Load Balancing in Heterogeneous Server Clusters: Insights From a Product-Form Queueing Model

IEEE/ACM International Symposium on Quality of Service

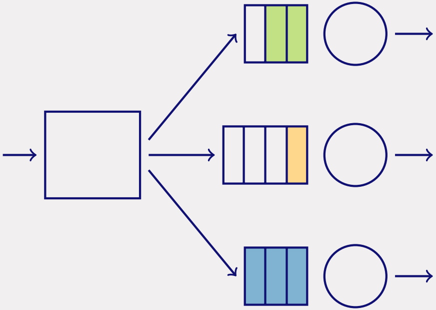
Mark van der Boor and Céline Comte

Eindhoven University of Technology

Heterogeneous server cluster

Model

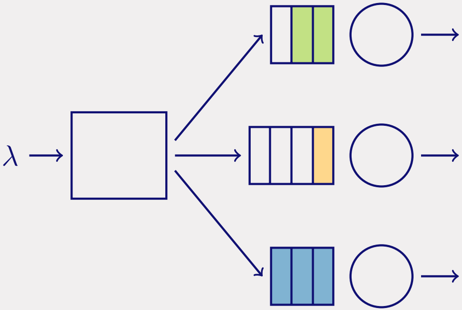
- Dispatcher, n servers, jobs



Heterogeneous server cluster

Model

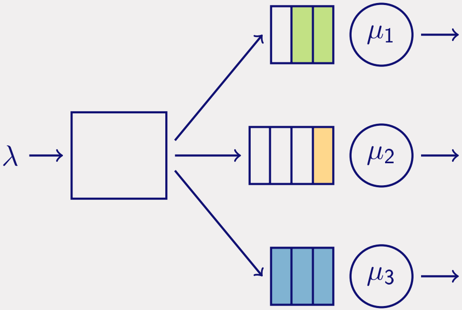
- Dispatcher, n servers, jobs
- Poisson arrival process with rate λ



Heterogeneous server cluster

Model

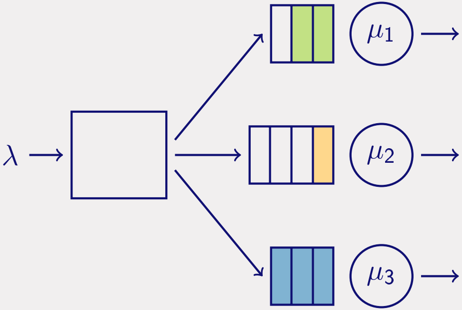
- Dispatcher, n servers, jobs
- Poisson arrival process with rate λ
- Service time exponential with rate μ_i, with $\mu_1 > \mu_2 > \ldots > \mu_n$



Heterogeneous server cluster

Model

- Dispatcher, n servers, jobs
- Poisson arrival process with rate λ
- Service time exponential with rate μ_i, with $\mu_1 > \mu_2 > \ldots > \mu_n$
- Buffer of length $\ell_i < \infty$

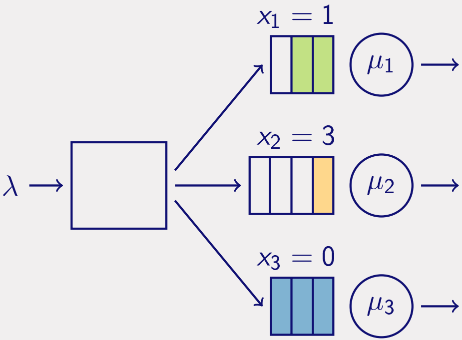


Heterogeneous server cluster

Model

- Dispatcher, n servers, jobs
- Poisson arrival process with rate λ
- Service time exponential with rate μ_i, with $\mu_1 > \mu_2 > \ldots > \mu_n$
- Buffer of length $\ell_i < \infty$

State: $x = (x_1, x_2, \ldots, x_n)$
x_i = number of available slots at server i



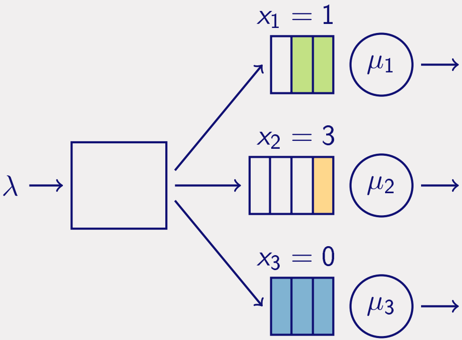
Heterogeneous server cluster

Model

- Dispatcher, n servers, jobs
- Poisson arrival process with rate λ
- Service time exponential with rate μ_i, with $\mu_1 > \mu_2 > \ldots > \mu_n$
- Buffer of length $\ell_i < \infty$

