Heavy-Traffic Universality of Redundancy Systems with Data Locality Constraints

YEQT 2021

Ellen Cardinaels, Sem Borst, Johan van Leeuwaarden
Heavy-Traffic Universality

of Redundancy Systems

with Data Locality Constraints
Heavy-Traffic Universality (result)

of Redundancy Systems (tool: product-form expressions)

with Data Locality Constraints (motivation)
Motivation

Question: How to model such a system?
Motivation

**Question:** How to *model* such a system?
Motivation
Supermarket Model
Motivation
Supermarket Model
Motivation

Data Locality Constraints

Question: How to assign jobs to servers?
Motivation
Data Locality Constraints

arrival process

job types

.servers

MOVIE I MOVIE II MOVIE II MOVIE III

N = 2 servers

$N \lambda$

$p_R$ $p_B$ $p_Y$

$\mu_1$ $\mu_2$

How to assign jobs to servers?
Motivation
Data Locality Constraints

JSQ($d$) *
speed aware JSQ/JIQ †
Redundancy
...

Question: How to assign jobs to servers?

$^*$ Rutten and Mukherjee, ’20 (large $N$ reg.)
$^†$ Weng et al., ’20 (large $N$ reg.)
Redundancy Systems

Assignment policy

redundancy cancel-on-completion (c.o.c.)

Question: How to analyze such a system?
Redundancy Systems
Assignment policy

redundancy cancel-on-completion (c.o.c.)

How to analyze such a system?
Redundancy Systems
Assignment policy

redundancy cancel-on-completion (c.o.c.)

fictitious central queue

scan for compatible jobs

\[ N \lambda \rightarrow \]

Question: How to analyze such a system?
Redundancy Systems
Assignment policy

redundancy cancel-on-completion (c.o.c.)

{\[ N \lambda \rightarrow \text{fictitious central queue} \rightarrow B_2 \rightarrow R_1 \rightarrow B_1 \rightarrow \text{scan for compatible jobs} \]}

Question: How to analyze such a system?
Redundancy Systems
Assignment policy

redundancy cancel-on-completion (c.o.c.)

How to analyze such a system?
Redundancy Systems

Assignment policy

redundancy cancel-on-completion (c.o.c.)


Question: How to analyze such a system?
Redundancy Systems

Product-form expression

\[ \pi(c_1, c_2, \ldots, c_n) = C \prod_{i=1}^{n} \frac{N \lambda p_{c_i}}{\mu(c_1, \ldots, c_i)} \]

Gardner et al., ’16; Ayesta et al., ’18
Redundancy Systems

Product-form expression

\[ \pi(c_1, c_2, \ldots, c_n) = C \prod_{i=1}^{n} \frac{N\lambda p_{c_i}}{\mu(c_1, \ldots, c_i)} \]

\[ \pi(Y, B, R, B) = C \left( \frac{N\lambda p_Y}{\mu_2} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right) \left( \frac{N\lambda p_R}{\mu_1 + \mu_2} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right) \]

**Question:** How to analyze this system using PF expressions?

‡Gardner et al., ’16; Ayesta et al., ’18
Redundancy Systems

Product-form expression

Difﬁcult: aggregate states according to the number of jobs.

\[ \mathbb{P}\{3 \text{ jobs in the system}\} = \pi(Y, B, R) + \pi(R, R, R) + \pi(B, R, R) + \pi(R, R, B) + \pi(R, Y, B) + \ldots \]
Redundancy Systems

Product-form expression

**Difficult:** aggregate states according to the number of jobs.

\[ \mathbb{P}\{3 \text{ jobs in the system}\} \]
\[ = \pi(Y, B, R) + \pi(R, R, R) + \pi(B, R, R) + \pi(R, R, B) + \pi(R, Y, B) + \ldots \]

\( \pi(Y, B, R) = C \left( \frac{N\lambda p_Y}{\mu_2} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right) \left( \frac{N\lambda p_R}{\mu_1 + \mu_2} \right) \)

\( \pi(R, R, R) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_R}{\mu_1} \right) \)

\[ \text{different job types} \]
Redundancy Systems

Product-form expression

Difficult: aggregate states according to the number of jobs.

\[ P\{3 \text{ jobs in the system}\} = \pi(Y, B, R) + \pi(R, R, R) + \pi(B, R, R) + \pi(R, R, B) + \pi(R, Y, B) + \ldots \]

\[
\begin{align*}
\pi(Y, B, R) &= C \left( \frac{N \lambda p_Y}{\mu_2} \right) \left( \frac{N \lambda p_B}{\mu_1 + \mu_2} \right) \left( \frac{N \lambda p_R}{\mu_1 + \mu_2} \right) \\
\pi(R, R, R) &= C \left( \frac{N \lambda p_R}{\mu_1} \right) \left( \frac{N \lambda p_R}{\mu_1} \right) \left( \frac{N \lambda p_R}{\mu_1} \right) \quad \text{different job types} \\
\pi(B, R, R) &= C \left( \frac{N \lambda p_B}{\mu_1 + \mu_2} \right) \left( \frac{N \lambda p_R}{\mu_1 + \mu_2} \right) \left( \frac{N \lambda p_R}{\mu_1 + \mu_2} \right) \quad \text{order of the jobs} \\
\pi(R, R, B) &= C \left( \frac{N \lambda p_R}{\mu_1} \right) \left( \frac{N \lambda p_R}{\mu_1} \right) \left( \frac{N \lambda p_B}{\mu_1 + \mu_2} \right)
\end{align*}
\]
Redundancy Systems

Product-form expression

Easier: aggregate states according to the job types and their order of first occurrences.

