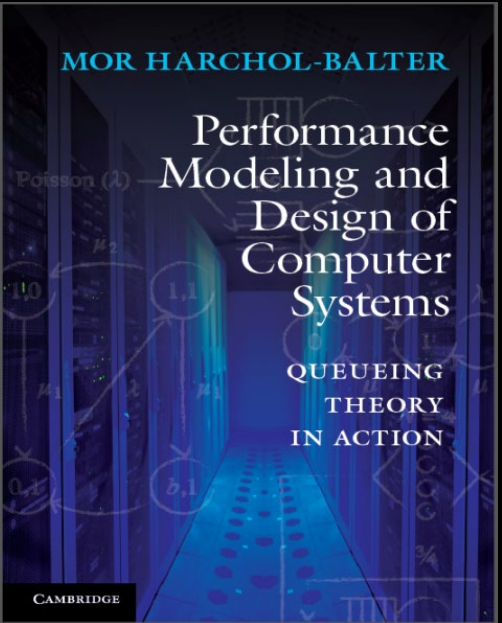
Queueing with Redundant Requests

Mor Harchol-Balter

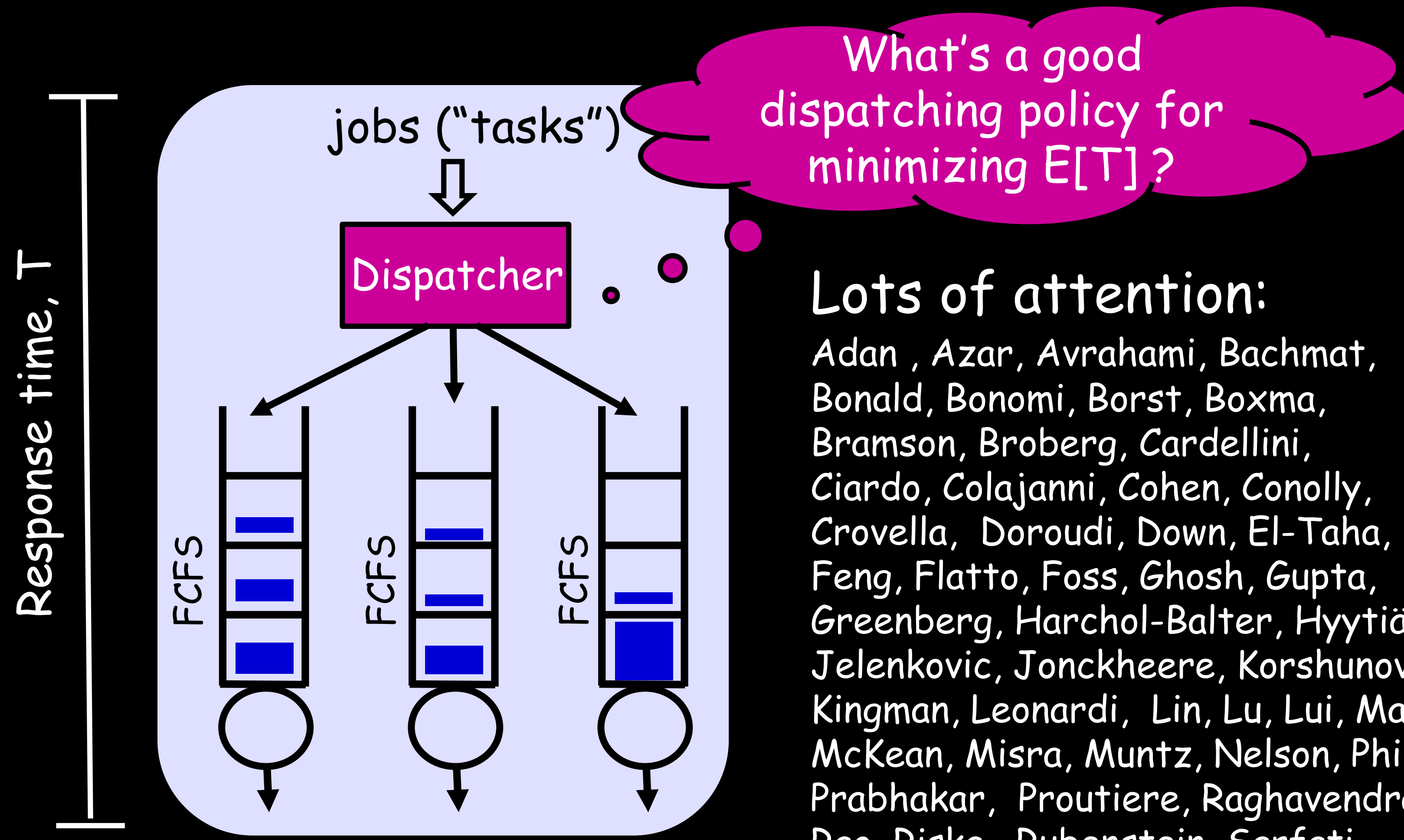
Professor of Computer Science, CMU

Joint with:
Kristy Gardner
Alan Scheller-Wolf



Faculty opening
@ CMU
in queueing theory!

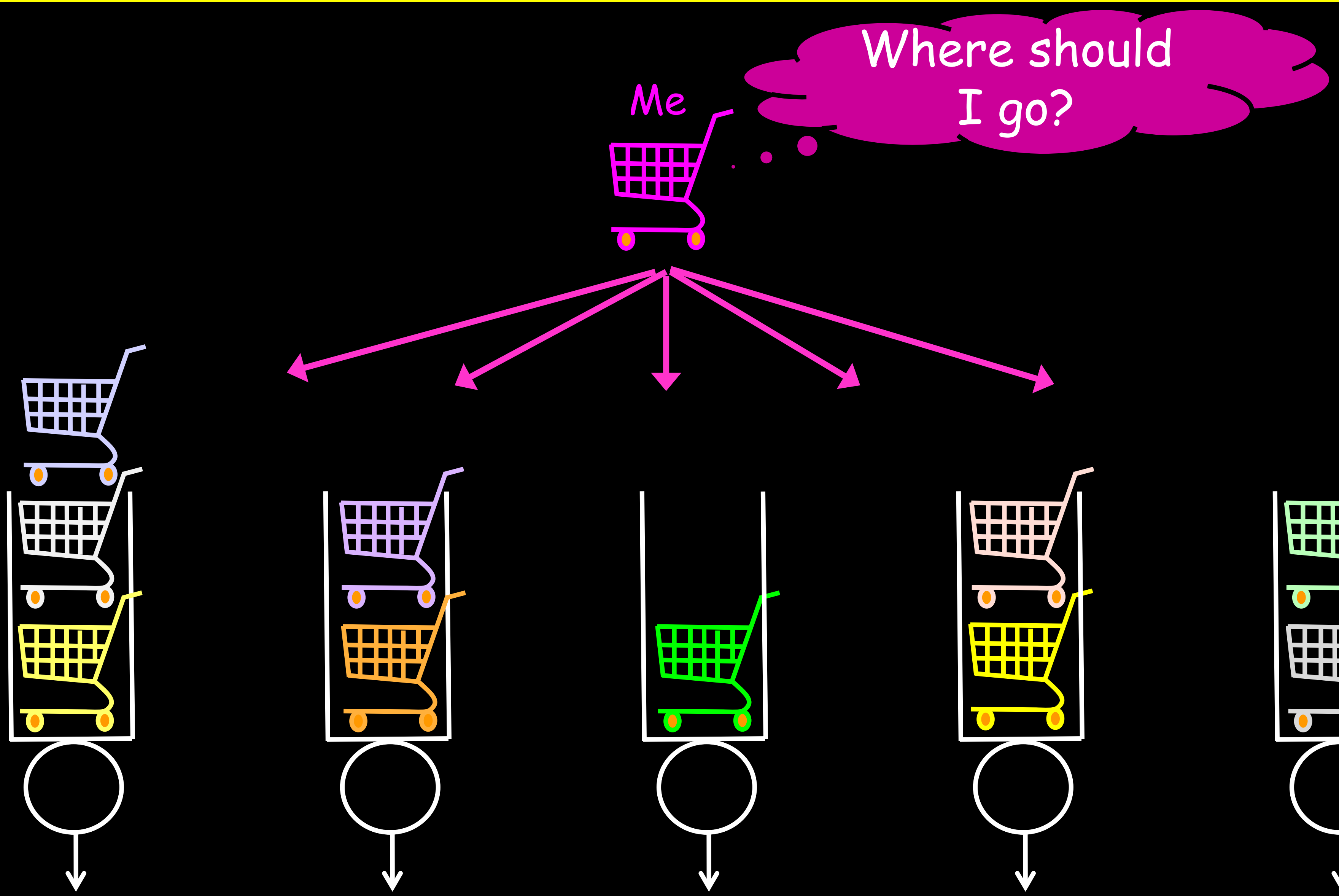
Task Assignment Problem is very Old



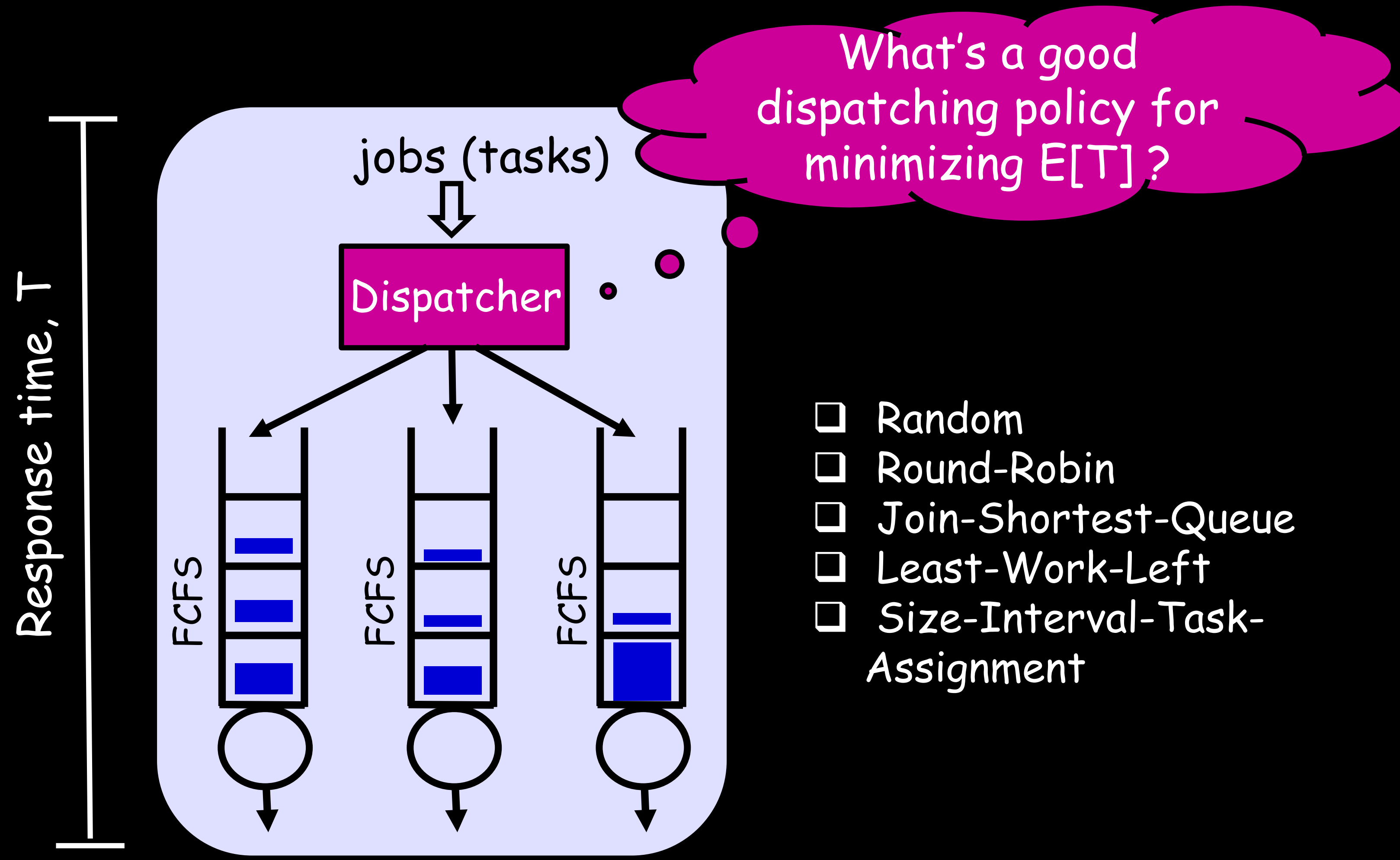
Lots of attention:

Adan , Azar, Avrahami, Bachmat, Bonald, Bonomi, Borst, Boxma, Bramson, Broberg, Cardellini, Ciardo, Colajanni, Cohen, Conolly, Crovella, Doroudi, Down, El-Taha, Feng, Flatto, Foss, Ghosh, Gupta, Greenberg, Harchol-Balter, Hyytiä, Jelenkovic, Jonckheere, Korshunov, Kingman, Leonardi, Lin, Lu, Lui, Maddah, McKean, Misra, Muntz, Nelson, Philips, Prabhakar, Proutiere, Raghavendra, Raz, Rao, Riska, Rubenstein, Sarfati, Schroeder, Smirni, Stanford, Squillante, Tari, Towsley, Tsitsiklis, van der Wal, Virtamo, Wessels, Whitt, Xia, Yao, Yechialli, Young, Yu, Zhang, Zijm, …

Same problem in Supermarket



Job size distribution, X, plays big role



- Random
- Round-Robin
- Join-Shortest-Queue
- Least-Work-Left
- Size-Interval-Task-Assignment

