Performance of Balanced Fairness in Resource Pools: A Recursive Approach

Céline Comte
Joint work with Thomas Bonald and Fabien Mathieu

LINCS Seminar
January 24, 2018

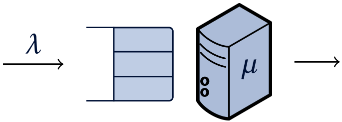
Our objective

Understanding the impact of complex server interactions in large-scale resource pools with parallel processing

www.rambus.com/data-center/

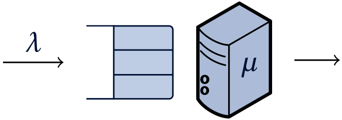
Processor-sharing

- “Service policy where the customers, clients or jobs are all served simultaneously, each receiving an equal fraction of the service capacity available” (Wikipedia)

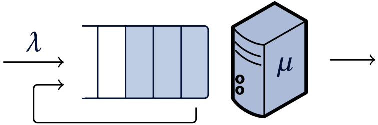


Processor-sharing

- "Service policy where the customers, clients or jobs are all served simultaneously, each receiving an equal fraction of the service capacity available" (Wikipedia)

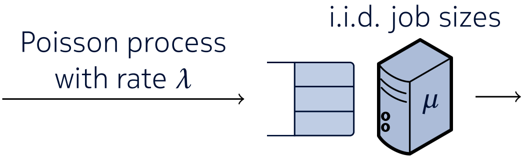


- "Emerged as an idealisation of round-robin scheduling algorithms" (Aalto et al., 2007)



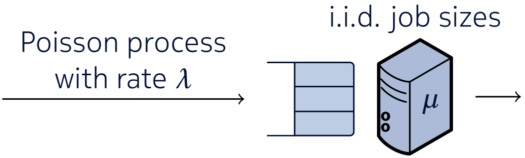
Insensitivity

- Performance only depends on the load $\rho = \frac{\lambda}{\mu}$



Insensitivity

- Performance only depends on the load $\rho = \frac{\lambda}{\mu}$

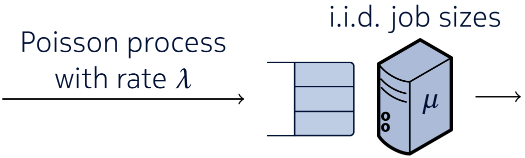


- Probability that the system is empty

$$\psi = 1 - \rho$$

Insensitivity

- Performance only depends on the load $\rho = \frac{\lambda}{\mu}$



- Probability that the system is empty

$$\psi = 1 - \rho$$

- Mean number of jobs in the system

$$L = \frac{\rho}{1-\rho}$$

NOKIA Bell Labs
TELECOM ParisTech

Outline

Resource allocation

New formula for performance prediction

Applications

- Gain of differentiation
- Impact of locality

Outline

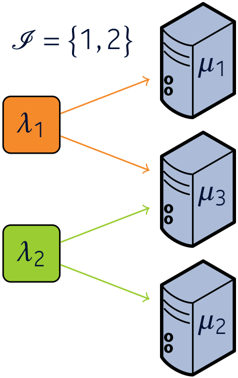
Resource allocation

New formula for performance prediction

Applications

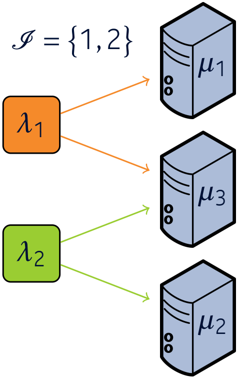
- Gain of differentiation
- Impact of locality

Resource pool with parallel processing



  Public NOKIA Bell Labs TELECOM ParisTech

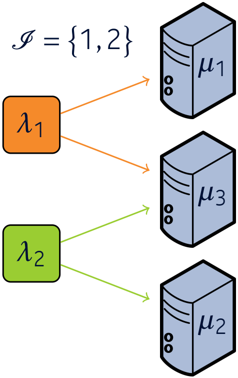
Resource pool with parallel processing



Arrivals

- Class i has traffic intensity λ_i
- Poisson arrival process
- i.i.d. job sizes

Resource pool with parallel processing



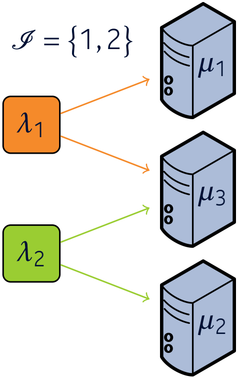
Arrivals

- Class i has traffic intensity λ_i
- Poisson arrival process
- i.i.d. job sizes

Service

- Server k has capacity μ_k
- Parallel processing

Resource pool with parallel processing



Arrivals

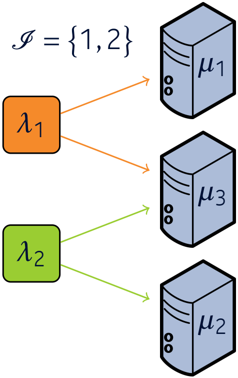
- Class i has traffic intensity λ_i
- Poisson arrival process
- i.i.d. job sizes

Service

- Server k has capacity μ_k
- Parallel processing

State $x = (x_i : i \in \mathscr{I}) \in \mathbb{N}^{\mathscr{I}}$

Resource pool with parallel processing



Arrivals

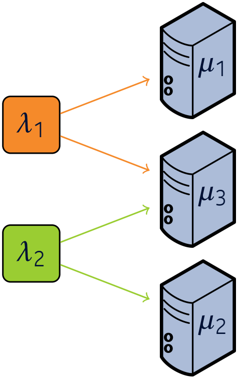
- Class i has traffic intensity λ_i
- Poisson arrival process
- i.i.d. job sizes

Service

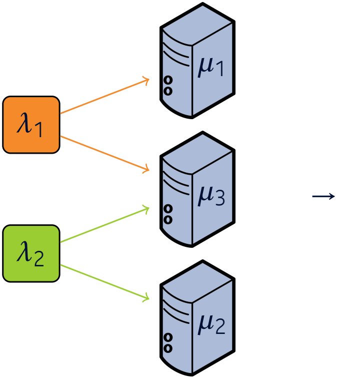
- Server k has capacity μ_k
- Parallel processing

State $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{N}^{\mathscr{I}}$

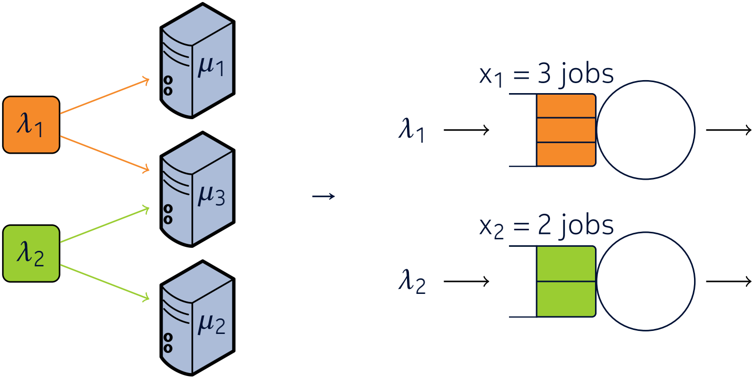
Network of processor-sharing queues



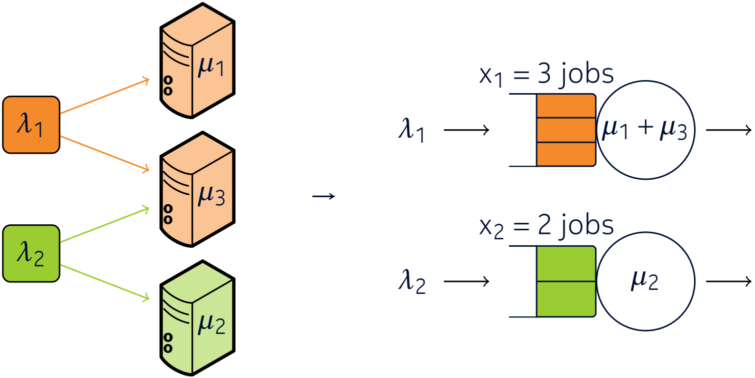
Network of processor-sharing queues



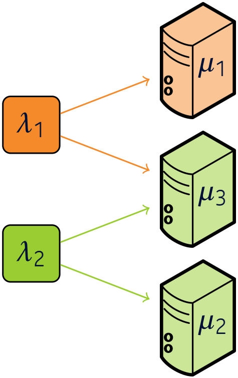
Network of processor-sharing queues

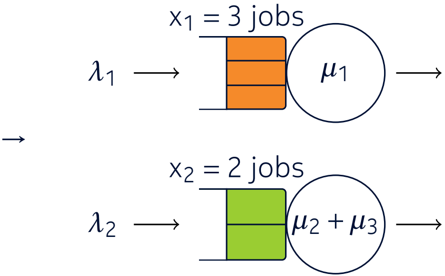


Network of processor-sharing queues

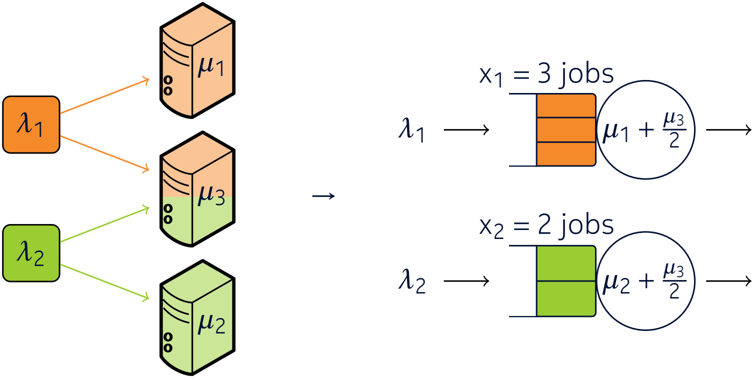


Network of processor-sharing queues

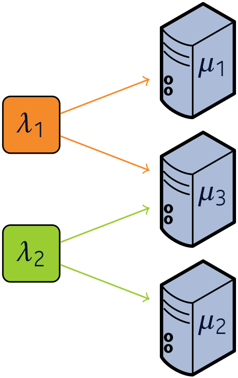




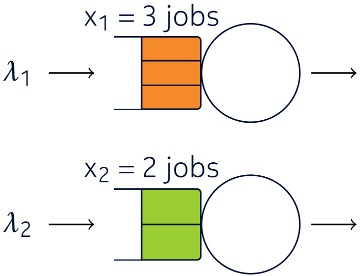
Network of processor-sharing queues



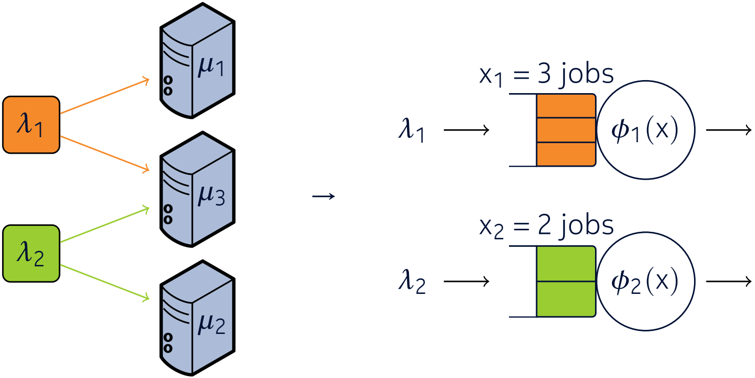
Network of processor-sharing queues



→

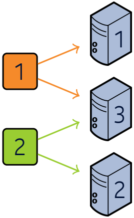


Network of processor-sharing queues



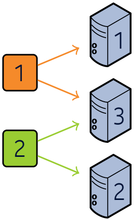
Define $\phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix}$ for each $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Balanced fairness

