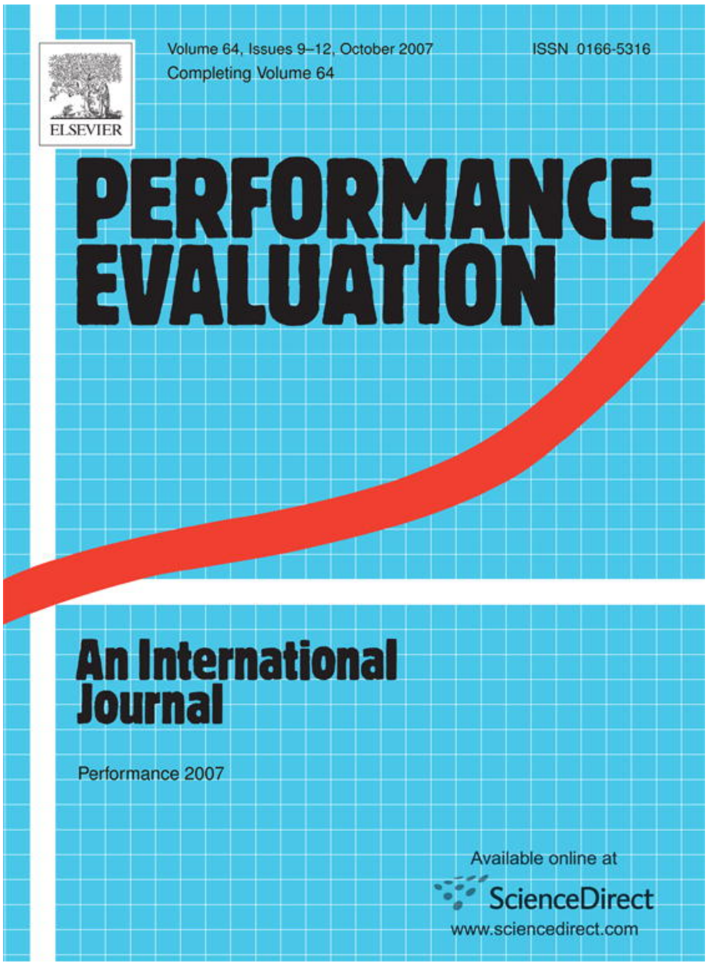

Volume 64, Issues 9–12, October 2007
Completing Volume 64
ISSN 0166-5316
ELSEVIER
PERFORMANCE EVALUATION
An International Journal
Performance 2007
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect

Performance Evaluation 64 (2007) 1062–1081

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

Analysis of join-the-shortest-queue routing for web server farms

Varun Gupta[a], Mor Harchol Balter[a,*], Karl Sigman[b], Ward Whitt[b]

[a] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[b] Department of Industrial Engineering and Operations Research, Columbia University, NY 10027, USA

Available online 28 June 2007

Abstract

Join the Shortest Queue (JSQ) is a popular routing policy for server farms. However, until now all analysis of JSQ has been limited to First-Come-First-Serve (FCFS) server farms, whereas it is known that web server farms are better modeled as Processor Sharing (PS) server farms. We provide the first approximate analysis of JSQ in the PS server farm model for general job-size distributions, obtaining the distribution of queue length at each queue. To do this, we approximate the queue length of each queue in the server farm by a one-dimensional Markov chain, in a novel fashion. We also discover some interesting insensitivity properties of PS server farms with JSQ routing, and discuss the near-optimality of JSQ.

Keywords: Shortest queue routing; JSQ; Processor sharing; Insensitivity; Single-queue approximation

1. Introduction

1.1. Motivation

The server farm is a popular architecture of computing centers. A server farm consists of a front-end *router/dispatcher* which receives all the incoming requests (jobs), and dispatches each job to one of a collection of *servers* which do the actual processing. The dispatcher employs a routing policy (also called a "task assignment policy", or TAP), which decides when and to which server an incoming request should be routed. Server farms afford low cost (many slow servers are cheaper than one fast server), high scalability (it is easy to add and remove servers) and high reliability (failure of individual servers does not bring the whole system down). One of the most important design goals of a server farm is choosing a routing policy which will yield low response times; the response time is the time from the arrival of a request to its completion.

