Performance Degradation in Parallel-Server Systems with Shared Resources

Esa Hyytiä
University of Iceland
esa@hi.is

Rhonda Righter
University of California Berkeley
rrighter@ieor.berkeley.edu

ABSTRACT

Parallel server systems are ubiquitous. Multicore CPUs are in practically every personal device from mobile handsets to high-end desktop PCs. At larger scale, data centers consist of a huge number of physical servers often shared by multiple users (for economic reasons). Moreover, the simultaneous users are typically unaware of each other due to reasons that can be technical (cf. security & privacy), practical (coordination layer would add complexity) and business related (usage can be business sensitive information). This results in server-side variability in terms of unpredictable response times. We study means for tackling these challenges. In particular, we consider a model where multiple users (dispatchers) route their jobs to a pool of servers using different (dispatching) policies. The goal is to determine how different policies interact: whether users' decisions support each other, or if some decisions are simply counterproductive. The lack of coordination is shown to increase, e.g., the mean response times, with two common and robust dispatching policies: the static Size-Interval-Task Assignment (SITA) and the dynamic Round-Robin (RR). We refer to this phenomenon as the price of ignorance.

CCS CONCEPTS

• **Mathematics of computing** → **Queueing theory**; • **Information systems** → *Data centers.*

KEYWORDS

job dispatching, coordination, parallel servers, Round-Robin, SITA

ACM Reference Format:
Esa Hyytiä and Rhonda Righter. 2020. Performance Degradation in Parallel-Server Systems with Shared Resources. In *13th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '20), May 18–20, 2020, Tsukuba, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3388831.3388853

1 INTRODUCTION

We focus on large systems where multiple users share the same computing resources for economic reasons. By user we mean a general entity that generates computing jobs. It may correspond, e.g., to a single person running a batch of Monte Carlo simulations, or a company processing web page requests of their clients. Typical examples are computer centers and data centers in general.

VALUETOOLS '20, May 18–20, 2020, Tsukuba, Japan

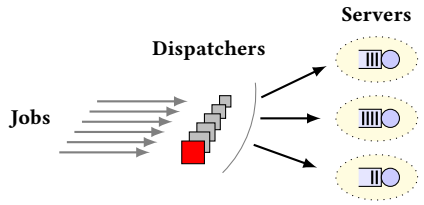

Figure 1: Multiple dispatching systems, unaware of each other, utilizing the same pool of servers.

Recently, the server-side variability in processing times has been acknowledged as one important performance factor [1]. This variability can be, e.g., due to other users sharing the same server (cf. virtual machines), background traffic in network, data locality, etc. These phenomena are hard to predict, and one practical solution proposed is job replication [6, 7, 12]. However, the models studied, e.g., in [5–7], do not explicitly model the source of server-side variability. In contrast, the model we consider in this paper, includes this aspect explicitly in the form of *competing* dispatchers.

It turns out that systems with multiple dispatchers have received far less attention than systems with a single dispatcher. This is somewhat surprising as one main argument for static policies (i.e., policies that are independent of the state of the servers) is the *scalability* in terms of parallel dispatchers. Recently, Doncel et al. [2] study the static Size-Interval-Task Assignment (SITA) policies for systems where coordination between the dispatchers is assumed. In particular, it is assumed that each dispatcher is allocated its own set of servers, and therefore no (stochastic) interactions are present. In contrast, we consider the situation where dispatchers, unaware of each other, *share* the same set of servers. We give exact closed-form results that quantify the performance with SITA, for Round-Robin (RR) we resort to simulation experiments and analysis in the heavy traffic limit. Both policies reveal interesting and different behavior as a function of the number of servers, the number of dispatchers and the offered load.

2 MODEL AND PRELIMINARIES

The system depicted in Figure 1 consists of k dispatchers each receiving jobs according to a Poisson process at rate $\lambda' = \Lambda/k$. Job sizes are i.i.d. and generally distributed random variables, $X_i \sim X$. The task of a dispatcher is to route each job immediately upon an arrival to one of the servers. The server pool consists of n identical first-come-first-served (FCFS) servers. The offered load is thus $\rho = \Lambda\,\mathrm{E}[X]/n$, which is assumed to be less than one for stability. The notation is summarized in Table 1.

The basic performance metric is the mean waiting time. The key dimension we explore in this paper is the *performance degradation*

Table 1: Notation:

n	the number of servers
k	the number of dispatchers
Λ	the total job arrival rate to the system
λ'	the job arrival rate per dispatcher, $\lambda' = \Lambda/k$
$\tilde{\lambda}$	the (random) job arrival rate to a single server
X	the (random) service time distribution
ρ	the offered load per server, $\rho = \Lambda\,\mathrm{E}[X]/n$
$\eta(x)$	the nominal cumulative load, $\eta(x) = \int_0^x t \cdot f(t)\,dt$

due to uncoordinated dispatching decisions, i.e., *the price of ignorance*. We assume that the dispatchers are completely unaware of each other. There are no status updates or initial communication prior to the start of operation. Moreover, we assume that no information about the total number of dispatchers k is available to any individual dispatcher, and the pool of servers is just an unordered set. This may arise, e.g., in computing centers whenever multiple parties submit their tasks concurrently independently of each other.

The random split (RND) assigns jobs at random and thus all dispatchers make statistically the same decision. With SITA, the decision depends on the size of the job and the dispatcher-specific numbering of the servers. With RR, the decision depends on which server the given dispatcher sent the previous job to. Thus, with all three policies, the dispatching decision is independent of the state of the servers, but with SITA and RR the destination of a new job depends on the dispatcher handling it. It turns out that the performance decreases as a function of k due to the lack of coordination for these two policies. Quantifying the performance deterioration (i.e., the price of ignorance) with SITA and RR is the main contribution of this paper.

3 STATIC POLICIES

A dispatching policy is *static* if its decision is independent of past decisions and the state of the queues. Consequently, these policies scale extremely well as no communication is needed between the dispatchers and servers, nor among the dispatchers. The downside is that their performance is typically worse than that of an adequate dynamic policy. First we recap the situation with a single dispatcher, and then consider a system with multiple dispatchers.