E.g., $R$ and $B$:

$$\pi(R, B) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)$$
Redundancy Systems
Product-form expression

**Easier:** aggregate states according to the job types and their order of first occurrences.

E.g., $R$ and $B$:

$$\pi(R, B) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)$$

$$\pi(R, \{R\}^m, B) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_R}{\mu_1} \right)^m \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)$$

$m, n_R, n_B \in \mathbb{N}$
Redundancy Systems
Product-form expression

**Easier:** aggregate states according to the job types and their order of first occurrences.

E.g., $R$ and $B$:

$$\pi(R, B) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)$$

$$\pi(R, \{R\}^m, B) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_R}{\mu_1} \right)^m \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)$$

$$\pi(R, \{R\}^m, B, \{R\}^{n_R}, \{B\}^{n_B}) = C \left( \frac{N\lambda p_R}{\mu_1} \right) \left( \frac{N\lambda p_R}{\mu_1 + \mu_2} \right)^m \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)^{n_R} \left( \frac{N\lambda p_B}{\mu_1 + \mu_2} \right)^{n_B}$$

$m, n_R, n_B \in \mathbb{N}$
Redundancy Systems
Product-form expression

Probability generating function:

\[
\mathbb{E} \left[ Z_R^Q Z_B^Q Z_Y^Q \right] = C + C \sum_{m=1}^{3} \sum_{s \in S_m} \prod_{j=1}^{m} \left( \frac{N \lambda p_{S_j} s_j}{\mu(S_1, \ldots, S_j)} \right) \left( 1 - \frac{N \lambda}{\mu(S_1, \ldots, S_j)} \sum_{i=1}^{j} p_{S_i} s_i \right)^{-1}
\]

\(|Z_R|, |Z_B|, |Z_Y| \leq 1\).
Heavy-Traffic Universality

Result

\[ \left(1 - \frac{\lambda}{\mu}\right) (Q_R, Q_B, Q_Y) \xrightarrow{d} \text{Exp}(1) (p_R, p_B, p_Y) \]

as \( \lambda \uparrow \mu \).
Heavy-Traffic Universality

Result

\[ \left(1 - \frac{\lambda}{\mu}\right) (Q_R, Q_B, Q_Y) \xrightarrow{d} \text{Exp}(1) (\rho_R, \rho_B, \rho_Y) \]

as \( \lambda \uparrow \mu \).
Heavy-Traffic Universality

Result

Stability condition for the PF:

\[ \lambda N \rho_R \ < \ \mu_1 \]
\[ \lambda N \rho_Y \ < \ \mu_2 \]
\[ \lambda N \rho_B \ < \ \mu_1 + \mu_2 \]
\[ \lambda N(p_R + p_B) \ < \ \mu_1 + \mu_2 = \mu_N \]
\[ \lambda N(p_B + p_Y) \ < \ \mu_1 + \mu_2 = \mu_N \]
\[ \lambda \ < \ \mu \]

HT result: No local bottlenecks allowed.
Concluding remarks

General result for all systems with no local bottlenecks.

Heavy-Traffic Universality (result) of Redundancy Systems (tool: product-form expressions) with Data Locality Constraints (motivation)


Extensions: Relaxed condition & Redundancy cancel-on-start
Concluding remarks

General result for all systems with no local bottlenecks.

Heavy-Traffic Universality (result) of Redundancy Systems (tool: product-form expressions) with Data Locality Constraints (motivation)

Extensions: Relaxed condition & Redundancy cancel-on-start

Thank you!
Heavy-Traffic Universality
Systems without local bottlenecks

Stability conditions for PF expressions:

\[
N \lambda \sum_{S \in T} p_S < \sum_{n \in \bigcup_{S \in T} S} \mu_n
\]

Condition for no local bottlenecks:

\[
\sum_{S \in T} p_S < \frac{1}{N \mu \sum_{n \in \bigcup_{S \in T} S} \mu_n}
\]
Heavy-Traffic Universality

Systems without local bottlenecks

General result for *all* systems with no local bottlenecks.

\[ N \lambda \]

arrival process

\[ p_R, p_R, p_R, p_B \]

job types

\[ \rho_R = \frac{1 - \epsilon}{N} \]

\[ \rho_B = \epsilon \]

N servers
Heavy-Traffic Universality
Systems with local bottlenecks

(partial) state space collapse and a truncated system:

\[
\left( \left( 1 - \frac{\lambda}{\lambda^*} \right) (Q_s)_{s \in \mathcal{T}^*}, (Q_s)_{s \notin \mathcal{T}^*} \right) \xrightarrow{d} \left( \text{Exp}(1) \left( \frac{p_s}{\rho_{\mathcal{T}^*}} \right)_{s \in \mathcal{T}^*}, (Q_s^*)_{s \notin \mathcal{T}^*} \right)
\]

as \( \lambda \uparrow \lambda^* \)