$ and $c_1 \to 0$.

Individuals:

- Only one server is used.
- As $r_1 \downarrow \overline{\eta}$, $E[N_1^I] \to \infty$, but $c_1 \to 0$, and overall $c_1 E[N_1^I] \to \overline{\eta}/2$.

Sketch of proof: $PoA = \dfrac{\sum_{j=1}^{C} c_j E[N_j^I]}{\min_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j^G]}$

Assume $c_j = r_j = 1$, for $j = 2, \dots, C$.

We take $r_1 \downarrow \overline{\eta}$ and $c_1 \to 0$.

Individuals:

- Only one server is used.
- As $r_1 \downarrow \overline{\eta}$, $E[N_1^I] \to \infty$, but $c_1 \to 0$, and overall $c_1 E[N_1^I] \to \overline{\eta}/2$.

Global optimal:

- All servers are used.
- As $r_1 \downarrow \overline{\eta}$ the global optimal tends to route everything towards server 1, thus $\sum_{j=2}^{C} c_j E[N_j^G] \to 0$.
- $\rho_1 = 1 - o(\sqrt{r_1 - \overline{\eta}})$ and it turns out that $c_1 E[N_1^G] \to 0$.
- Thus for the global optimum, as $r_1 \downarrow \overline{\eta}$, $\sum_{j=1}^{C} c_j E[N_j^G] \to 0$.

Conclusions and Future work

- Centralized setting
  - Existence of a **size-unaware** optimal routing policy
  - Characterize set of useful servers and the optimal load on each

Conclusions and Future work

- Centralized setting
  - Existence of a **size-unaware** optimal routing policy
  - Characterize set of useful servers and the optimal load on each
- Decentralized setting
  - Potential game
  - Characterize set of useful servers and the optimal load on each

Conclusions and Future work

- Centralized setting
  - Existence of a **size-unaware** optimal routing policy
  - Characterize set of useful servers and the optimal load on each
- Decentralized setting
  - Potential game
  - Characterize set of useful servers and the optimal load on each
- Compare the two settings
  - Decentralized solution uses more servers than centralized solution
  - Price of Anarchy is unbounded.

Conclusions and Future work

- Centralized setting
  - Existence of a **size-unaware** optimal routing policy
  - Characterize set of useful servers and the optimal load on each
- Decentralized setting
  - Potential game
  - Characterize set of useful servers and the optimal load on each
- Compare the two settings
  - Decentralized solution uses more servers than centralized solution
  - Price of Anarchy is unbounded.
- Future work
  - Alternative back-end scheduling disciplines: SRPT, LAS etc.
  - Non-atomic selfish setting: Each class chooses a routing strategy to minimize its own total weighted delay.