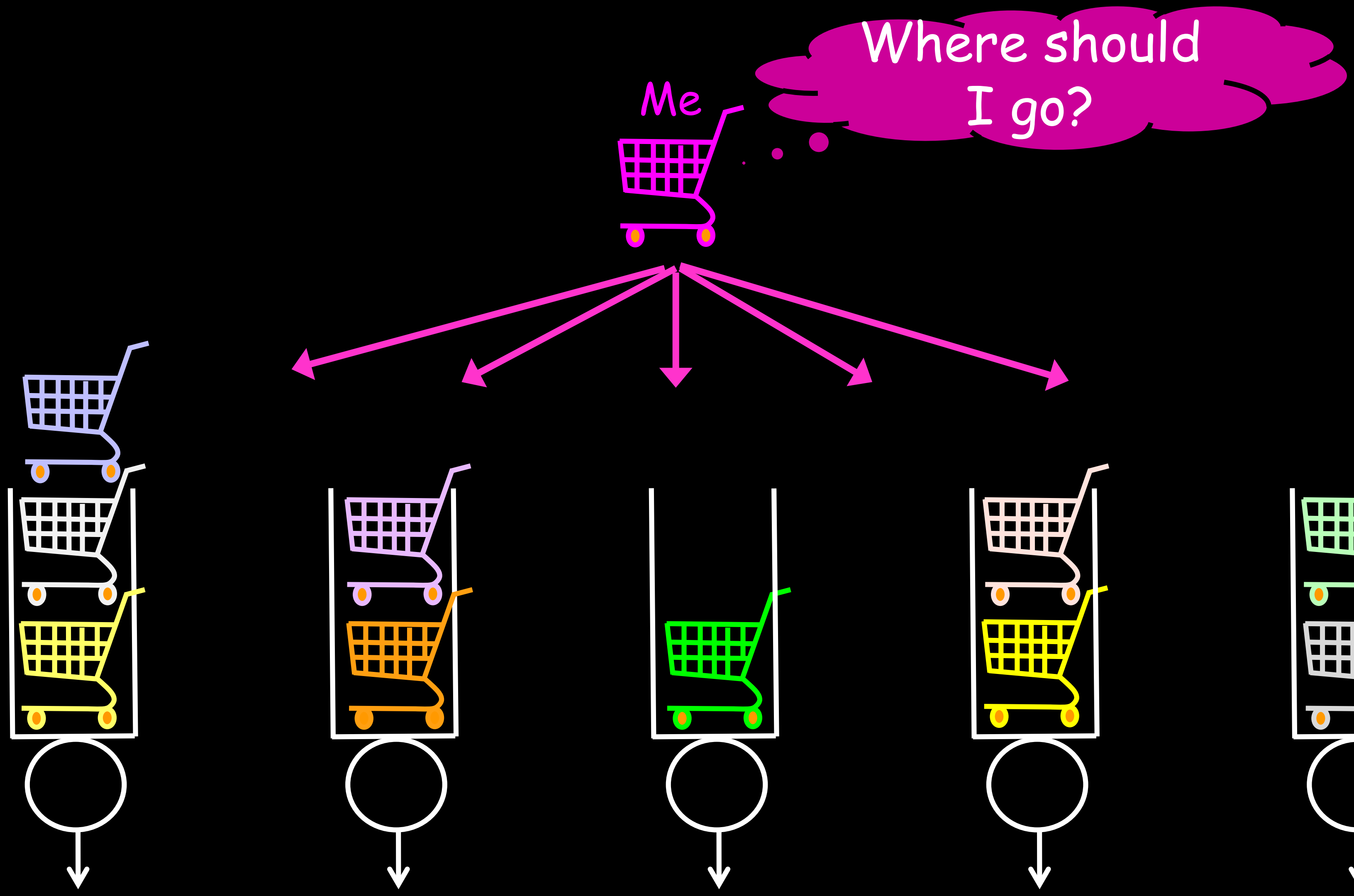
Knowing X is not enough

Suppose you know job size distribution, X, …

And you even know **exact job sizes** …

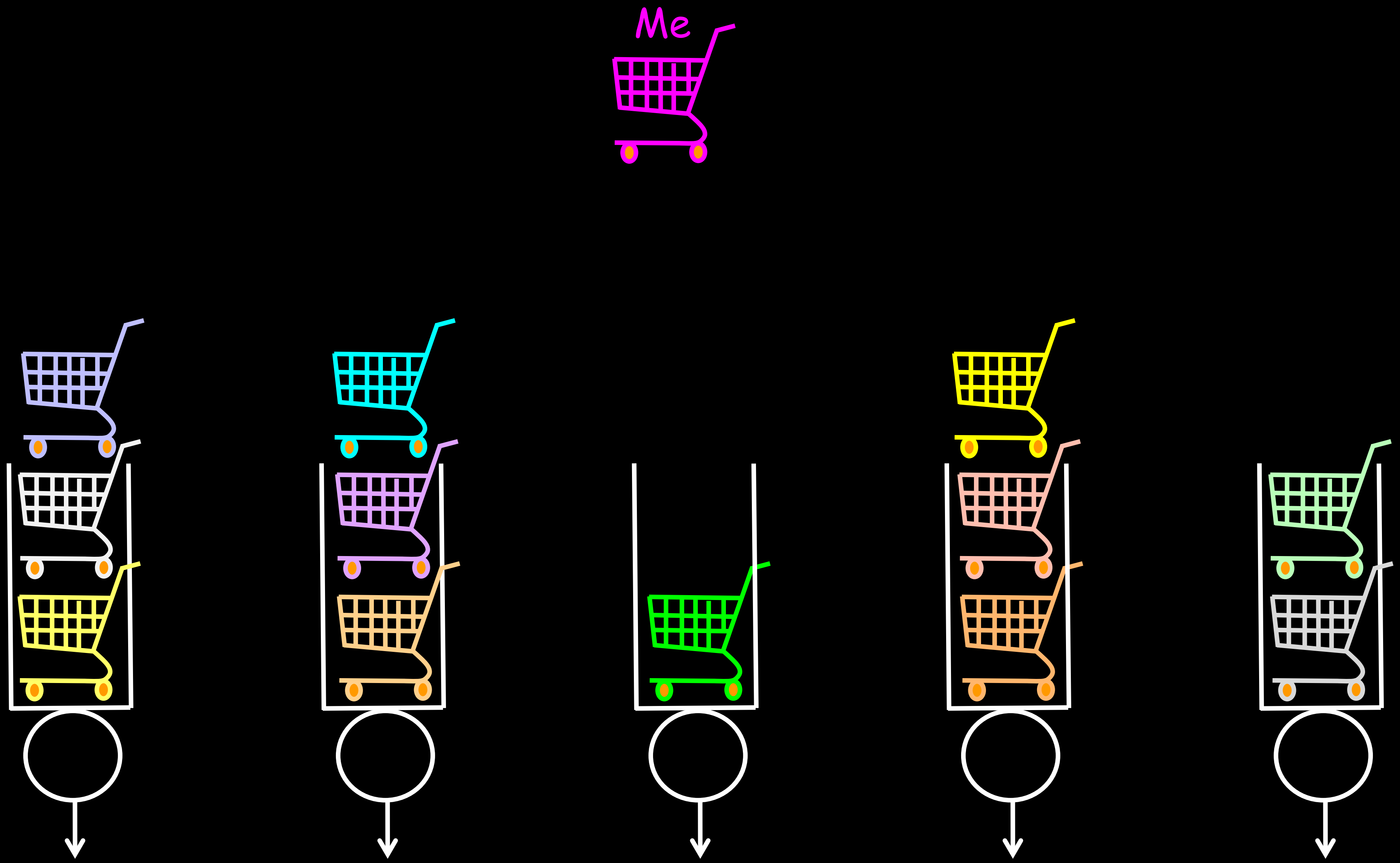
Claim: Still insufficient for good task assignment.