We are motivated by web server farm architectures serving static requests. Requests for files (or HTTP pages) arrive at a front-end dispatcher. The dispatcher then *immediately* routes the request to one of the servers in the farm for processing using a routing policy. It is important that the dispatcher does not hold back the arriving connection request, or the client will time out and possibly submit more requests. The bottleneck resource at a web server is often the uplink bandwidth. This bandwidth is shared by all files requested in a round-robin manner with a small granularularity,

[*] Corresponding author.
E-mail address: harchol@cs.cmu.edu (M. Harchol Balter).

0166-5316/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2007.06.012

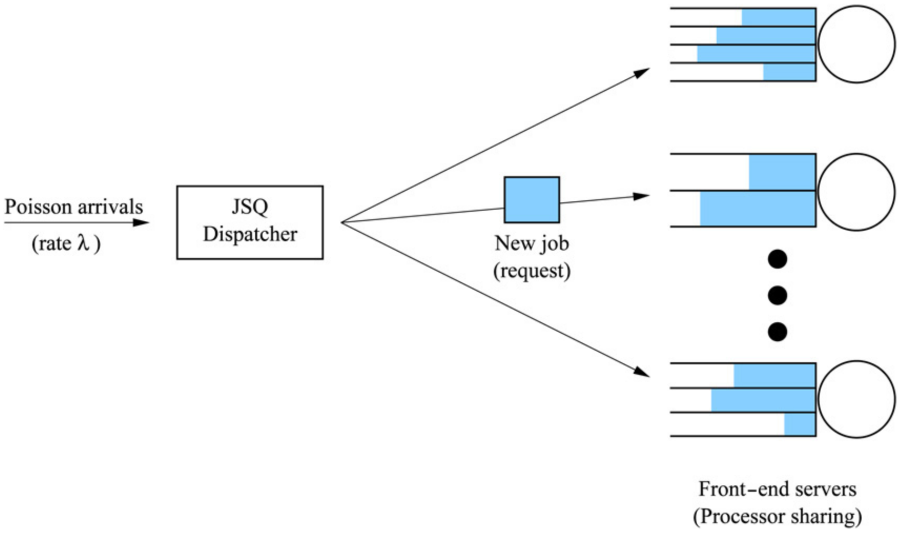


Fig. 1. Server farm with front-end dispatcher and K identical processor sharing back-end servers.

which is well-modeled by the idealized processor sharing (PS) scheduling policy [18]. Under PS scheduling, the server splits its capacity equally over the requests it is processing, giving an equal share of its capacity to each of the current requests at every instant of time. We are thus interested in a *PS server farm with immediate dispatch*. Time sharing servers are beneficial in that they allow "short jobs" to get processed quickly, without being stuck waiting behind long jobs. This is particularly important, since measurements have shown that requested files' sizes, and the associated service requirements, are highly variable, (e.g., heavy-tailed [4,10]).

The *Join-the-Shortest-Queue* (JSQ) routing policy is the most popular routing policy used in PS server farms today; e.g., it is used in Cisco Local Director, IBM Network Dispatcher, Microsoft Sharepoint and F5 Labs BIG/IP. Under JSQ, an incoming request is routed to the server with the least number of unfinished requests. Thus, JSQ strives to balance load across the servers, reducing the probability of one server having several jobs while another server sits idle. From the point of view of a new arrival, it is a *greedy policy* for the case of PS servers, because the arrival would prefer sharing a server with as few jobs as possible. We refer to a PS server farm with JSQ routing as a *JSQ/PS server farm*.

