Performance Degradation in Parallel-Server Systems with Shared Resources and Lack of Coordination

Esa Hyytiä and Rhonda Righter

November 8, 2021

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). The workload patterns of different users may also vary significantly. This results in unpredictable and often long 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 uncoordinated static 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 proper coordination is shown to increase the mean response times, with two common and robust dispatching policies: Size-Interval-Task Assignment (SITA) and Round-Robin (RR). We refer to this phenomenon as the price of ignorance.

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. We assume that server state information is unavailable to the users, and that they dispatch jobs to servers upon arrival. Multiple dispatchers may be needed for different reasons. For example, the sheer volume of jobs may be so large that also the (dispatching) load must be shared between multiple units. In this case, it makes sense to assume that the operation between different units can be coordinated. In contrast, our focus is in scenarios where multiple parties, unaware of each other, share the same resources without coordination.

Most of the literature for state-independent routing has assumed a single dispatcher. The most common static dispatching rules are random (RND) or Bernoulli splitting and round robin (RR), and, when job sizes are know, Size-Interval-Task-Assigntment (SITA). These can be shown to be optimal given certain informational constraints, and have been shown to work well in practice, see, e.g., [1, 2, 3] and [4, 5, 6, 7].

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. [8] 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, in [9], we considered the situation where dispatchers, unaware of each other, *share* the same set of servers. The dispatchers do not have a common ordering of the servers and dispatcher-specific workloads are homogeneous (arrival process and service time distribution). In this paper, an extended version of [9], we consider a significantly more general setting where the dispatcher-specific workloads are heterogeneous. Moreover, in addition to the scenario where the pool of servers is unordered and each dispatcher independently orders them randomly (as in [9]), we also consider the scenario where servers have a common order for all dispatchers and dispatchers assign their respective size-intervals accordingly to the same servers. The latter obviously improves the performance. We give exact closed-form results that quantify the performance with SITA in all possible cases. 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. We define the *"price of ignorance,"* to measure the performance degradation due to multiple uncoordinated

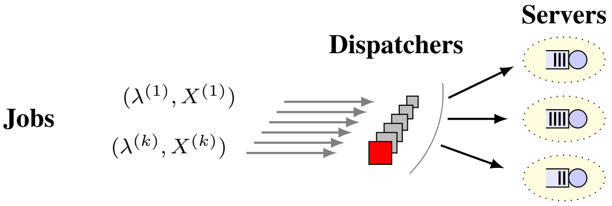

Figure 1: Multiple dispatching systems, unaware of each other, utilizing the same pool of servers.

Table 1: Notation:

n	the number of servers
k	the number of dispatchers
$\lambda^{(i)}$	the job arrival rate to dispatcher i
Λ	the total job arrival rate to the system, $\Lambda = \sum_i \lambda^{(i)}$
$X^{(i)}$	the (random) service time distribution at dispatcher i
X	the (random) service time distribution to the system
$\tilde{\lambda}$	the (random) job arrival rate to a fixed server
$\tilde{X}$	the (random) service time distribution at a fixed server
ρ	the offered load per server, $\rho = \Lambda\,\mathrm{E}[X]/n$
ρ_i	the offered load originating from dispatcher i
$\eta(x)$	the nominal cumulative load, $\eta(x) = \int_0^x t \cdot f(t)\,dt$

dispatchers, and show how it degrades in the number of dispatchers. We find, surprisingly, that under RR in heavy traffic the price of ignorance vanishes. This is not true for SITA.

We note that if dynamic policies, such as join-the-shortest queue (JSQ), are permitted, dispatcher coordination is not an issue. However, in many applications it is not possible to transmit server state information from all the servers to all the dispatchers. Attractive solutions that address the scalability exist. For example, one could employ the power-of-d or the join-the-idle-queue (JIQ) policies that scale well, see, e.g., [10, 11, 12] and [13]. However, both need appropriate feedback from the servers that may not be available.

Another important approach to dispatching is job replication [14, 15, 16, 17, 18, 19, 20]. For example, with redundancy-d, a dispatcher forwards a copy of the job to d randomly chosen servers. The copy that finishes first is used. The idea is that this way most jobs find a fast way through the system. However, due to multiple copies, load in the system tends to increase. Moreover, job replication requires extra communication between servers for cancellation message passing. In this paper, we focus on policies that do not require communication between servers and dispatchers, nor among dispatchers, nor among servers.

The rest of the paper is organized as follows. First, in Section 2, we introduce our model for the multi-dispatcher system. In Sections 3 and 4, we analyze SITA and RR policies, and Section 5 concludes the paper.

2 Model and Preliminaries

The system depicted in Figure 1 consists of k dispatchers receiving jobs according to Poisson processes at rates $\lambda^{(1)}, \ldots, \lambda^{(k)}$. Similarly, the job-size distributions are dispatchers specific, $X^{(1)}, \ldots, X^{(k)}$. The total arrival rate to the system is $\Lambda = \lambda^{(1)} + \ldots + \lambda^{(k)}$, and the job-sizes at the system level obey the corresponding compound distribution denoted by X. Dispatchers have *homogeneous workloads* when $\lambda^{(i)} = \Lambda/k$ and $X^{(i)} \sim X$.

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 servers. First-come-first-served (FCFS) scheduling is assumed. With FCFS, it is well-known that variability in job sizes translates to long delays (cf. the Pollaczek-Khinchine formula for the mean waiting time). The offered load (per server) is thus

$$\rho = \frac{\Lambda \mathrm{E}[X]}{n} = \rho_1 + \ldots + \rho_k,$$

where $\rho_i = \lambda^{(i)}\,\mathrm{E}[X^{(i)}]/n$ denotes the load originating from dispatcher i. It is assumed that $\rho < 1$ for stability. The notation is summarized in Table 1.