Supermarket

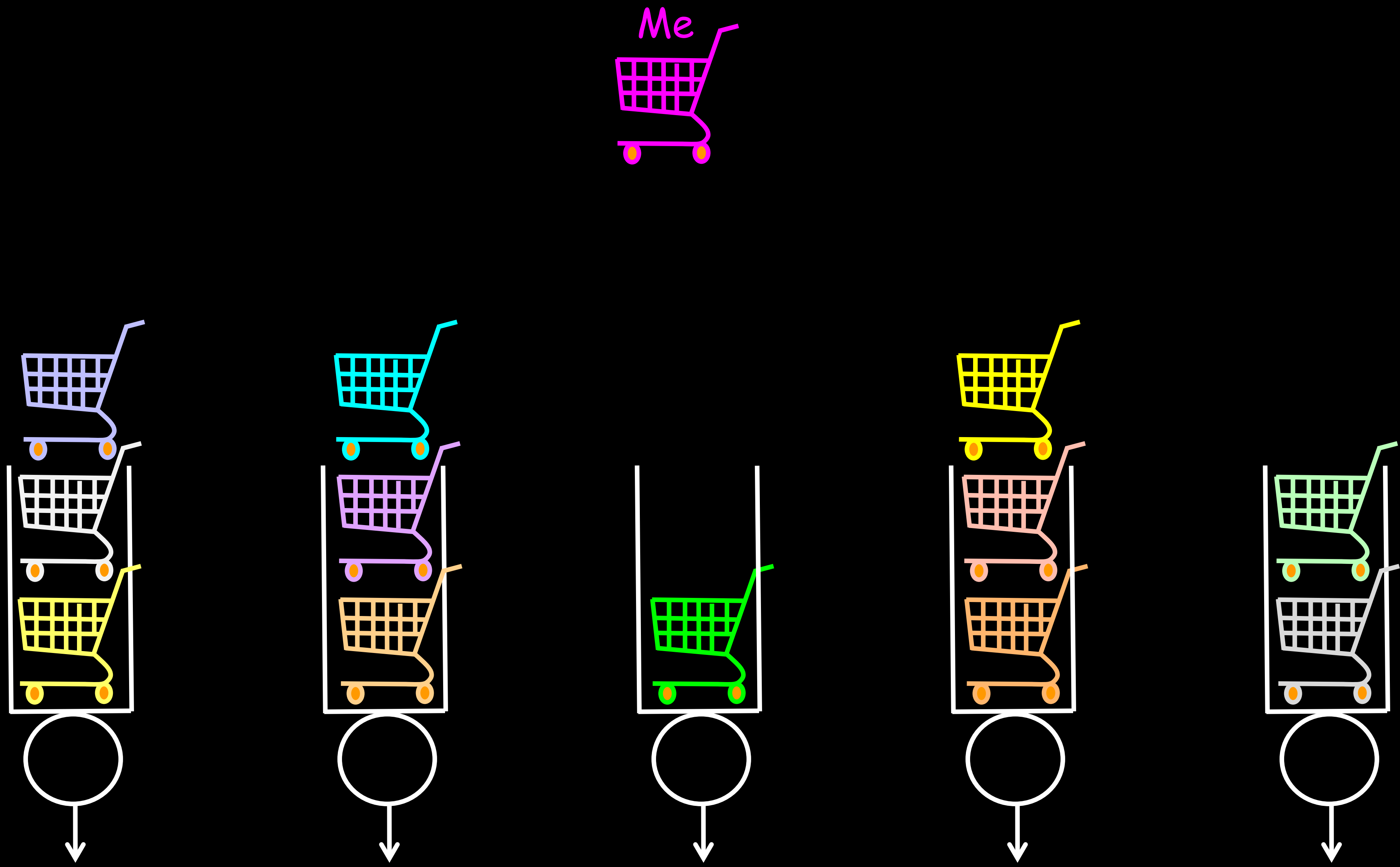


Problem: What you see ≠ What you get

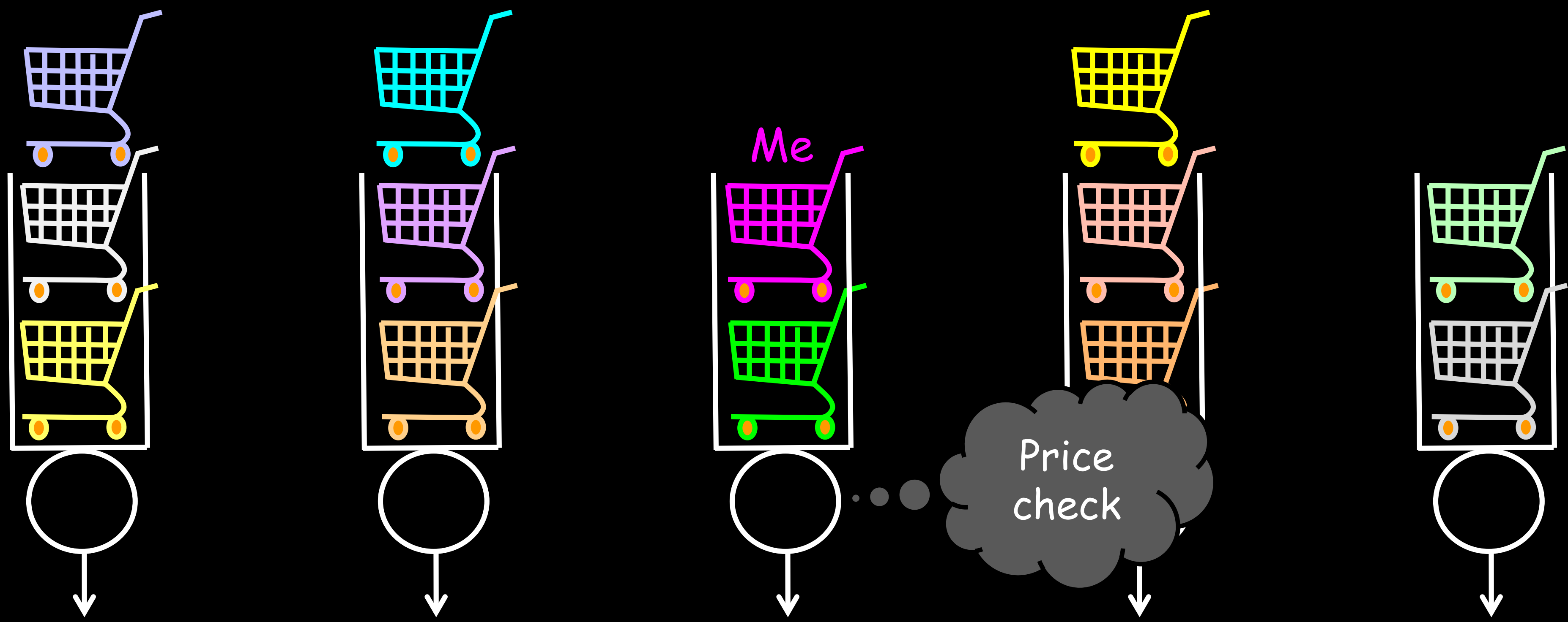
My Supermarket Experience



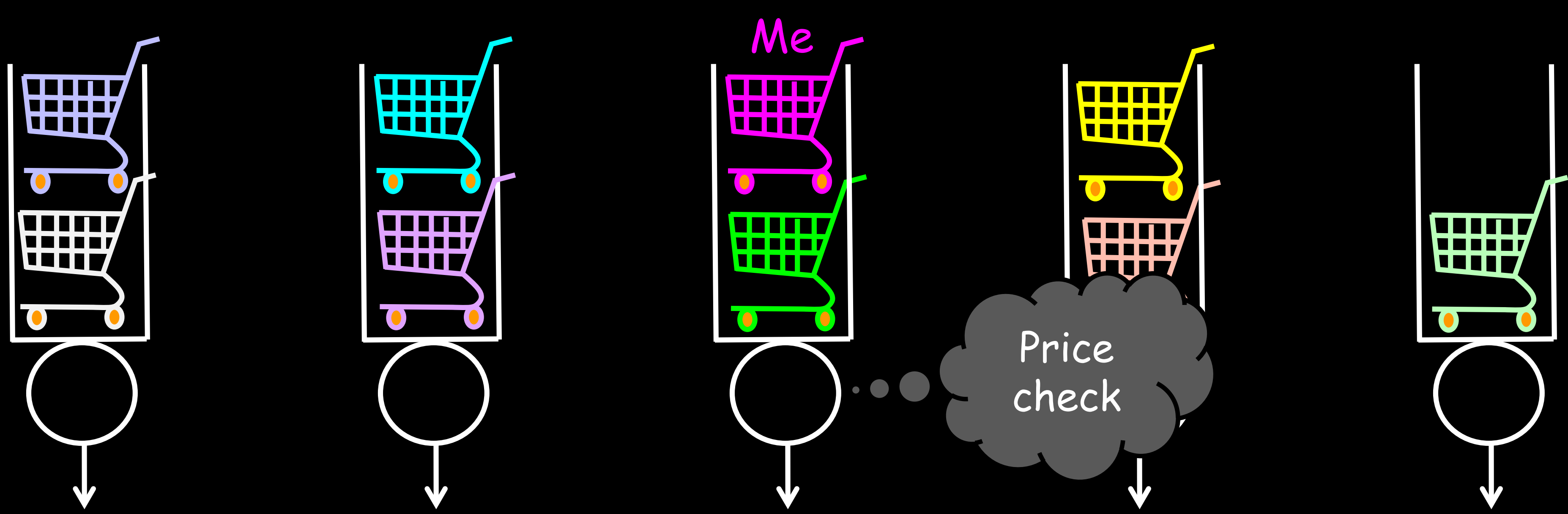
My Supermarket Experience



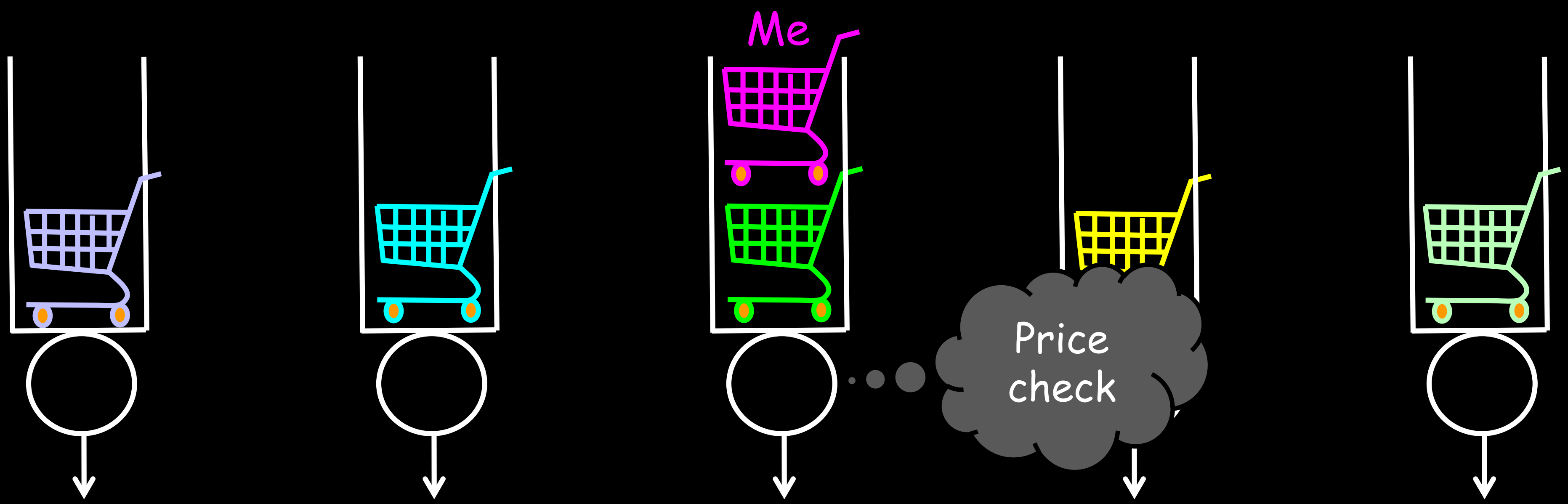
My Supermarket Experience



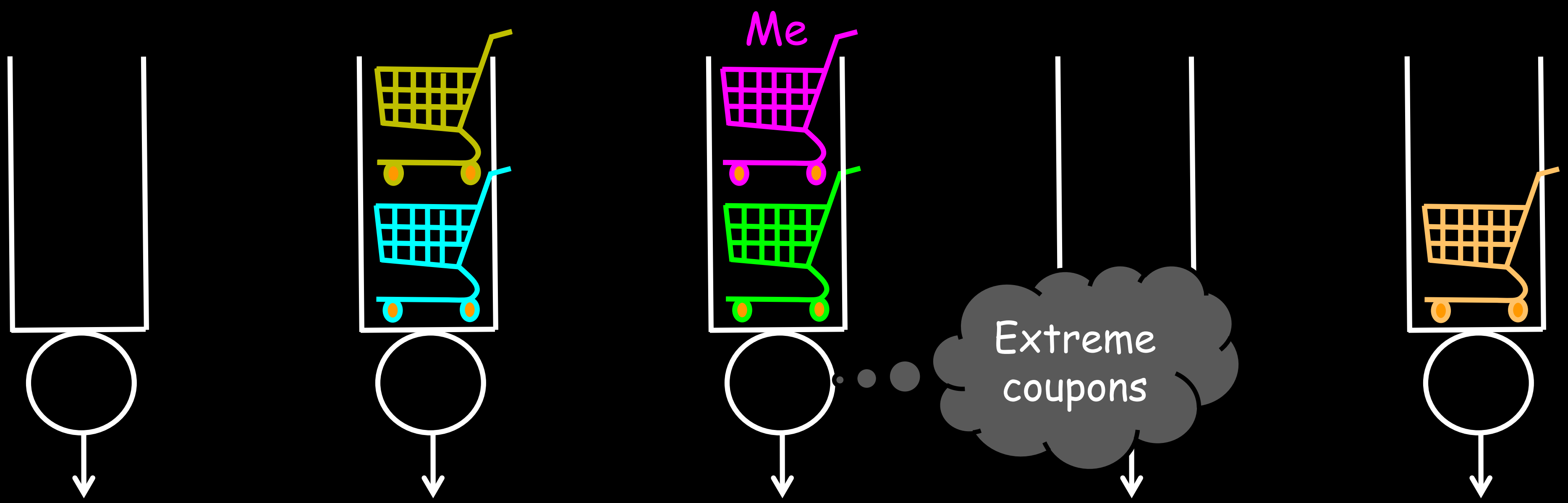
My Supermarket Experience



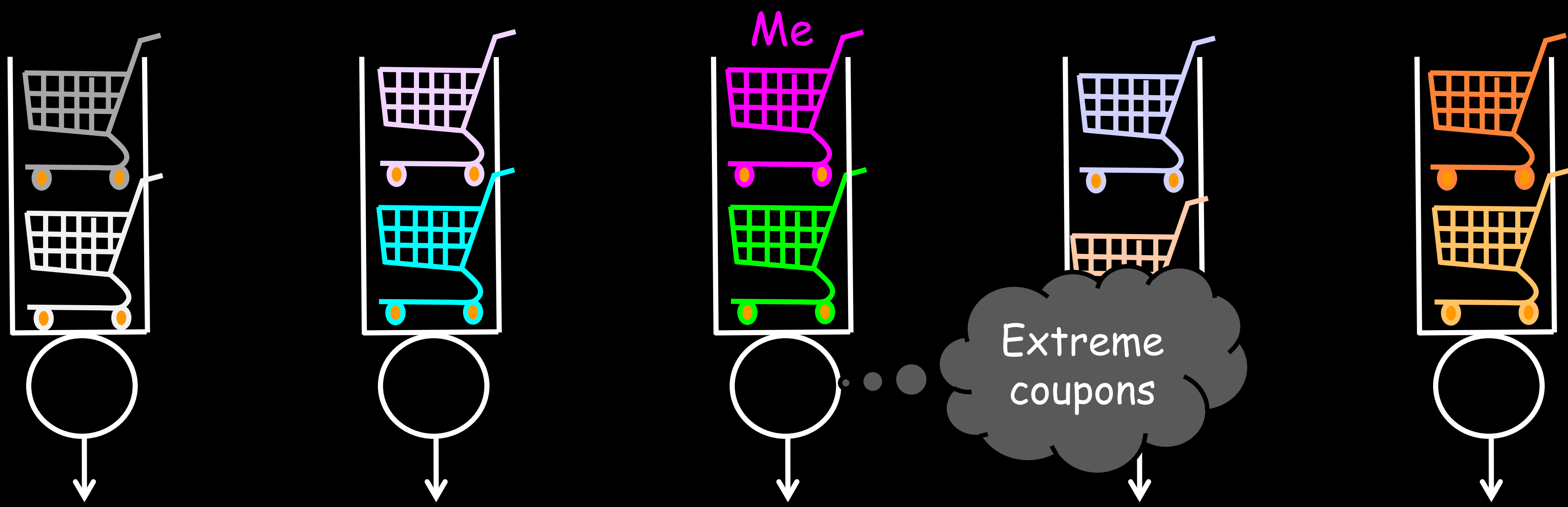
My Supermarket Experience



My Supermarket Experience

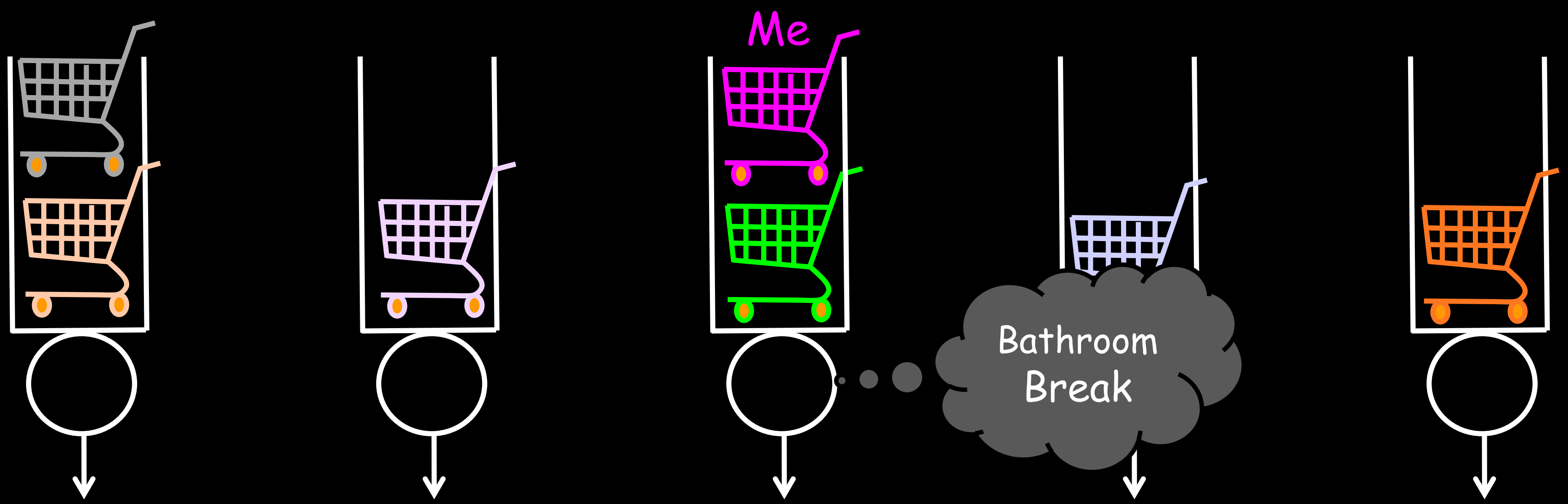


My Supermarket Experience



My Supermarket Experience

Problem: Server-side variability



Server-side variability can dominate a job's runtime

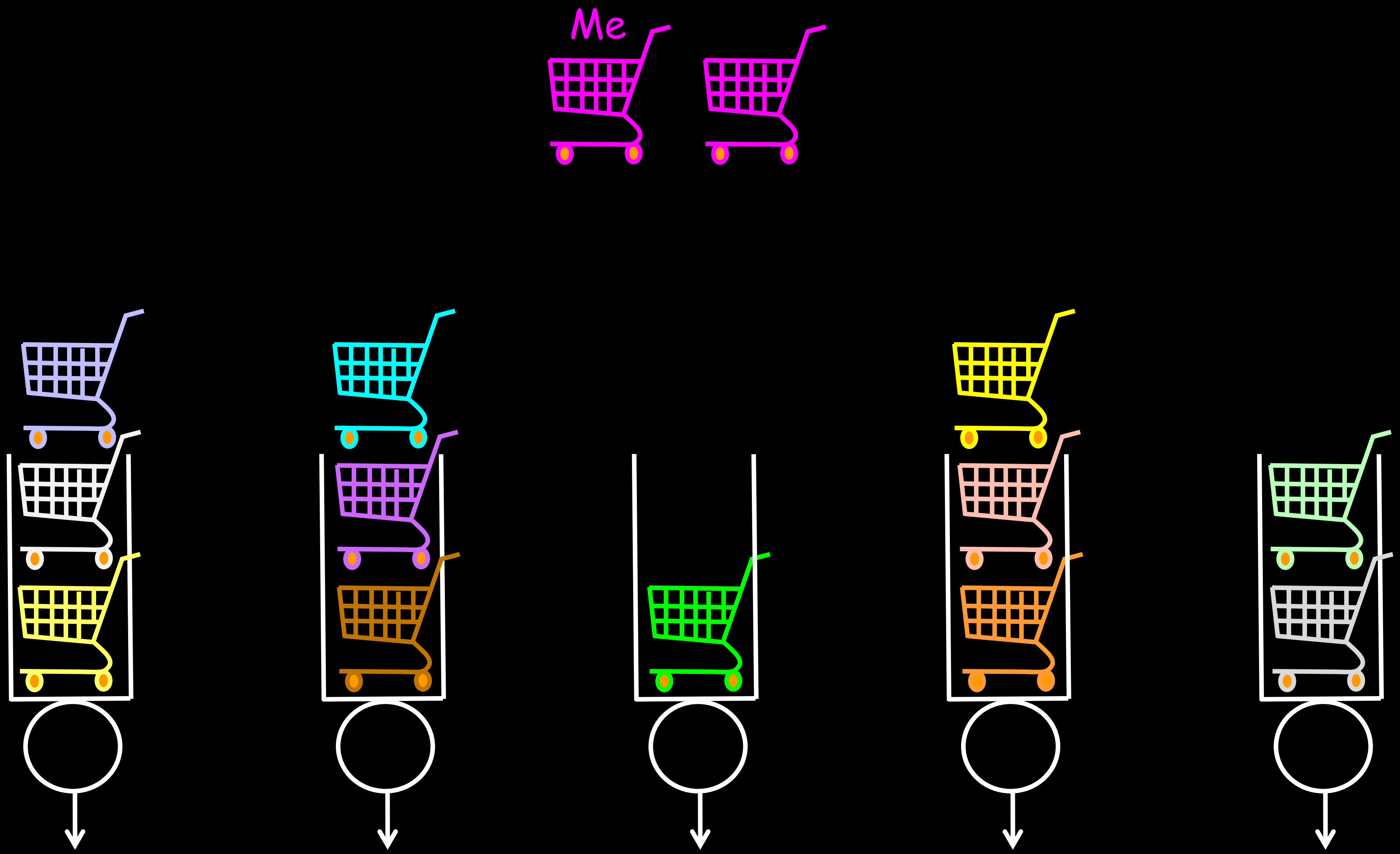
… can be more relevant than inherent job size, X

