# On the Efficiency of Non-Cooperative Load Balancing

**J. Doncel** (jdoncel@laas.fr)
U. Ayesta, O. Brun, B. Prabhu

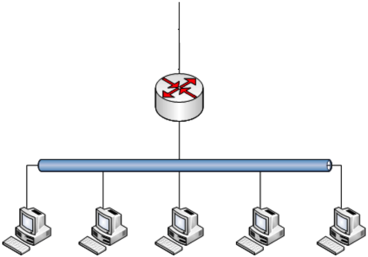
LAAS-CNRS

Networking Conference
New York, USA
23rd May 2013

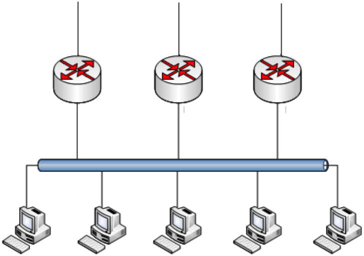
# Outline

# Introduction

Routing problem in server farms



(a) Centralized architecture.



(b) Non-cooperative decentralized architecture.

Decentralized architecture based on autonomous, selfish agents: each one minimizes the sojourn time of its jobs

# Introduction

## Comparison of both settings:

Problem addressed using the Price of Anarchy (PoA)

$$PoA = \frac{\text{decentralized setting worst performance}}{\text{optimal performance}} \geq 1$$

# Introduction

## Comparison of both settings:

Problem addressed using the Price of Anarchy (PoA)

$$PoA = \frac{\text{decentralized setting worst performance}}{\text{optimal performance}} \geq 1$$

## From previous results

Selfish routing can be inefficient

- [Ayesta, Brun, Prabhu]: $PoA \leq \sqrt{K}$ (sqrt of num dispatchers)
- [Haviv, Roughgarden]: $PoA \leq S$ (num servers)

Heavy-traffic is always the most inefficient situation

# Introduction

## Comparison of both settings:

Problem addressed using the Price of Anarchy (PoA)

$$PoA = \frac{\text{decentralized setting worst performance}}{\text{optimal performance}} \geq 1$$

## From previous results

Selfish routing can be inefficient

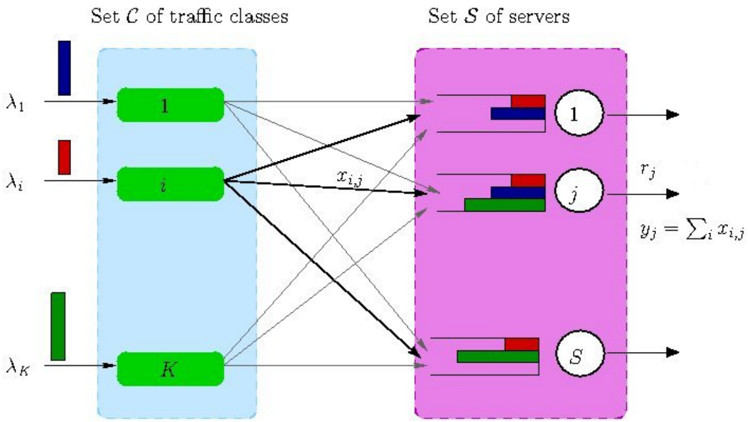
- [Ayesta, Brun, Prabhu]: $PoA \leq \sqrt{K}$ (sqrt of num dispatchers)
- [Haviv, Roughgarden]: $PoA \leq S$ (num servers)

Heavy-traffic is always the most inefficient situation

## We show that

- Selfish routing is almost always efficient
- The worst case traffic condition is not the heavy-traffic

# Model Description



# Optimization Problem

## For each dispatcher $i$

$$\underset{\mathbf{x}_i}{\text{minimize}}\ T_i(\mathbf{x}) = \sum_{j \in \mathcal{S}} \frac{x_{ij}}{r_j - y_j}$$

$$\text{s. t.} \sum_{j \in \mathcal{S}} x_{ij} = \lambda_i,\ \ i = 1, \ldots, K$$

$$\text{and} \quad 0 \leq x_{ij} \leq r_j,\ \forall j \in \mathcal{S}$$

# Optimization Problem

## For each dispatcher $i$

$$\underset{\mathbf{x}_i}{\text{minimize}} \; T_i(\mathbf{x}) = \sum_{j \in \mathcal{S}} \frac{x_{ij}}{r_j - y_j}$$

$$\text{s. t.} \; \sum_{j \in \mathcal{S}} x_{ij} = \lambda_i, \; i = 1, \ldots, K$$

$$\text{and} \quad 0 \leq x_{ij} \leq r_j, \; \forall j \in \mathcal{S}$$

## Decentralized setting: Nash Equilibrium

No dispatcher has incentive to change the strategy

# Performance and comparison

## Performance

Performance of the decentralized setting:

$$D_K(\boldsymbol{\lambda}, \mathbf{r}) = \sum_{i \in \mathcal{C}} T_i(\mathbf{x}) = \sum_{j \in \mathcal{S}} \frac{y_j}{r_j - y_j},$$

where $\mathbf{x}$ is the NEP.