The basic performance metric is the mean waiting time for service (in queue). The key dimension we explore in this paper is the *performance degradation* due to uncoordinated dispatching decisions, i.e., *the price of ignorance*.

We assume that the dispatchers are completely unaware of each other. The pool of servers is either an unordered set or an ordered set. In the unordered case, each dispatcher uses a random numbering of the servers i.e., no coordination for the role of each server can take place. In the latter case, the pool of servers is assumed to be numbered and the dispatching policy can assign different "roles" based on this identification number. With SITA, this means assigning jobs of similar size to the same server. Otherwise, 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.

These scenarios may arise, e.g., in computing centers whenever multiple parties submit their tasks concurrently and independently of each other. The policy of the computer center may request users, e.g., to submit short jobs to certain servers.

The random split dispatching rule (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. If the pool of servers is ordered, then SITA can utilize this information to combine jobs of similar type together. With RR, the decision depends on which server the given dispatcher sent its 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 in these different scenarios 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 in the number of servers 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 for completeness (see [9]), and then consider a system with multiple dispatchers.

3.1 Single Dispatcher

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.

Two popular static dispatching policies are the Bernoulli split (RND) and the size-interval-task-assignment (SITA) [21].

Definition 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 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 [6] when job sizes are known and past decisions are unknown. 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

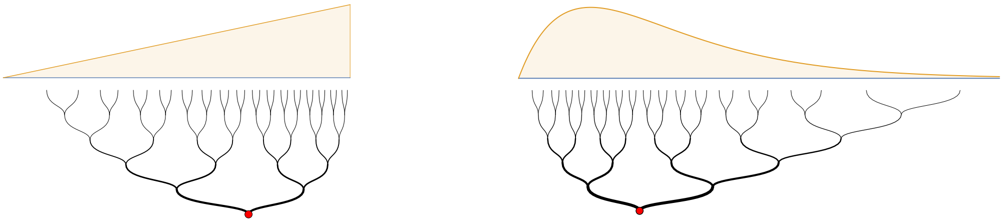

Figure 2: Thresholds ξ_i with SITA when $X \sim \mathrm{U}(0,2)$ and $X \sim \mathrm{Exp}(\mu)$ for $n = 2, 2^2, \ldots, 2^5$ servers. The upper curves correspond to $x\, f(x)$ and $\int_{\xi_{i-1}}^{\xi_i} x\, f(x)\, dx$ is equal to $\mathrm{E}[X]/n$ for all i.

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. (If we have sets of servers with different speeds, our analysis can be applied for dispatching within each homogeneous set.) 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 (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 into 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 1 *The mean waiting time with SITA is given by,*

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

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,*

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

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

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

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

Remark 3.1 *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 1

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

Proof: *From* (6)*, we need to show that (for $n \geq 2$)*

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

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 as the number of servers grows large.

Corollary 1 *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 2 *Assume X is a continuous random variable with a continuous pdf $f(x)$ such that $\mathrm{E}[X]$ and $\mathrm{E}[X^2]$ are finite. The mean waiting time for single-dispatcher systems with SITA in the limit as $n \to \infty$ with ρ fixed 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),*

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 with unbounded support so that

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

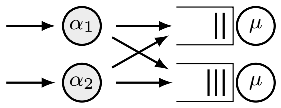

Figure 3: Two dispatchers routing jobs to a small pool of two servers.

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 (when* n *doubles, each size-interval is split into two). 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

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

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

3.2 Two Dispatchers

Suppose next that the system comprises two servers and two dispatchers, both routing an equal amount of work using SITA (see Figure 3). 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.

3.3 Multi-Dispatcher System without Any Coordination

Next we will generalize the system in several ways. First, the number of dispatchers is arbitrary. Second, the arrival rates to dispatchers can be heterogeneous. Third, also the job size distributions at different dispatchers are heterogeneous.

More specifically, let us assume that the system comprises k independent dispatchers routing jobs to n servers, as illustrated in Figure 1. Dispatcher i receives jobs according to a Poisson process at rate $\lambda^{(i)}$. Service time distributions, denoted by $X^{(i)}$, are also dispatcher-specific. The offered load to the system is

$$\rho = \frac{\Lambda\,\mathrm{E}[X]}{n} = \frac{1}{n} \sum_i \lambda^{(i)}\,\mathrm{E}[X^{(i)}].$$

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.

In the most elementary case, each dispatcher uses RND, and the mean waiting time in the system is

$$\mathrm{E}[W^{\mathrm{RND}}] = \frac{\sum_{i=1}^{k} \lambda^{(i)} \, \mathrm{E}[(X^{(i)})^2]}{2(1-\rho)n}. \tag{12}$$

Next we assume that dispatchers operate independently of each other, and according to SITA. In particular, each of them splits its own service time distribution $X^{(i)}$ into n intervals so that the load is balanced. Let $I_j^{(i)}$ denote the j^{th} size-interval of dispatcher i, and $X_j^{(i)}$ the corresponding conditional random variable, $X_j^{(i)} = (X^{(i)} | X^{(i)} \in I_j^{(i)})$. Unaware of each other, each dispatcher numbers the n servers independently at random. The mean waiting time in this system is given by the following result.

