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

Examples: cloud, manufacturing...
Heterogeneous server cluster

Model

- Dispatcher, $n$ servers, jobs
- Poisson arrival process with rate $\lambda$
Heterogeneous server cluster

Model
- Dispatcher, \( n \) servers, jobs
- Poisson arrival process with rate \( \lambda \)
- Service time exponential with rate \( \mu_i \), with \( \mu_1 > \mu_2 > \ldots > \mu_n \)
Heterogeneous server cluster

Model

- Dispatcher, $n$ servers, jobs
- Poisson arrival process with rate $\lambda$
- Service time exponential with rate $\mu_i$, 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 \( \lambda \)
- Service time exponential with rate \( \mu_i \), 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 $\lambda$
- Service time exponential with rate $\mu_i$, 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, ...

\[
\mu_1 x_1 = 1 \\
\mu_2 x_2 = 3 \\
\mu_3 x_3 = 0
\]

\( \lambda \)
Heterogeneous server cluster

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

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

$\lambda \rightarrow \begin{array}{ccc}
  x_1 &=& 1 \\
  x_2 &=& 3 \\
  x_3 &=& 0 \\

d\rightarrow \mu_1 \rightarrow \\
d\rightarrow \mu_2 \rightarrow \\
d\rightarrow \mu_3 \rightarrow 
\end{array}$
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.

\[
\pi(x) = \beta(\ell)(x_1 + \ldots + x_n) \prod_{i=1}^{n} (\mu_i / \lambda)
\]

**Loss probability**:

\[
1 - \beta(\ell) = \sum_{x_1 + \ldots + x_n \leq \ell} (x_1 + \ldots + x_n) \prod_{i=1}^{n} (\mu_i / \lambda)
\]
Stationary distribution

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

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

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

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

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

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

Loss probability:

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

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

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

Contributions:
- \( \hat{\text{Low-trac analysis:}} \quad \lambda \ll \mu_1 + \ldots + \mu_n \)
- \( \hat{\text{Heavy-trac analysis:}} \quad \lambda \gg \mu_1 + \ldots + \mu_n \)
- \( \hat{\text{Monotonicity result:}} \quad \lambda \text{ increases} \quad \mu_1 \xrightarrow{x_1 = 1} \mu_2 \xrightarrow{x_2 = 3} \mu_3 \xrightarrow{x_3 = 0} \lambda \)
Problem and contributions

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

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

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

\[ x_1 = 1 \quad \mu_1 \rightarrow \]
\[ x_2 = 3 \quad \mu_2 \rightarrow \]
\[ x_3 = 0 \quad \mu_3 \rightarrow \]
Problem and contributions

Question: Given $\lambda$, $\mu_1$, $\mu_2$, ..., $\mu_n$, and $L = \ell_1 + \ell_2 + \ldots + \ell_n$, how to choose $\ell_1$, $\ell_2$, ..., $\ell_n$ 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: $\lambda$ increases
Analytical results

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

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

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

\[ \mu_1 = 1 \quad \mu_2 = 3 \]
Analytical results

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

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

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

![Diagram](image-url)
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 $\frac{\ell_1}{L} \approx \frac{\mu_1}{\mu_1 + \mu_2}$ and $\frac{\ell_2}{L} \approx \frac{\mu_2}{\mu_1 + \mu_2}$.

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

**Monotonicity:** The optimal buffer length of the fastest server, in terms of the loss probability, is decreasing with the arrival rate $\lambda$. 
Numerical results

\[ L = 20 \]
\[ \mu_1 = 0.9 \]
\[ \mu_2 = 0.1 \]
Numerical results

\[ L = 20 \]
\[ \mu_1 = 0.9 \]
\[ \mu_2 = 0.1 \]

Loss probability

Buffer length of the fastest server
Numerical results

\[ L = 20 \]
\[ \mu_1 = 0.9 \]
\[ \mu_2 = 0.1 \]

![Graph showing loss probability vs. buffer length of the fastest server.](image)

\[ \lambda = 0.25 \]
\[ \lambda = 0.5 \]
Numerical results

\[ L = 20 \]
\[ \mu_1 = 0.9 \]
\[ \mu_2 = 0.1 \]

Loss probability

Buffer length of the fastest server

\[ \lambda = 0.25 \]
\[ \lambda = 0.5 \]
\[ \lambda = 0.75 \]
Numerical results

$L = 20$

$\mu_1 = 0.9$

$\mu_2 = 0.1$

Loss probability

Buffer length of the fastest server

$\lambda = 0.25$  $\lambda = 0.5$  $\lambda = 0.75$

$\lambda = 1$  $\lambda = 1.25$  $\lambda = 1.5$

$\lambda = 1.75$  $\lambda = 2$
Numerical results

$L = 20$

$\mu_1 + \mu_2 = 1$
Numerical results

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

Arrival rate

Optimal buffer length of the fastest server

- $$\mu_1 = 0.5$$
Numerical results

\[ L = 20 \]
\[ \mu_1 + \mu_2 = 1 \]

Arrival rate

Optimal buffer length of the fastest server

\[ \mu_1 = 0.5 \]  \[ \mu_1 = 0.6 \]
Numerical results

$L = 20$

$\mu_1 + \mu_2 = 1$

Arrival rate

Optimal buffer length of the fastest server

$\mu_1 = 0.5$  $\mu_1 = 0.6$  $\mu_1 = 0.7$
Numerical results

\[
L = 20 \\
\mu_1 + \mu_2 = 1
\]

![Graph showing the optimal buffer length of the fastest server as a function of the arrival rate for different values of \(\mu_1\). The graph includes lines for \(\mu_1 = 0.5\), \(\mu_1 = 0.6\), \(\mu_1 = 0.7\), and \(\mu_1 = 0.8\). The optimal buffer length \(L\) is set to 20.](image-url)
Numerical results

\[
\begin{align*}
L &= 20 \\
\mu_1 + \mu_2 &= 1
\end{align*}
\]

Arrival rate

Optimal buffer length of the fastest server

\[
\begin{align*}
\mu_1 &= 0.5 \\
\mu_1 &= 0.6 \\
\mu_1 &= 0.7 \\
\mu_1 &= 0.8 \\
\mu_1 &= 0.9
\end{align*}
\]
Numerical results

\[ L = 20 \]
\[ \mu_1 + \mu_2 = 1 \]

Arrival rate vs. Optimal buffer length of the fastest server for different values of \( \mu_1 \):
- \( \mu_1 = 0.5 \)
- \( \mu_1 = 0.6 \)
- \( \mu_1 = 0.7 \)
- \( \mu_1 = 0.8 \)
- \( \mu_1 = 0.9 \)
- \( \mu_1 = 0.95 \)
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.

\[\begin{align*}
\lambda & \rightarrow x_1 = 1 \\
\lambda & \rightarrow x_2 = 3 \\
\lambda & \rightarrow x_3 = 0 \\
\mu_1 \rightarrow & \\
\mu_2 \rightarrow & \\
\mu_3 \rightarrow & 
\end{align*}\]
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.