Example: Webpage download

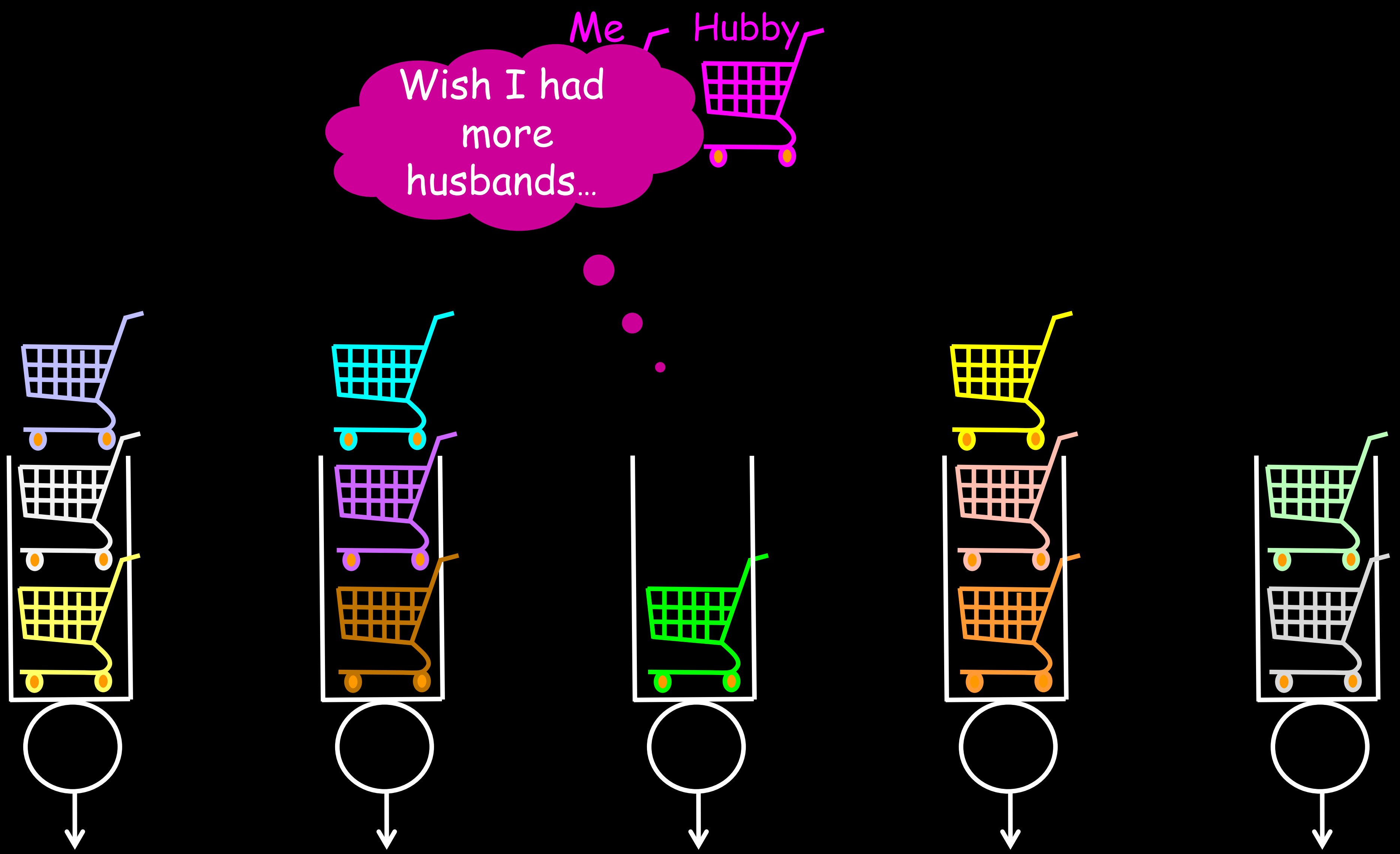
- Typically very fast: $X \approx 10$ ms
- But can be 1 s if server is slow

When server-side variability dominates, need new task assignment policy

Redundancy!



Redundancy!



Redundancy in Computer Systems

NEW **Computer Systems Redundancy Research:**

- Berkeley Dolly System [Ananthanarayanan et al. 2012]
- Google "Tail at Scale" 2013 [Dean, Barroso 2013]
- Berkeley Sparrow paper 2013 [Ousterhout et al. 2013]
- DNS and Database query systems 2013 [Vulimiri et al. 2013]
- GRASS 2014 [Ananthanarayanan et al. 2014]
- Hopper 2015 [Ren et al. 2015]

Computer Systems ≠ SuperMarket

Same job runs on multiple servers at once.
Wait for 1st copy to complete.

Motivation: Server-side variability:

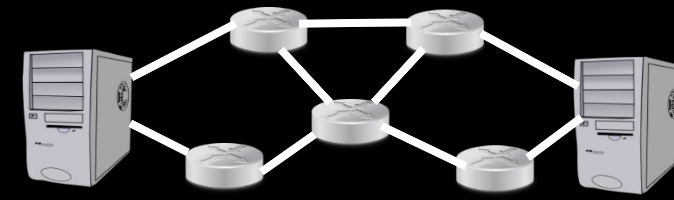
Same job can take 27X longer on one machine than another [Xu]

- Background load
- Garbage collection
- Network interrupts
- Disk head location
- Cache contents

Redundancy in Our Lives

Dolly LATE Hopper Mantri GRASS

Job Replication in Computer Systems



Redundant Packet Transmisison

UNOS

UNITED NETWORK FOR ORGAN SHARING

Multiple Listing
for Kidneys/Livers

Multi-listing at
Daycare Centers

Why Redundancy Rocks!

1. Redundancy → Experience queue with less work
2. Redundancy → Experience lower server slowdown

Both important under high server-side variability

But redundancy can also hurt …

HOW MUCH redundancy is best?

Analyzing Redundancy is Not Easy…

Requires tracking all copies of a job

Approximations

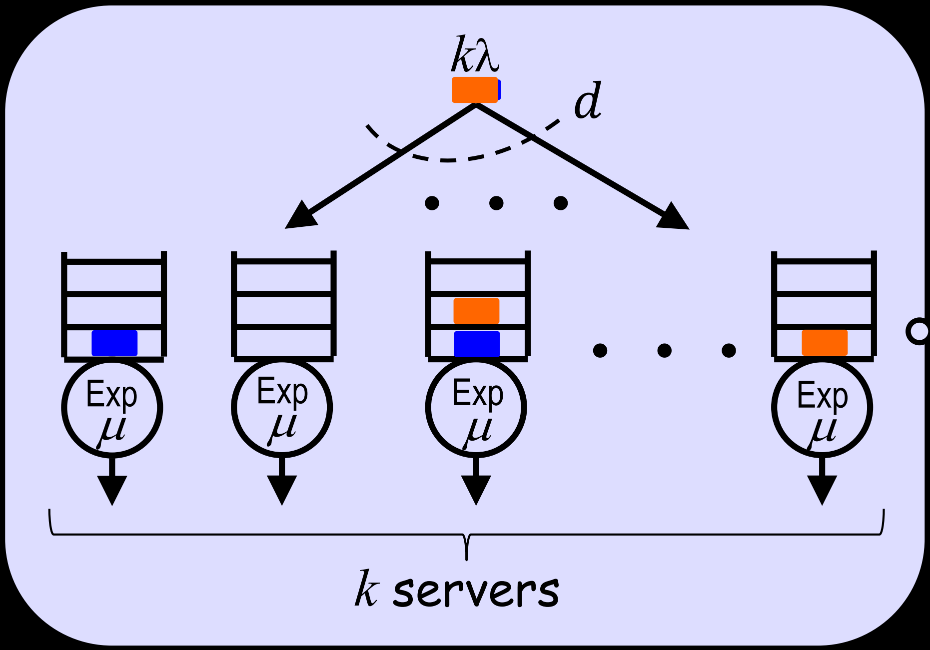
- [Koole, Righter 2009]
- [Joshi, Liu, Soljanin 2012]
- [Shah, Lee, Ramchandran 2012]
- [Huang, Pawar, Zhang, Ramchandran 2012]
- [Vulimiri, Godfrey, Mittal, Sherry, Ratnasamy, Shenker 2013]
- [Shah, Lee, Ramchandran 2013]
- [Joshi, Liu, Soljanin 2014]
- [Kumar, Tandon, Clancy 2014]
- [Sun, Koksal, Shroff 2016]

Exact Analysis

- [Gardner, Doroudi, Harchol-Balter, Hyytiä, Scheller-Wolf, Zbarsky] –Sigmetrics 2015
- [Gardner, Harchol-Balter, Scheller-Wolf, Zbarsky] – Operations Research 2017.
- [Bonald, Comte 2017]

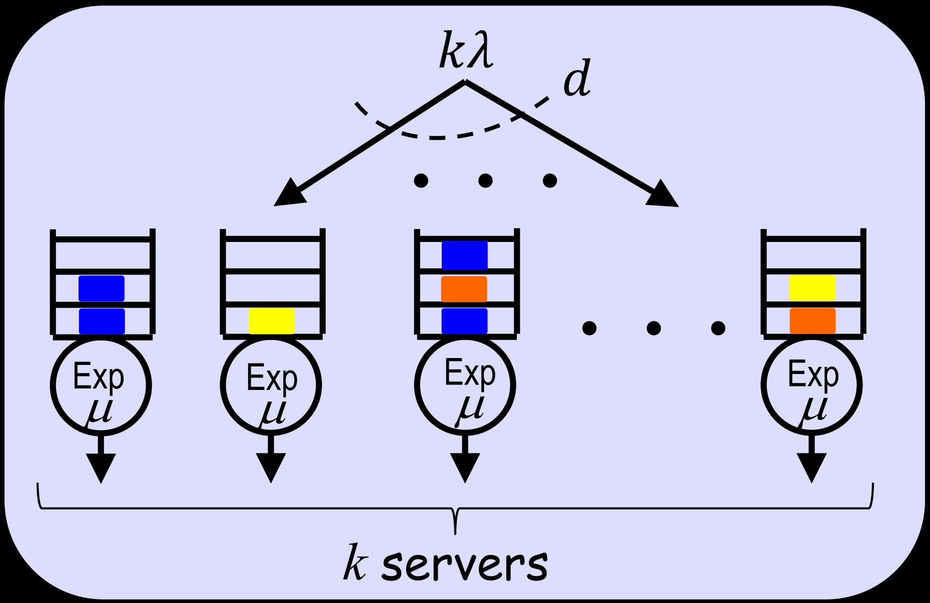
Example: Redundancy-d

[Gardner, Harchol-Balter, Scheller-Wolf, Zbarsky 2016]



- Every arrival is sent to d servers at random. Job is "done" as soon as 1^{st} copy completes.
- Poisson arrivals with rate $k\lambda$
- Independent runtimes (service times), Exponentially-distributed with rate μ

Markov Chain Analysis of Redundancy-d



Each job chooses random subset of d servers.

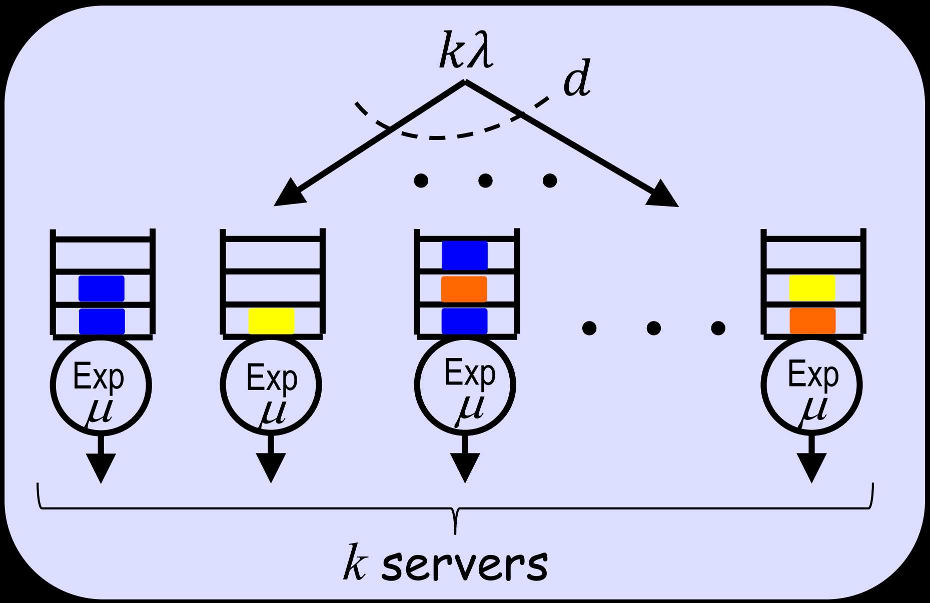
subset chosen $\Leftrightarrow$ "color"

c_i : color of i^{th} arrival

Thm:

$$\pi_{\underbrace{(c_m, c_{m-1}, \ldots, c_1)}_{\text{All current jobs in order of arrival}}} = C \cdot \prod_{i=1}^{m} \frac{\lambda_{c_i}}{\mu \cdot (\#\text{ servers used by first } i \text{ jobs})}$$