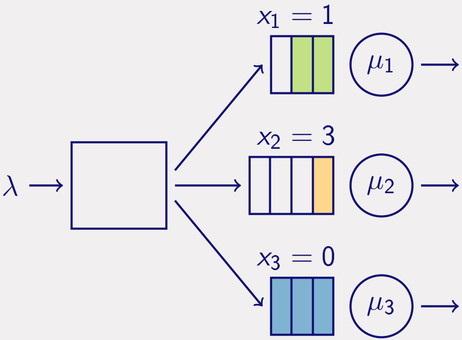
State: $x = (x_1, x_2, \ldots, x_n)$
x_i = number of available slots at server i

Examples: cloud, manufacturing...



Heterogeneous server cluster

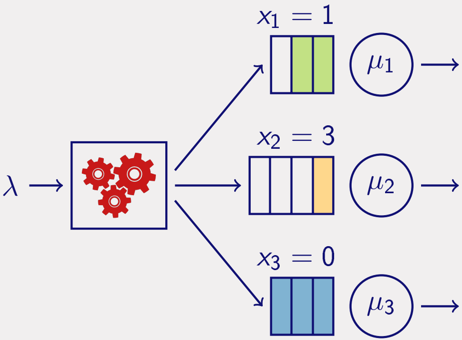
Scheduling: Any non-anticipating policy
Processor-sharing, first-come-first-served, ...



Heterogeneous server cluster

Scheduling: Any non-anticipating policy
Processor-sharing, first-come-first-served, ...

Load balancing: Immediate and irrevocable
Choose server i with probability $\frac{x_i}{x_1+\ldots+x_n}$



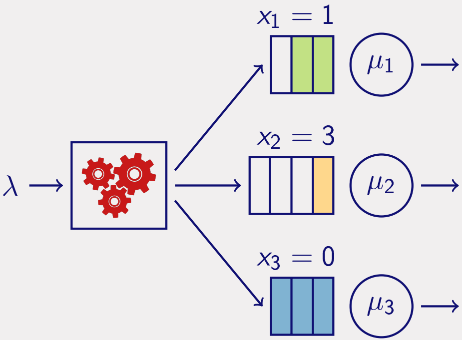
Heterogeneous server cluster

Scheduling: Any non-anticipating policy
Processor-sharing, first-come-first-served, ...

Load balancing: Immediate and irrevocable
Choose server i with probability $\frac{x_i}{x_1+\ldots+x_n}$

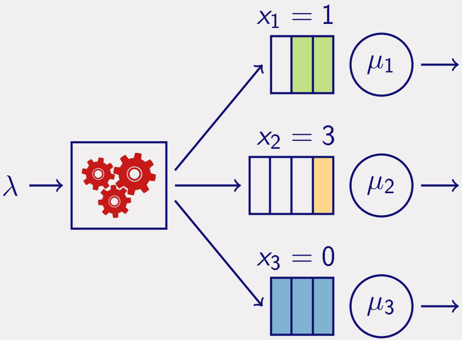
Relations with other algorithms:

- Insensitive (Bonald et al., 2004)
- Join-idle-queue (Lu et al., 2011)
- Join-below-threshold (Zhou et al., 2018)
- Idle-one-queue (Gupta and Walton, 2019)



Stationary distribution

The evolution of the state $x = (x_1, \ldots, x_n)$ defines a continuous-time Markov chain.

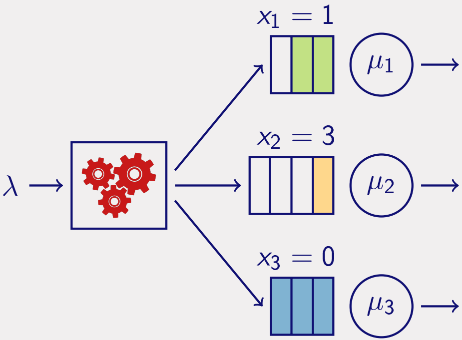


Stationary distribution

The evolution of the state $x = (x_1, \ldots, x_n)$ defines a continuous-time Markov chain.

Stationary distribution: For $x \leq \ell$,

$$\pi(x) = \beta(\ell) \binom{x_1 + \ldots + x_n}{x_1, \ldots, x_n} \prod_{i=1}^{n} \left(\frac{\mu_i}{\lambda}\right)^{x_i}.$$



Stationary distribution

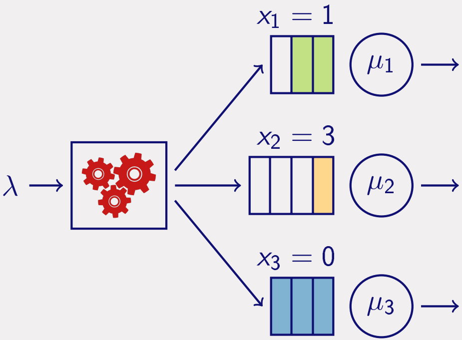
The evolution of the state $x = (x_1, \ldots, x_n)$ defines a continuous-time Markov chain.