Proposition 3 *The expected waiting time in the system with k independent SITA dispatchers with heterogeneous workloads and n parallel servers is*

$$\mathrm{E}[W^{\mathit{SITA}}] = \frac{1}{2(1-\rho)} \sum_{i=1}^{k} \lambda^{(i)} \left[\frac{1-p_i}{n} \, \mathrm{E}[(X^{(i)})^2] + p_i \sum_{j=1}^{n} (b_j^{(i)})^2 \, \mathrm{E}[(X_j^{(i)})^2] \right], \tag{13}$$

where $p_i = \lambda^{(i)}/\Lambda$ is the probability that a job is from dispatcher i, $b_j^{(i)}$ denotes the probability that a job from dispatcher i belongs to the j^{th} size-interval, $b_j^{(i)} = \mathrm{P}\{X^{(i)} \in I_j^{(i)}\}$.

Proof: *We first determine the mean number of waiting jobs in a fixed queue, and then apply Little's result to the whole system. The situation is depicted in Figure 4. The mean number of jobs in the queue at an arbitrary server is*

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

and hence the mean number of jobs in all queues is

$$\mathrm{E}[N_q] = n \cdot \frac{\mathrm{E}[\tilde{\lambda}^2 \, \tilde{X}^2]}{2(1-\rho)}. \tag{14}$$

What remains is to determine $\mathrm{E}[\tilde{\lambda}^2 \, \tilde{X}^2]$. Let random variable Z_i denote the index of the size-interval that dispatcher i assigns to the given queue (see Figure 4). The Z_i are independent and uniformly distributed discrete random variables, $\mathrm{P}\{Z_i = j\} = 1/n$, $j = 1, \ldots, n$. Let $\lambda_j^{(i)}$ and $X_j^{(i)}$ be the arrival rate and random job size that dispatcher i sends to the fixed queue given $Z_i = j$. Then

$$\begin{aligned}
\mathrm{E}[\tilde{\lambda}^2 \, \tilde{X}^2] &= \mathrm{E}\left[\left(\sum_{i=1}^{k} \lambda_{Z_i}^{(i)} \right)^2 \cdot \frac{\sum_i \lambda_{Z_i}^{(i)} \mathrm{E}[(X_{Z_i}^{(i)})^2]}{\sum_i \lambda_{Z_i}^{(i)}} \right] \\
&= \mathrm{E}\left[\left(\sum_{i=1}^{k} \lambda_{Z_i}^{(i)} \right) \cdot \left(\sum_{i} \lambda_{Z_i}^{(i)} \mathrm{E}[(X_{Z_i}^{(i)})^2] \right) \right] \\
&= \mathrm{E}\left[\sum_{i=1}^{k} (\lambda_{Z_i}^{(i)})^2 \, \mathrm{E}[(X_{Z_i}^{(i)})^2] + \sum_{i=1}^{k} \lambda_{Z_i}^{(i)} \, \mathrm{E}[(X_{Z_i}^{(i)})^2] \cdot \sum_{j \neq i} \lambda_{Z_j}^{(j)} \right] \\
&= \sum_{i=1}^{k} \mathrm{E}\left[(\lambda_{Z_i}^{(i)})^2 \, \mathrm{E}[(X_{Z_i}^{(i)})^2] \right] + \sum_{i=1}^{k} \mathrm{E}\left[\lambda_{Z_i}^{(i)} \, \mathrm{E}[(X_{Z_i}^{(i)})^2] \right] \cdot \sum_{j \neq i} \mathrm{E}\left[\lambda_{Z_j}^{(j)} \right].
\end{aligned} \tag{15}$$

First,

$$\sum_{i=1}^{k} \mathrm{E}\left[(\lambda_{Z_i}^{(i)})^2 \, \mathrm{E}[(X_{Z_i}^{(i)})^2] \right] = \sum_{i=1}^{k} \frac{(\lambda^{(i)})^2}{n} \sum_{j=1}^{n} (b_j^{(i)})^2 \, \mathrm{E}[(X_j^{(i)})^2].$$

Second,

$$\sum_{i=1}^{k} \mathrm{E}\left[\lambda_{Z_i}^{(i)} \, \mathrm{E}[(X_{Z_i}^{(i)})^2] \right] = \sum_{i=1}^{k} \frac{\lambda^{(i)}}{n} \sum_{j=1}^{n} b_j^{(i)} \, \mathrm{E}[(X_j^{(i)})^2] = \sum_{i=1}^{k} \frac{\lambda^{(i)}}{n} \mathrm{E}[(X^{(i)})^2].$$

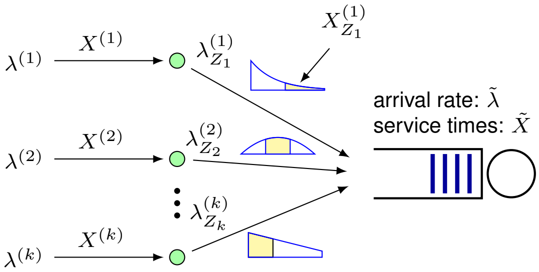

Figure 4: Derivation of Proposition 3, where a fixed server receives a randomly chosen compound arrival process.