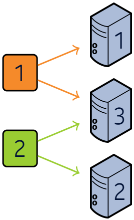


Balanced fairness

The most efficient insensitive resource allocation



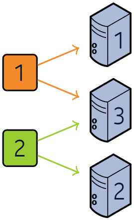
Balanced fairness



The most efficient insensitive resource allocation

- Introduced for dimensioning data networks (Bonald and Proutière, 2003)

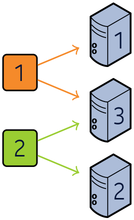
Balanced fairness



The most efficient insensitive resource allocation

- Introduced for dimensioning data networks (Bonald and Proutière, 2003)
- Good approximation of proportional fairness

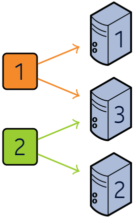
Balanced fairness



The most efficient insensitive resource allocation

- Introduced for dimensioning data networks (Bonald and Proutière, 2003)
- Good approximation of proportional fairness
- Recently applied to server pools

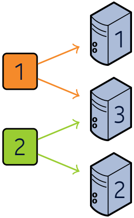
Balanced fairness



The most efficient insensitive resource allocation

- Introduced for dimensioning data networks (Bonald and Proutière, 2003)
- Good approximation of proportional fairness
- Recently applied to server pools
  - Content-delivery networks (Shah and de Veciana, 2015)

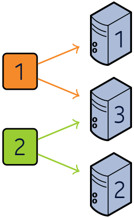
Balanced fairness



The most efficient insensitive resource allocation

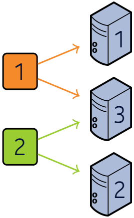
- Introduced for dimensioning data networks (Bonald and Proutière, 2003)
- Good approximation of proportional fairness
- Recently applied to server pools
  - Content-delivery networks (Shah and de Veciana, 2015)
  - Computer clusters (Bonald and Comte, 2017)

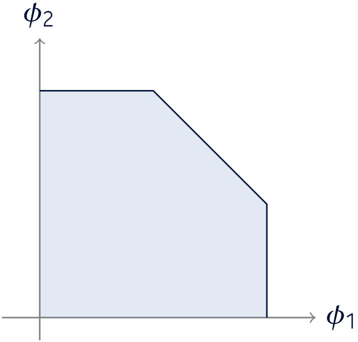
Balanced fairness



  Public NOKIA Bell Labs TELECOM ParisTech

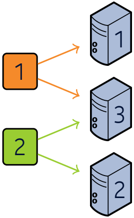
Balanced fairness

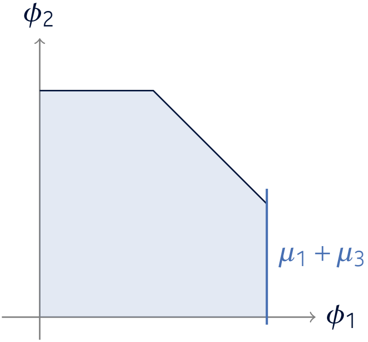




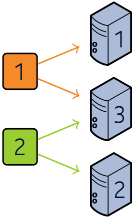
  Public NOKIA Bell Labs TELECOM ParisTech

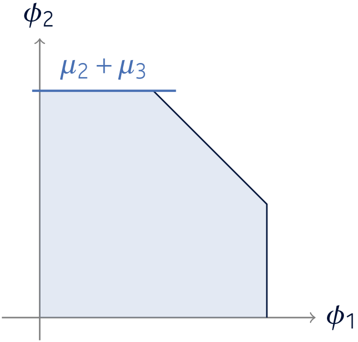
Balanced fairness





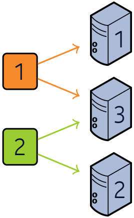
Balanced fairness

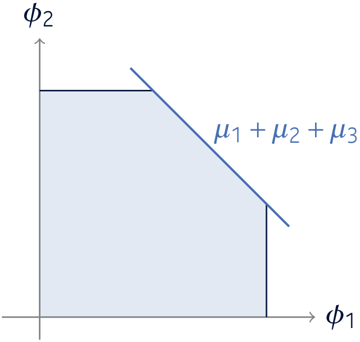




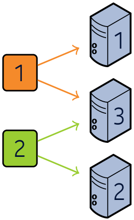
  Public NOKIA Bell Labs TELECOM ParisTech

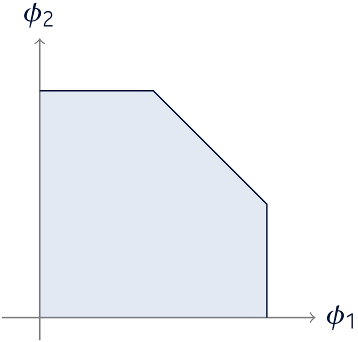
Balanced fairness





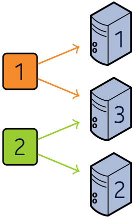
Balanced fairness

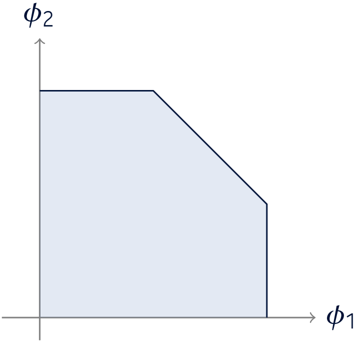




Balanced fairness

- Balance property

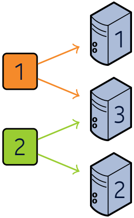


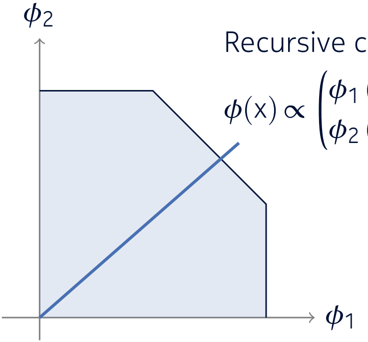


  Public NOKIA Bell Labs TELECOM ParisTech

Balanced fairness

- Balance property





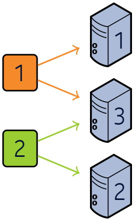
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

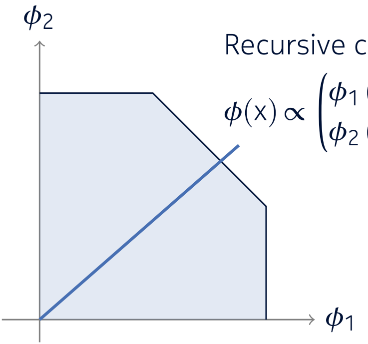
  Public

NOKIA Bell Labs TELECOM ParisTech

Balanced fairness



- Balance property
- Maximize the resource utilization

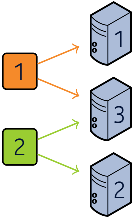


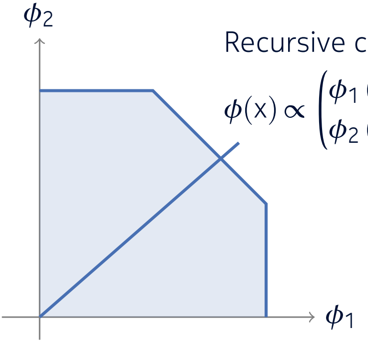
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

Balanced fairness

- Balance property
- Maximize the resource utilization

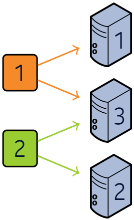




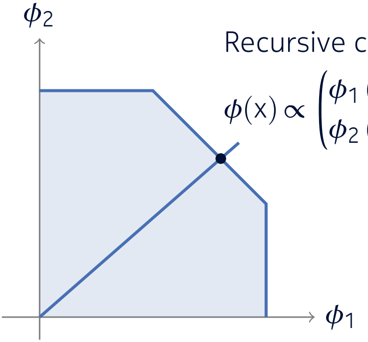
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

Balanced fairness



- Balance property
- Maximize the resource utilization



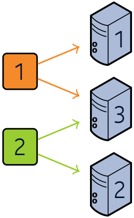
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

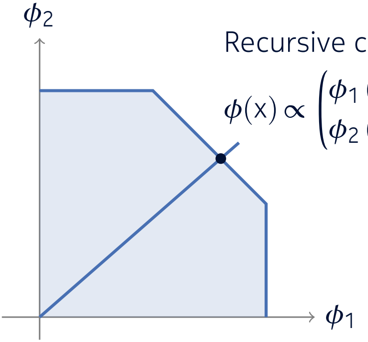
  Public

NOKIA Bell Labs TELECOM ParisTech

Balanced fairness



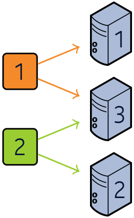
- Stabilizes the maximum set of admissible traffic intensities



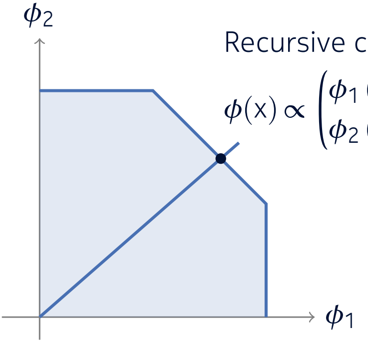
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

Balanced fairness



- Explicit stationary distribution of the system state x

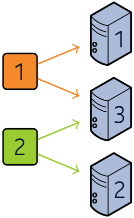


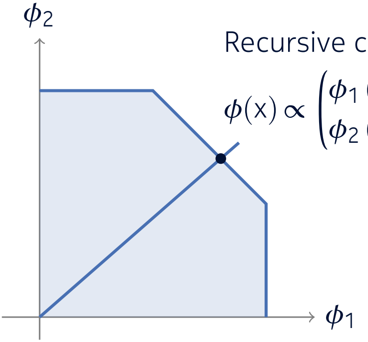
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x-e_2) \\ \phi_2(x-e_1) \end{pmatrix}$$

  Public

In resource pools



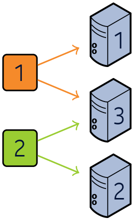


Recursive construction

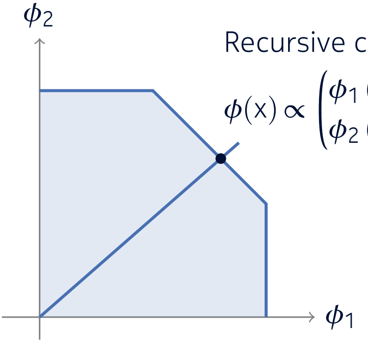
$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

NOKIA Bell Labs TELECOM ParisTech

In resource pools



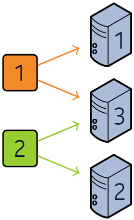
- Pareto-efficiency
  (Shah and de Veciana, 2015)



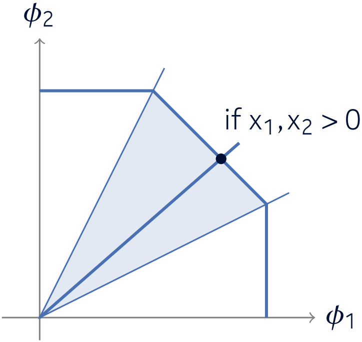
Recursive construction

$$\phi(x) \propto \begin{pmatrix} \phi_1(x - e_2) \\ \phi_2(x - e_1) \end{pmatrix}$$

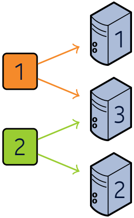
In resource pools



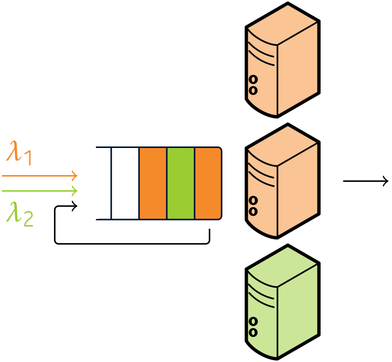
- Pareto-efficiency (Shah and de Veciana, 2015)