Centralized architecture: $\lambda_1 = \bar{\lambda} \Rightarrow D_1(\bar{\lambda}, \mathbf{r})$

Measuring:

$$\frac{D_K(\boldsymbol{\lambda}, \mathbf{r})}{D_1(\bar{\lambda}, \mathbf{r})} \geq 1$$

# Some Definitions

## Inefficiency

For a fixed data-center architecture ($S$ and capacities)

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda} < \bar{r},\ \boldsymbol{\lambda} \in \Lambda(\bar{\lambda})} \frac{D_K(\boldsymbol{\lambda}, \mathbf{r})}{D_1(\bar{\lambda}, \mathbf{r})},$$

where $\bar{r} = \sum_{j \in \mathcal{S}} r_j$.

# Some Definitions

## Inefficiency

For a fixed data-center architecture ($S$ and capacities)

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda}<\bar{r},\ \boldsymbol{\lambda}\in\Lambda(\bar{\lambda})} \frac{D_K(\boldsymbol{\lambda},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})},$$

where $\bar{r} = \sum_{j\in\mathcal{S}} r_j$.

## Price of Anarchy

$$PoA\ (K,S) = \sup_{\mathbf{r}} I_K^S(\mathbf{r})$$

# Worst Case Traffic Conditions

**Previous Result [Ayesta et al]**

The worst case occurs when each player routes exactly the same amount of traffic.

**Corollary**

*We focus on the total amount of incoming traffic*

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda}<\bar{r},\ \boldsymbol{\lambda}\in\Lambda(\bar{\lambda})} \frac{D_K(\boldsymbol{\lambda},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})} = \sup_{\bar{\lambda}<\bar{r}} \frac{D_K(\frac{\bar{\lambda}}{K}e,\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$$

*where $e$ is the all-ones vector.*

# Example

Server farm of $S = 800$ servers with 4 different values

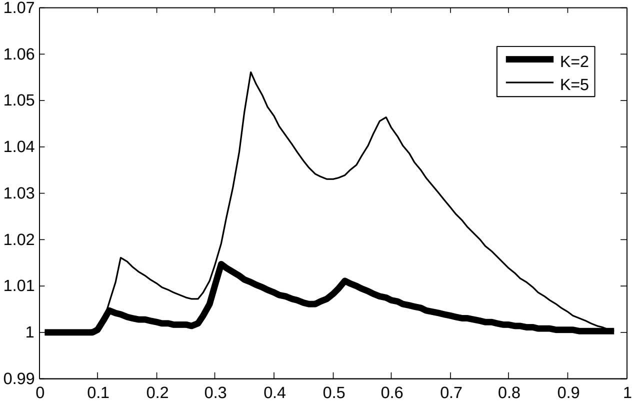



Figure: Evolution of $\frac{D_K(\frac{\bar{\lambda}}{K}e,\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ over the load of the system (K=2 and K=5)

# Inefficiency is not in HT

## Proposition

*If the total traffic intensity $\bar{\lambda}$ is such that the centralized and the decentralized setting use the same number of servers (more than one), then the ratio of the social costs $D_K(\frac{\bar{\lambda}}{K}e, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ is decreasing with $\bar{\lambda}$.*

## Corollary

For a sufficient high load all the servers will be used by both settings, then heavy-traffic regime is not the worst case

# Inefficiency is not in HT

## Proposition

*If the total traffic intensity $\bar{\lambda}$ is such that the centralized and the decentralized setting use the same number of servers (more than one), then the ratio of the social costs $D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ is decreasing with $\bar{\lambda}$.*

## Corollary

For a sufficient high load all the servers will be used by both settings, then heavy-traffic regime is not the worst case

## Theorem

For a fixed $K < \infty$,

$$\lim_{\bar{\lambda} \to \bar{r}} \frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r})}{D_1(\bar{\lambda}, \mathbf{r})} = 1.$$

# 2 classes of servers

## Server farm with two classes of servers

$S_1$ servers of capacity $r_1$
$S_2$ servers of capacity $r_2$, where $r_1 > r_2$

## Definition

Let $\bar{\lambda}^{OPT}$ be a threshold value of the total incoming traffic such that

- if $\bar{\lambda} \leq \bar{\lambda}^{OPT}$ the centralized setting uses only the "fast" servers,
- if $\bar{\lambda} > \bar{\lambda}^{OPT}$ all servers are used by the centralized setting.