Finally,

$$\sum_{j\neq i} \mathrm{E}\left[\lambda^{(j)}_{Z_j}\right] = \sum_{j\neq i}\sum_{s} \frac{1}{n}\lambda^{(j)}_s = \frac{1}{n}(\Lambda - \lambda_i).$$

Substituting these into (15)*, together with* (14)*, a yields*

$$\mathrm{E}[N_q] = \frac{1}{2(1-\rho)}\left(\sum_{i=1}^{k}(\lambda^{(i)})^2\sum_{j=1}^{n}(b^{(i)}_j)^2\,\mathrm{E}[(X^{(i)}_j)^2] + \frac{1}{n}\sum_{i=1}^{k}\lambda^{(i)}\,\mathrm{E}[(X^{(i)})^2](\Lambda-\lambda_i)\right). \qquad (16)$$

Therefore,

$$\mathrm{E}[W] = \frac{\mathrm{E}[N_q]}{\Lambda} = \frac{1}{2(1-\rho)}\sum_{i=1}^{k}\lambda^{(i)}\left[\frac{1-p_i}{n}\,\mathrm{E}[(X^{(i)})^2] + p_i\sum_{j=1}^{n}(b^{(i)}_j)^2\,\mathrm{E}[(X^{(i)}_j)^2]\right].$$

□

Corollary 2 *The mean waiting time with SITA can be expressed as*

$$\mathrm{E}[W^{\mathit{SITA}}] = \mathrm{E}[W^{\mathit{RND}}] - \frac{1}{2(1-\rho)}\sum_{i=1}^{k}\lambda^{(i)}p_i\left[\frac{1}{n}\,\mathrm{E}[(X^{(i)})^2] - \sum_{j=1}^{n}(b^{(i)}_j)^2\,\mathrm{E}[(X^{(i)}_j)^2]\right],$$

Note that with $k = 1$, (13) reduces to (4). Let us next consider what happens when the number of dispatchers is scaled up by factor m, $m = 1, 2, 3, \ldots$, while keeping the *"workload profile"* and offered load the same. That is, suppose the arrival rates are $\lambda^{(i+jm)} = \hat{\lambda}^{(i)}/m$ and the job size distributions are $X^{(i+jm)} = \hat{X}^{(i)}/m$, where $j = 0, \ldots, (m-1)$, and $\hat{\lambda}^{(i)}$ and $\hat{X}^{(i)}$ denote the arrival rate and job size distribution of class i (workload profile). Hence, also $p_i = \hat{p}_i/m$.

Corollary 3 *When the number of dispatchers is scaled by factor m for a fixed ρ and workload profile, we have*

$$\mathrm{E}[W^{\mathit{SITA}}] = \frac{1}{2(1-\rho)}\sum_{i=1}^{k}\frac{\lambda^{(i)}}{n}\left[(1-\frac{p_i}{m})\,\mathrm{E}[(X^{(i)})^2] + \frac{p_i}{m}n\sum_{j=1}^{n}(b^{(i)}_j)^2\,\mathrm{E}[(X^{(i)}_j)^2]\right], \qquad (17)$$

and because, for all i,

$$\frac{1}{n}\mathrm{E}[(X^{(i)})^2] > \sum_{j=1}^{n}(b^{(i)}_j)^2\,\mathrm{E}[(X^{(i)}_j)^2],$$

(see the proof of Proposition 1), it follows that $\mathrm{E}[W^{\mathit{SITA}}]$ increases in m. In particular, when $m\to\infty$,

$$\mathrm{E}[W^{\mathit{SITA}}] \to \mathrm{E}[W^{\mathit{RND}}].$$

Similarly, we can study the situation when the number of servers becomes large. In this case, we have the following result.

Corollary 4 *Keeping the ρ_i fixed as $n\to\infty$, we obtain in the limit that*

$$\mathrm{E}[W^{\mathit{SITA}}] \to \frac{1}{2(1-\rho)}\sum_{i}\rho_i\,\mathrm{E}[X^{(i)}]\left(1 + (1-p_i)\,c_v^2(X^{(i)})\right),$$

where $\rho_i = \lambda^{(i)}\,\mathrm{E}[X^{(i)}]$, $\rho = \rho_1 + \ldots + \rho_k$, and $c_v^2(X^{(i)})$ denotes the squared coefficient of variation of $X^{(i)}$.

3.4 Multi-Dispatcher System with Ordered Servers

Let us next assume that the server pool is ordered and each dispatchers knows which server is suppose to process which size interval (i.e., where to assign short jobs, and where to assign long jobs). If the dispatchers were homogeneous, then the performance would be the same as the performance with just one dispatcher. However, we assume that the arrival processes to dispatchers are heterogeneous, which will induce a performance degradation.

In this case, we can immediately write the solution,

$$\mathrm{E}[W^{\text{SITA}}_{\text{ordered}}] = \sum_{s=1}^{n} \frac{\Lambda_s}{\Lambda} \cdot \frac{\sum_{i=1}^{k} \lambda_s^{(i)}\, \mathrm{E}[(X_s^{(i)})^2]}{2(1-\rho)}, \tag{18}$$

where Λ_s denotes the total arrival rate to server s, $\Lambda_s = \lambda_s^{(1)} + \ldots + \lambda_s^{(k)}$. The ratio Λ_s/Λ is the probability that a job is routed to server s, and that is then multiplied by the server-specific mean waiting time.

3.5 Multi-Dispatcher System with Identical Workloads and Unordered Servers

Now we assume that all dispatchers receive jobs according to the same process and use SITA. That is, $\lambda^{(i)} = \Lambda/k$ and $X^{(i)} \sim X$. In this case, also 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. As in Section 3.3, we assume that dispatchers operate independently and assign the n job size intervals randomly to the n servers.