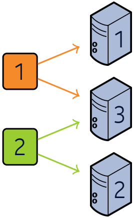
Scheduling



- Idealisation of an extension of round-robin scheduling algorithm (Bonald and Comte, 2017)



Related work



- Recursion on the set of active classes
  - Proposed in (Bonald and Virtamo, 2004) and (Shah and de Veciana, 2015)
  - Exponential complexity in general
  - Polynomial complexity under “poly-symmetry” (Bonald et al., 2017)

- Explicit formulas in specific configurations (Gardner et al., 2017)

Outline

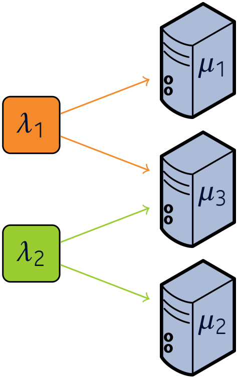
Resource allocation

New formula for performance prediction

Applications

- Gain of differentiation
- Impact of locality

Resource pool with parallel processing



Arrivals

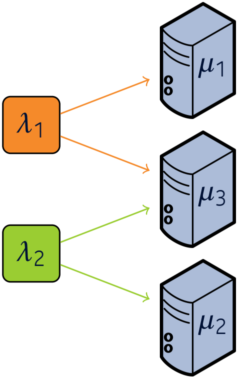
- Class i has traffic intensity λ_i
- Poisson arrival process
- i.i.d. job sizes

Service

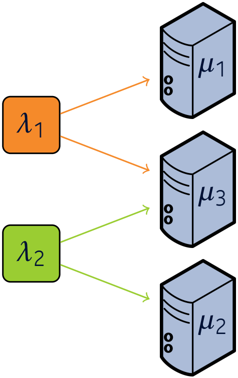
- Server k has capacity μ_k
- Parallel processing

Resources are allocated according to balanced fairness

Resource pool with parallel processing

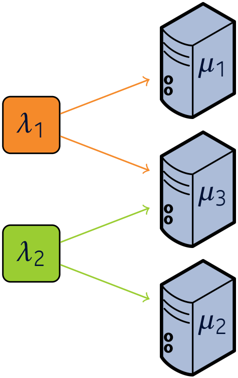


Resource pool with parallel processing



Additional notations

Resource pool with parallel processing

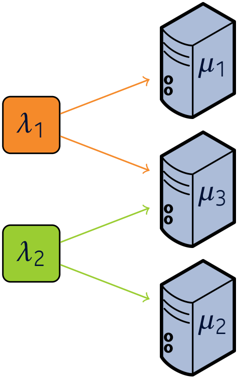


Additional notations

- $\mathscr{I}$ set of classes

  Public NOKIA Bell Labs TELECOM ParisTech

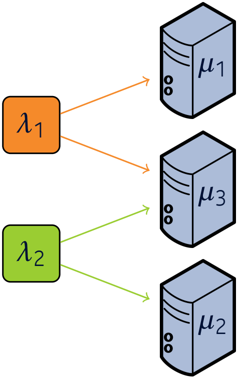
Resource pool with parallel processing



Additional notations

- $\mathscr{I}$ set of classes
- $\mathscr{K}$ set of servers

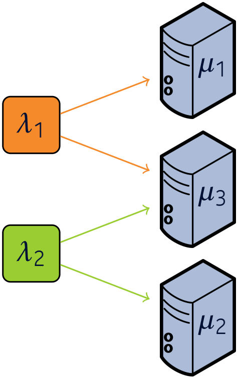
Resource pool with parallel processing



Additional notations

- $\mathscr{I}$ set of classes
- $\mathscr{K}$ set of servers
- $\mathscr{I}_k$ set of classes that can be processed by server k

Resource pool with parallel processing



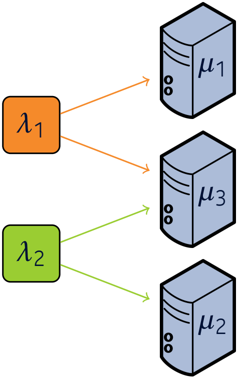
Additional notations

- $\mathscr{I}$ set of classes
- $\mathscr{K}$ set of servers
- $\mathscr{I}_k$ set of classes that can be processed by server k
- $\rho = \frac{\sum_{i \in \mathscr{I}} \lambda_i}{\sum_{k \in \mathscr{K}} \mu_k}$ load of the system

NOKIA Bell Labs

TELECOM ParisTech

Resource pool with parallel processing



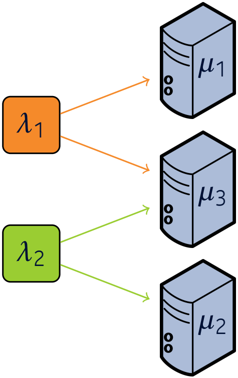
Additional notations

- $\mathscr{I}$ set of classes
- $\mathscr{K}$ set of servers
- $\mathscr{I}_k$ set of classes that can be processed by server k
- $\rho = \frac{\sum_{i \in \mathscr{I}} \lambda_i}{\sum_{k \in \mathscr{K}} \mu_k}$ load of the system

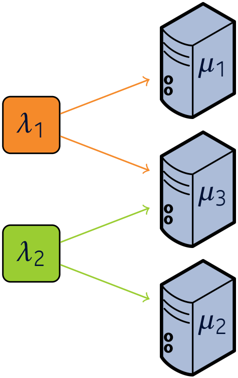
Stability: for all $\mathscr{L} \subseteq \mathscr{K}$ with $\mathscr{L} \neq \emptyset$,

$$\sum_{i \in \mathscr{I} \setminus \bigcup_{k \in \mathscr{K} \setminus \mathscr{L}} \mathscr{I}_k} \lambda_i < \sum_{k \in \mathscr{L}} \mu_k$$

Server idling

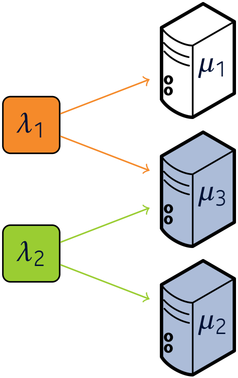


Server idling



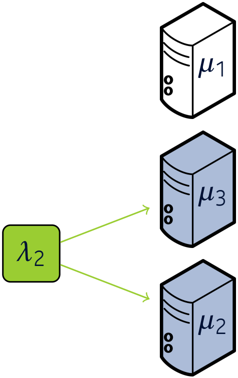
Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathcal{I}_k$

Server idling



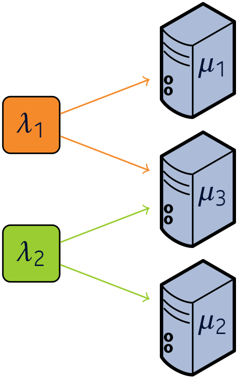
Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathscr{I}_k$

Server idling



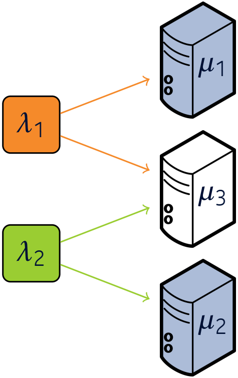
Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathcal{I}_k$

Server idling



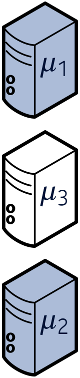
Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathcal{I}_k$

Server idling



Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathscr{I}_k$

Server idling

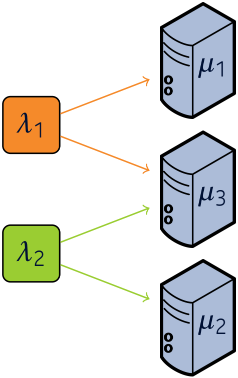


Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathscr{I}_k$

  Public

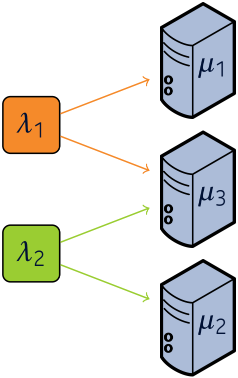
NOKIA Bell Labs TELECOM ParisTech

Server idling



Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathscr{I}_k$

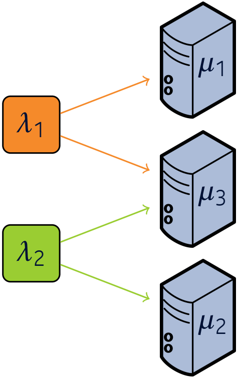
Server idling



Conditionally on server k being idle, the stationary system behaves like the same system without traffic generated by the classes in $\mathscr{I}_k$

Special case where $\mathscr{I}_k = \mathscr{I}$

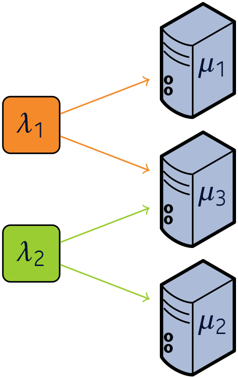
Server idling



Conditional probabilities

$$\psi = \psi_{k}\psi_{I-k}$$

Server idling

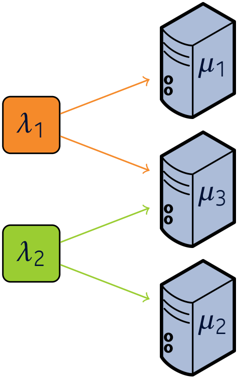


Conditional probabilities

$$\psi = \psi_k \psi_{I-k}$$

Probability of an empty system

Server idling



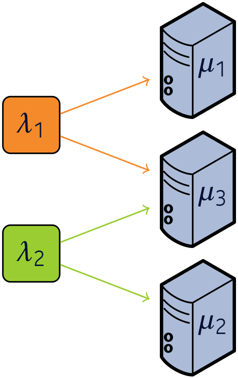
Conditional probabilities

$$\psi = \psi_k \psi_{I-k}$$

Probability of an empty system

Probability that server k is idle

Server idling



Conditional probabilities

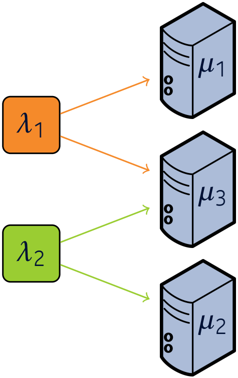
$$\psi = \psi_k \psi_{I-k}$$

Probability of an empty system

Probability that server k is idle

Conditional probability

Server idling



Conditional probabilities

$$\psi = \psi_k \psi_{|-k}$$

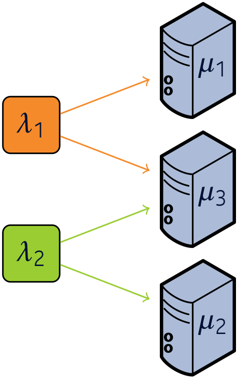
Probability of an empty system

Probability that server k is idle

Conditional probability

$$\psi_{|-k} = \mathbb{P}\left(\begin{array}{c}\text{the system is empty,}\\ \text{given that server}\\ \text{k is idle}\end{array}\right)$$

Server idling



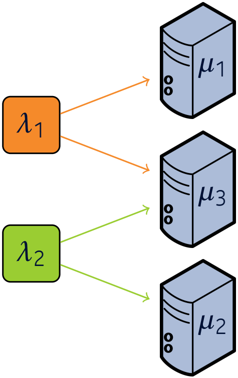
Conditional probabilities

$$\psi = \psi_k \psi_{|-k}$$

- ψ: Probability of an empty system
- ψ_k: Probability that server k is idle
- $\psi_{|-k}$: Conditional probability