Stationary distribution: For $x \leq \ell$,

$$\pi(x) = \beta(\ell) \binom{x_1 + \ldots + x_n}{x_1, \ldots, x_n} \prod_{i=1}^{n} \left(\frac{\mu_i}{\lambda}\right)^{x_i}.$$

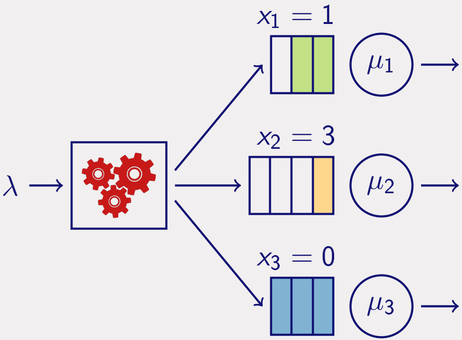
Loss probability:

$$\frac{1}{\beta(\ell)} = \sum_{x \leq \ell} \binom{x_1 + \ldots + x_n}{x_1, \ldots, x_n} \prod_{i=1}^{n} \left(\frac{\mu_i}{\lambda}\right)^{x_i}.$$



Problem and contributions

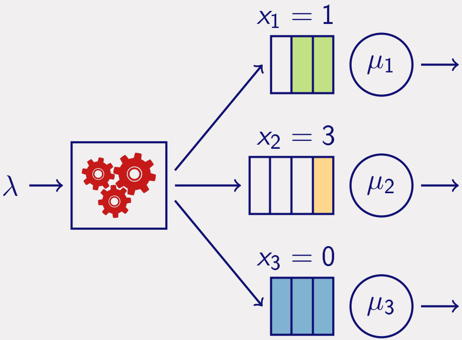
Question: Given λ, $\mu_1, \mu_2, \ldots, \mu_n$, and $L = \ell_1 + \ell_2 + \ldots + \ell_n$, how to choose ℓ_1, ℓ_2, $\ldots$, ℓ_n to minimize the loss probability?



Problem and contributions

Question: Given λ, $\mu_1, \mu_2, \ldots, \mu_n$, and $L = \ell_1 + \ell_2 + \ldots + \ell_n$, how to choose ℓ_1, ℓ_2, $\ldots$, ℓ_n to minimize the loss probability?

Motivation: Trade-off loss probability vs. $\{$mean response time, communication cost$\}$.



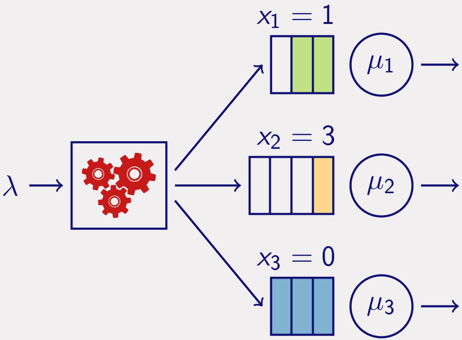
Problem and contributions

Question: Given λ, $\mu_1, \mu_2, \ldots, \mu_n$, and $L = \ell_1 + \ell_2 + \ldots + \ell_n$, how to choose ℓ_1, ℓ_2, $\ldots$, ℓ_n to minimize the loss probability?

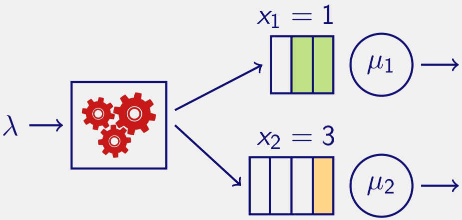
Motivation: Trade-off loss probability vs. $\{$mean response time, communication cost$\}$.

Contributions:

- Low-traffic analysis: $\lambda \ll \mu_1 + \ldots + \mu_n$
- Heavy-traffic analysis: $\lambda \gg \mu_1 + \ldots + \mu_n$
- Monotonicity result: λ increases

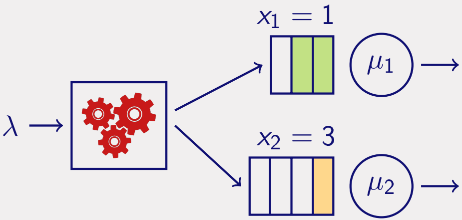


Analytical results



Analytical results

Low traffic: There is $\lambda_* > 0$ such that, for $\lambda \leq \lambda_*$, the loss probability is minimized when $\frac{\ell_1}{L} \simeq \frac{\mu_1}{\mu_1+\mu_2}$ and $\frac{\ell_2}{L} \simeq \frac{\mu_2}{\mu_1+\mu_2}$.