1.2. Model and notation

We model the arrival process of jobs as a stationary Poisson process.[1] We assume that there is a single dispatcher (router) and K identical PS servers with unlimited waiting space, as depicted in Fig. 1. We assume that routing is immediate using the JSQ policy. Ties are broken by randomly choosing (with equal probabilities) among the servers with the fewest jobs. No jockeying is allowed between the servers (once a job is dispatched to a server, it stays there until completion). A job's *size* (service requirement) is defined as the time taken by a job to run on a server in isolation.

Consequently, the JSQ/PS server farm acts as an $M/G/K/JSQ/PS$ queueing model, with JSQ denoting the policy used to route arrivals to the servers and PS denoting the scheduling rule (service discipline) used by each server. Jobs arrive as a Poisson stream (the M) with rate λ and are routed immediately to one of the K servers with the fewest jobs. The service requirements are drawn independently from a general distribution with mean μ^{-1} (the G) and service is performed at each server according to PS. We define the load of this system, ρ, as the per-server load $\rho = \lambda/(K\mu)$. We sometimes use the extra notation $M(\lambda)/G(\mu)/K/JSQ/PS$ to denote that the average arrival rate is λ and the mean job size is μ^{-1}. We will use N to denote the random variable for the queue length of a *single* PS queue in the server farm.

[1] This is consistent with measurements, except that measurements invariably show that the arrival rate varies strongly by time of day. However, in the short time scale over which we analyze performance (say, minutes), the arrival rate usually can be regarded as constant. The request pattern of individual users is typically far from Poisson, but as in many applications, a Poisson arrival process becomes justified because the overall arrival process is the superposition of relatively sparse arrival processes from many nearly independent users. We can then invoke the classical limit theorem establishing convergence to the Poisson process, as in Proposition 9.2.VII on p. 285 of Daley and Vere-Jones [11].

1.3. Contributions/outline

Despite the ubiquity of JSQ/PS server farms, no one has yet analyzed the performance of JSQ in this setting. The existing analysis of JSQ involves *First-Come-First-Serve (FCFS) server farms*, where the servers employ FCFS scheduling. Within the JSQ/FCFS setting, almost all analysis is restricted to 2 servers, often with exponentially-distributed job sizes. For more than 2 servers, while some very appealing approximations exist, the accuracy of those approximations decreases as the number of servers is increased or as the job-size distribution becomes more variable. Prior work is detailed in Section 2.

In this paper we provide the first analysis of the JSQ/PS model. In particular, we provide a way to calculate the approximate steady-state distribution of queue length (number of jobs in the system) at any server, which also yields the mean response time via Little's Law. While our analysis is approximate, the accuracy of our approximation is extremely good: $<3\%$ error for mean response time and only slightly more for the second moment of queue length. More importantly, the error does *not* seem to increase beyond 3% with increased numbers of servers, or with an increase in job-size variability.

1.3.1. SQA

We accomplish this goal in what we believe is an interesting innovative way. In Section 3 we introduce a new approximation technique for server farms, which we call the *single-queue approximation* (SQA). Besides being useful for JSQ/PS server farms, SQA should apply to a much larger class of multi-server systems with state-dependent routing policies. The key idea behind SQA is the following: instead of analyzing the entire multi-server model, we just concentrate on a single queue in the server farm, say queue Q, and model its behavior *independently* of all the other queues. To capture the effect of the other queues, without directly considering them, we model the arrival process into queue Q by a stochastic point process with state-dependent rates. In particular, we assume that the arrival process into queue Q has stochastic intensity $\lambda(N_Q(t))$, where $N_Q(t)$ is the queue length of Q at time t and $\lambda(n)$ is the long-run arrival rate when Q has n customers in the original multi-server model.

We provide strong theoretical support for SQA: In Theorem 3.1 of Section 3 we prove that SQA is in fact *exact* when the job-size distribution is exponential (given exact conditional arrival rates). Thus,

$$M/M/K/JSQ/PS \overset{SQA}{\equiv} M_n/M/1/PS,$$

where equivalence denotes equivalent steady-state queue-length distributions.