Let $\bar{\lambda}^{NE}$ be a threshold value of the total incoming traffic such that

- if $\bar{\lambda} \leq \bar{\lambda}^{NE}$ the decentralized setting uses only the "fast" servers,
- if $\bar{\lambda} > \bar{\lambda}^{NE}$ all servers are used by the decentralized setting.

# 2 classes of servers

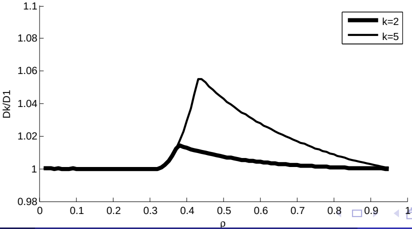
## Proposition

*$\bar{\lambda}^{OPT} < \bar{\lambda}^{NE}$ and the ratio $D_K(\frac{\bar{\lambda}}{K}e, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ is*

- *equal to 1 for $0 \leq \bar{\lambda} \leq \bar{\lambda}^{OPT}$*
- *strictly increasing over the interval $(\bar{\lambda}^{OPT}, \bar{\lambda}^{NE})$*
- *and strictly decreasing over the interval $(\bar{\lambda}^{NE}, \bar{r})$*

## Theorem

*Inefficiency is achieved when $\bar{\lambda} = \bar{\lambda}^{NE}$*



# For a given architecture

Let $\alpha = \frac{S_1}{S_2}$ and $\beta = \frac{r_1}{r_2} > 1$ parameters of a data-center

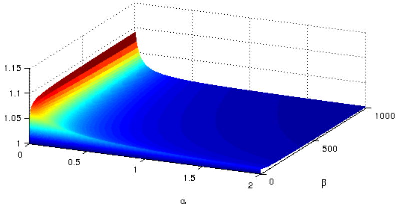
$I_K^S(\mathbf{r})$ does not depend on $S$ and only on $\alpha$ and $\beta$

$\Rightarrow$ Notation: $I_K(\alpha, \beta)$

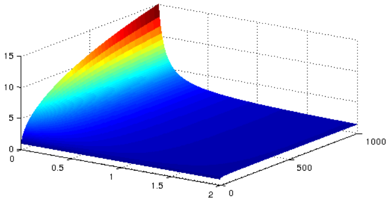
Evaluating the ratio $\frac{D_K(\frac{\bar{\lambda}}{K}e,\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ in $\bar{\lambda} = \bar{\lambda}^{NE}$

$$I_K(\alpha,\beta) = \frac{1}{2}\frac{\sqrt{(K-1)^2+4K\beta}-(K+1)}{\frac{(\frac{1}{\alpha}+\sqrt{\beta})^2}{\frac{1}{\alpha}+\frac{2\beta}{\sqrt{(K-1)^2+4K\beta}-(K-1)}}-(\frac{1}{\alpha}+1)}$$

# For a given architecture




(a) K=2




(b) K=1000

## Conclusion

The worst inefficiency occurs when the slower servers are infinitely more numerous and infinitely slower than the faster ones.

# 2 classes: PoA

## Proposition

$$PoA(K,S) = \sup_{\alpha,\beta} I_K(\alpha,\beta) = \sup_{\beta} I_K(\frac{1}{S-1},\beta) = lim_{\beta\to\infty} I_K(\frac{1}{S-1},\beta)$$

## Proposition

*For a server farm with two server classes and K dispatchers*

$$PoA(K,S) \leq min(\frac{K}{2\sqrt{K}-1}, S)$$

## Conclusion

PoA high when $K$ and $S$ large, but inefficiency is low!!

# Conclusions

Arbitrary architecture:

- Inefficiency is not in heavy-traffic
- Obtained at low loads

Two classes of servers:

- Characterize the traffic conditions for inefficiency
- A refined upper-bound on the PoA
- Non-cooperative load-balancing is almost always efficient
- Give the parameters of a data-center to achieve the PoA

# Thank you

Thank you for your attention.