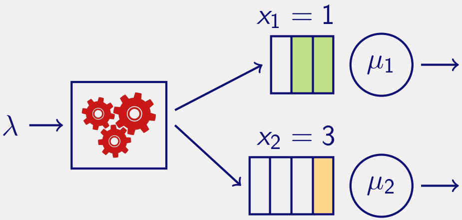


TU/e

Analytical results

Low traffic: There is $\lambda_* > 0$ such that, for $\lambda \leq \lambda_*$, the loss probability is minimized when $\frac{\ell_1}{L} \simeq \frac{\mu_1}{\mu_1+\mu_2}$ and $\frac{\ell_2}{L} \simeq \frac{\mu_2}{\mu_1+\mu_2}$.

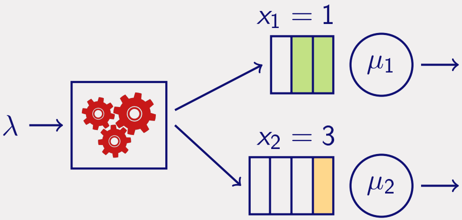
Heavy traffic: There is $\lambda^* > 0$ such that, for $\lambda \geq \lambda^*$, the loss probability is minimized when $\frac{\ell_1}{L} \simeq \frac{1}{2}$ and $\frac{\ell_2}{L} \simeq \frac{1}{2}$.



Analytical results

Low traffic: There is $\lambda_* > 0$ such that, for $\lambda \leq \lambda_*$, the loss probability is minimized when $\frac{\ell_1}{L} \simeq \frac{\mu_1}{\mu_1+\mu_2}$ and $\frac{\ell_2}{L} \simeq \frac{\mu_2}{\mu_1+\mu_2}$.

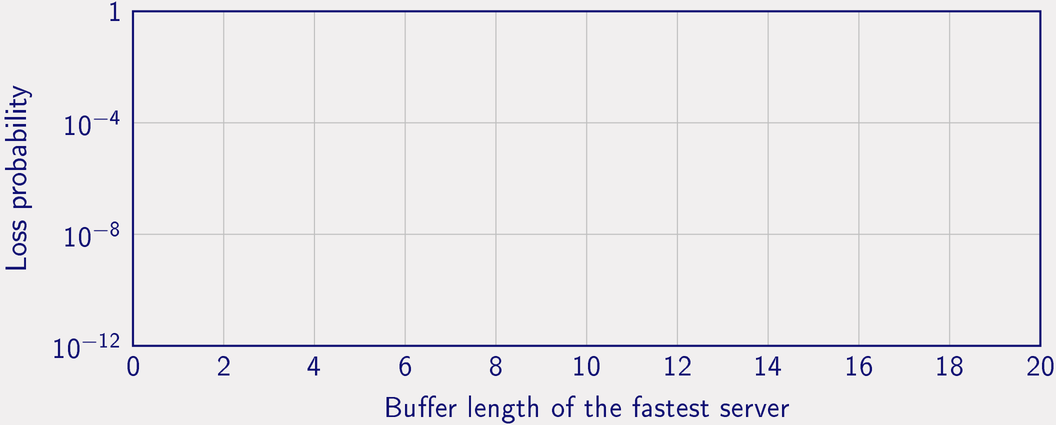
Heavy traffic: There is $\lambda^* > 0$ such that, for $\lambda \geq \lambda^*$, the loss probability is minimized when $\frac{\ell_1}{L} \simeq \frac{1}{2}$ and $\frac{\ell_2}{L} \simeq \frac{1}{2}$.



Monotonicity: The optimal buffer length of the fastest server, in terms of the loss probability, is decreasing with the arrival rate λ.