1.3.2. Near-insensitivity

Turning to general job-size distributions, in Section 4, we investigate the sensitivity of the $M/G/K/JSQ/PS$ to the variability of G. In Section 4.1, we prove that under a class of distributions, the *degenerate hyperexponential* (H_2^*), the mean response time of a JSQ/PS server farm depends on the job-size distribution only through its mean. That is, even when the parameters of the degenerate hyperexponential are set to create very high variability, mean response time is unaffected, as we prove in Theorem 4.1. Coupled with the above equation, we now have:

$$M/H_2^*/K/JSQ/PS \equiv M/M/K/JSQ/PS \overset{SQA}{\equiv} M_n/M/1,$$

where equivalence denotes equivalent queue-length distributions.

To examine other job-size distributions, we resort to extensive simulations of a wide class of distributions, including hyperexponential distributions, Erlang distributions, Weibull distributions, the deterministic distribution and bimodal distributions (mixture of two point masses). We find, see Section 4.2, that the JSQ/PS system shows *near-insensitivity* to the variability of the job-size distribution. Coupled with the above equation, we now have:

$$M/G/K/JSQ/PS \approx M/M/K/JSQ/PS \overset{SQA}{\equiv} M_n/M/1,$$

where the approximation is quite close for at least the first two moments of queue length.

In Section 4.3, we discuss *intuition* for the near-insensitivity of JSQ/PS server farms. First, we note that the insensitivity of the $M/G/1/PS$ queue (which is well-known) extends also to the (state-dependent) $M_n/G/1/PS$ queue (a less known fact). Second, we demonstrate that the conditional arrival rates—the $\lambda(n)$—are also nearly insensitive to the job-size distribution. Finally, we point out that the fact that the $M/G/K/JSQ/PS$ server farm exhibits near-

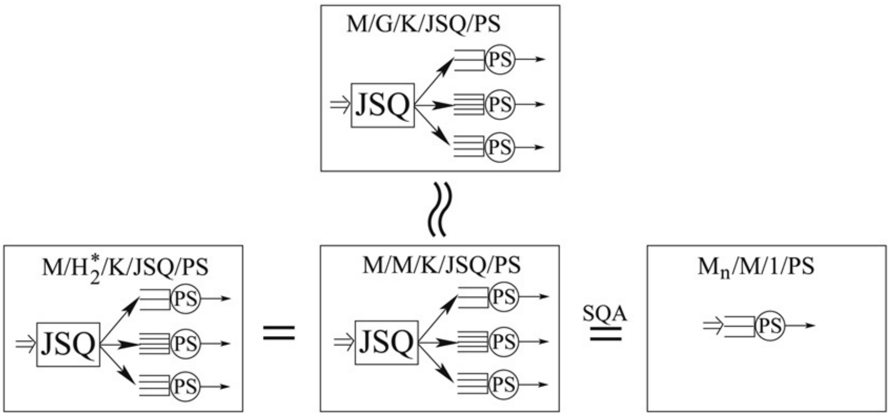


Fig. 2. Pictorial view of results in the paper.

insensitivity is non-trivial, since very similar routing policies for PS server farms, like Least-Work-Left (sending the job to the host with the least total work), or Round-Robin, are highly sensitive to the job-size distribution.

1.3.3. Conditional arrival rates

At this point, it appears that we have a method for analyzing JSQ/PS server farms with general job-size distributions: As shown in Fig. 2, we approximate the $M/G/K/JSQ/PS$ by an $M/M/K/JSQ/PS$, which we prove is equivalent to an $M_n/M/1$, which we then solve. However, there is an important unresolved issue: We have not explained how to derive the conditional arrival rates, the $\lambda(n)$'s, into the $M_n/M/1$ queue. This is the subject of Section 5. To determine the $\lambda(n)$'s, we began by measuring them through extensive simulation experiments. Fortunately, we found stunning regularity in the results. We observed that $\lambda(n) \approx \mu\rho^K$ for all $n \geq 3$. We further support this observation by Theorem 5.1, showing that