Corollary 5 *The mean waiting time in a multi-dispatcher system with k uncoordinated SITA dispatchers with homogeneous workload sharing n identical servers is given by*

$$\mathrm{E}[W^{\textit{SITA}}] = \frac{\rho}{2(1{-}\rho)\,\mathrm{E}[X]} \left[\frac{k-1}{k}\,\mathrm{E}[X^2] + \frac{n}{k} \sum_{i=1}^{n} (b_i)^2 \mathrm{E}[X_i^2] \right], \tag{19}$$

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

Proof: *This result follows from Proposition 3 as a special case. Namely, we let*

$$\lambda_s^{(i)} = \lambda_s,$$
$$X^{(i)} \sim X,$$

so that $X_s^{(i)} \sim X_s$ and $p_i = \lambda^{(i)}/\Lambda = 1/k$. Substituting these into (13) *gives*

$$\begin{aligned}
\mathrm{E}[W^{\textit{SITA}}] &= \frac{1}{2(1-\rho)} \sum_{i=1}^{k} \lambda \left[\frac{1-1/k}{n}\,\mathrm{E}[X^2] + \frac{1}{k} \sum_{j=1}^{n} (b_j)^2\, \mathrm{E}[(X_j)^2] \right], \\
&= \frac{\lambda k/n}{2(1-\rho)} \left[\frac{k-1}{k}\,\mathrm{E}[X^2] + \frac{n}{k} \sum_{j=1}^{n} (b_j)^2\, \mathrm{E}[(X_j)^2] \right].
\end{aligned}$$

The offered load is $\rho = \lambda k\,\mathrm{E}[X]/n$, so that

$$\frac{\lambda k}{n} = \frac{\rho}{\mathrm{E}[X]},$$

which yields (19). □

Note that with $k = 1$, (19) reduces to (4). Corollary 5 has several important further corollaries, especially for larger systems. First, comparing the expressions (19) (for SITA) and (3) (for RND) reveals the following compact relationship that generalizes the result of Remark 3.1 to systems with $k > 1$ dispatchers:

Corollary 6 *The mean waiting time in a multi-dispatcher system with k SITA dispatchers and n servers is given by*

$$\mathrm{E}[W^{\textit{SITA}}] = \mathrm{E}[W^{\textit{RND}}] \left(1 - \tfrac{r_n}{k}\right), \tag{20}$$

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

The next corollary is an immediate consequence of Corollary 6, for systems with a large number of dispatchers, $k \gg 1$:

Corollary 7 (Many dispatchers limit) *For any fixed load $\rho < 1$, when the number of servers n is kept constant, $\mathrm{E}[W^{\mathit{SITA}}]$ is increasing in k, and when the number of dispatchers tends to infinity,*

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

Remark 3.2 *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 2 for $k \geq 1$ dispatchers.

Corollary 8 (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*

$$\mathrm{E}[W^{\mathit{SITA}}] \to \frac{\rho}{2(1-\rho)\mathrm{E}[X]} \left[\mathrm{E}[X^2] - \frac{\mathrm{Var}[X]}{k} \right], \; \textit{as } n \to \infty. \tag{21}$$

Proof: *With SITA, it holds that $\mathrm{E}[X_i] b_i = \mathrm{E}[X]/n$. Arguing as in Proposition 2, 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* (19) *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 (19) *we can deduce that*

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

which yields (21). □

Remark 3.3 *Comparing* (20) *and* (21) *reveals that*

$$\lim_{n \to \infty} r_n = \frac{\mathrm{Var}[X]}{\mathrm{E}[X^2]} = \frac{c_v^2(X)}{c_v^2(X) + 1}, \tag{22}$$

where $c_v^2(X)$ 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 (20).

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

$$\Delta \triangleq \mathrm{E}[W_{\text{uc}}] - \mathrm{E}[W_{\text{c}}],$$

where the subscript uc refers to the uncoordinated system, and subscript c to the system with coordination (equivalently, with $k = 1$). For SITA with homogeneous workloads we have,

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

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):

Definition 4 (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,*

$$\gamma \triangleq \frac{\mathrm{E}[W_{uc}]}{\mathrm{E}[W_c]}.$$

Table 2: Performance degradation with SITA due to multiple uncoordinated dispatchers. Coefficients $\mathbf{r_n}$ in the limit $n \to \infty$ are obtained from (22) : $1/4$, $1/2$ and $5/6$.

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

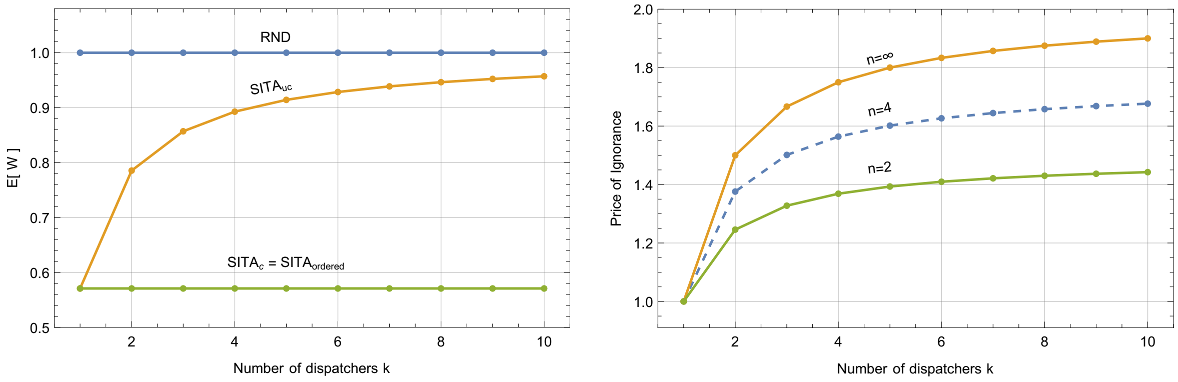

Figure 5: Left: The mean waiting time with RND, SITA without coordination (SITA$_{\mathrm{uc}}$) and SITA with coordination (SITA$_{\mathrm{c}}$) with homogeneous exponentially distributed workloads with $n = 4$ servers and $\rho = 0.5$. With homogeneous workloads, the full coordination is achieved if the servers are ordered. Right figure shows the price of ignorance for systems with 2, 4 and ∞ servers.