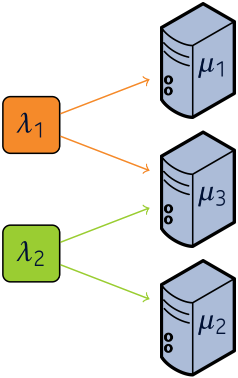
$$\psi_{|-k} = \mathbb{P}\left(\begin{array}{c}\text{the subsystem without}\\ \text{traffic generated by the}\\ \text{classes in } \mathscr{I}_k \text{ is idle}\end{array}\right)$$

Probability of an empty system



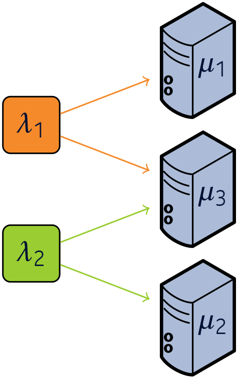
$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$$

Probability of an empty system



$$\psi = (1-\rho) \times \frac{\sum_{k\in\mathcal{K}} \mu_k}{\sum_{k\in\mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$$

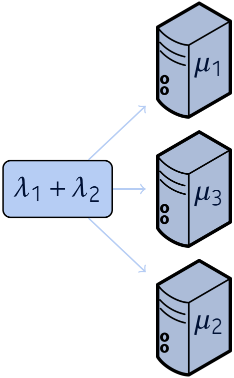
Probability of an empty system



$$\psi = (1 - \rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$$

Complete resource pooling

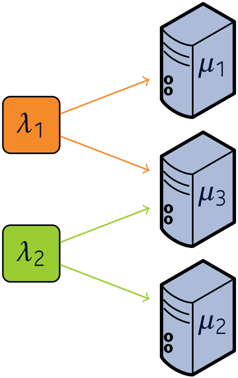
Probability of an empty system



$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$$

Complete resource pooling

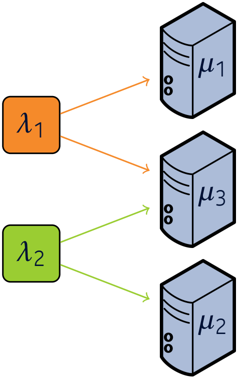
Probability of an empty system



$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$$

Complete resource pooling

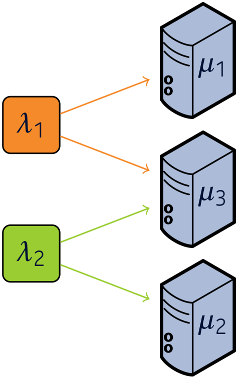
Probability of an empty system



$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{I-k}}}$$

Complete resource pooling

Probability of an empty system



$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{I-k}}}$$

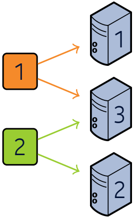
Complete resource pooling

Overhead due to incomplete pooling

  Public

NOKIA Bell Labs
TELECOM ParisTech

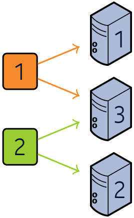
Proof



Proof

- Conservation equation

$$\sum_{i \in \mathscr{I}} \lambda_i = \sum_{k \in \mathscr{K}} \mu_k (1 - \psi_k),$$



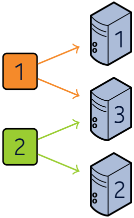
  Public

Proof

- Conservation equation

$$\sum_{i \in \mathscr{I}} \lambda_i = \sum_{k \in \mathscr{K}} \mu_k (1 - \psi_k),$$

$$\text{i.e.} \quad \sum_{k \in \mathscr{K}} \mu_k \psi_k = \sum_{k \in \mathscr{K}} \mu_k - \sum_{i \in \mathscr{I}} \lambda_i,$$

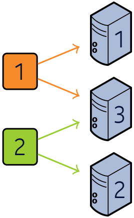


Proof

- Conservation equation

$$\sum_{i\in\mathscr{I}} \lambda_i = \sum_{k\in\mathscr{K}} \mu_k (1-\psi_k),$$

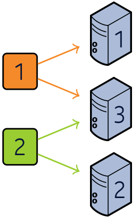
$$\text{i.e.} \quad \sum_{k\in\mathscr{K}} \mu_k \psi_k = \sum_{k\in\mathscr{K}} \mu_k - \sum_{i\in\mathscr{I}} \lambda_i,$$



- Conditional probabilities

$$\psi = \psi_k \psi_{\mathscr{I}-k} \quad \rightarrow \quad \psi_k = \frac{\psi}{\psi_{\mathscr{I}-k}}$$

Proof



- Conservation equation

$$\sum_{i \in \mathscr{I}} \lambda_i = \sum_{k \in \mathscr{K}} \mu_k (1 - \psi_k),$$

$$\text{i.e.} \quad \sum_{k \in \mathscr{K}} \mu_k \psi_k = \sum_{k \in \mathscr{K}} \mu_k - \sum_{i \in \mathscr{I}} \lambda_i,$$

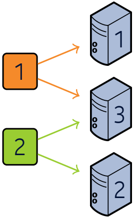
- Conditional probabilities

$$\psi = \psi_k \psi_{I-k} \quad \rightarrow \quad \psi_k = \frac{\psi}{\psi_{I-k}}$$

- Substitution

$$\sum_{k \in \mathscr{K}} \mu_k \frac{\psi}{\psi_{I-k}} = \sum_{k \in \mathscr{K}} \mu_k - \sum_{i \in \mathscr{I}} \lambda_i$$

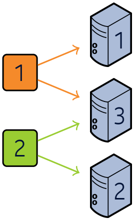
Proof



- Substitution

$$\sum_{k \in \mathcal{K}} \mu_k \frac{\psi}{\psi_{|-k}} = \sum_{k \in \mathcal{K}} \mu_k - \sum_{i \in \mathcal{I}} \lambda_i,$$

Proof



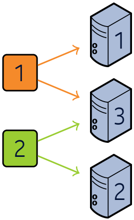
- Substitution

$$\sum_{k \in \mathcal{K}} \mu_k \frac{\psi}{\psi_{|-k}} = \sum_{k \in \mathcal{K}} \mu_k - \sum_{i \in \mathcal{I}} \lambda_i,$$

that is,

$$\psi = \frac{\sum_{k \in \mathcal{K}} \mu_k - \sum_{i \in \mathcal{I}} \lambda_i}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}.$$

Proof



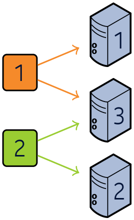
- Substitution

$$\sum_{k\in\mathcal{K}} \mu_k \frac{\psi}{\psi_{|-k}} = \sum_{k\in\mathcal{K}} \mu_k - \sum_{i\in\mathcal{I}} \lambda_i,$$

that is,

$$\psi = \frac{\sum_{k\in\mathcal{K}} \mu_k - \sum_{i\in\mathcal{I}} \lambda_i}{\sum_{k\in\mathcal{K}} \frac{\mu_k}{\psi_{|-k}}} \times \frac{\sum_{k\in\mathcal{K}} \mu_k}{\sum_{k\in\mathcal{K}} \mu_k}.$$

Proof



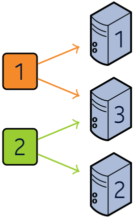
- Substitution

$$\sum_{k \in \mathcal{K}} \mu_k \frac{\psi}{\psi_{|-k}} = \sum_{k \in \mathcal{K}} \mu_k - \sum_{i \in \mathcal{I}} \lambda_i,$$

that is,

$$\psi = \frac{\sum_{k \in \mathcal{K}} \mu_k - \sum_{i \in \mathcal{I}} \lambda_i}{\sum_{k \in \mathcal{K}} \mu_k} \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}.$$

Proof



- Substitution

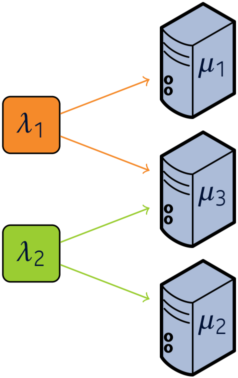
$$\sum_{k\in\mathcal{K}} \mu_k \frac{\psi}{\psi_{|-k}} = \sum_{k\in\mathcal{K}} \mu_k - \sum_{i\in\mathcal{I}} \lambda_i,$$

that is,

$$\psi = \frac{\sum_{k\in\mathcal{K}} \mu_k - \sum_{i\in\mathcal{I}} \lambda_i}{\sum_{k\in\mathcal{K}} \mu_k} \times \frac{\sum_{k\in\mathcal{K}} \mu_k}{\sum_{k\in\mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}.$$

□

Probability of an empty system

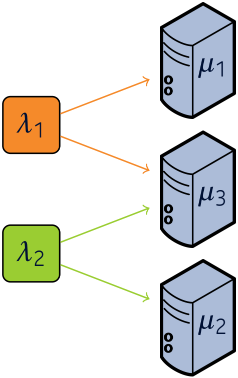


$$\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{I-k}}}$$

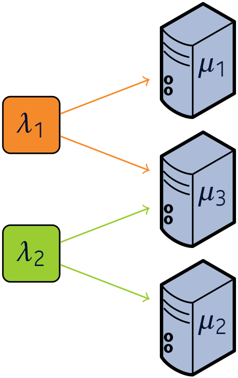
Complete resource pooling

Overhead due to incomplete pooling

Toy example



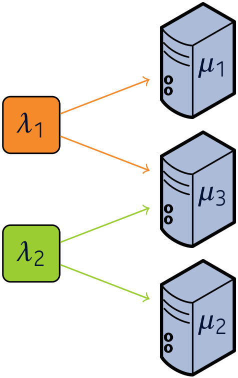
Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

Toy example

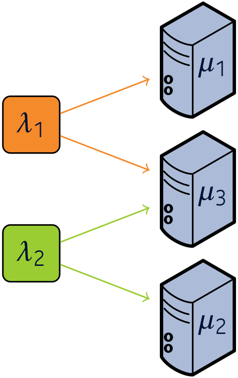


$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

Toy example



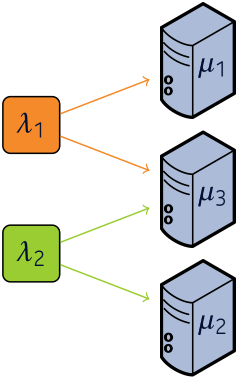
$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

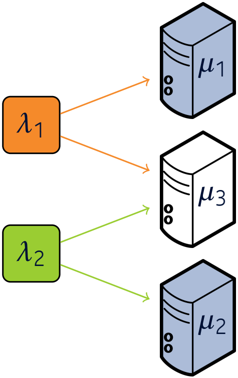
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} =$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

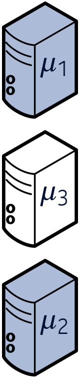
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} =$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

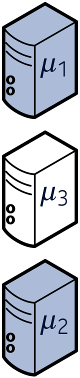
with $\rho = \dfrac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$

$$\psi_{|-1} =$$

$$\psi_{|-3} =$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

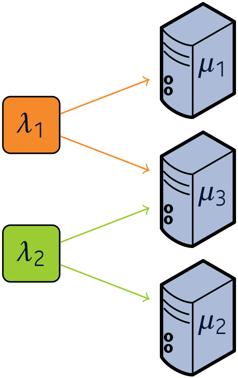
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

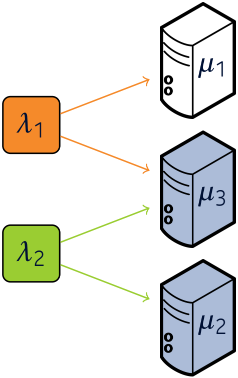
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

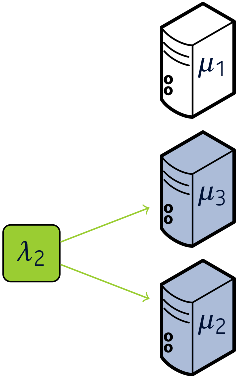
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