3.1 Single Dispatcher – Warmup and Recap

Let us consider a system comprising a single dispatcher routing jobs to n servers. Jobs arrive according to a Poisson process at rate Λ and their sizes are i.i.d. random variables, denoted by X. The n servers are identical and follow the FCFS queueing discipline. Given the dispatcher employs a static policy, the system decomposes into n independent M/G/1 queues, and the mean waiting time for any fixed server is given by the Pollaczek-Khinchine (PK) formula,

$$\mathrm{E}[\tilde{W}] = \frac{\tilde{\lambda}\,\mathrm{E}[\tilde{X}^2]}{2(1-\tilde{\rho})} = \frac{\tilde{\rho}}{2(1-\tilde{\rho})} \times \frac{\mathrm{E}[\tilde{X}^2]}{\mathrm{E}[\tilde{X}]}, \tag{1}$$

where $\tilde{\lambda}$, $\tilde{X}$ and $\tilde{\rho} = \tilde{\lambda}\,\mathrm{E}[\tilde{X}]$ denote the arrival rate, job size and the offered load at the given server, which all depend on the static policy splitting the Poisson stream of new jobs among the n servers.

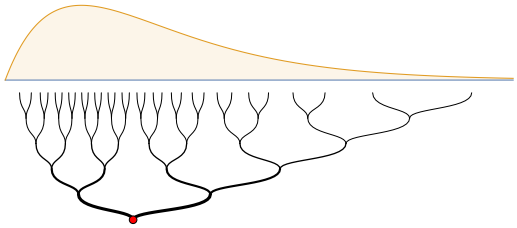

Figure 2: Thresholds ξ_i with SITA for $\mathrm{X} \sim \mathrm{Exp}(\mu)$ for $\mathrm{n} = 2, 2^2, \ldots, 2^6$ servers. The upper curve corresponds to $\mathrm{x\,f(x)}$ and $\int_{\xi_{i-1}}^{\xi_i} x\,f(x)\,dx$ is equal to $\mathrm{E[X]/n}$ for all i.

Two popular static dispatching policies are the Bernoulli split (RND) and the size-interval-task-assignment (SITA) [8].

Definition 3.1. **Random Bernoulli split (RND)** routes jobs independently at random according to probabilities $p_1, \ldots, p_n$, one for each server, such that $p_1 + \ldots + p_n = 1$. In general, the probabilities can be dispatcher specific.

Definition 3.2. **Size-interval-task-assignment (SITA)** has $n+1$ threshold parameters $\xi_0 < \ldots < \xi_n$ that split the possible job sizes into n disjoint intervals,

$$[\xi_0, \xi_1),\ [\xi_1, \xi_2),\ \ldots,\ [\xi_{n-1}, \xi_n).$$

SITA routes a job to server i if its size belongs to the i^{th} size-interval. It is customary that $\xi_0 = 0$ and $\xi_n = \infty$. With multiple dispatchers, the size intervals can be dispatcher specific.

It is worth noting that SITA, with optimized thresholds (SITA-opt), has been shown to be the optimal static policy for FCFS servers with respect to the mean response time [4]. In this paper, however, we focus on load balancing versions of RND and SITA. In our context, this is a fair assumption as the dispatching policies are assumed to be unaware of each other. Thus, as a "*gentlemen's agreement*", they are expected to balance the load by utilizing all servers equally. Moreover, we assume that the number of servers, n, is known, and they are identical. Therefore, we will choose the parameters of RND and SITA so that the load for each server is $\rho = \Lambda\,\mathrm{E}[X]/n$. That is, our RND uses $p_i = 1/n$ for all i. Similarly, SITA with equal load has well-defined thresholds that depend solely on the job size distribution.

Definition 3.3 (Nominal cumulative load). For (continuous) job size distribution $f(t)$, the *nominal cumulative load* is

$$\eta(x) \triangleq \int_0^x t \cdot f(t)\,dt. \tag{2}$$

Thus, $\lim_{x\to\infty} \eta(x) = \mathrm{E}[X]$. Given $\eta(x)$, the SITA thresholds for n identical servers are obtained by solving for ξ_i from

$$\eta(\xi_i) = \frac{i}{n}\,\mathrm{E}[X], \qquad i = 1, \ldots, (n-1).$$

As an example, with $X \sim \mathrm{Exp}(\mu)$, we have

$$\eta(x) = (1 - (1+\mu x)e^{-\mu x})/\mu,$$

from which SITA thresholds ξ_i can be easily determined. The resulting thresholds are illustrated in Figure 2.

We call jobs that fall in the i^{th} interval type i jobs. Letting X_i be the job size of a type i job,

$$X_i = (X \mid \xi_{i-1} \le X < \xi_i),$$

and letting b_i be the probability a random job is a type i job, we have $b_i\,\mathrm{E}[X_i] = \mathrm{E}[X]/n$ for all i. Note that the thresholds ξ_i, and therefore also X_i and b_i, are independent of ρ.

From (1), The mean waiting time with RND is given by,

$$\mathrm{E}[W^{\mathrm{RND}}] = \frac{\rho}{2(1-\rho)} \times \frac{\mathrm{E}[X^2]}{\mathrm{E}[X]}. \tag{3}$$

Lemma 3.4. *The mean waiting time with SITA is given by,*

$$\mathrm{E}[W^{SITA}] = \frac{\rho}{2(1-\rho)} \times \frac{n}{\mathrm{E}[X]} \sum_{i=1}^{n} b_i s_i, \tag{4}$$

where

$$b_i \triangleq \int_{\xi_{i-1}}^{\xi_i} f(x)\,dx, \quad \textit{and} \quad s_i \triangleq \int_{\xi_{i-1}}^{\xi_i} x^2 f(x)\,dx. \tag{5}$$

Proof. The mean waiting time with SITA is,

$$\mathrm{E}[W^{\mathrm{SITA}}] = \sum_{i=1}^{n} b_i \cdot \frac{\rho}{2(1-\rho)} \cdot \frac{\mathrm{E}[X_i^2]}{\mathrm{E}[X_i]},$$

and as $b_i\,\mathrm{E}[X_i] = \mathrm{E}[X]/n$,

$$\mathrm{E}[W^{\mathrm{SITA}}] = \frac{\rho}{2(1-\rho)} \times \frac{n}{\mathrm{E}[X]} \sum_{i=1}^{n} (b_i)^2 \cdot \mathrm{E}[X_i^2].$$

Substituting $s_i = b_i\,\mathrm{E}[X_i^2]$ yields (4). □

Remark 3.5. *The relative performance improvement with SITA over RND in terms of mean waiting time does not depend on* Λ,

$$\frac{\mathrm{E}[W^{SITA}]}{\mathrm{E}[W^{RND}]} = \beta(X,n) \qquad \forall\,\Lambda, \tag{6}$$

where,

$$\beta(X,n) \triangleq \frac{n}{\mathrm{E}[X^2]} \sum_{i=1}^{n} b_i s_i,$$

is independent of the offered load ρ.