Markov Chain Analysis of Redundancy-d



Each job chooses random subset of d servers.

subset chosen ⟺ "color"

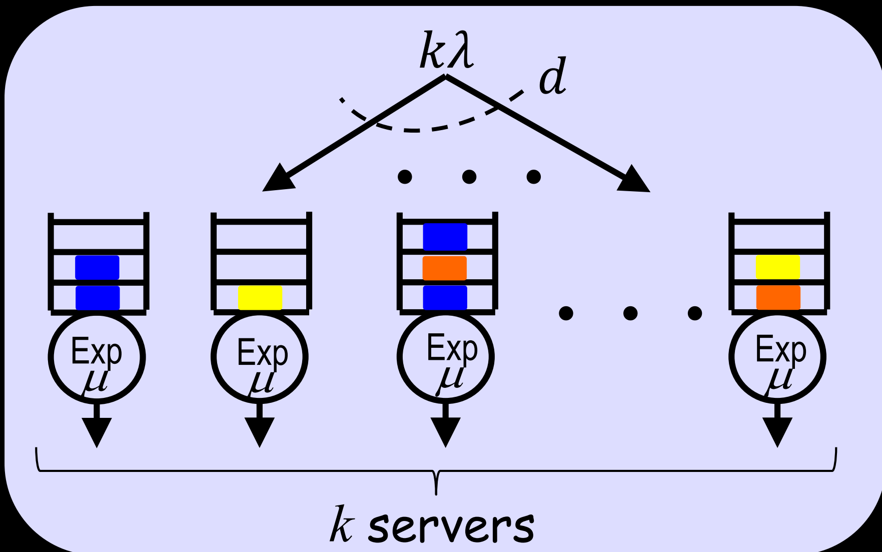
c_i : color of i^{th} arrival

$$\pi_{(B,Y,O,B)} = C \cdot \frac{\lambda_{blue}}{4\mu} \cdot \frac{\lambda_{yellow}}{4\mu} \cdot \frac{\lambda_{orange}}{3\mu} \cdot \frac{\lambda_{blue}}{2\mu}$$

Thm:

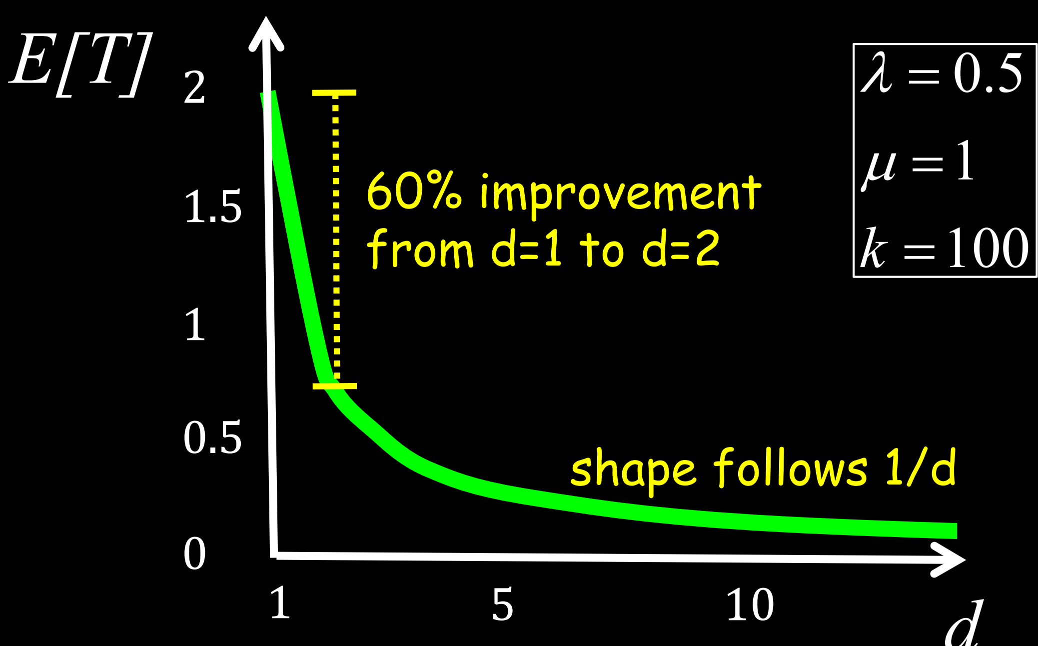
$$\pi_{\underbrace{(c_m, c_{m-1}, \ldots, c_1)}_{\text{All current jobs in order of arrival}}} = C \cdot \prod_{i=1}^{m} \frac{\lambda_{c_i}}{\mu \cdot (\#\text{ servers used by first } i \text{ jobs})}$$

Results of Exact Analysis: Redundancy-d

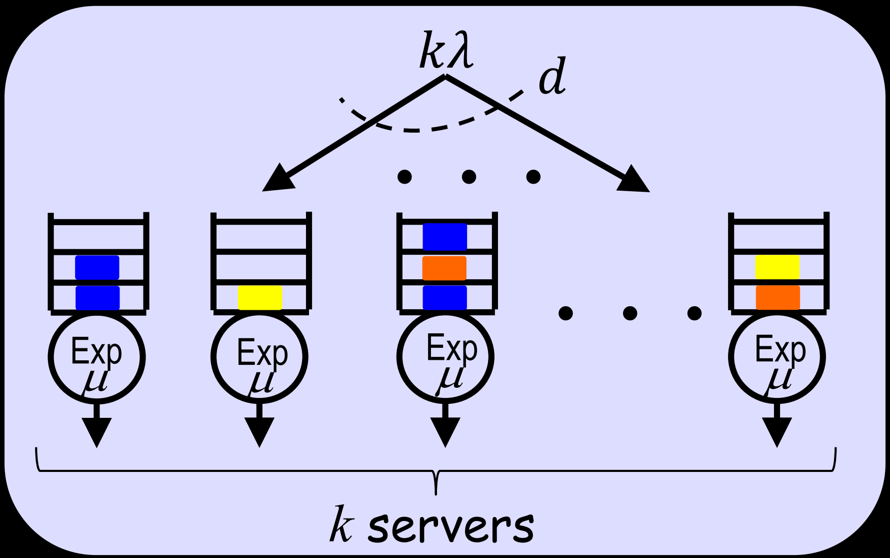


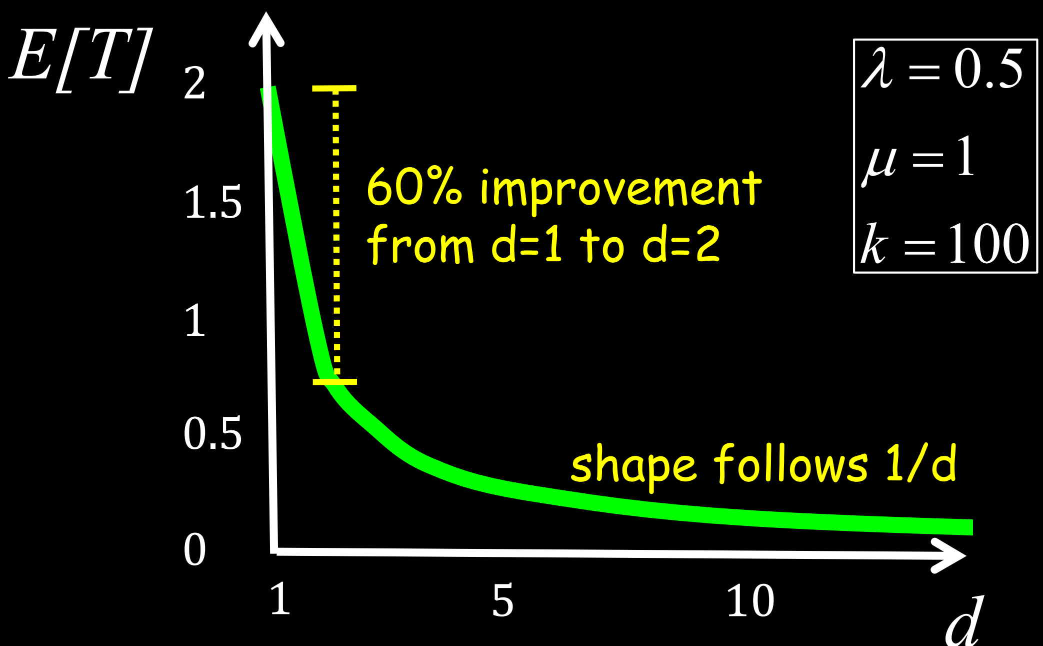
Thm:

$$E[T]^{k,d} = \sum_{i=d}^{k} \frac{\binom{i-1}{d-1}}{\binom{k}{d}} \cdot \frac{1}{\mu d - \binom{i-1}{d-1}\lambda_{c_i}}$$



Results of Exact Analysis: Redundancy-d



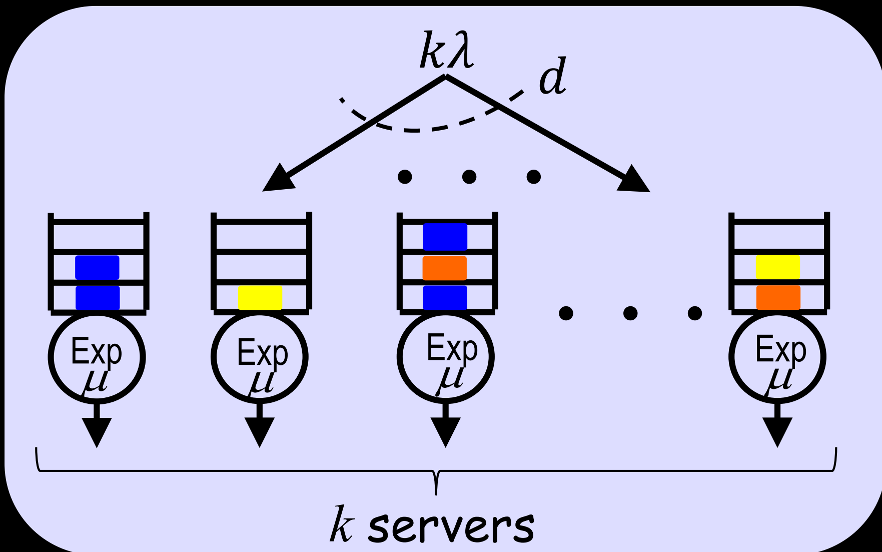


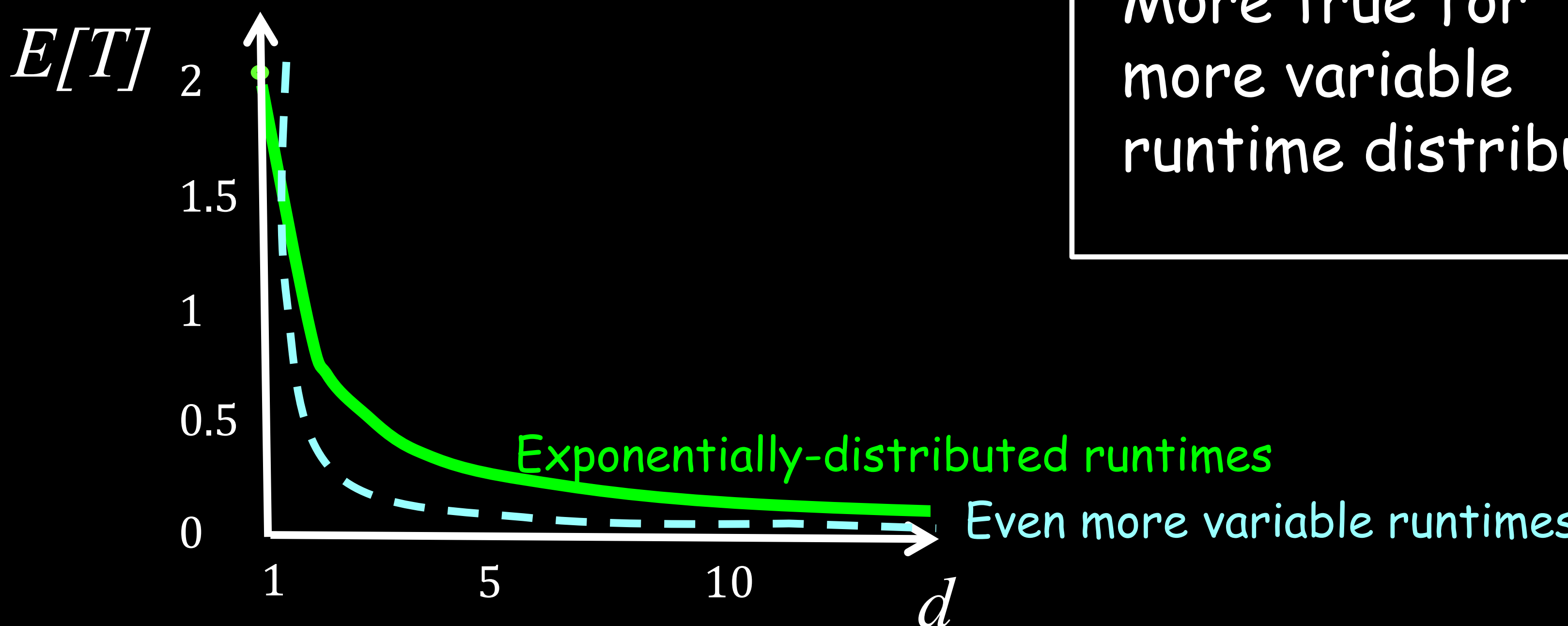
But WHY?

Is more redundancy always better?

Maybe only true for exponential runtimes?