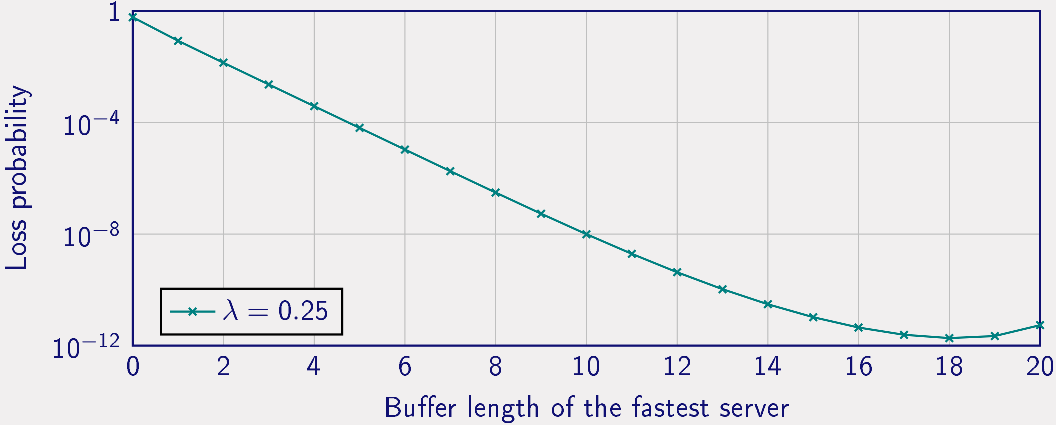
$L = 20$
$\mu_1 = 0.9$
$\mu_2 = 0.1$

Numerical results



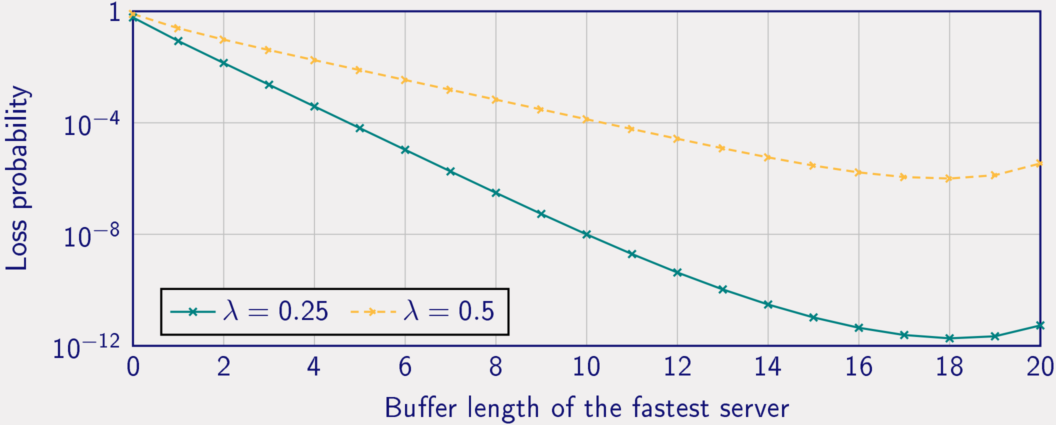
$L = 20$
$\mu_1 = 0.9$
$\mu_2 = 0.1$

Numerical results



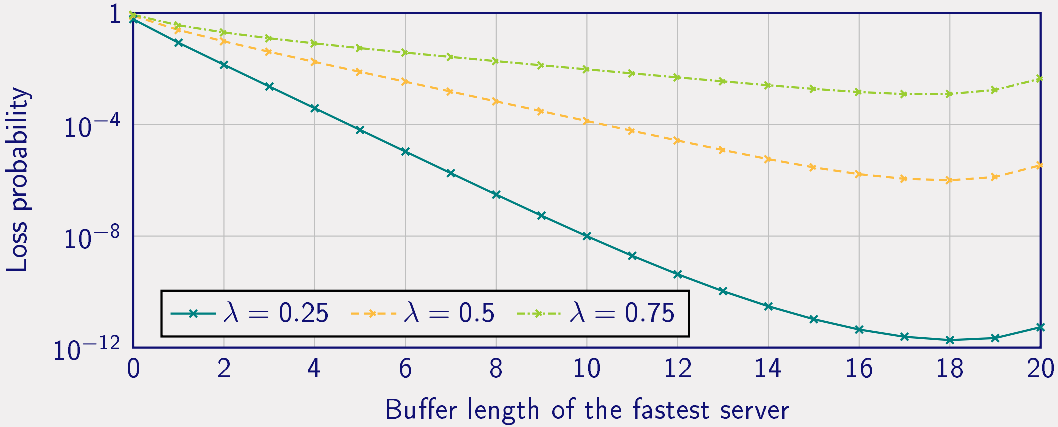
Numerical results

$L = 20$
$\mu_1 = 0.9$
$\mu_2 = 0.1$

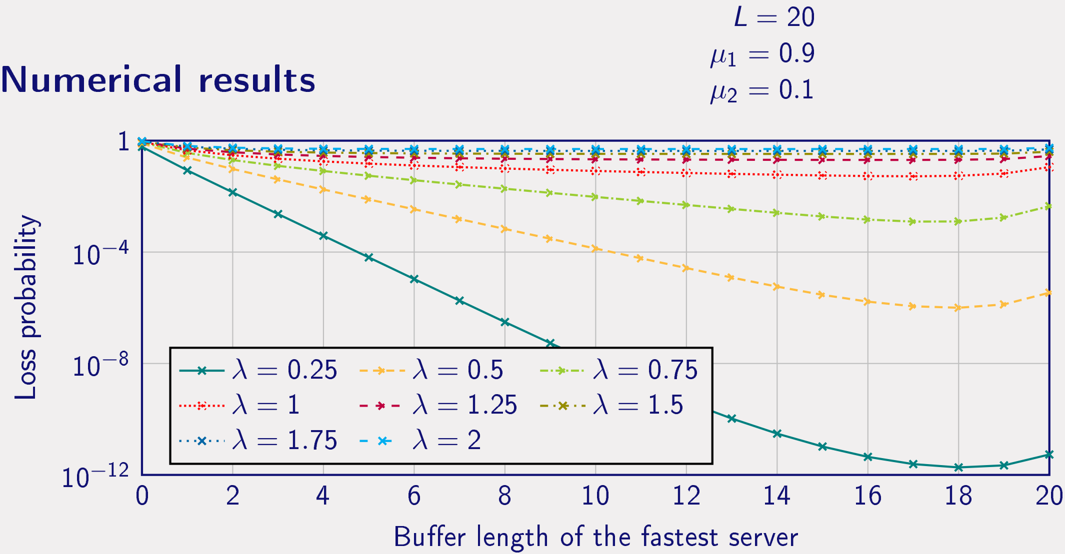


Numerical results

$$L = 20 \quad \mu_1 = 0.9 \quad \mu_2 = 0.1$$

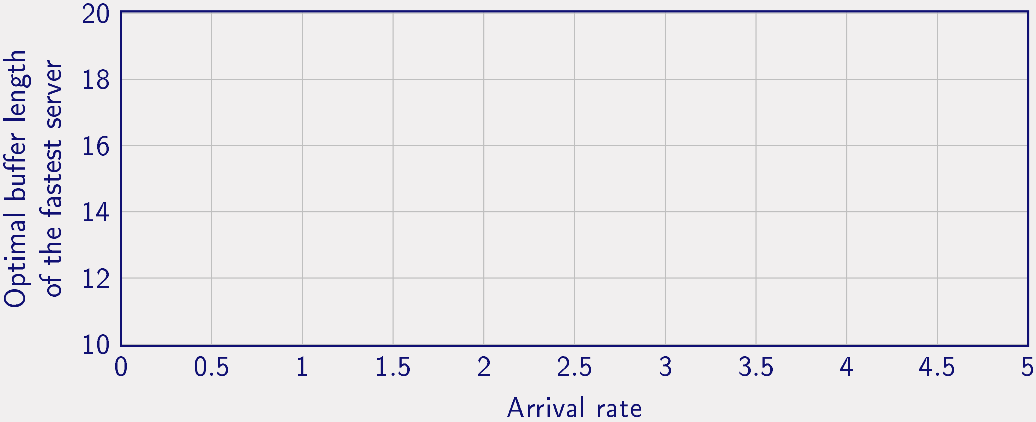