Proposition 3.6.

$$\mathrm{E}[W^{SITA}] < \mathrm{E}[W^{RND}]. \tag{7}$$

Proof. From (6), we need to show that (for $n \ge 2$)

$$\sum_{i=1}^{n} b_i s_i < \frac{\mathrm{E}[X^2]}{n} = \sum_{i=1}^{n} \frac{1}{n} s_i.$$

Because ξ_i is strictly increasing in i, so is $\mathrm{E}[X_i]$. As $b_i\,\mathrm{E}[X_i] = \mathrm{E}[X]/n$ for all i, we must have b_i strictly decreasing in i. Thus it is sufficient to show that s_i is strictly increasing in i. This follows from

$$\xi_{i-1} \int_{\xi_{i-1}}^{\xi_i} x\,f(x)\,dx < \int_{\xi_{i-1}}^{\xi_i} x^2 f(x)\,dx < \xi_i \int_{\xi_{i-1}}^{\xi_i} x\,f(x)\,dx,$$

yielding

$$\xi_{i-1}\frac{\mathrm{E}[X]}{n} < s_i < \xi_i \frac{\mathrm{E}[X]}{n}, \tag{8}$$

which implies that s_i is strictly increasing in i. □

The following corollary shows that the performance with SITA improves when the size of the system is scaled.

Corollary 3.7. *Consider dispatching systems with n and mn servers,* $m = 2, 3, \ldots$, *both routing jobs according to SITA under the same load* ρ. *The mean waiting time is lower in the larger system.*

Proposition 3.8. *The mean waiting time for single-dispatcher systems with SITA in the limit as* $n \to \infty$ *is*

$$\mathrm{E}[W^{SITA}] \to \frac{\rho\,\mathrm{E}[X]}{2(1-\rho)}, \quad \textit{as} \quad n \to \infty. \tag{9}$$

Proof. Let us first assume that X is supported on a bounded interval (u, v), and the pdf $f(x)$ is a (piecewise) continuous function. Then $\eta(v) = \mathrm{E}[X]$, and we can set $\xi_0 = u$ and $\xi_n = v$ so that the lengths of all size-intervals, $\Delta_i = \xi_i - \xi_{i-1}$, tend to zero as n increases. The key observation is that in the limit the variance at each server goes to 0, so $\mathrm{E}[X^2] \to \mathrm{E}[X]^2$, which yields the result.

More precisely, the mean waiting time with SITA is (4),

$$\mathrm{E}[W] = \frac{\rho}{2(1-\rho)\,\mathrm{E}[X]} \times n \sum_{i=1}^{n} b_i s_i. \tag{10}$$

For large n, $b_i\xi_i \to \mathrm{E}[X]/n$, and $s_i \to \xi_i^2\, f(\xi_i)\,\Delta_i$, so that

$$\sum_i n b_i s_i = \mathrm{E}[X] \sum_i \xi_i\, f(\xi_i)\,\Delta_i \to \mathrm{E}[X]^2.$$

Substituting the above into (10) yields (9).

Consider next the case where X is a continuous random variable with a (piecewise) continuous pdf $f(x)$ and unbounded support so that

$$\mathrm{P}\{X > x\} > 0, \quad \forall\, x > 0. \tag{11}$$

Moreover, we assume that $\mathrm{E}[X]$ and $\mathrm{E}[X^2]$ are finite.

The expression (10) for the mean waiting time can be written as,

$$\mathrm{E}[W] = \frac{\rho}{2(1-\rho)\,\mathrm{E}[X]} \times \left[n \sum_{i=1}^{n-1} b_i s_i + n b_n s_n \right].$$

The sum includes a finite interval $[\xi_0, \xi_{n-1})$, which will be covered by arbitrarily small intervals as n increases. In contrast, given (11), we have $\xi_n = \infty$, so we need to show that $n b_n s_n \to 0$ as $n \to 0$.

According to the load balancing, $b_i\mathrm{E}[X_i] = \mathrm{E}[X]/n$, and we have

$$n b_n s_n = \frac{\mathrm{E}[X]}{b_n\,\mathrm{E}[X_n]} \cdot b_n s_n = \frac{\int_{\xi_{n-1}}^{\xi_n} f(x)\,dx}{\int_{\xi_{n-1}}^{\xi_n} x\,f(x)\,dx} \cdot s_n\,\mathrm{E}[X].$$

As $\int_{\xi_{n-1}}^{\xi_n} x\,f(x)\,dx > \xi_{n-1} \int_{\xi_{n-1}}^{\xi_n} f(x)\,dx$, we obtain

$$n b_n s_n < \frac{s_n\,\mathrm{E}[X]}{\xi_{n-1}} < \frac{\mathrm{E}[X^2]\,\mathrm{E}[X]}{\xi_{n-1}}.$$

Given the support of X is unbounded, it follows that $\xi_{n-1} \to \infty$ as $n \to \infty$, and therefore $n b_n s_n \to 0$ as $n \to \infty$. □

It is straightforward to show that this result holds also if $f(x) = 0$ in some sub-interval(s). Recall that the mean waiting time with RND, given by (3), holds for any n. It follows that

$$1 \ge \frac{\mathrm{E}[W^{\mathrm{SITA}}]}{\mathrm{E}[W^{\mathrm{RND}}]} \ge \frac{\mathrm{E}[X]^2}{\mathrm{E}[X^2]},$$

where the equality on the left holds for $n = 1$, and on the right in the limit when $n \to \infty$.

3.2 Multi-Dispatcher Systems

3.2.1 Two Dispatchers. Suppose first that the system comprises two servers and two dispatchers, both routing an equal amount of work using SITA (see Figure 1). Consequently, both dispatchers either send their short jobs (long jobs) to the same server, or the opposite servers. If the dispatchers manage to agree on the same server for short (and long) jobs, the system reduces to a single-dispatcher system with SITA. In contrast, if the server assignment is the opposite, both servers receive both short and long jobs, and the system reduces to a system with RND. Given the dispatchers assign servers at random, the chances that the dispatchers order the servers the same way is 0.5. Hence, the *expected* gain from using SITA, without any *coordination* when configuring SITA for both dispatchers, is 50% of that achieved with a single dispatcher.