$$\frac{\lambda(n)}{\mu} \to \rho^K, \quad \text{as } n \to \infty$$

in the case where $K = 2$. Given the above observations, it suffices to determine only the three remaining parameters: $\lambda(0)$, $\lambda(1)$ and $\lambda(2)$, which we determine using a combination of analysis and simulation. We thus obtain closed-form expressions for all the conditional arrival rates $\lambda(n)$ as a function of the three parameters λ, μ and K. With those formulas, we can get a closed-form solution for the queue-length distribution, since the approximating $M_n/M/1$ model is a repeating birth-and-death process.

1.3.4. High accuracy

In Section 6, we demonstrate the remarkable accuracy of our approximation method under a wide array of job-size distributions, where we use our derived conditional arrival rates $\lambda(n)$. We show that our analytical approximation method is always within 2.5% of simulation estimates for mean queue length and response time, under all job-size distributions examined. Furthermore, this percentage error does not appear to increase as K is increased from 2 to 16. The maximum error only rises from 2.5% to 3.5% when we look at the second moment of queue length.

1.3.5. Where prior work fits in

Fig. 2 demonstrates pictorially some of the results in this paper. It is important to note that one is not forced to use the SQA approximation in the rightmost equality of Fig. 2. Once we know that: $M/G/K/JSQ/PS \approx M/M/K/JSQ/PS \equiv M/M/K/JSQ/FCFS$, we can then apply any known method in the literature to solve the $M/M/K/JSQ/FCFS$, not just SQA. As mentioned earlier, the literature is full of methods for analyzing the $M/M/K/JSQ/FCFS$ for the case of $K = 2$; even for $K > 2$, there are some attractive approximations by Nelson and Philips [24] or by Lin and Raghavendra [22].

1.3.6. Near-optimality of JSQ

We end the paper in Section 7 by presenting simulation results comparing JSQ with other routing policies in the PS server farm setting. We find that JSQ is impressively close to achieving optimality, despite using far less information about jobs than the other routing policies against which it is compared.

2. Prior work

We emphasize again that there has been no previous mathematical analysis of the $M/G/K/JSQ/PS$ model. However, Bonomi [5] conducted a simulation study for the special case of two servers. He showed that, among all policies that base their decisions only on the queue lengths at the servers, JSQ minimizes the mean response time for the PS scheduling rule and exponential service requirements. Bonomi also proposed policies that improve slightly upon JSQ (5% improvement), for some general job-size distributions, by exploiting the remaining service times of jobs. He showed via simulation that common load-balancing schemes that perform well for JSQ/FCFS do not perform well for JSQ/PS. Bonomi observed that, while Least-Work-Left (LWL) is good for FCFS, it is not good for PS. However, we find that LWL is not always bad; see Fig. 7.

In contrast, there is a lot of work on the JSQ/FCFS model (recall that under exponential workloads, JSQ/FCFS is equivalent to JSQ/PS with respect to the stationary queue-length distribution). However, even the $M/M/K/JSQ/FCFS$ model remains quite intractable. Several authors, including Weber [26], Winston [29], and Ephremides et al. [13], consider the optimality of JSQ for FCFS servers in certain constrained settings involving a job-size distribution with non-decreasing failure rate and various assumptions on not knowing job sizes a priori. Note, however, that JSQ is far from optimal for FCFS servers with highly-variable job sizes [9,17].

Almost all papers analyzing JSQ/FCFS performance are limited to 2 servers, an exponential job-size distribution and the mean response time metric. Among the classic papers are Kingman [20] and Flatto and McKean [14]. They use generating functions to derive the joint probability distribution of queue lengths and express the mean response time as an infinite sum, which in practice requires truncation to compute. Adan et al. [2] show that Kingman's result can be derived more intuitively via the compensation approach. Approximations for the mean response time have also been obtained by state space truncation of the Markov chain [16,8,25,23]. Heavy traffic approximations for JSQ/FCFS also exist and are evaluated in [15,21]. Lastly, Boxma and Cohen [6] obtain a functional representation for the mean response time using boundary value approach. These methods are exact. However they are not always computationally efficient and do not generalize to higher values of K.