Corollary 9 (Price of ignorance with SITA) *With SITA and homogeneous workloads, 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 3.4 *Note that the price of ignorance for RND is* 1*, i.e., (lack of) coordination has no effect. Moreover, from* (6)*, the price of using RND relative to SITA is*

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

3.6 Numerical Examples

In this section, we give some numerical examples that illustrate our results and the price of ignorance with the SITA dispatching policy.

Example 3.5 *Let us first assume that workloads at dispatchers are identical (see Section 3.5). 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*

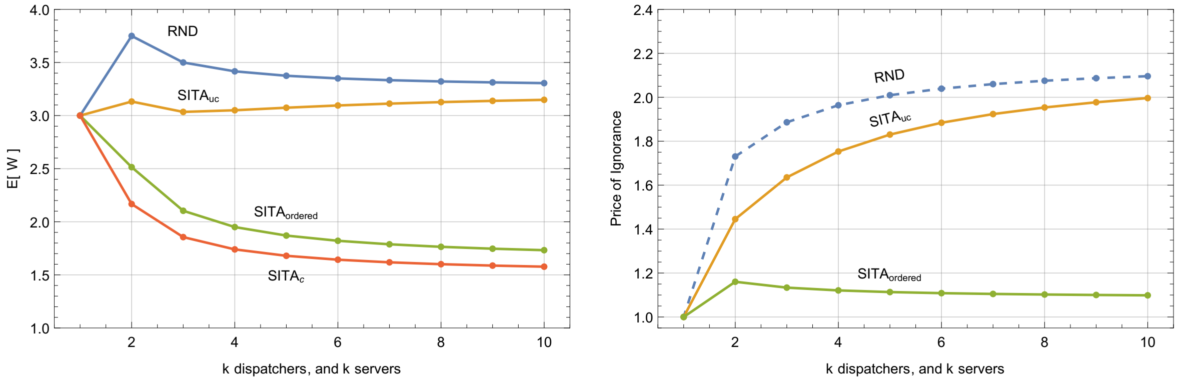

Figure 6: Performance with heterogeneous exponentially distributed workloads with k dispatchers and k servers when $\rho = 0.6$. If the servers are ordered, the performance decreases only moderately in this case. The figure on the left depicts the mean waiting time, and the figure on the right the price of ignorance. Note that as $k = n$ increases, also the overall workload changes.

observe that the marginal gain from adding one more server quickly diminishes. Moreover, from (20) *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!*

Example 3.6 *Suppose we have $n = 4$ servers and k dispatchers with equal exponentially distributed workloads: $\lambda^{(i)} = \Lambda/k$, and $X^{(i)} \sim$ Exp(1). Figure 5(left) depicts the mean waiting time with RND, SITA$_{uc}$ and SITA$_c$ for $\rho = 0.5$ as a function of k. The price of ignorance is evident and the performance with SITA$_{uc}$ quickly deteriorates to the proximity of RND. The numerical values are in agreement with the values obtained for Exp(1) distribution in the previous example. Figure 5(right) depicts the price of ignorance when $n = 2$, 4 and ∞ (large system limit).*

Example 3.7 *Let us next fix the load, $\rho = 0.6$, and scale the system so that $n = k$. Moreover, the service time distributions are dispatcher specific: $X^{(i)} \sim$ Exp(μ_i), where*

$$\frac{1}{\mu_i} = 1 + 2\frac{i-1}{k-1},$$

so that the mean is 2. Note that the overall workload distribution offered to the system changes as k increases. Figure 6 depicts the mean waiting time and price of ignorance with RND, SITA$_{uc}$, SITA$_{ordered}$ and SITA$_c$ as a function of k. The price of ignorance is again striking without any coordination. However, if the servers are ordered, i.e., there is a common understanding between the dispatchers on what kind of jobs each server should receive, then the performance degradation is reasonable across all values of k. This suggests that the companies who offer computing resources to their customers should let their customers know about preferred usage of servers.

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., [1, 2, 3]. 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 5 *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.

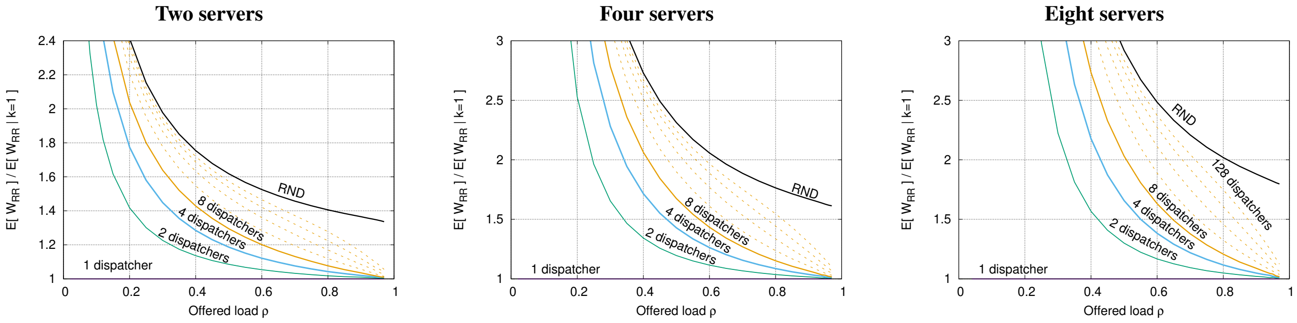

Figure 7: Simulation results with $n \in \{2, 4, 8\}$ servers and $k \in \{1, 2, 4, 8, \ldots, 128\}$ dispatchers with RR. For every $\rho < 1$, the performance deteriorates to that of RND as k increases. However, in the heavy traffic limit, the price of ignorance vanishes.