3.2.2 General Case. Suppose our system comprises k independent dispatchers routing jobs to n servers, as illustrated in Figure 1. Given the dispatchers have static dispatching policies, the system again decomposes into n independent parallel FCFS M/G/1 queues and the PK mean value results can be utilized.

Let us next assume that all dispatchers use SITA. The size intervals are thus chosen identically, each constituting a type i flow with parameters (λ_i, X_i), where $i = 1, \ldots, n$, and $\lambda_i = b_i\Lambda/k$ is the rate of type i jobs per dispatcher. Dispatchers operate independently and assign the n job size intervals randomly to the n servers.

Let us consider an arbitrary server and let Z_i denote the number of dispatchers that route type i jobs to this server. We refer to the vector $\mathbf{Z} = (Z_1, \ldots, Z_n)$ as the *configuration*, even though it describes only the flow of jobs routed to a single server. Then $\mathbf{Z}$ obeys the multinomial distribution with constraint $Z_1 + \ldots, Z_n = k$. Moreover, since the size intervals (the order of servers) are chosen uniformly, we have $Z_i \sim \mathrm{Bin}(k, p)$, where $p = 1/n$.

Proposition 3.9. *The mean waiting time in a multi-dispatcher system with k uncoordinated SITA dispatchers sharing n identical servers is given by*

$$\mathrm{E}[W^{SITA}] = \frac{\rho}{2(1-\rho)\,\mathrm{E}[X]}\left[\frac{k-1}{k}\,\mathrm{E}[X^2] + \frac{n}{k}\sum_{i=1}^{n}(b_i)^2\mathrm{E}[X_i^2]\right], \quad (12)$$

where the b_i and X_i depend on n, but not on k.

Proof. As all servers behave statistically the same way, we can focus on, say, server 1. The random configuration $\mathbf{Z}$ induces an arrival rate $\tilde{\lambda}$ and job size distribution $\tilde{X}$ for the given server. The mean number of waiting jobs in this server is

$$\mathrm{E}[\tilde{N}_q] = \frac{1}{2(1-\rho)} \cdot \mathrm{E}[\tilde{\lambda}^2\,\tilde{X}^2], \quad (13)$$

where the fact that the load remains constant under every configuration is utilized. Conditioning on $\mathbf{Z}$ gives

$$(\tilde{\lambda} \mid \mathbf{Z}) = \sum_i \lambda_i Z_i.$$

Similarly, for the second moment of $\tilde{X}$ we have,

$$\mathrm{E}[\tilde{X}^2 \mid \mathbf{Z}] = \frac{\sum_i \lambda_i Z_i\,\mathrm{E}[X_i^2]}{(\tilde{\lambda} \mid \mathbf{Z})}.$$

The conditional expectation, needed for the PK formula (13), is

$$\begin{aligned}\mathrm{E}[\tilde{\lambda}^2\,\tilde{X}^2 \mid \mathbf{Z}] &= \left(\sum_{i=1}^{n}\lambda_i Z_i\right)\cdot\left(\sum_{i=1}^{n}\lambda_i Z_i\,\mathrm{E}[X_i^2]\right),\\ &= \sum_{i=1}^{n}(\lambda_i)^2\mathrm{E}[X_i^2]Z_i^2 + \sum_{i\neq j}\lambda_i\lambda_j\,\mathrm{E}[X_j^2]\,Z_iZ_j. \quad (14)\end{aligned}$$

Given that $\mathbf{Z}$ obeys the multinomial distribution, we have

$$\begin{aligned}\mathrm{E}[Z_i] &= k\cdot\frac{1}{n},\\ \mathrm{V}[Z_i] &= k\cdot\frac{1}{n}\cdot\frac{n-1}{n},\\ \mathrm{Cov}(Z_i, Z_j) &= -k\cdot\frac{1}{n^2}, \qquad (i\neq j).\end{aligned}$$

Thus,

$$\begin{aligned}\mathrm{E}[(Z_i)^2] &= \frac{k(k+n-1)}{n^2},\\ \mathrm{E}[Z_iZ_j] &= \frac{k(k-1)}{n^2}. \qquad (i\neq j).\end{aligned}$$

Substituting these and the relation $\lambda_i = b_i\Lambda/k$ into (14) yields

$$\begin{aligned}\mathrm{E}[\tilde{\lambda}^2\,\tilde{X}^2] &= \frac{\Lambda^2}{n^2k}\Bigg[(k+n-1)\sum_i(b_i)^2\mathrm{E}[X_i^2]\\ &\qquad + (k-1)\sum_{i\neq j}b_ib_j\,\mathrm{E}[X_j^2]\Bigg].\end{aligned}$$

For the latter sum, we have

$$\sum_{i\neq j}b_ib_j\,\mathrm{E}[X_j^2] = \sum_{i=1}^{n}b_i\mathrm{E}[X_i^2]\sum_{j:j\neq i}b_j = \sum_i(1-b_i)b_i\mathrm{E}[X_i^2].$$

Therefore,

$$\begin{aligned}\mathrm{E}[\tilde{\lambda}^2\,\tilde{X}^2] &= \frac{\Lambda^2}{n^2k}\left[n\sum_i(b_i)^2\,\mathrm{E}[X_i^2] + (k-1)\sum_i b_i\,\mathrm{E}[X_i^2]\right],\\ &= \frac{\Lambda^2}{n^2k}\left[n\sum_i(b_i)^2\mathrm{E}[X_i^2] + (k-1)\,\mathrm{E}[X^2]\right]. \quad (15)\end{aligned}$$

According to Little's result,

$$\mathrm{E}[W^{\mathrm{SITA}}] = \frac{n\,\mathrm{E}[\tilde{N}_q]}{\Lambda} = \frac{n}{2(1-\rho)\Lambda}\cdot\mathrm{E}[\tilde{\lambda}^2\,\tilde{X}^2],$$

and substituting (15) then gives

$$\mathrm{E}[W^{\mathrm{SITA}}] = \frac{\Lambda}{2(1-\rho)n}\left[\frac{n}{k}\sum_i(b_i)^2\mathrm{E}[X_i^2] + \frac{k-1}{k}\,\mathrm{E}[X^2]\right],$$

which yields (12) as $\rho = \Lambda\,\mathrm{E}[X]/n$. □

Note that with $k = 1$, (12) reduces to (4). Proposition 3.9 has several important corollaries especially for larger systems. First, comparing the expressions (12) (for SITA) and (3) (for RND) reveals the following compact relationship that generalizes the result of Remark 3.5 to systems with $k > 1$ dispatchers:

Corollary 3.10. *The mean waiting time in a multi-dispatcher system with k SITA dispatchers and n servers is given by*

$$\mathrm{E}[W^{SITA}] = \mathrm{E}[W^{RND}]\left(1 - \frac{r_n}{k}\right), \quad (16)$$

where r_n is a load independent factor that depends solely on n and X,

$$r_n \triangleq 1 - \beta(X,n) = 1 - \frac{n}{\mathrm{E}[X^2]} \sum_{i=1}^{n} (b_i)^2 \mathrm{E}[X_i^2],$$

and $\mathrm{E}[W^{RND}]$ *is given by* (3).