Results of Exact Analysis



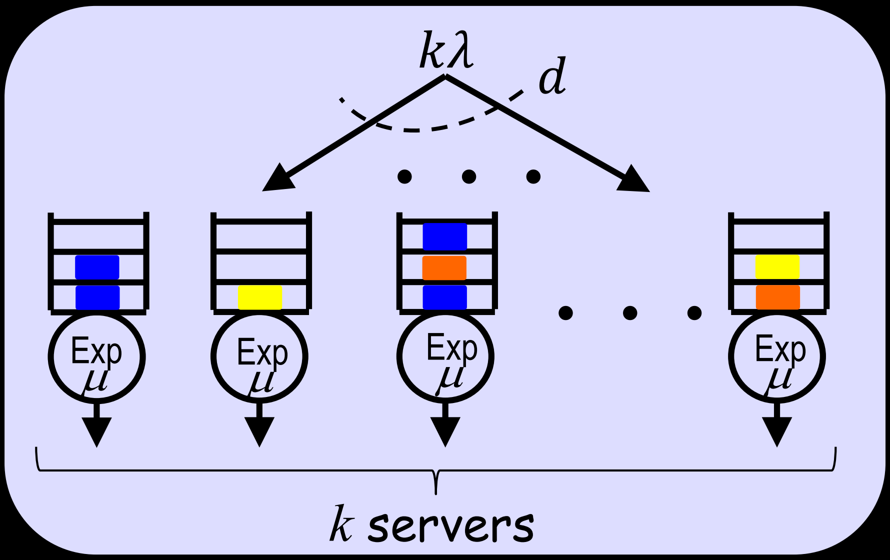


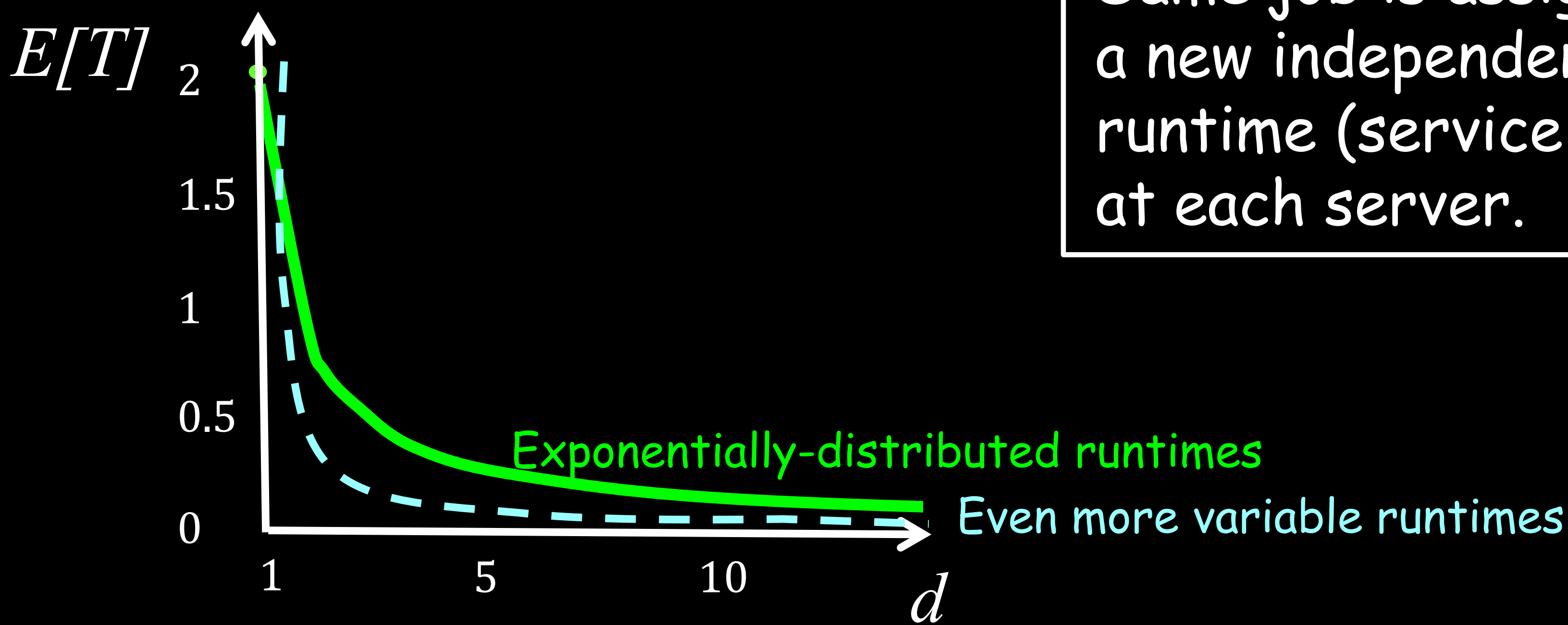
<u>But WHY?</u>

Is more redundancy always better?

More true for more variable runtime distributions.

Results of Exact Analysis





Problem:

Independent Runtime Model

Same job is assigned a new independent runtime (service time) at each server.

Prior Analytical Work assumes Independent Runtime Model (IRM)

Approximations

- [Koole, Righter 2009]
- [Joshi, Liu, Soljanin 2012]
- [Shah, Lee, Ramchandran 2012]
- [Huang, Pawar, Zhang, Ramchandran 2012]
- [Vulimiri, Godfrey, Mittal, Sherry, Ratnasamy, Shenker 2013]
- [Shah, Lee, Ramchandran 2013]
- [Joshi, Liu, Soljanin 2014]
- [Kumar, Tandon, Clancy 2014]
- [Sun, Koksal, Shroff 2016]

Exact Analysis

- [Gardner, Doroudi, Harchol-Balter, Scheller-Wolf, Zbarsky 2015]
- [Gardner, Harchol-Balter, Scheller-Wolf, Zbarsky 2016]
- [Bonald, Comte 2017]

IRM is reasonable if inherent job size is negligible.

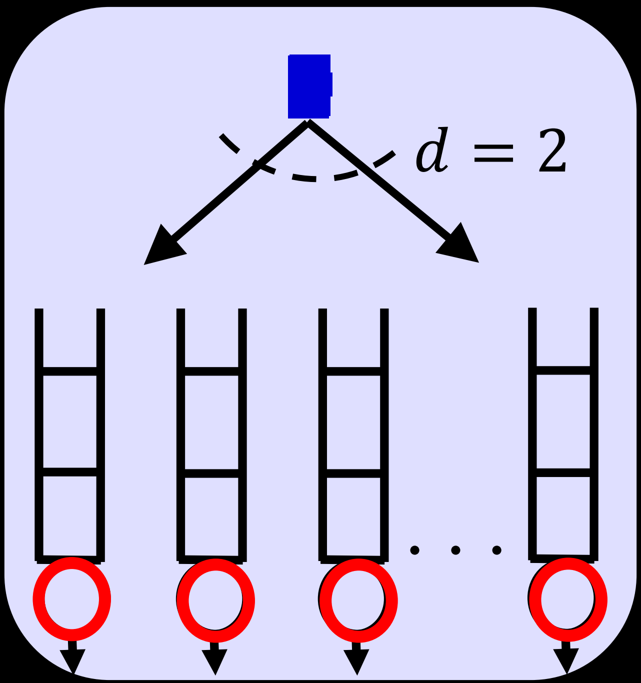
Unreasonable, otherwise.

Introducing S&X model

server slowdown

inherent job size

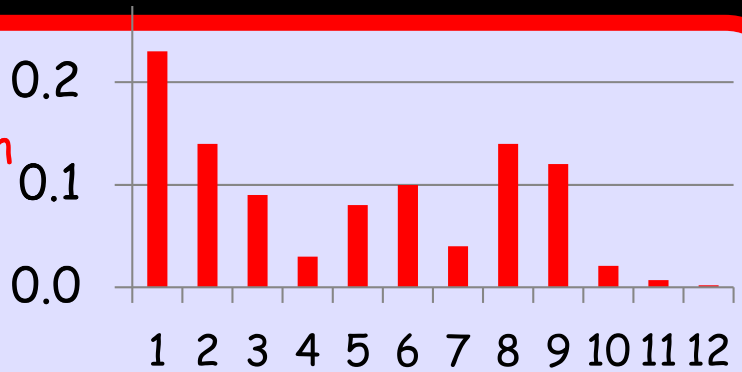
Ex: Redundancy-d



Traditional Q-theory handles only <u>single</u> "service time" variable. INADEQUATE for redundancy.

Empirical server slowdown distribution [Dolly '12]

$E[S] = 4.7$



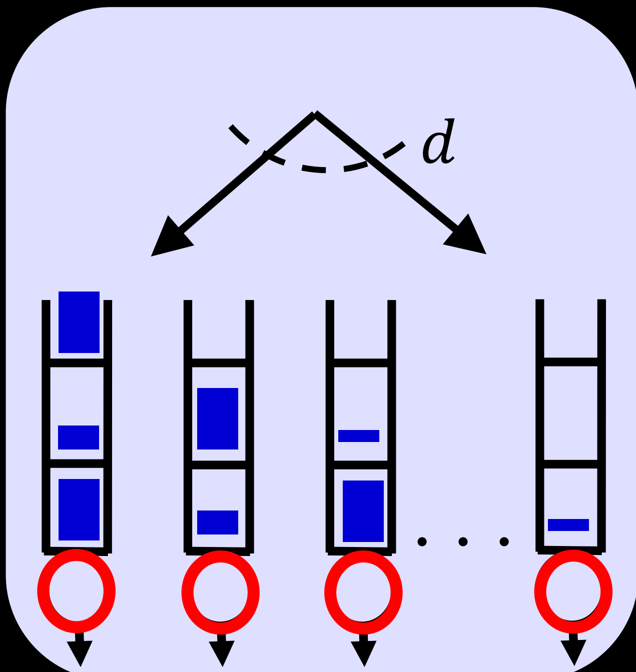
[Gardner, Harchol-Balter, Scheller-Wolf, *MASCOTS 2016*]

Introducing S&X model

server slowdown

inherent job size

Ex: Redundancy-d



When $d = 1$,

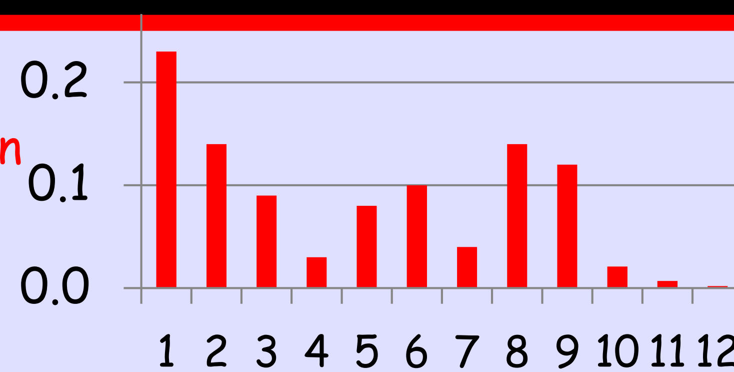
$$\text{Runtime} = R = S \cdot X$$

$$\text{Resp.Time} = T = T_Q + R$$

When $d > 1$, experience lowest of d response times.

Empirical server slowdown distribution [Dolly '12]

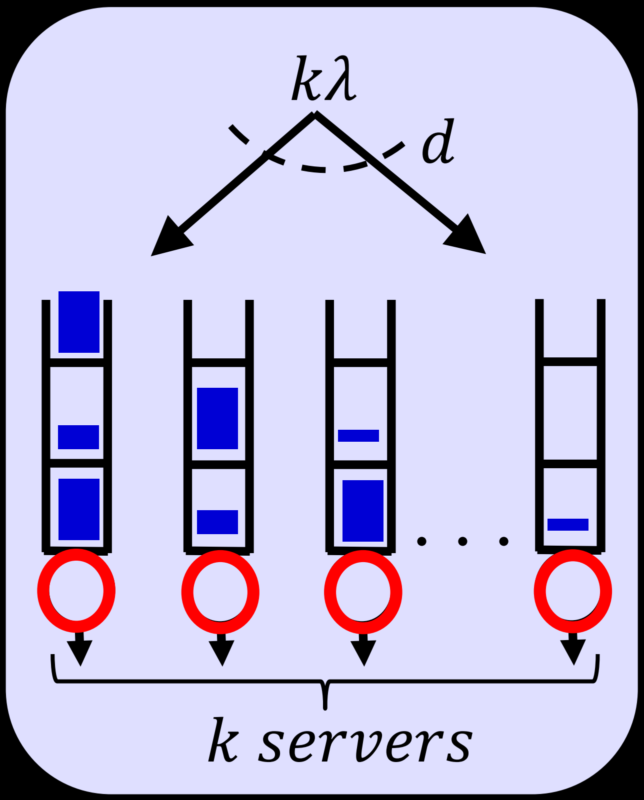
$E[S] = 4.7$



Redundancy in S&X: pros/cons

server slowdown

inherent job size



+ Redundancy ➔ see queue with lesser work

+ Redundancy ➔ see lower server slowdown

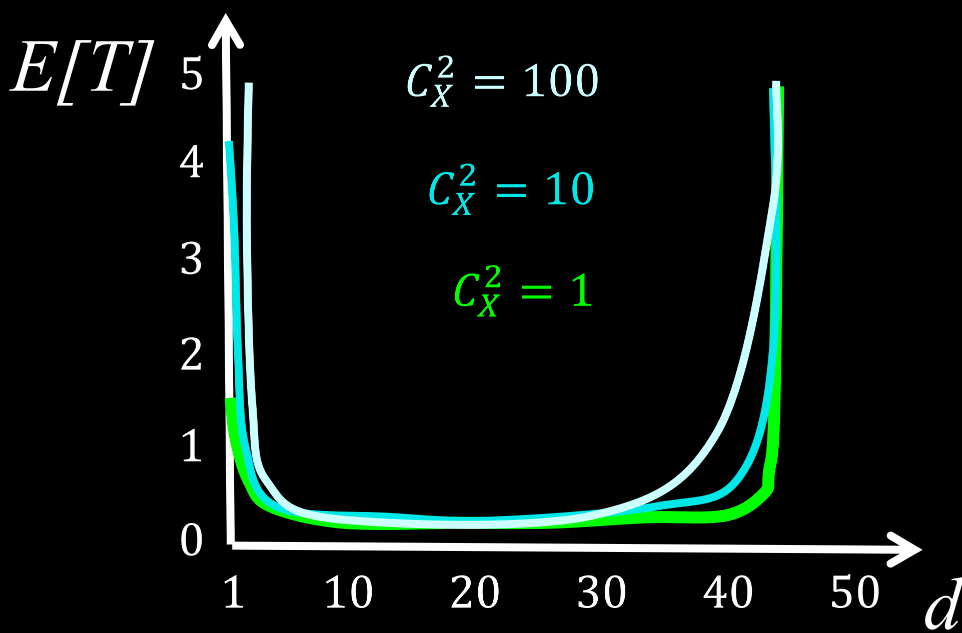
- Job with large inherent size adds lots of load.