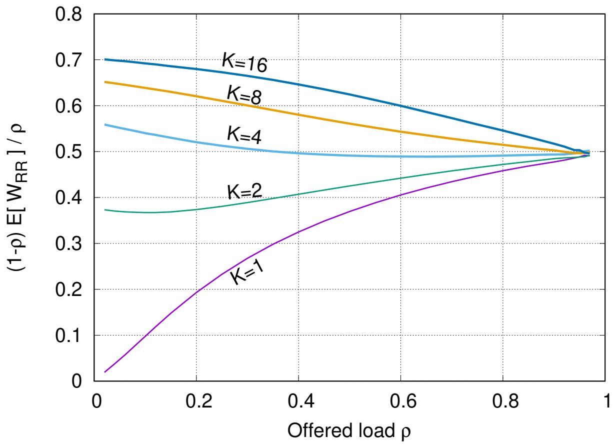

Figure 8: Simulation results with generalized round-robin and heterogeneous servers.

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 7 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. 4)

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

and the x-axis is the offered load ρ. The service times are exponentially distributed, $X \sim \text{Exp}(1)$. 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 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.

4.1.1 Heterogeneous Service Rates

Next we consider a heterogeneous scenario where Server 1 is twice as fast as Servers 2 and 3, i.e., $n = 3$. Dispatchers are aware of the relative service rates and apply a generalized round-robin policy with pattern $1, 2, 1, 3, 1, 2, \ldots$ that balances the load. (See [22] for generalized RR for servers with heterogeneous speeds.) The job size distribution is exponential with unit mean.

Figure 8 depicts the mean waiting time $\mathrm{E}[W]$ scaled by $\rho/(1-\rho)$ as a function of ρ. Similarly as before, the performance deteriorates as the number of dispatchers increases. However, all curves seems to converge at the heavy-traffic limit when $\rho \to 1$. Moreover, the performance in this limit appears to be the same as with (standard) round-robin with 4 homogeneous servers.

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 randomly 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 $\mathrm{Var}[C_m] = mp(1-p)$ 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 2 *The number of jobs routed to Server 1 with k dispatchers is bounded:*

$$\left\lfloor \frac{m-(n-1)(k-1)}{n} \right\rfloor \le C_m^{(k)} \le \left\lceil \frac{m+(n-1)(k-1)}{n} \right\rceil . \tag{23}$$

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

$$\left\lfloor \frac{m}{n} \right\rfloor \le C_m^{(1)} \le \left\lceil \frac{m}{n} \right\rceil . \tag{24}$$

Proof: *As* (24) *follows directly from* (23)*, we can focus on the case of k dispatchers and* (23)*. 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 3 *With k RR dispatchers, the variance of $C_m^{(k)}$ is bounded,*

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

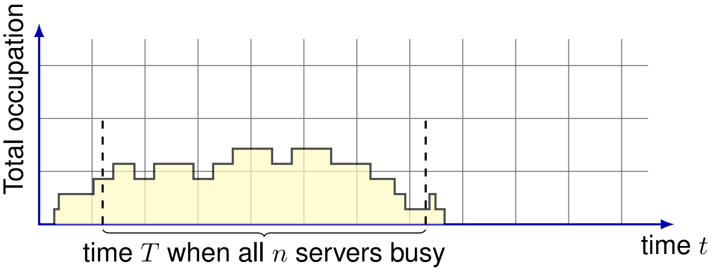

Figure 9: Illustration of a long busy period with RR.

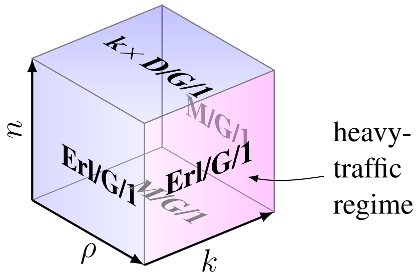

Figure 10: The *"Round-Robin cube"* illustrates the multi-dispatcher Round-Robin system at different limits.

Proof: *Relaxing the bound* (23) *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*

$$\mathrm{Var}[C_m^{(k)}] \leq k^2.$$

□

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 *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* (23)*, 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 9, 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 10 *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 $k\times$D/G/1 system. Moreover, the $k\times$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 10 summarizes the behavior of the multi-dispatcher Round-robin system in the different limits.