The next corollary is an immediate consequence of Corollary 3.10, for systems with a large number of dispatchers, $k \gg 1$:

Corollary 3.11 (Many dispatchers limit). *For any fixed load $\rho < 1$, when the number of servers n is kept constant,* $\mathrm{E}[W^{SITA}]$ *is increasing in k and when the number of dispatchers tends to infinity,*

$$\mathrm{E}[W^{SITA}] \to \mathrm{E}[W^{RND}] \quad as \quad k \to \infty.$$

Remark 3.12. *With* coordination *the k dispatchers would act as a single dispatcher, with the mean waiting time given by* (4)*. This gives the* best-case *performance with uncoordinated SITA dispatchers.*

Let us next consider the many server limit $n \to \infty$, thus generalizing Proposition 3.8 for $k \geq 1$ dispatchers.

Corollary 3.13 (Many servers limit). *For any fixed load $\rho < 1$, when the number of dispatchers k is kept constant while the number of servers n tends to infinity, we have*

$$\mathrm{E}[W^{SITA}] \to \frac{\rho}{2(1-\rho)\mathrm{E}[X]} \left[\mathrm{E}[X^2] - \frac{\mathrm{V}[X]}{k} \right], \; as \; n \to \infty. \quad (17)$$

Proof. With SITA, it holds that $\mathrm{E}[X_i]b_i = \mathrm{E}[X]/n$. Arguing as in Proposition 3.8, we have $\mathrm{E}[X_i^2] = \mathrm{E}[X_i]^2$ in the limit $n \to \infty$, i.e. the variability in each size-interval eventually vanishes (given $\mathrm{E}[X^2]$ is finite). Thus, the term $(b_i)^2\mathrm{E}[X_i^2]$ in the latter sum in (12) converges to $\mathrm{E}[X]^2/n^2$, and

$$\frac{n}{k} \sum_{i=1}^{n} (b_i)^2 \mathrm{E}[X_i^2] \to \frac{\mathrm{E}[X]^2}{k}.$$

Therefore, from (12) we can deduce that

$$\mathrm{E}[W^{\mathrm{SITA}}] \to \frac{\rho}{2(1-\rho)\,\mathrm{E}[X]} \left[\frac{(k-1)\mathrm{E}[X^2] + \mathrm{E}[X]^2}{k} \right], \text{ as } n \to \infty,$$

which yields (17). □

Remark 3.14. *Comparing* (16) *and* (17) *reveals that*

$$\lim_{n\to\infty} r_n = \frac{\mathrm{V}[X]}{\mathrm{E}[X^2]} = \frac{c_v^2}{c_v^2 + 1}, \quad (18)$$

where c_v^2 denotes the squared coefficient of variation of the job sizes.

These corollaries imply that systems with multiple SITA dispatchers, unaware of each other, scale well as a function of the number of servers n, whereas the performance quickly deteriorates with increasing number of dispatchers, k, as quantified by (16).

One performance metric characterizing the increase is the (absolute) increase in the mean waiting time,

$$\Delta \triangleq \mathrm{E}[W \mid \text{uncoordinated}] - \mathrm{E}[W \mid \text{coordinated}].$$

With SITA,

$$\Delta_{\mathrm{SITA}} = \mathrm{E}[W^{\mathrm{RND}}] \cdot r_n \cdot (1 - 1/k),$$

where the first factor depends on the system parameters (base level performance), the second is a function of X and the number of servers n, and the third depends only on the number of dispatchers k. However, instead of absolute increase, one is often interested in the proportional increase (which is a dimensionless quantity):

Table 2: Performance degradation with SITA due to multiple uncoordinated dispatchers. Coefficients r_n in the limit $n \to \infty$ are obtained from (18) : 1/4, 1/2 and 5/6.

	$\mathrm{E}[W^{\mathrm{SITA}}]/\mathrm{E}[W^{\mathrm{RND}}]$		
n	U(0, 2)	Exp(1)	Weibull(1/2, 1/2)
$n = 1$	1	1	1
$n = 2$	$1 - 0.121/k$	$1 - 0.330/k$	$1 - 0.632/k$
$n = 3$	$1 - 0.163/k$	$1 - 0.399/k$	$1 - 0.722/k$
$n = 4$	$1 - 0.185/k$	$1 - 0.429/k$	$1 - 0.758/k$
$n = 5$	$1 - 0.198/k$	$1 - 0.446/k$	$1 - 0.777/k$
$n = 6$	$1 - 0.206/k$	$1 - 0.456/k$	$1 - 0.788/k$
$n = 7$	$1 - 0.212/k$	$1 - 0.464/k$	$1 - 0.796/k$
$n = 8$	$1 - 0.217/k$	$1 - 0.469/k$	$1 - 0.802/k$
$n = 9$	$1 - 0.221/k$	$1 - 0.473/k$	$1 - 0.806/k$
$n = 10$	$1 - 0.224/k$	$1 - 0.476/k$	$1 - 0.809/k$
⋮		⋮	
$n = \infty$	$1 - 0.250/k$	$1 - 0.500/k$	$1 - 0.833/k$

Definition 3.15 (Price of ignorance). Let us define the price of ignorance for a dispatching policy α as the ratio of the mean waiting times for that policy without coordination and with it,

$$\gamma \triangleq \frac{\mathrm{E}[W \mid \text{uncoordinated}]}{\mathrm{E}[W \mid \text{coordinated}]}.$$

Corollary 3.16 (Price of ignorance with SITA). *With SITA, the price of ignorance is*

$$\gamma_{SITA} = \frac{1 - r_n/k}{1 - r_n},$$

which tends to $(1 - r_n)^{-1}$ at the many dispatcher limit $k \to \infty$.