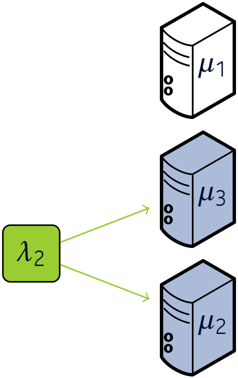
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} =$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

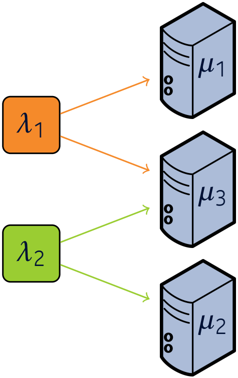
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} = 1 - \rho_{|-1} \quad \text{with} \quad \rho_{|-1} = \frac{\lambda_2}{\mu_2 + \mu_3}$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

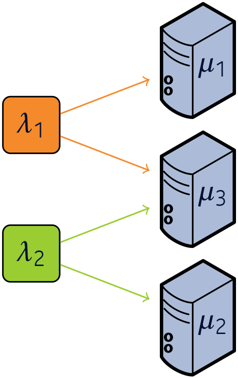
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} = 1 - \rho_{|-1} \quad \text{with} \quad \rho_{|-1} = \frac{\lambda_2}{\mu_2 + \mu_3}$$

$$\psi_{|-3} = 1$$

$$\psi_{|-2} =$$

Toy example



$$\psi = (1-\rho) \times \frac{\mu_1 + \mu_2 + \mu_3}{\frac{\mu_1}{\psi_{|-1}} + \frac{\mu_2}{\psi_{|-2}} + \frac{\mu_3}{\psi_{|-3}}},$$

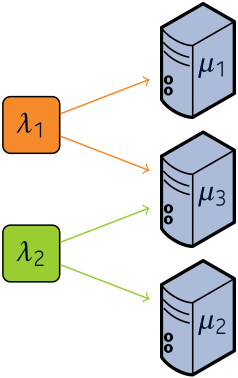
$$\text{with } \rho = \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 + \mu_3}$$

$$\psi_{|-1} = 1 - \rho_{|-1} \quad \text{with} \quad \rho_{|-1} = \frac{\lambda_2}{\mu_2 + \mu_3}$$

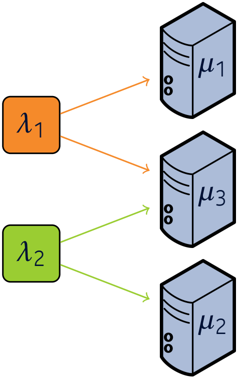
$$\psi_{|-3} = 1$$

$$\psi_{|-2} = 1 - \rho_{|-2} \quad \text{with} \quad \rho_{|-2} = \frac{\lambda_1}{\mu_1 + \mu_3}$$

Mean number of jobs

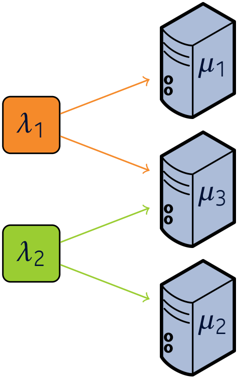


Mean number of jobs



$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho}\frac{\sum_{k\in\mathcal{K}} \mu_k \psi_k L_{I-k}}{\sum_{k\in\mathcal{K}} \mu_k}$$

Mean number of jobs

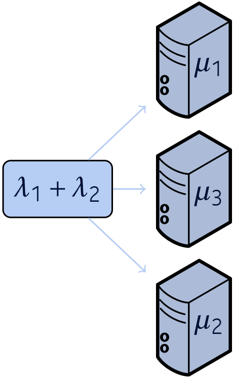


$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho}\frac{\sum_{k\in\mathcal{K}} \mu_k \psi_k L_{|-k}}{\sum_{k\in\mathcal{K}} \mu_k}$$

Complete resource pooling

  Public NOKIA Bell Labs TELECOM ParisTech

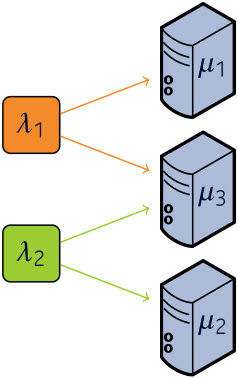
Mean number of jobs



$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{\sum_{k \in \mathcal{K}} \mu_k \psi_k L_{|-k}}{\sum_{k \in \mathcal{K}} \mu_k}$$

Complete resource pooling

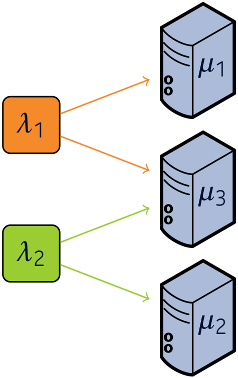
Mean number of jobs



$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{\sum_{k \in \mathcal{K}} \mu_k \psi_k L_{I-k}}{\sum_{k \in \mathcal{K}} \mu_k}$$

Complete resource pooling

Mean number of jobs



$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{\sum_{k \in \mathcal{K}} \mu_k \psi_k L_{I-k}}{\sum_{k \in \mathcal{K}} \mu_k}$$

Complete resource pooling

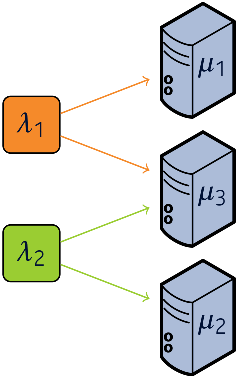
Overhead due to incomplete pooling

Public

NOKIA Bell Labs

TELECOM ParisTech

Mean number of jobs



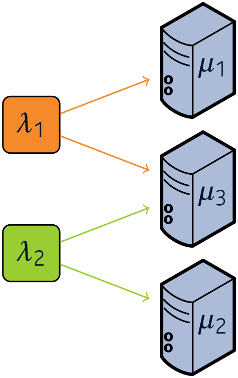
$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{\sum_{k \in \mathcal{K}} \mu_k \psi_k L_{I-k}}{\sum_{k \in \mathcal{K}} \mu_k}$$

Complete resource pooling

Overhead due to incomplete pooling

- Exponential complexity in the number of servers in general

Mean number of jobs



$$L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{\sum_{k \in \mathcal{K}} \mu_k \psi_k L_{l-k}}{\sum_{k \in \mathcal{K}} \mu_k}$$

Complete resource pooling

Overhead due to incomplete pooling

- Exponential complexity in the number of servers in general
- Polynomial in “nice” systems

Outline

Resource allocation

New formula for performance prediction

Applications

- Gain of differentiation
- Impact of locality

  Public

NOKIA Bell Labs

TELECOM ParisTech

Outline

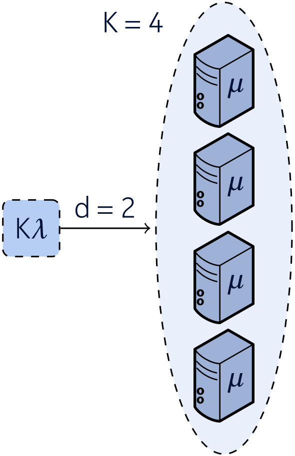
Resource allocation

New formula for performance prediction

Applications

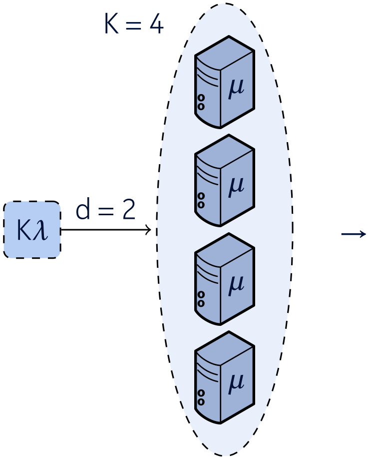
- Gain of differentiation
- Impact of locality