Numerical results



$L = 20$
$\mu_1 + \mu_2 = 1$

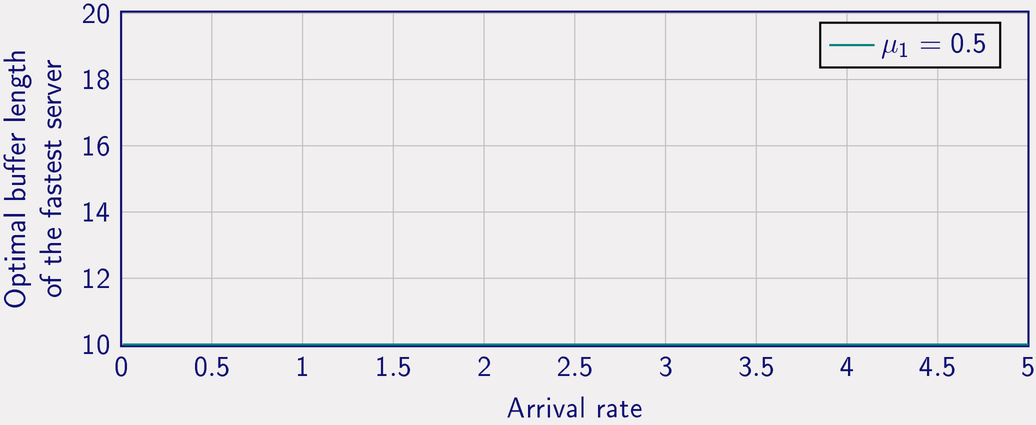
Numerical results



$L = 20$

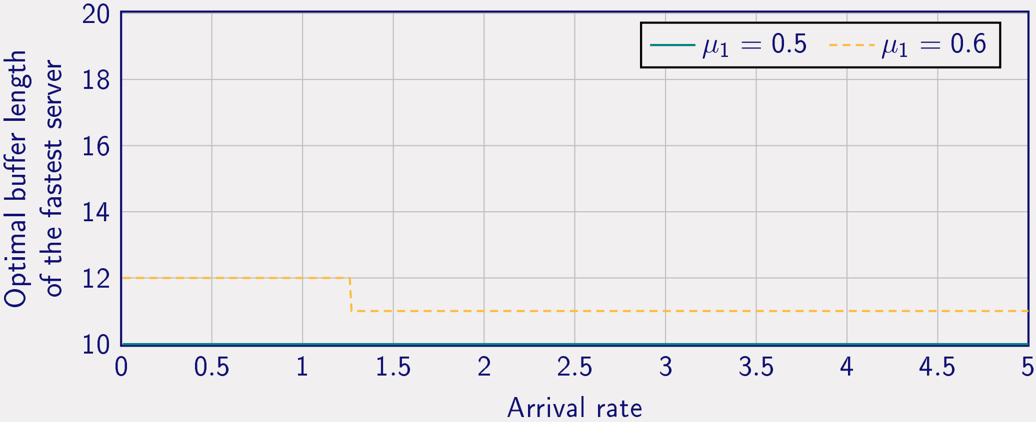
$\mu_1 + \mu_2 = 1$

Numerical results



$$L = 20$$
$$\mu_1 + \mu_2 = 1$$

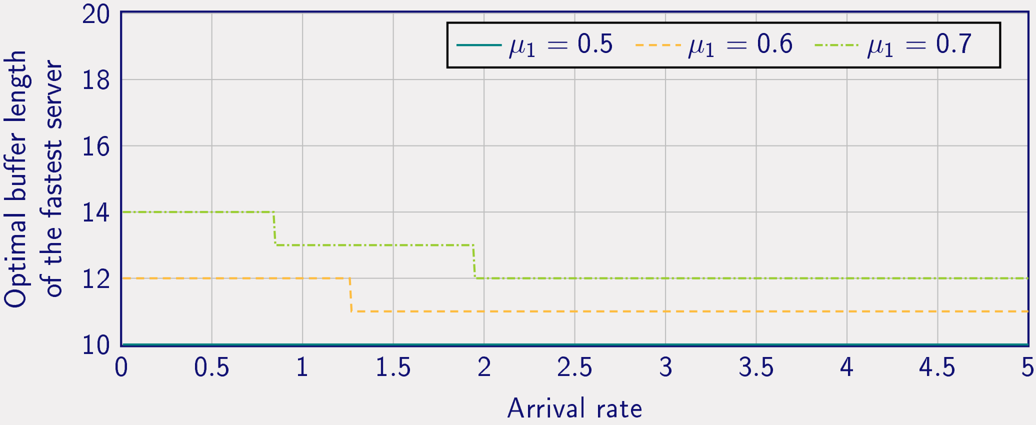
Numerical results



$L = 20$

$\mu_1 + \mu_2 = 1$

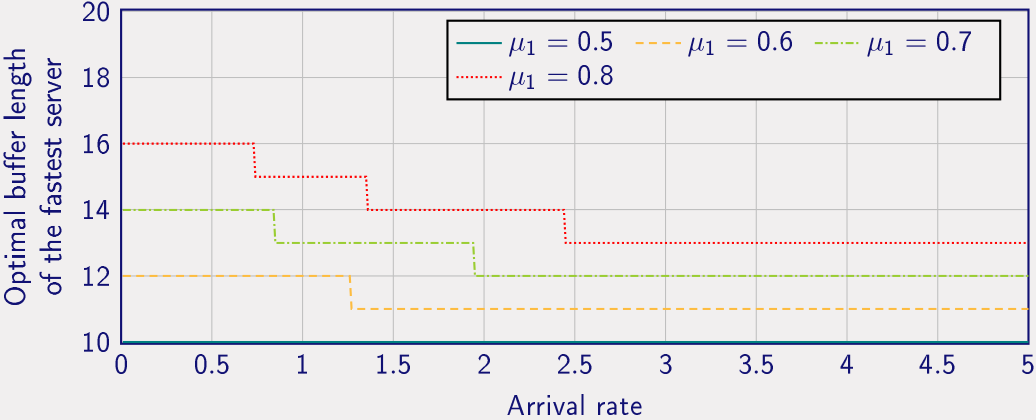
Numerical results



$L = 20$

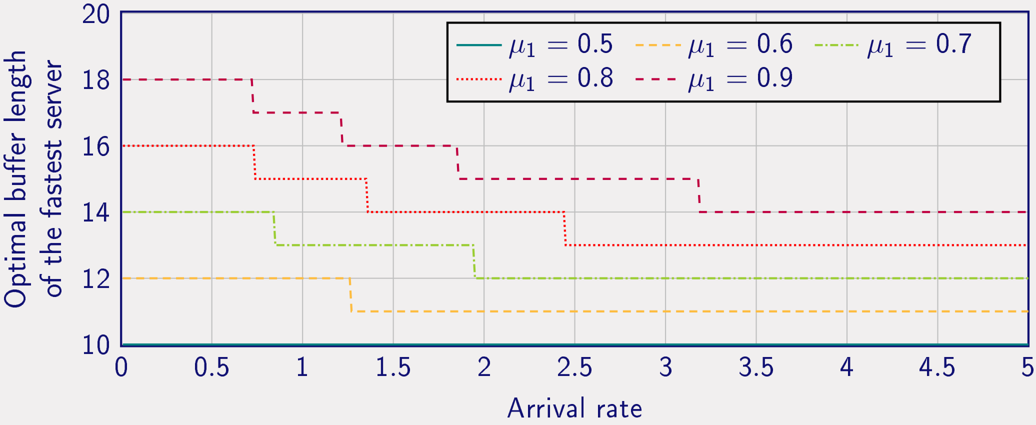