Remark 3.17. *Note that the price of ignorance for RND is* 1*, i.e., (lack of) coordination has no effect. Moreover, from* (3.10)*, the price of using RND relative to SITA is*

$$\frac{\mathrm{E}[W^{RND}]}{\mathrm{E}[W^{SITA}]} = 1 - r_n/k.$$

Example 3.18. *Expressions for the (scaled) mean waiting time with k SITA dispatchers and n shared servers are given in Table 2 for $X \sim Exp(1)$, $X \sim U(0,2)$ and $X \sim Weibull(1/2, 1/2)$. Note that $n = 1$ corresponds to the performance relative to RND. We observe that the marginal gain from adding one more server quickly diminishes. Moreover, from* (16) *we can deduce that, e.g., with $X \sim Exp(1)$, $k \geq 5$ implies that the performance with SITA is within* 10% *of that with RND no matter how large n is. That is, benefits from applying SITA vanish quickly when k increases a bit, which discourages its use in multi-user environments!*

4 ROUND-ROBIN POLICY

In this section, we revisit the popular Round-Robin (RR) routing policy, where dispatchers assign jobs sequentially to the servers. RR is known to be optimal in several settings, where limited information is available, see, e.g., [3, 10, 11]. Implementation-wise, as with static

policies, RR scales extremely well in the numbers of servers and dispatchers, as only a dispatcher-specific local state information is needed for routing decisions (i.e., the server the previous job was routed to). Formally, RR is defined as follows:

Definition 4.1. The **Round-Robin (RR)** routing policy assigns jobs sequentially to n servers, $1, 2, \ldots, n, 1, 2, \ldots$. In multi-dispatcher systems, each dispatcher follows its own *phase*.

The interesting aspect in our context is that in the presence of multiple RR dispatchers, their relative phases vary constantly because the arrival times are random. Consequently, this affects the performance of the system and our goal in this section is to quantify the performance degradation in this case.

4.1 Simulation Experiments with Round-Robin

Unfortunately systems with multiple RR dispatchers are difficult to analyze, and thus we resort to simulation experiments. Figure 3 depicts simulation results with RR for $n = 2, 4, 8$ servers. In each case, we vary the number of dispatchers $k = 1, 2, 4, 8, \ldots, 128$. Each sample is based on a simulation run consisting of about 100M jobs. The y-axis corresponds to the relative mean waiting time (i.e., the price of ignorance, Def. 3.15)

$$\gamma_{\mathrm{RR}} = \frac{\mathrm{E}[W^{\mathrm{RR}}]}{\mathrm{E}[W^{\mathrm{RR}} \mid k = 1]},$$

and the x-axis is the offered load ρ. The service times are exponentially distributed, $X \sim \mathrm{Exp}(1)$. For comparison, the performance with RND is also shown.

In general, introducing multiple RR-dispatchers degrades the mean performance. We can make two interesting observations: (i) For any finite $\rho < 1$, the performance with k RR-dispatchers converges to that of RND when $k \to \infty$. This is the same observation as we made with multiple SITA-dispatchers. (ii) In contrast, as the offered load ρ tends to 1, i.e., in the heavy-traffic limit, all curves corresponding to RR with different numbers of dispatchers k converge to the same point as with one dispatcher! This suggests that no matter how large k is, RR still introduces some order in the system that is extremely valuable under heavy load. Note that RND has a significantly worse performance in this limit.

Thus, SITA and RR behave quite differently when it comes to multiple dispatchers. With SITA, the performance loss is independent of ρ, but increases as a function of k until it is equal to that of RND. RR also suffers from multiple dispatchers, especially under low load, but at a varying degree depending on the offered load. In particular, at the heavy traffic limit the price of ignorance vanishes and k RR dispatchers, unaware of each other, still work seamlessly together.

4.2 Convergence in the Heavy-traffic Limit

This peculiar behavior that a system with multiple RR-dispatchers and a system with a single (centralized) RR-dispatcher become equivalent (in terms of the mean waiting time) can be explained by sample path arguments. Let us refer to the single dispatcher system as System A and to the multi-dispatcher system as System B. Suppose both systems receive the same arrival pattern, which System B splits at random to its k dispatchers. We refer to time periods when all servers are busy as *busy periods*, and let T denote length of a busy period. As $\rho \to 1$, the queue lengths in both systems start to increase rapidly. The largest contributions to the mean waiting time are incurred during long busy periods, during which a large number of jobs are routed to n servers.

Let us consider m consecutive arrivals and let random variable C_m denote the number of jobs the dispatcher(s) assign to Server 1. Note that with RND and SITA, C_m has no other bounds than $0 \le C_m \le m$, i.e., the support of C_m grows linearly as a function of m. In fact, C_m obeys a binomial distribution, $C_m \sim \mathrm{Bin}(p, m)$, where p is the probability of routing a job to Server 1 (with RND, $p = 1/n$), and the variance $\mathrm{V}[C_m] = mp(1-p)$ grows linearly as a function of m.

With RR, the support of C_m is much smaller, and in particular, bounded, and so is the variance of C_m.

Lemma 4.2. *The number of jobs routed to Server 1 with k dispatchers is bounded:*

$$\left\lfloor \frac{m-(n-1)(k-1)}{n} \right\rfloor \le C_m^{(k)} \le \left\lceil \frac{m+(n-1)(k-1)}{n} \right\rceil. \tag{19}$$

In the case of a single RR dispatcher, the bounds reduce to

$$\left\lfloor \frac{m}{n} \right\rfloor \le C_m^{(1)} \le \left\lceil \frac{m}{n} \right\rceil. \tag{20}$$