Randomized assignment

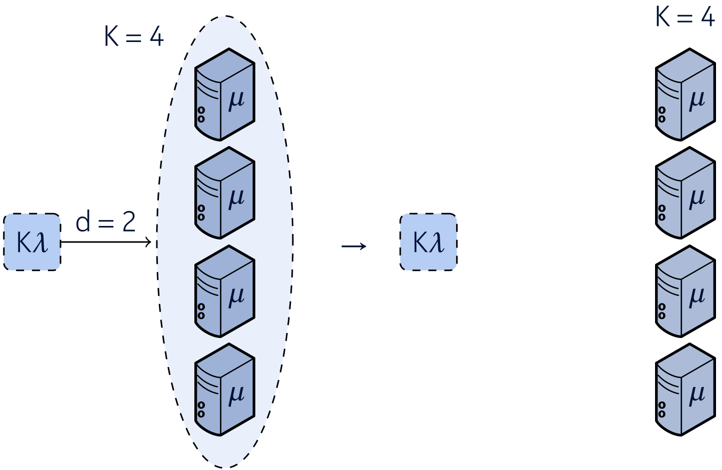


NOKIA Bell Labs
TELECOM ParisTech

Randomized assignment

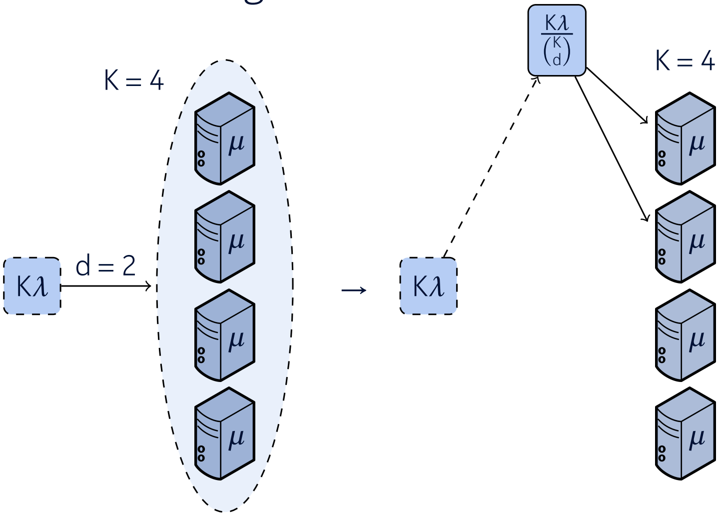


Randomized assignment

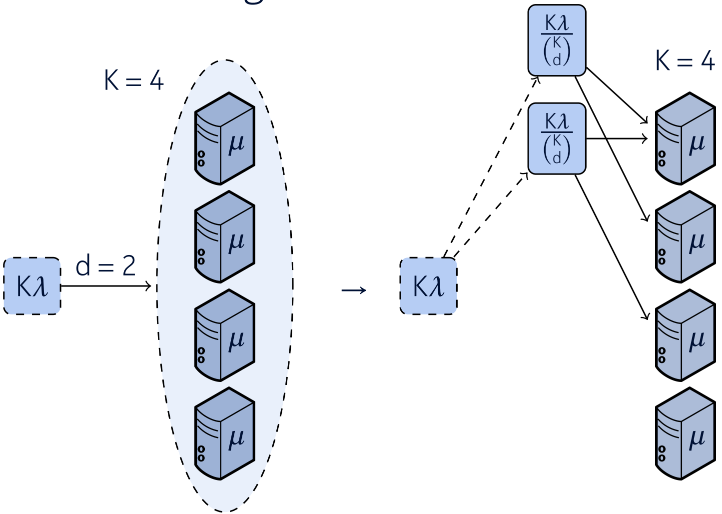


  Public NOKIA Bell Labs TELECOM ParisTech

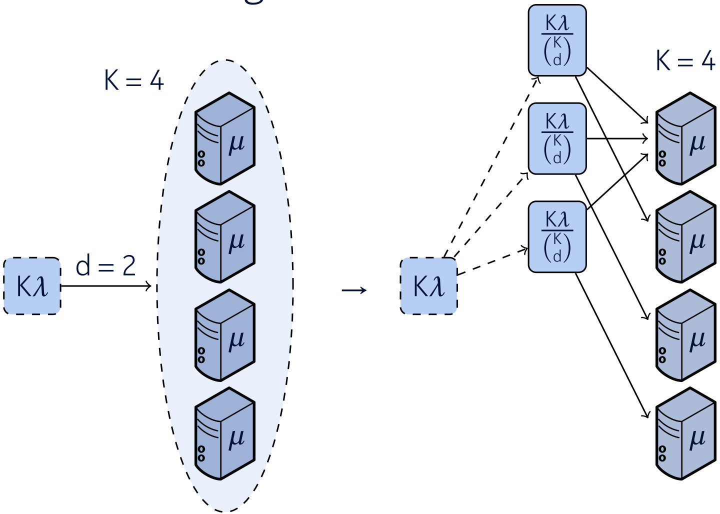
Randomized assignment



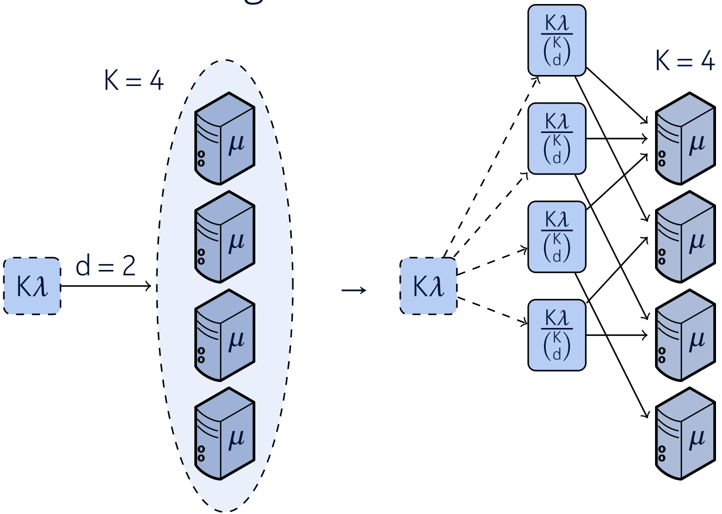
Randomized assignment



Randomized assignment

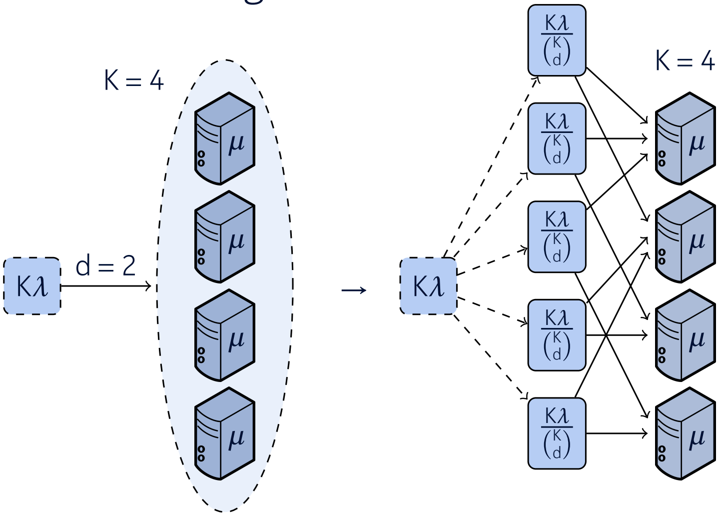


Randomized assignment

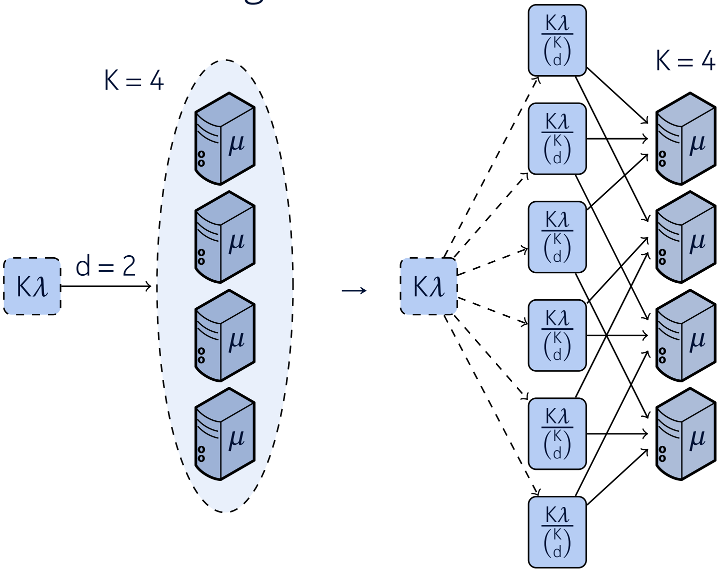


   NOKIA Bell Labs TELECOM ParisTech

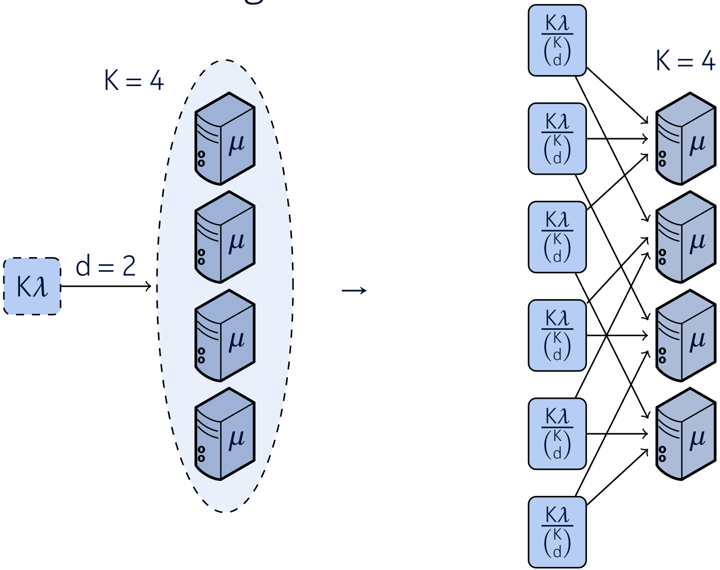
Randomized assignment



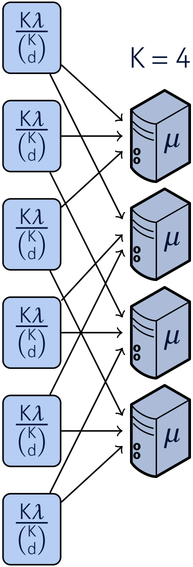
Randomized assignment



Randomized assignment

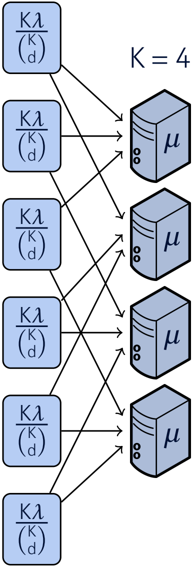


Homogeneous pool



Homogeneous pool

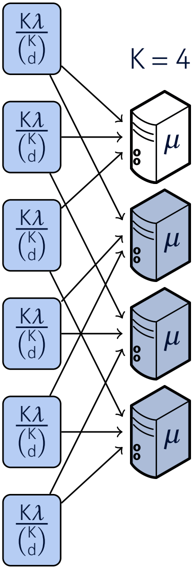
- All servers are exchangeable



  Public NOKIA Bell Labs TELECOM ParisTech

Homogeneous pool

- All servers are exchangeable

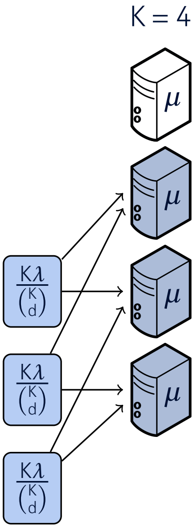


  Public

NOKIA Bell Labs TELECOM ParisTech

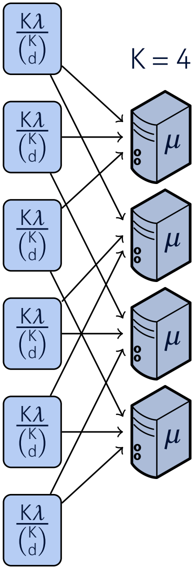
Homogeneous pool

- All servers are exchangeable



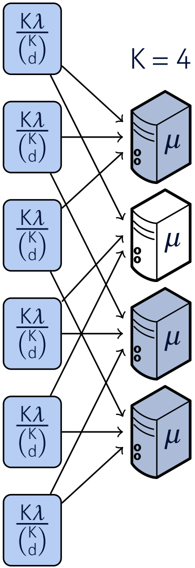
Homogeneous pool

- All servers are exchangeable



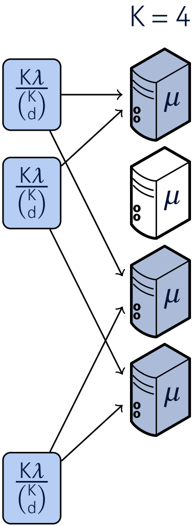
Homogeneous pool

- All servers are exchangeable



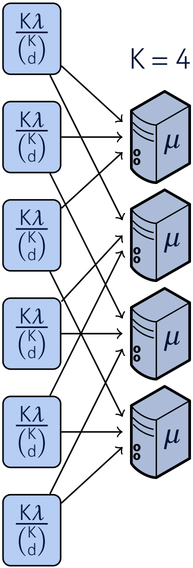
Homogeneous pool

- All servers are exchangeable



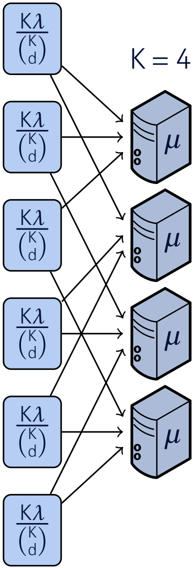
Homogeneous pool

- All servers are exchangeable



Homogeneous pool

- All servers are exchangeable
  The subsystem is again a homogeneous pool



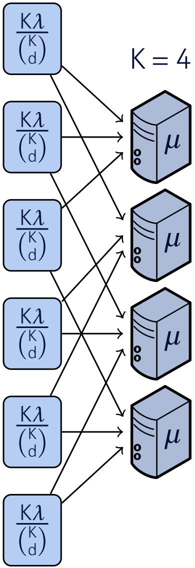
  Public

NOKIA Bell Labs

Homogeneous pool

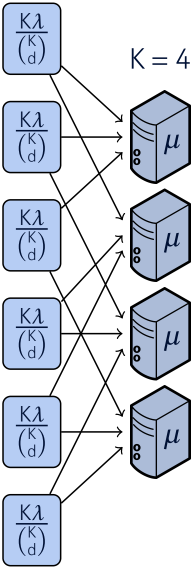
- All servers are exchangeable
  The subsystem is again a homogeneous pool

- $\psi = (1-\rho) \times \frac{\sum_{k \in \mathcal{K}} \mu_k}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$



Homogeneous pool

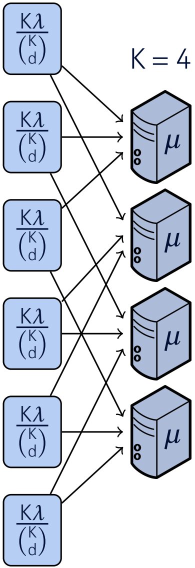
- All servers are exchangeable
  The subsystem is again a homogeneous pool
- $\psi = (1-\rho) \times \dfrac{K\mu}{\sum_{k \in \mathcal{K}} \frac{\mu_k}{\psi_{|-k}}}$