$\mu_1 + \mu_2 = 1$

Numerical results



$$L = 20$$
$$\mu_1 + \mu_2 = 1$$

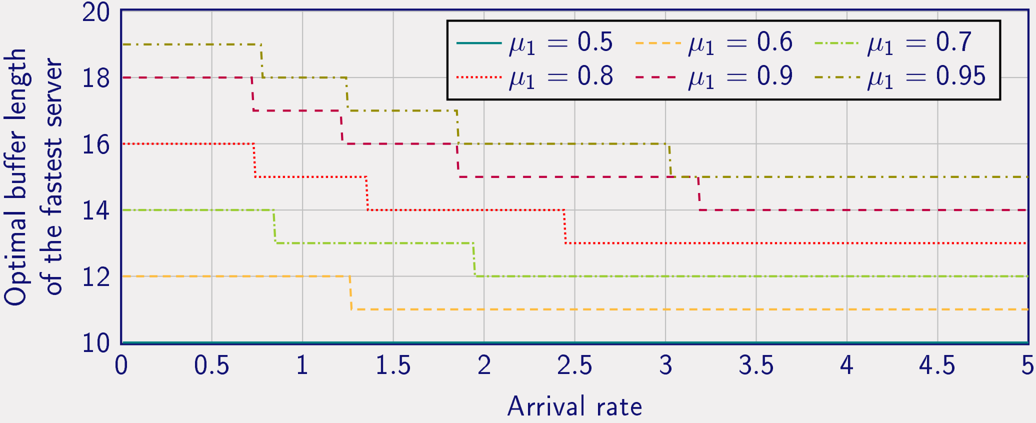
Numerical results



$L = 20$

$\mu_1 + \mu_2 = 1$

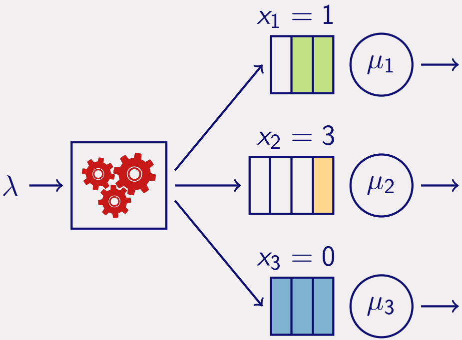
Numerical results



Conclusion

Contributions

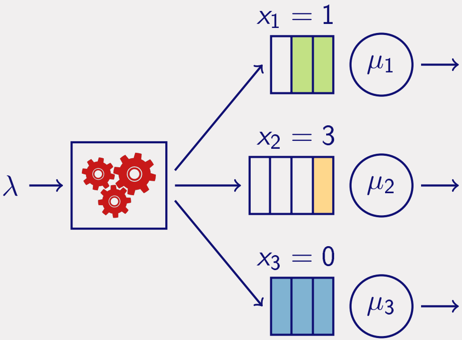
- Analysis of a randomized load-balancing algorithm in heterogeneous server clusters.
- Understanding of the optimal buffer lengths in terms of the loss probability.
- Developed new analytical methods.



Conclusion

Contributions

- Analysis of a randomized load-balancing algorithm in heterogeneous server clusters.
- Understanding of the optimal buffer lengths in terms of the loss probability.
- Developed new analytical methods.



Future works

- Optimize for other performance metrics.
- Generalize our results to other models that account for locality constraints.