Proof. As (20) follows directly from (19), we can focus on the case of k dispatchers and (19). For the lower bound, the slowest possible progress for $C_m^{(k)}$ is obtained with a pattern where the first $k(n-1)$ jobs are routed elsewhere before Server 1 receives its first job. After that, every n^{th} job is routed to Server 1, yielding

$$\left\lceil \frac{m-k(n-1)}{n} \right\rceil \le C_m^{(k)},$$

which reduces to

$$\left\lfloor \frac{m-(n-1)(k-1)}{n} \right\rfloor \le C_m^{(k)}.$$

The upper bound can be deduced similarly. The fastest possible progress for $C_m^{(k)}$ is attained with a pattern where the first k jobs are all routed to Server 1, and after that every n^{th} job, yielding

$$C_m^{(k)} \le k + \left\lfloor \frac{m-k}{n} \right\rfloor,$$

which is equivalent to

$$C_m^{(k)} \le \left\lceil \frac{m+(n-1)(k-1)}{n} \right\rceil.$$

□

Lemma 4.3. *With k RR dispatchers, the variance of $C_m^{(k)}$ is bounded,*

$$\mathrm{V}[C_m^{(k)}] < k^2.$$

Proof. Relaxing the bound (19) a bit, we can write

$$\frac{m}{n} - k < C_m^{(k)} < \frac{m}{n} + k,$$

and as the mean is within the same interval of length $2k$, we have

$$\mathrm{V}[C_m^{(k)}] \le k^2.$$

□

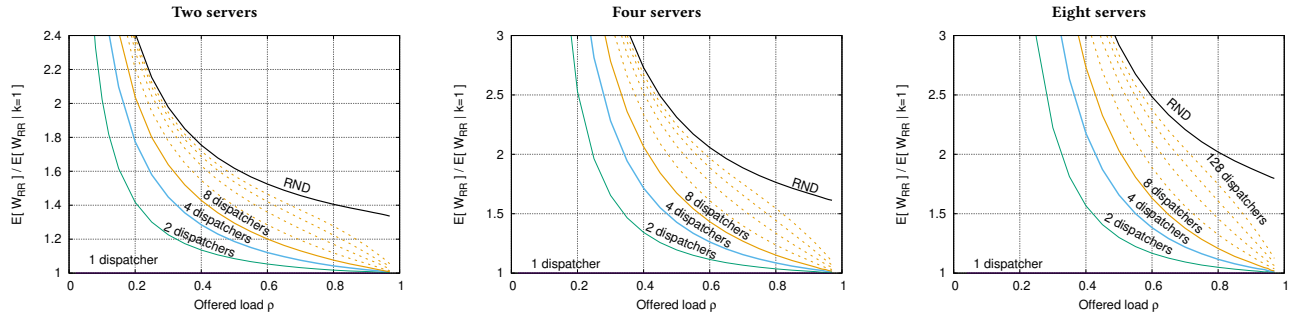

Figure 3: Simulation results with $n \in \{2, 4, 8\}$ servers and $k \in \{1, 2, 4, 8, \ldots, 128\}$ dispatchers with RR. For every $\rho < 1$, the performance deteriorates to that of RND as k increases. However, in the heavy traffic limit, the price of ignorance vanishes.

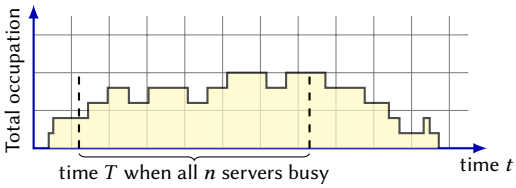

Figure 4: Illustration of a long busy period with RR.

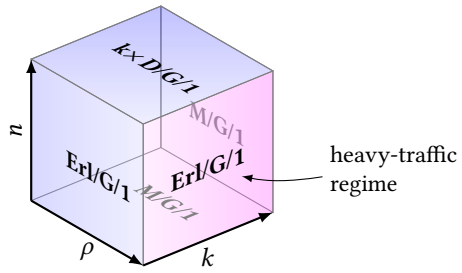

Figure 5: The *"Round-Robin cube"* illustrates the multi-dispatcher Round-Robin system at different limits.

Thus, even though multiple RR dispatchers introduce some randomness in a short timescale, at a longer timescale, "exactly" every n^{th} job will get assigned to Server 1 (and similarly to any other server). This is the sole reason why the performance of the two systems converges in the heavy traffic limit, where all (relevant) busy periods are (very) long.

Proposition 4.4. *The mean waiting time in a system with k RR dispatchers becomes equal to the mean waiting time with a single RR dispatcher in the heavy traffic limit where* $\rho \to 1$.

Proof. In the heavy-traffic regime, the main contribution to the mean waiting time is incurred during the long busy periods. Suppose that during such a busy period each dispatcher in System B routes many more than n jobs, so that in total $N \gg kn$ jobs are routed to the n servers. At the same time, the single dispatcher of System A routes basically the same N jobs to the n servers (just in a slightly different order in short timescale). According to the bound (19), there is very little room to *wiggle* and in any longer timescale, all servers receive the same number of jobs in both systems. As the jobs routed to different servers are statistically identical with RR, the total number of jobs in the two systems, on average, follow the same pattern. That is, referring to Figure 4, apart from the initial and final "transient" at the start and at the end of the busy period, which become negligible in the heavy traffic limit, the two systems behave essentially the same way in terms of the total number of jobs in the system. Thus, according to Little's law, the mean waiting times become equal as $\rho \to 1$. □

Corollary 4.5. *The price of ignorance with Round-Robin vanishes in the heavy-traffic limit where* $\rho \to 1$.

Considering the limit as $n \to \infty$, we have that dispatchers route jobs at practically constant time intervals to each server (phases between the dispatchers vary at much slower time scale), yielding the k×D/G/1 system. Moreover, the k×D/G/1 system is upper bounded by the D/G/1 queue with k-sized batch arrivals, yielding a bound for the price of ignorance in this limit.

The cube in Figure 5 summarizes the behavior of the multi-dispatcher Round-robin system in the different limits.

4.3 Light traffic

In contrast to the heavy-traffic limit, in light traffic the price of ignorance can be arbitrarily large. The mean waiting time with a single RR dispatcher in the light traffic regime (when $\rho \approx 0$) is [9],

$$\mathrm{E}[W^{\mathrm{RR}} \mid k = 1] \approx (n\rho)^n \cdot \frac{1}{\mu}.$$

With multiple dispatchers, the situation becomes worse and it is straightforward to show that

$$\mathrm{E}[W^{\mathrm{RR}} \mid k > 1] \approx \rho \, \frac{k-1}{k} \cdot \frac{1}{\mu}, \qquad \forall\, n > 1.$$

Consequently, in light traffic the price of ignorance is

$$\gamma_{\mathrm{RR}} \approx \frac{k-1}{n^n \rho^{n-1} k}, \qquad n, k > 1,$$

which tends to infinity as $\rho \to 0$.