Homogeneous pool

- All servers are exchangeable
  The subsystem is again a
  homogeneous pool

- $\psi = (1 - \rho) \times \dfrac{K\mu}{K \times \frac{\mu}{\psi_{l-K}}}$

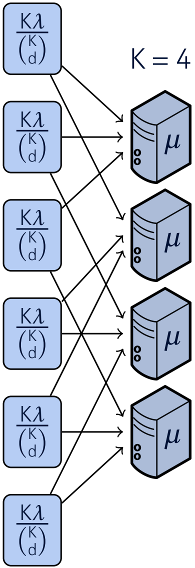


  Public

Homogeneous pool

- All servers are exchangeable
  The subsystem is again a homogeneous pool

- $\psi = (1 - \rho) \times \psi_{|-K}$

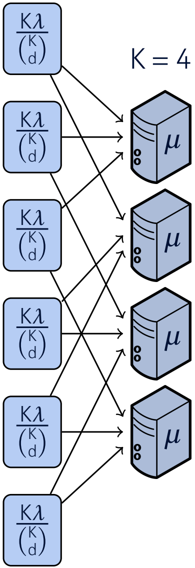


  Public NOKIA Bell Labs TELECOM ParisTech

Homogeneous pool

- All servers are exchangeable
  The subsystem is again a homogeneous pool

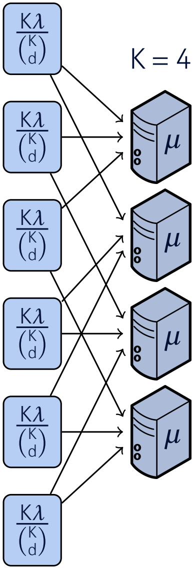
- $\psi = (1 - \rho) \times \psi_{I-K}$
- $L = \dfrac{\rho}{1-\rho} + \dfrac{1}{1-\rho} \dfrac{\sum_{k \in \mathcal{K}} \mu_k \frac{\psi}{\psi_{I-k}} L_{I-k}}{\sum_{k \in \mathcal{K}} \mu_k}$



Homogeneous pool

- All servers are exchangeable
  The subsystem is again a
  homogeneous pool

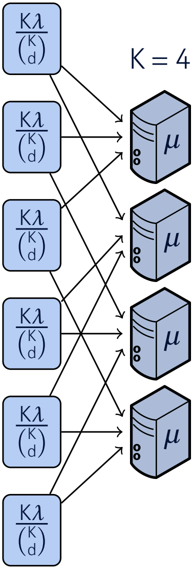
- $\psi = (1-\rho) \times \psi_{\mathrm{I}-\mathrm{K}}$
- $\mathrm{L} = \dfrac{\rho}{1-\rho} + \dfrac{1}{1-\rho}\dfrac{\sum_{k\in\mathcal{K}} \mu_k \frac{\psi}{\psi_{\mathrm{I}-k}} \mathrm{L}_{\mathrm{I}-k}}{\mathrm{K}\mu}$



Homogeneous pool

- All servers are exchangeable
  The subsystem is again a homogeneous pool

- $\psi = (1-\rho) \times \psi_{I-K}$
- $L = \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \frac{K\mu \frac{\psi}{\psi_{I-K}} L_{I-K}}{K\mu}$



  Public

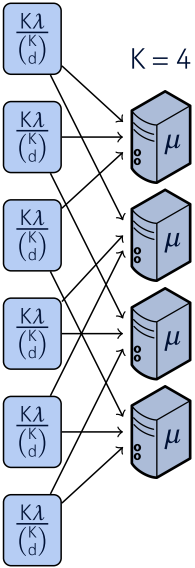
NOKIA Bell Labs TELECOM ParisTech

Homogeneous pool

- All servers are exchangeable
  The subsystem is again a homogeneous pool

- $\psi = (1 - \rho) \times \psi_{I-K}$

- $L = \dfrac{\rho}{1-\rho} + L_{I-K}$



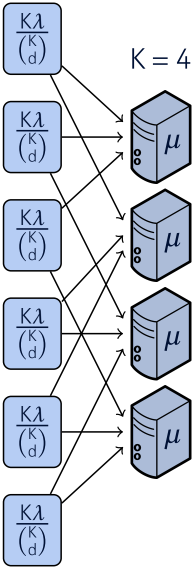
Homogeneous pool

- All servers are exchangeable
  The subsystem is again a
  homogeneous pool

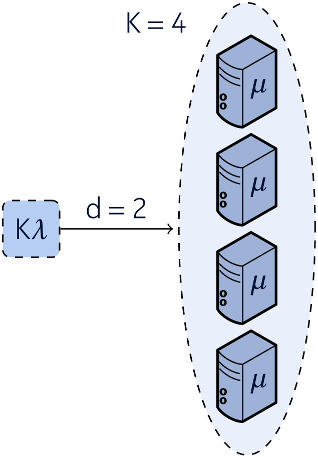
- $\psi = (1 - \rho) \times \psi_{I-K}$

- $L = \dfrac{\rho}{1-\rho} + L_{I-K}$

- New proof for the result of
  (Gardner et al., 2017a)

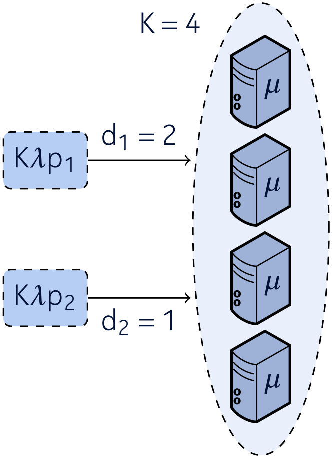


Randomized assignment



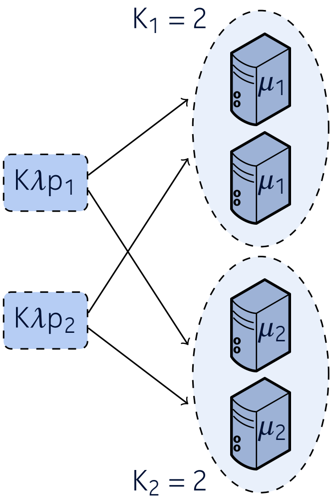
Time complexity
O(K)

Randomized assignment



Time complexity
O(NK)
N = number of job types

Randomized assignment



Time complexity

$O(NSK_1 \cdots K_S)$

N = number of job types

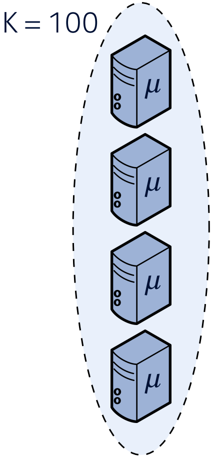
S = number of server pools

  Public

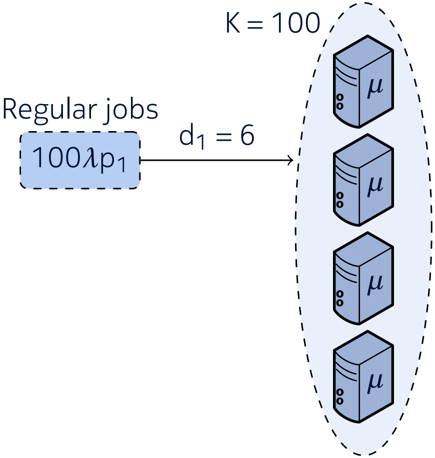
NOKIA Bell Labs TELECOM ParisTech

Gain of differentiation

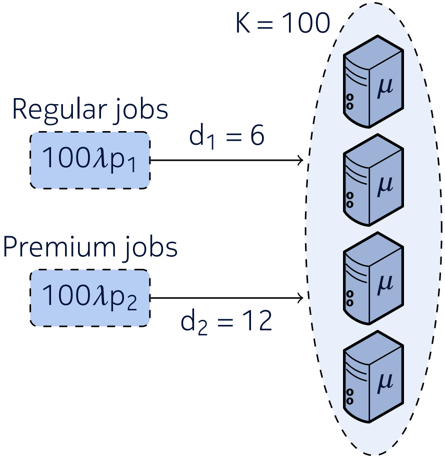
Gain of differentiation



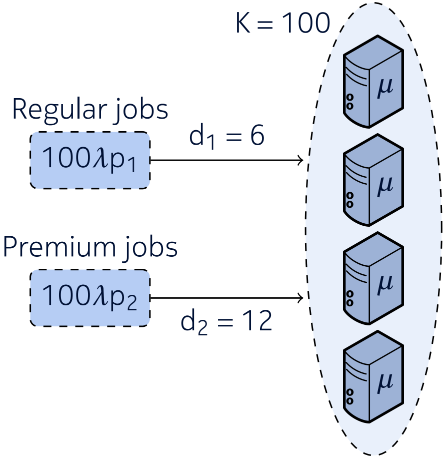
Gain of differentiation



Gain of differentiation



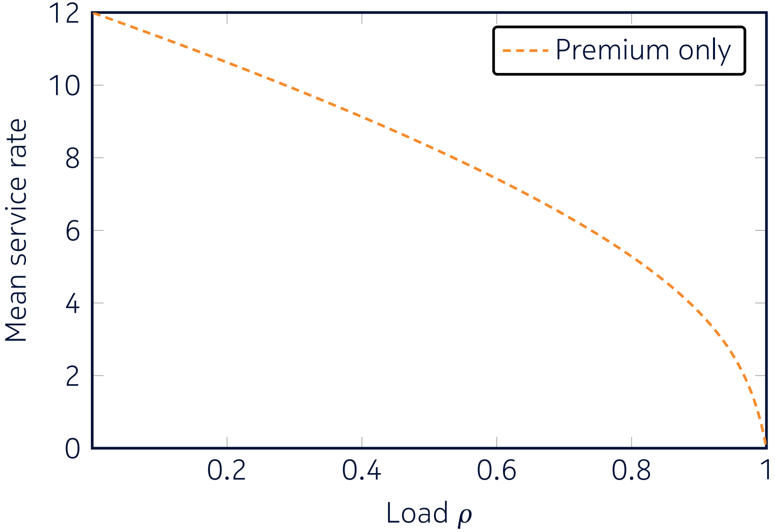
Gain of differentiation



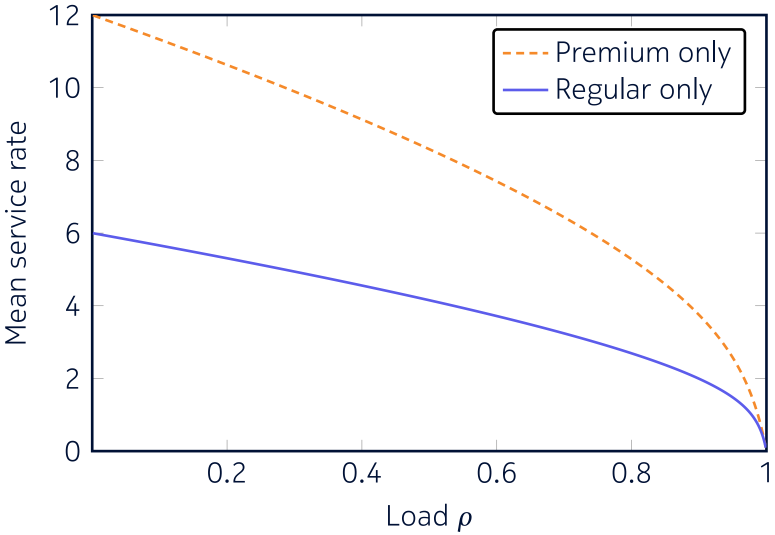
Study the impact of the job distribution on performance