Redundancy-d in S&X Model

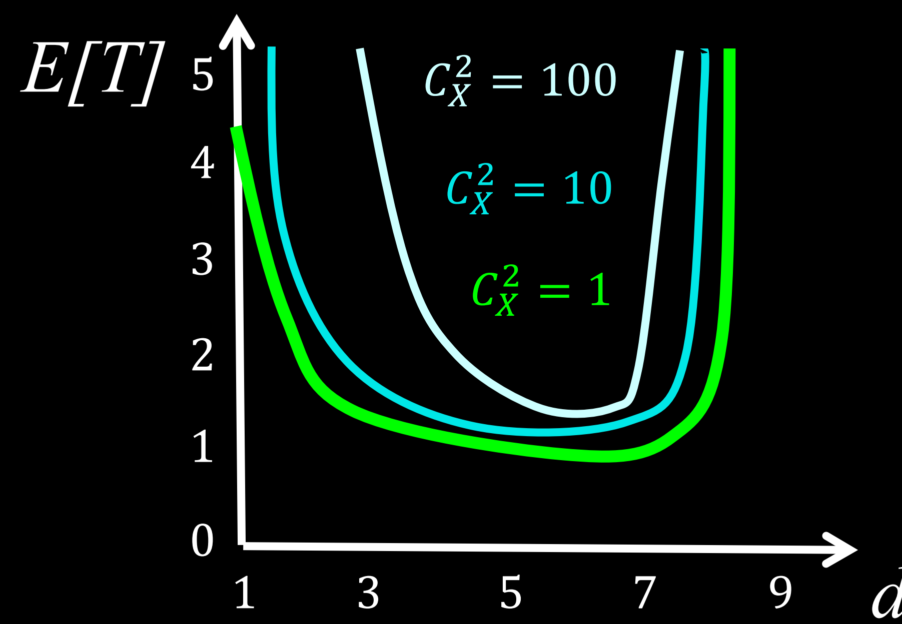
(simulation results: k = 1000 servers)

$S \sim$ Dolly empirical, $E[S] = 4.7$

$X \sim H_2$, $\mathrm{E[X]} = \frac{1}{4.7}$, $\mathrm{C}_X^2 = 1, 10, 100$



$\lambda = 0.3$



$\lambda = 0.7$

Issues

1. Robustness:
   Replication can create overload in S&X model
   Very sensitive to "right d"

2. Analytic Intractability:
   Can't analyze Redundancy-d in S&X model
   (our results are from simulation)

We propose one solution for both issues

Solution: RIQ algorithm

+ Limits extra load → No overload & More robust

+ Analytically tractable (approximately)

$\forall S \qquad \forall X \qquad \forall Z$

server slowdown

inherent job size

cancellation cost

Replicate-to-Idle-Queue (RIQ)

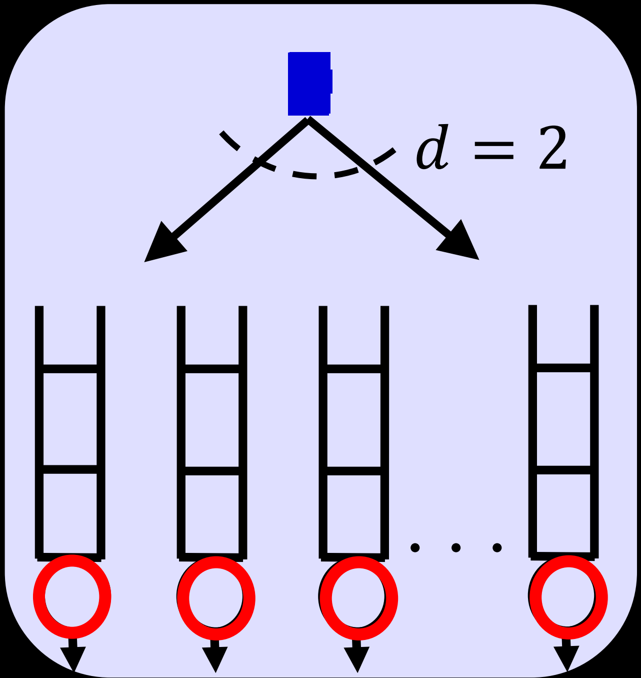
RIQ policy:

Arrival queries d random queues

- Replicate at all *idle* servers of d
- If none idle, pick random queue of d

RIQ intuition:
Only adding load if system can "take it."

RIQ analysis:
Replicas only affect first runtime in busy period → M/G/1/efs



RIQ Analysis Sketch

RIQ policy:

Arrival queries d random queues $(d \ll k)$

- If $i > 0$ idle ➔ replicate at all i
- If all busy ➔ go to random queue

$\rho = \Pr\{\text{Server is busy}\}$

Asymptotic Assumption: queues are independently busy with probability ρ

(recall $d \ll k$)

$$E[T]$$

$1-\rho^d$ $\qquad\qquad$ ρ^d

$E[T \mid \text{find} \geq 1 \text{ idle server}]$ $\qquad$ $E[T \mid \text{find all servers busy}]$

RIQ Analysis Sketch

RIQ policy:

Arrival queries d random queues

- If $i > 0$ idle → replicate at all i
- If all busy → go to random queue

$\rho = \Pr\{\text{Server is busy}\}$

Asymptotic Assumption: queues are independently busy with probability ρ

$$E[T]$$

$E[R(1)]$

$E[R(2)]$

$\binom{d}{i}(1-\rho)^i \rho^{d-i}$

$\ldots$

$E[R(i)]$

$)]$

ρ^d

$E[T \mid \text{find all servers busy}]$

Runtime on i servers

$$R(i) = \min(S_1, S_2, \ldots, S_i) \cdot X$$

RIQ Analysis Sketch

$$E[T]$$

$\rho = \Pr\{\text{Server is busy}\}$

$E[R(1)]$

$E[R(2)]$

...

$E[R(d)]$

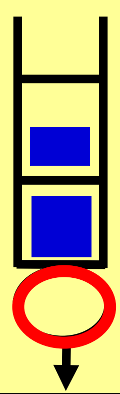
$i > 0$
idle
servers

ρ^d

$E[T \mid \text{find all servers busy}]$

➔ job goes to random queue

Random Queue $\equiv$ M*/G/1/efs



$$G_{\text{rest}} = R(1)$$

$$G_{\text{first}} = R(i) \quad \text{w.p.} \binom{d-1}{i-1}(1-\rho)^{i-1}\rho^{d-i}$$

RIQ Analysis Sketch

$$E[T]$$

$\rho = \Pr\{\text{Server is busy}\}$

$i > 0$ idle servers: $E[R(1)]$, $E[R(2)]$, …, $E[R(d)]$

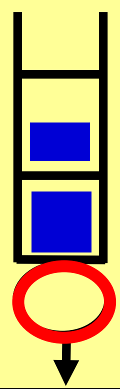
ρ^d: $E[T \mid \text{find all servers busy}]$

➔ job goes to random queue

Random Queue $\equiv$ M*/G/1/efs

$$\lambda_{\text{idle}} = \lambda k \cdot \frac{d}{k}$$

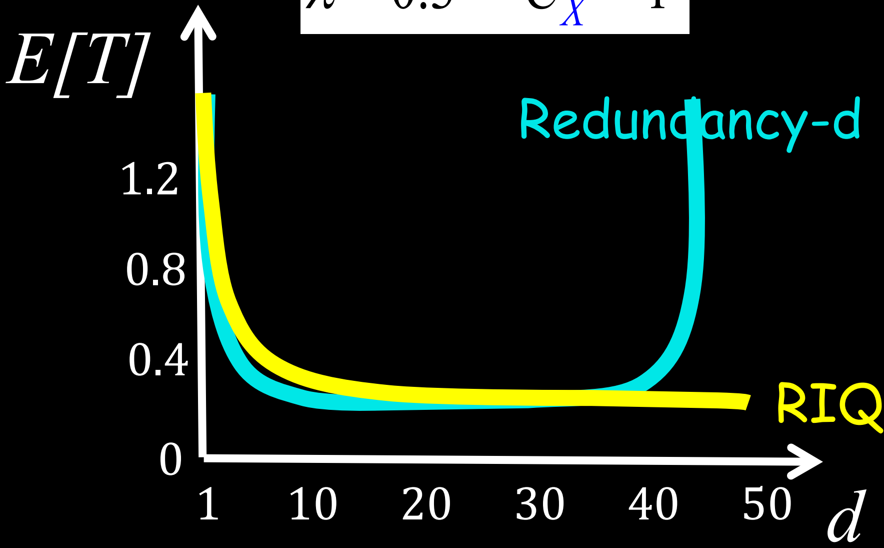
$$\lambda_{\text{busy}} = \lambda k \cdot \frac{d}{k} \cdot \rho^{d-1} \cdot \frac{1}{d}$$



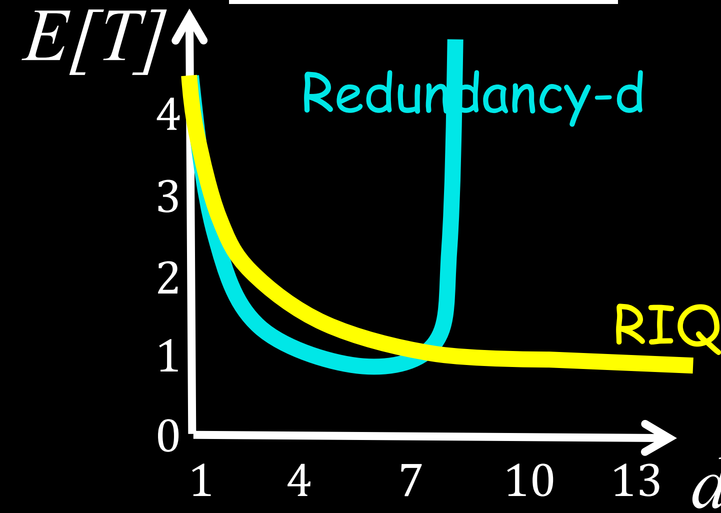
$$G_{\text{rest}} = R(1)$$

$$G_{\text{first}} = R(i) \quad \text{w.p. } \binom{d-1}{i-1}(1-\rho)^{i-1}\rho^{d-i}$$