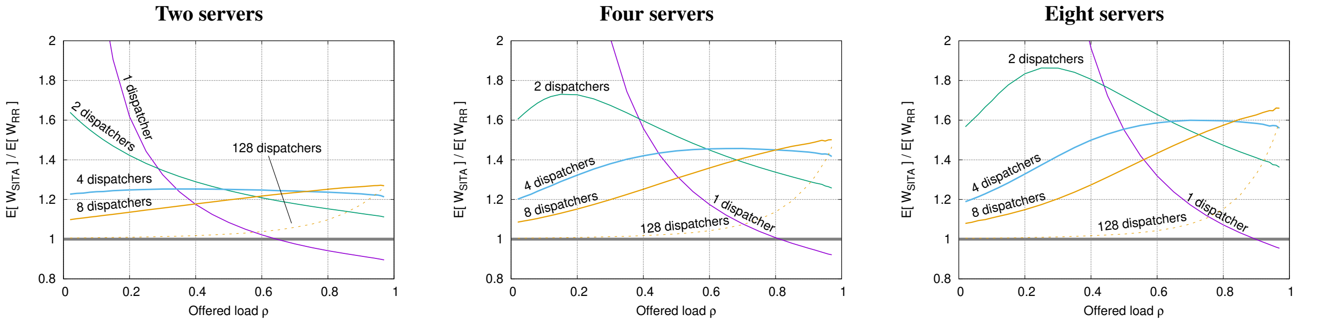

Figure 11: Simulation results with $n \in \{2, 4, 8\}$ servers and $k \in \{1, 2, 4, 8, 128\}$ dispatchers with RR and SITA. With $k > 1$ dispatchers, the performance with SITA is worse than with RR.

4.3 Light traffic

In contrast to the heavy-traffic limit, in light traffic the price of ignorance for RR can be arbitrarily large. The mean waiting time with a single RR dispatcher in the light traffic regime (when $\rho \approx 0$) is [23],

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

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 $\rho \to 0$.

4.4 Comparison between RR and SITA

Figure 11 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. Indeed, the (relative) performance of SITA would be better if $c_v^2(X) > 1$. 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 (19), 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

In large computing facilities, the available resources are often shared between 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*. We consider such systems when users apply either a static SITA or the classical round-robin dispatching policy. Both policies scale well and allow independent operation. With SITA, we consider two cases: either the servers are commonly ordered, or unordered. In the former scenario, all dispatchers can agree on where they send their respective "short" and "long" jobs, whereas in the latter, each dispatchers assigns job types to servers randomly. 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. Our results assume homogeneous servers. If there were pools of servers that are homogeneous within a pool, the results would carry over within each pool.

References

[1] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, Aug. 1980.

[2] Z. Liu and D. Towsley, "Optimality of the round-robin routing policy," *Journal of Applied Probability*, vol. 31, no. 2, pp. 466–475, Jun. 1994.

[3] Z. Liu and R. Righter, "Optimal load balancing on distributed homogeneous unreliable processors," *Operations Research*, vol. 46, no. 4, pp. 563–573, 1998.

[4] M. E. Crovella, M. Harchol-Balter, and C. D. Murta, "Task assignment in a distributed system: Improving performance by unbalancing load," in *Proceedings of SIGMETRICS '98*, Madison, Wisconsin, USA, Jun. 1998, pp. 268–269.

[5] M. Harchol-Balter, M. E. Crovella, and C. D. Murta, "On choosing a task assignment policy for a distributed server system," *Journal of Parallel and Distributed Computing*, vol. 59, pp. 204–228, 1999.

[6] H. Feng, V. Misra, and D. Rubenstein, "Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems," *Performance Evaluation*, vol. 62, no. 1-4, pp. 475–492, 2005.

[7] E. Bachmat and H. Sarfati, "Analysis of SITA policies," *Performance Evaluation*, vol. 67, no. 2, pp. 102–120, 2010.

[8] J. Doncel, S. Aalto, and U. Ayesta, "Performance degradation in parallel-server systems," *IEEE/ACM Transactions on Networking*, vol. 27, no. 02, pp. 875–888, Mar. 2019.

[9] E. Hyytiä and R. Righter, "Performance degradation in parallel-server systems with shared resources," in *13th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '20)*, Tsukuba, Japan, May 2020.

[10] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, pp. 1094–1104, Oct. 2001.

[11] O. Akgun, R. Righter, and R. Wolff, "The power of partial power of two choices," in *ACM SIGMETRICS*, San Jose, California, USA, Jun. 2011.

[12] T. Hellemans and B. Van Houdt, "On the power-of-d-choices with least loaded server selection," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 2, no. 2, pp. 27:1–27:22, Jun. 2018.

[13] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services,” *Performance Evaluation*, vol. 68, no. 11, pp. 1056–1071, 2011.

[14] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker, “Low latency via redundancy,” in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT’13)*. New York, NY, USA: ACM, 2013, pp. 283–294.

[15] J. Dean and L. A. Barroso, “The tail at scale,” *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013.

[16] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytiä, and A. Scheller-Wolf, “Reducing latency via redundant requests: Exact analysis,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, pp. 347–360, Jun. 2015, (ACM SIGMETRICS/Performance conference).

[17] ——, “Queueing with redundant requests: Exact analysis,” *Queueing Systems*, vol. 83, pp. 227–259, 2016.

[18] K. Gardner, M. Harchol-Balter, and A. Scheller-Wolf, “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)*, Sep. 2016, pp. 1–10.

[19] T. Bonald and C. Comte, “Balanced fair resource sharing in computer clusters,” *Performance Evaluation*, vol. 116, pp. 70–83, 2017.

[20] E. Anton, U. Ayesta, M. Jonckheere, and I. M. Verloop, “On the stability of redundancy models,” *Operations Research*, 2021.

[21] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[22] E. Altman, B. Gaujal, and A. Hordijk, “Balanced sequences and optimal routing,” *J. ACM*, vol. 47, no. 4, p. 752–775, Jul. 2000.

[23] E. Hyytiä and R. Righter, “Simulation and performance evaluation of mission critical dispatching systems,” *Performance Evaluation*, vol. 135, Nov. 2019.