1. Premium only
2. Regular only
3. Mixed

Gain of differentiation



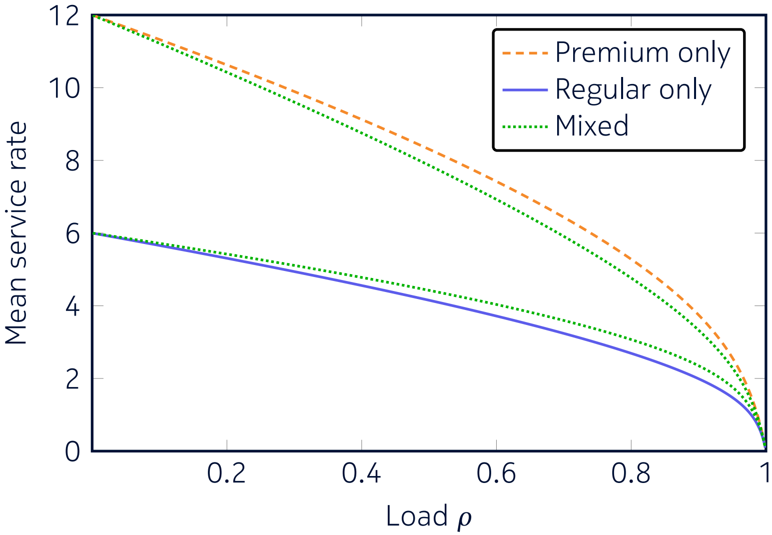
Gain of differentiation



  Public

NOKIA Bell Labs TELECOM ParisTech

Gain of differentiation



  Public

NOKIA Bell Labs TELECOM ParisTech

Outline

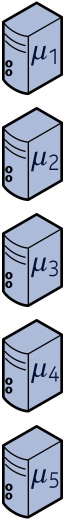
Resource allocation

New formula for performance prediction

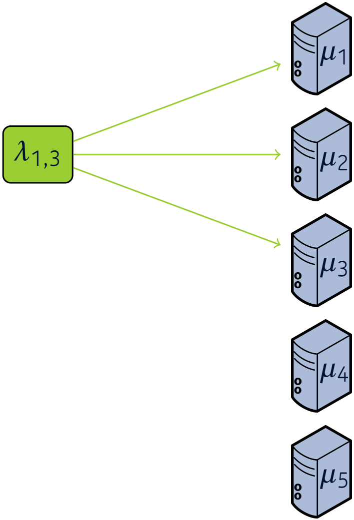
Applications

- Gain of differentiation
- Impact of locality

Line structure

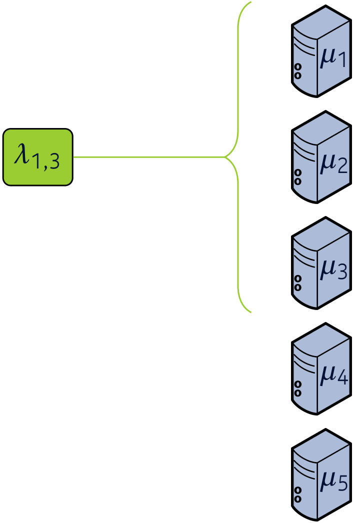


Line structure



  Public NOKIA Bell Labs TELECOM ParisTech

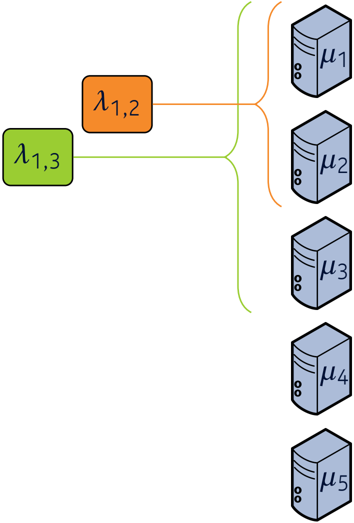
Line structure



  Public

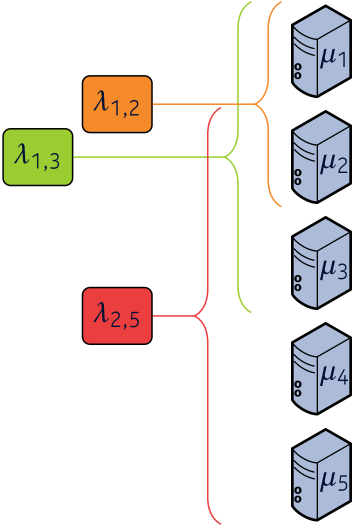
NOKIA Bell Labs TELECOM ParisTech

Line structure

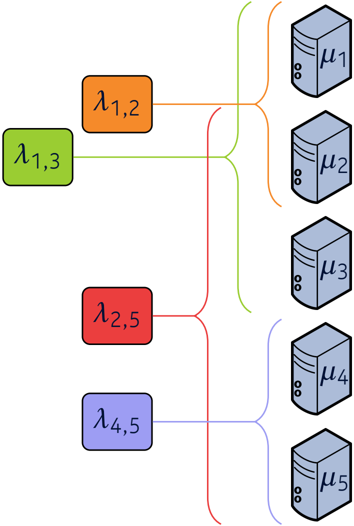


  Public

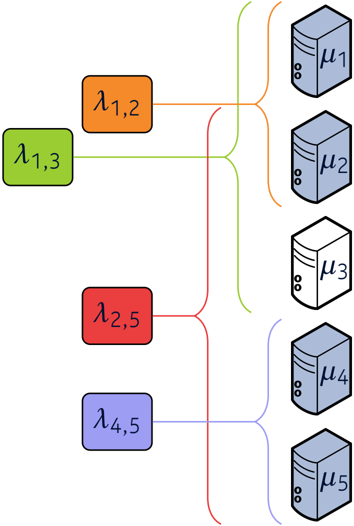
Line structure



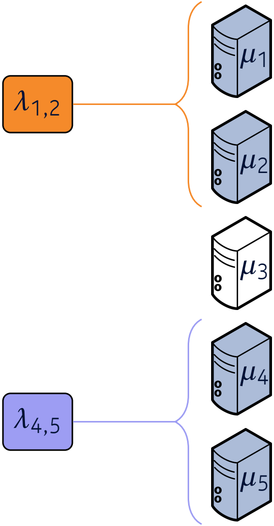
Line structure



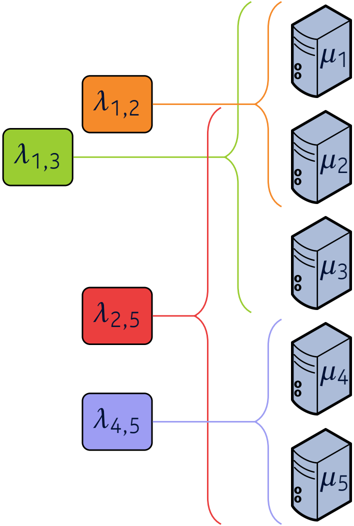
Line structure



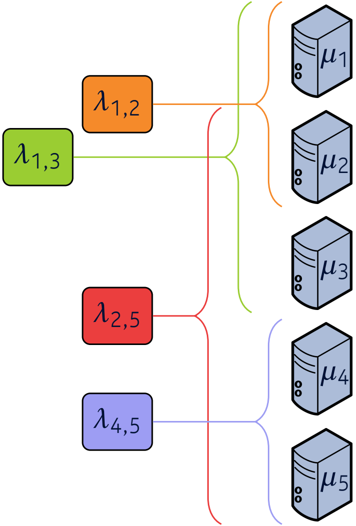
Line structure



Line structure



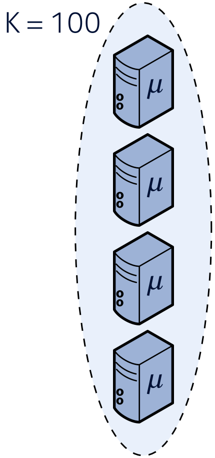
Line structure



Time complexity
$O(K^3)$ in general,
$O(K^2)$ in homogeneous pools

Impact of locality

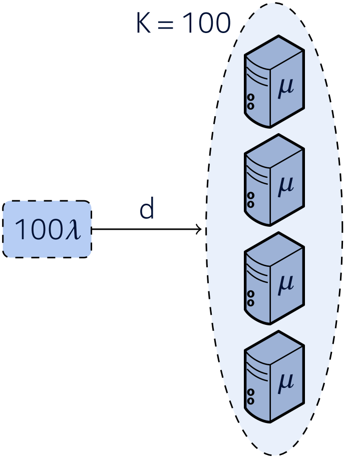
Impact of locality



  Public

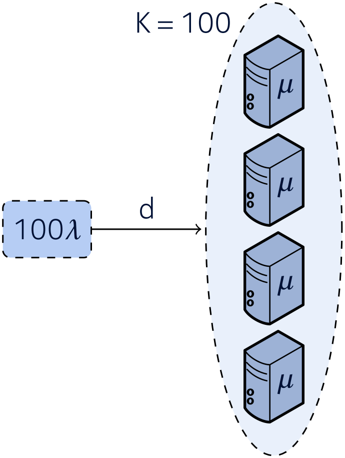
NOKIA Bell Labs TELECOM ParisTech

Impact of locality



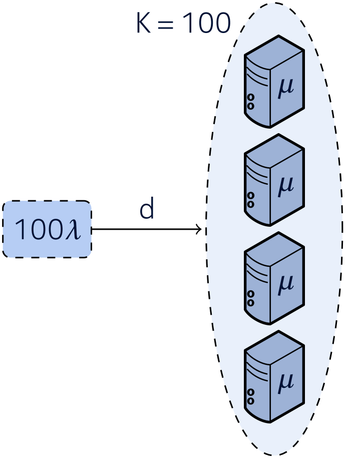
  Public NOKIA Bell Labs TELECOM ParisTech

Impact of locality



Load $\rho = \frac{\lambda}{\mu} = 0.9$

Impact of locality

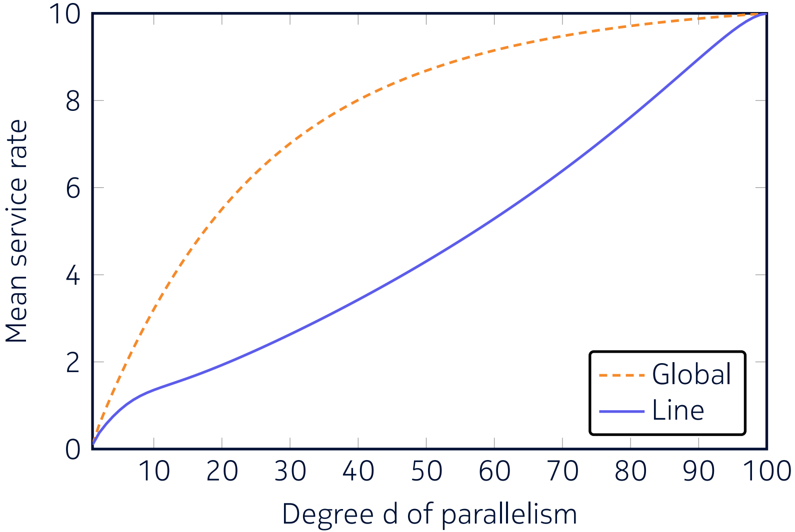


Load $\rho = \frac{\lambda}{\mu} = 0.9$

Study the impact of
locality on performance
under randomized assignment

① Global ② Line

Impact of locality



Conclusion

- New recursive formula to predict the performance of balanced fairness in an arbitrary compatibility graph

Conclusion

- New recursive formula to predict the performance of balanced fairness in an arbitrary compatibility graph
- Exponential time complexity in the number of servers in general

Conclusion

- New recursive formula to predict the performance of balanced fairness in an arbitrary compatibility graph
- Exponential time complexity in the number of servers in general
- Practically interesting configurations where the complexity is polynomial

Conclusion

- New recursive formula to predict the performance of balanced fairness in an arbitrary compatibility graph
- Exponential time complexity in the number of servers in general
- Practically interesting configurations where the complexity is polynomial
  - Randomized assignment

Conclusion

- New recursive formula to predict the performance of balanced fairness in an arbitrary compatibility graph
- Exponential time complexity in the number of servers in general
- Practically interesting configurations where the complexity is polynomial
  - Randomized assignment
  - Line, nested, ring pools