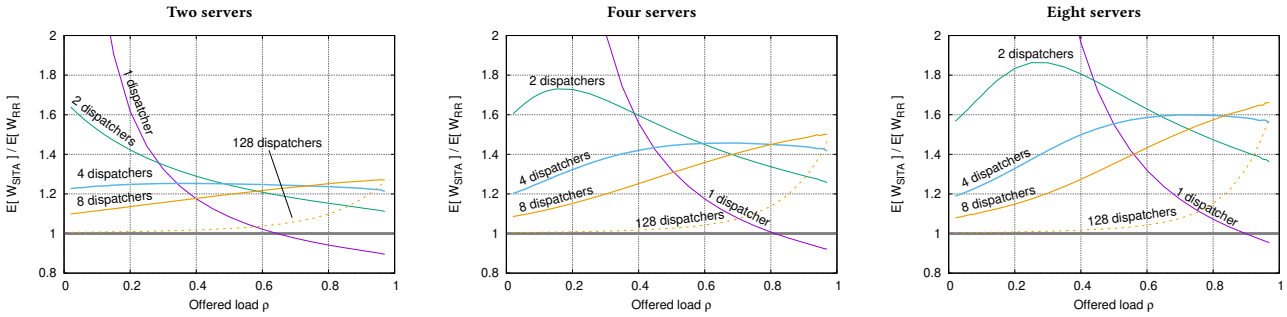

Figure 6: Simulation results with $n \in \{2, 4, 8\}$ servers and $k \in \{1, 2, 4, 8, 128\}$ dispatchers with RR and SITA. With k > 1 dispatchers, the performance with SITA is worse than with RR.

4.4 Comparison between RR and SITA

Figure 6 illustrates the performance ratio of SITA to RR with $n \in \{2, 4, 8\}$ servers, $k \in \{1, 2, 4, 8\}$ dispatchers and exponential service times. First we can observe that with a single dispatcher, the static SITA policy outperforms RR when the load is sufficiently high. Intuitively, at high load the variability in service times affects performance more than the variability in inter-arrival times. However, with $k \geq 2$, RR in consistently better than SITA. The difference is small with $n = 2$ servers, but tends to increase as a function of n.

From (12), we can deduce that the mean waiting time with SITA in light traffic behaves linearly,

$$\mathrm{E}[W^{\mathrm{SITA}}] \approx \frac{\rho}{2k\,\mathrm{E}[X]}\left((k-1)\mathrm{E}[X^2] + n\sum_i (b_i)^2 \mathrm{E}[X_i^2]\right),$$

and consequently, with $k > 1$ and $X \sim \mathrm{Exp}(\mu)$,

$$\frac{\mathrm{E}[W^{\mathrm{SITA}}]}{\mathrm{E}[W^{\mathrm{RR}}]} \approx 1 + \frac{n\mu^2}{2(k-1)}\sum_i (b_i)^2 \mathrm{E}[X_i^2].$$

That is, the ratio of the mean waiting times in the light traffic limit is $1 + g(n, X)/(k-1)$, where the SITA-specific factor $g(n, X)$ is independent of k. This dependence of the overall relative performance on k can be observed also from the simulation results.

5 CONCLUSIONS

Large computing systems often have multiple users operating their own dispatchers independently of each other (cf. cloud computing and virtual machines); this is one source of server-side variability. The assumed lack of coordination can lead to a performance degradation, a phenomenon we refer to as the *price of ignorance*. Our results show that: (i) The celebrated static policy SITA shows signs of weaknesses if dispatching policies are not coordinated. We give exact closed-form results and show that the performance without any coordination becomes equal to that of RND as the number of dispatchers increases (the many dispatchers limit). We also consider the many servers limit, where the performance advantage of SITA relative to RND is highest, and quantify the corresponding performance loss as a function of the number of dispatchers k. (ii) A similar pattern is observed with the Round-Robin policy, i.e., as the number of uncoordinated dispatchers increases, the performance decreases. However, in heavy traffic even the small amount of coordination provided by RR leads to a dramatic decrease in the mean waiting time and the price of ignorance vanishes.

ACKNOWLEDGMENTS

Our paper benefited from the extra time given as a consequence of the Corona virus.

REFERENCES

[1] Jeffrey Dean and Luiz André Barroso. 2013. The Tail at Scale. *Commun. ACM* 56, 2 (Feb. 2013), 74–80.
[2] J. Doncel, S. Aalto, and U. Ayesta. 2019. Performance Degradation in Parallel-Server Systems. *IEEE/ACM Transactions on Networking* 27, 02 (March 2019), 875–888.
[3] A. Ephremides, P. Varaiya, and J. Walrand. 1980. A simple dynamic routing problem. *IEEE Trans. Automat. Control* 25, 4 (Aug. 1980), 690–693.
[4] Hanhua Feng, Vishal Misra, and Dan Rubenstein. 2005. Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems. *Performance Evaluation* 62, 1-4 (2005), 475–492.
[5] K. Gardner, M. Harchol-Balter, and A. Scheller-Wolf. 2016. A Better Model for Job Redundancy: Decoupling Server Slowdown and Job Size. In *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 1–10.
[6] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytiä, and Alan Scheller-Wolf. 2015. Reducing Latency via Redundant Requests: Exact Analysis. *ACM SIGMETRICS Performance Evaluation Review* 43 (June 2015), 347–360. Issue 1. (ACM SIGMETRICS/Performance conference).
[7] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytiä, and Alan Scheller-Wolf. 2016. Queueing with Redundant Requests: Exact Analysis. *Queueing Systems* 83 (2016), 227–259. Issue 3.
[8] Mor Harchol-Balter. 2013. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action.* Cambridge University Press.
[9] Esa Hyytiä and Rhonda Righter. 2019. Simulation and performance evaluation of mission critical dispatching systems. *Performance Evaluation* 135 (Nov. 2019).
[10] Zhen Liu and Rhonda Righter. 1998. Optimal Load Balancing on Distributed Homogeneous Unreliable Processors. *Operations Research* 46, 4 (1998), 563–573.
[11] Zhen Liu and Don Towsley. 1994. Optimality of the Round-Robin Routing Policy. *Journal of Applied Probability* 31, 2 (June 1994), 466–475.
[12] Ashish Vulimiri, Philip Brighten Godfrey, Radhika Mittal, Justine Sherry, Sylvia Ratnasamy, and Scott Shenker. 2013. Low Latency via Redundancy. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT'13)* (Santa Barbara, California, USA). ACM, New York, NY, USA, 283–294.