$\lambda = 0.3 \quad C_X^2 = 1$

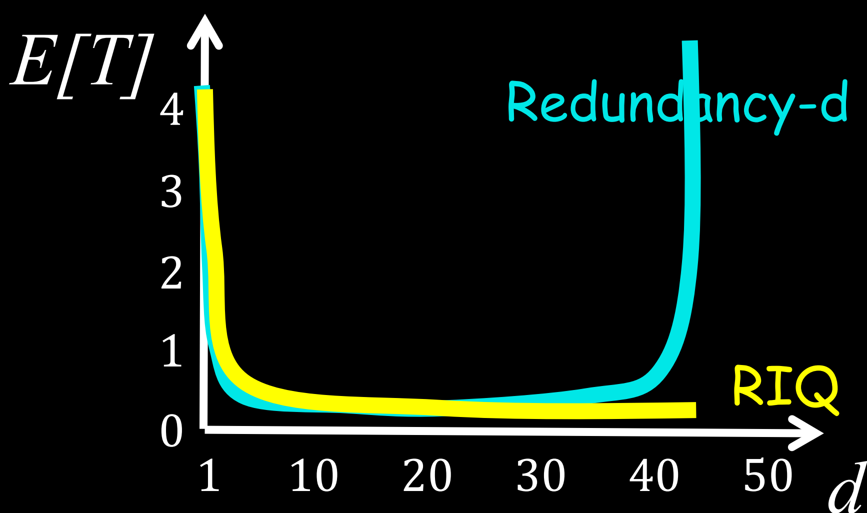


$\lambda = 0.7 \quad C_X^2 = 1$

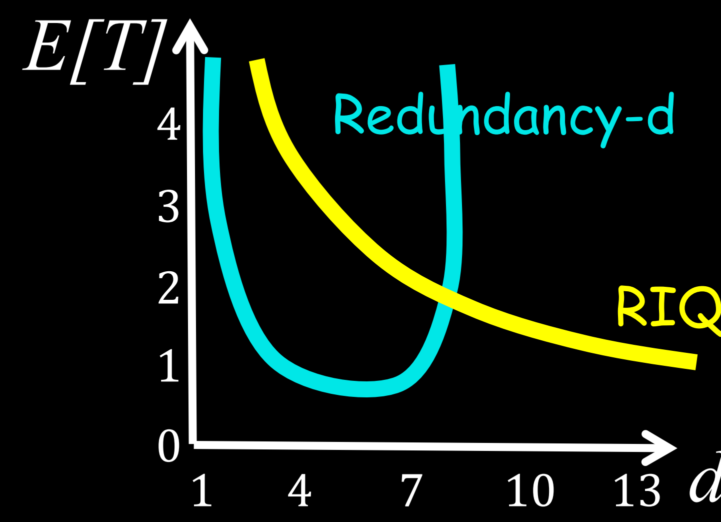


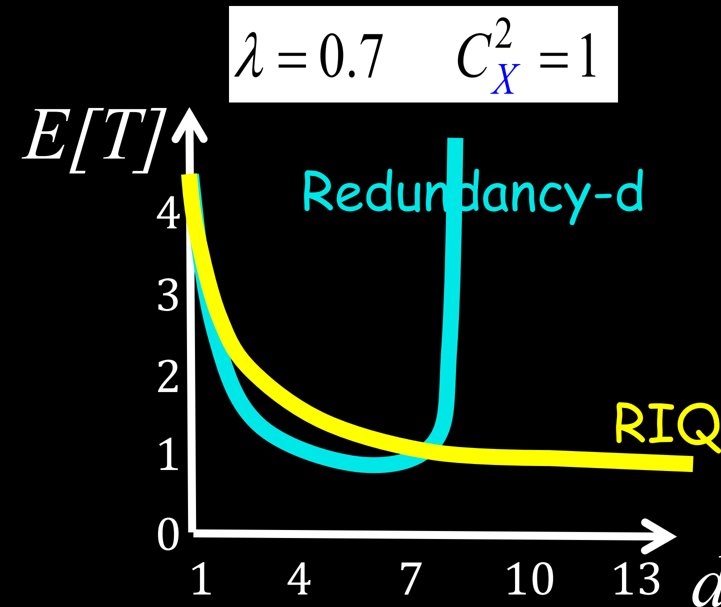
$k = 1000$ servers

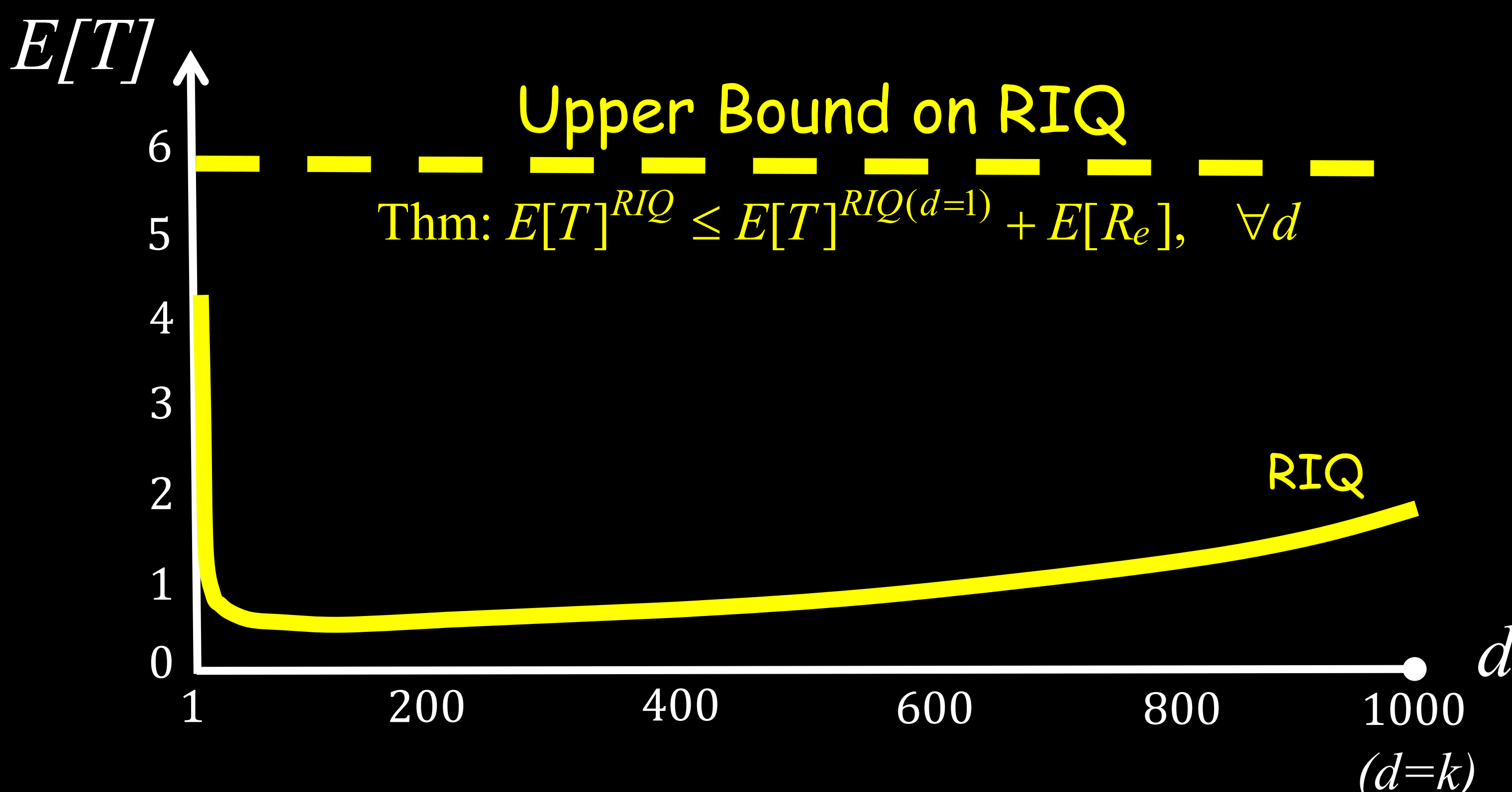
$\lambda = 0.3 \quad C_X^2 = 10$



$\lambda = 0.7 \quad C_X^2 = 10$

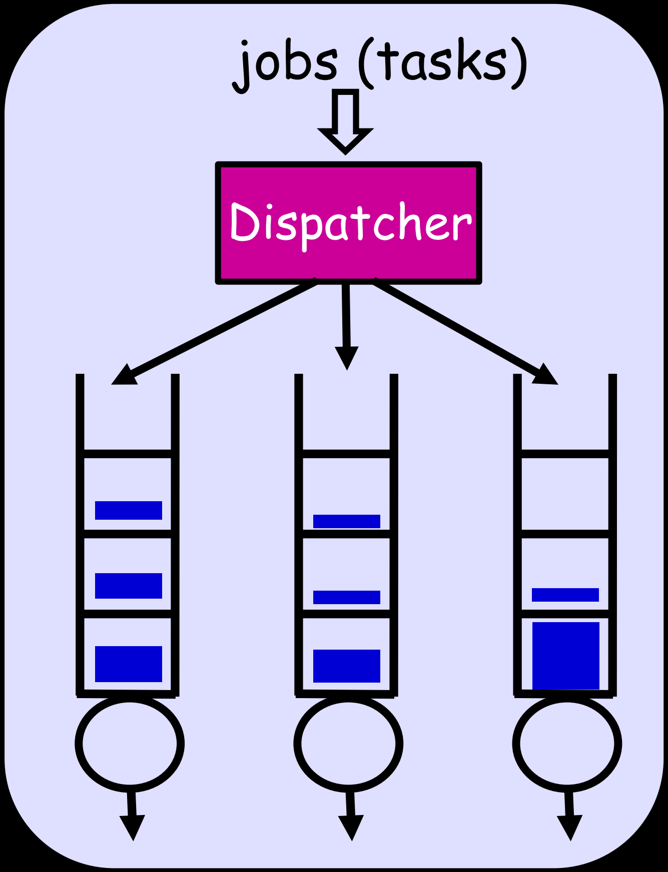






Thm: $E[T]^{RIQ} \le E[T]^{RIQ(d=1)} + E[R_e], \quad \forall d$

Conclusion 1/4

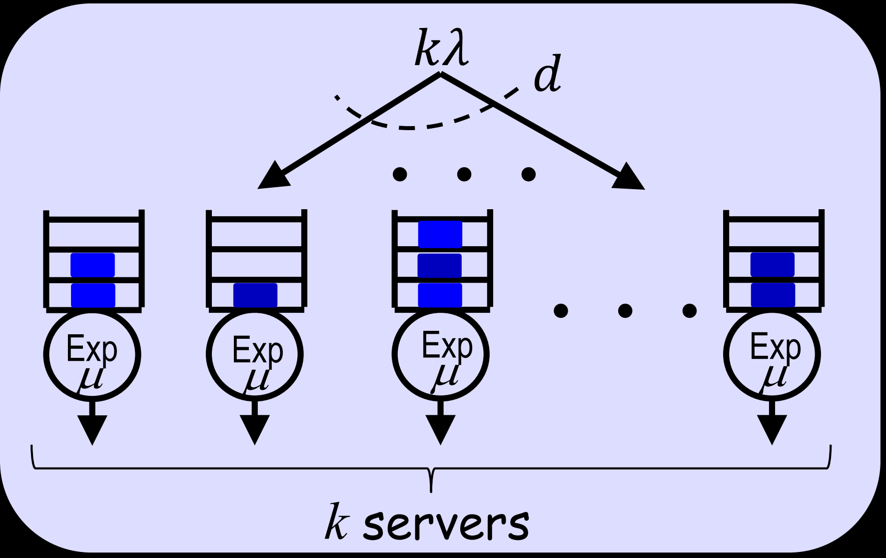


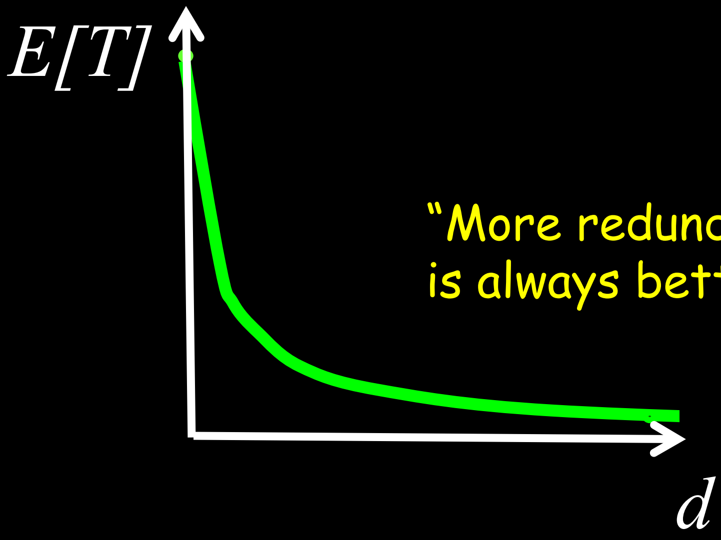
- Classic task assignment policies are inadequate in light of high server-side variability
- NEED REDUNDANCY!

Conclusion 2/4

- Traditional redundancy analysis, based on Independent Runtime Model, can lead to misleading conclusions.

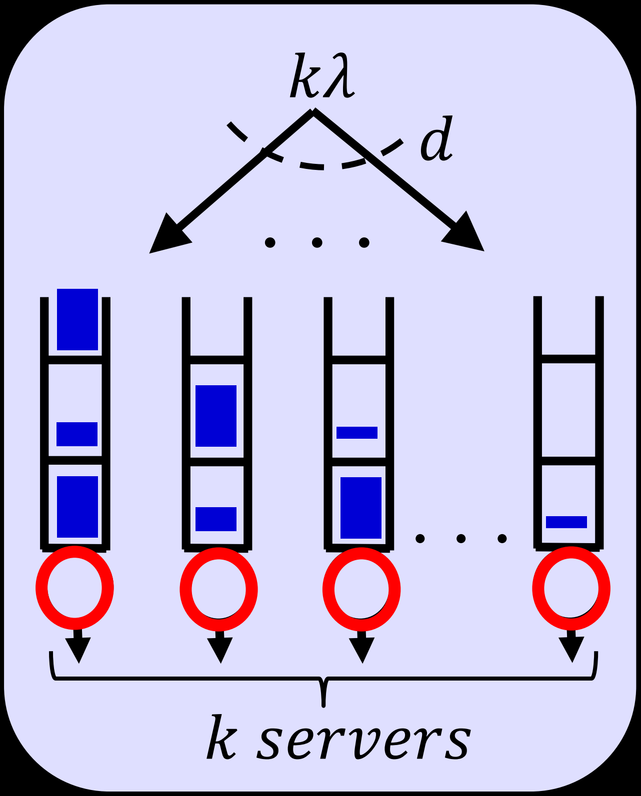
Redundancy-d





Conclusion 3/4

Redundancy-d in S&X

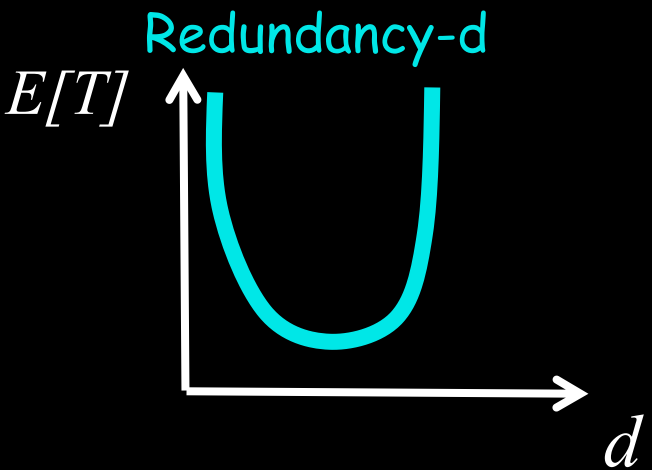


- Introduce S&X model

server slowdown

inherent job size

- Show via simulation that S&X leads to bathtubs



- Not robust!
  Also, not analytically tractable, so can't find "right" d

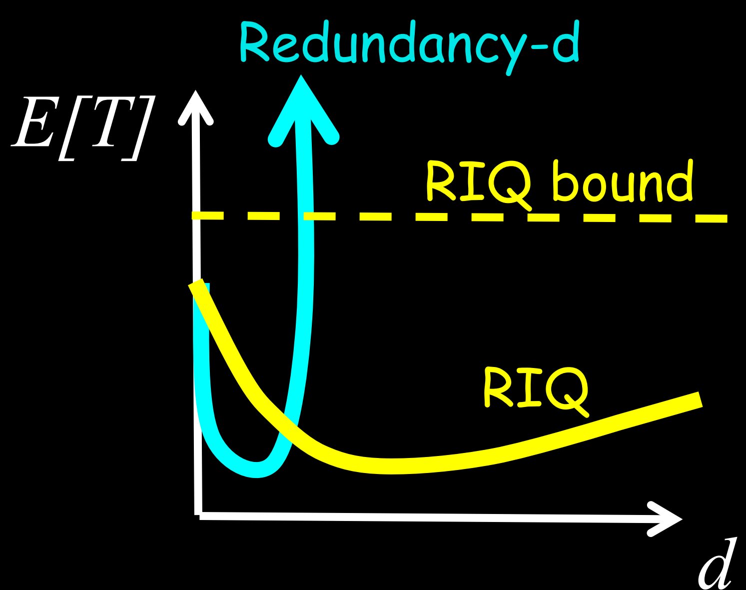
Conclusion 4/4

<u>Replicate-to-Idle-Queue policy</u>:

Arrival queries d random queues

- Replicate at all *idle* servers of d
- If none idle, pick random queue of d

- ❑ Introduced RIQ policy
- ❑ Analytically tractable in S&X model
- ❑ RIQ never in overload



Open questions on Replication

Math questions

- Convexity of $E[T]$ vs. d curve?
- What is the instability region for Redundancy-d?
- When does redundancy beat no redundancy?

Other replication algorithms

- "Replicate only small jobs" [Dolly]
- "Delay before replicating" [LATE, Mantri, Hopper]
- "Reserve servers for replicas"

More sophisticated models

- Jobs composed of multiple tasks
- Heterogeneous servers
- Correlated server slowdowns between consecutive jobs

Scheduling/Fairness

- PS queues
- Priority queues (kidneys)
- Class-based redundancy:
  - Pricing for redundancy
  - Which class to schedule first

Please email: harchol@cs.cmu.edu

REFERENCES

S&X Model + RIQ Algorithm

Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf. ``A Better Model for Job Redundancy: Decoupling Server Slowdown and Job Size."
In MASCOTS 2016.
Later Version: Transactions on Networking, vol. 25, no. 6, pp. 3353-3367, 2017.

IRM Model + Redundancy-d Algorithm

Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf, Mark Velednitsky, Sam Zbarsky. ``Redundancy-d: The Power of d Choices for Redundancy'' *Operations Research, vol. 65, no. 4, pp. 1078-1094.*

IRM Model + Class-based Redundancy

Kristen Gardner, Sam Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytia, Alan Scheller-Wolf. ``Queueing with redundant requests: exact analysis.'' *Queueing Systems: Theory and Applications* , vol. 83, no. 3, pp. 227-259, 2016.

Kristen Gardner, Sam Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytia, Alan Scheller-Wolf. ``Reducing Latency via Redundant Requests: Exact Analysis.'' *SIGMETRICS 2015.*