For analyzing JSQ/FCFS with more than $K = 2$ servers, again with exponential job sizes, only approximations exist. Again, the metric is mean response time. Nelson and Philips [24] use the following idea: They look at the steady-state probability of the $M/M/K/FCFS$ queue (with a central queue) as an estimate for the total number of jobs in the JSQ/FCFS system, and then assume that the jobs in the system are divided equally (within 1) among each of the queues. Lin and Raghavendra [22] follow the approach of approximating the number of busy servers by a binomial distribution and then also assume that the jobs are equally divided among each of the queues (within 1). Both approximations are reasonably accurate. Specifically, the Nelson and Philips method demonstrates error less than 8% for K up to 16 with exponentially-distributed job sizes. They also provide an empirically obtained correction factor which drops the error to 2%. Lin and Raghavendra method yields less than 3.5% error for exponentially-distributed job sizes and K up to 64. There are also some numerical methods papers that do not lead to a closed-form solution, but are accurate and computationally efficient for not-too-large K, see for example [1,23,3].

3. The single-queue approximation (SQA)

To understand SQA, it helps to recall that the main obstacle in analyzing routing policies such as JSQ is that the states of all the queues are correlated, necessitating a multidimensional state space for the system. Thus exact analysis requires that we work with the vector of queue lengths and possibly also the remaining service requirements of all jobs at each server. The SQA method allows one to approximate the marginal queue-length distribution of each queue in the server farm by modeling each queue independently of the other, thereby avoiding the above difficulties.

Consider a queue Q in the server farm. Under SQA, we model Q by a queue Q', where the arrival rate of jobs into Q' can depend only on the queue length of Q', and not on the state of any other queues. Thus SQA approximates each queue of the $M/G/K/JSQ/PS$ model by an associated $M_n/G/1/PS$ model, where M_n denotes a state-dependent Markovian arrival process. Specifically, at time t, the arrival process acts as a Poisson process with rate $\lambda(N_{Q'}(t))$, where $N_{Q'}(t)$ is the queue length of Q' at time t and $\{\lambda(n) : n \geq 0\}$ is a deterministic sequence with $\lambda(n)$ being the actual long-run arrival rate into queue Q (of the original server farm) conditioned on the queue length of Q being n. We define $\lambda(n)$ in Definition 3.1.

Definition 3.1. Given a general $M/G/K/\mathcal{R}/S$ model, the conditional arrival rate into one designated queue Q given that it has n jobs, $\lambda(n)$, is defined as

$$\lambda(n) = \lim_{t\to\infty} \frac{A_n(t)}{T_n(t)}, \tag{1}$$

where $A_n(t)$ is the number of arrivals into Q during the time interval $[0, t]$ that see n jobs at Q on arrival (excluding themselves), while $T_n(t)$ is the total time spent by Q with n jobs during the time interval $[0, t]$.

Formally, the arrivals form a stochastic point process with stochastic intensity $\lambda(N_{Q'}(t))$, as defined in Section II.3,5 in Brémaud [7].

The state-dependence in the arrival rate $\lambda(n)$ is intended to capture some of the dependence inherent in the full *M/G/K/JSQ/PS* model. Consider an *M/G/K/JSQ/PS* model with outside arrival rate λ. The average arrival rate into each queue is λ/K. However, if we condition on the fact that some designated queue has n jobs, then the arrival rate into that designated queue is no longer λ/K. In fact, with JSQ routing, we expect that the long-term arrival rates into that designated queue, $\lambda(n)$, should *decrease* as n increases, because it is likely that at least one other queue is shorter than the designated queue. This is precisely what happens: $\lambda(0)$ is larger than λ/K, but $\lambda(n)$ decreases as n increases. In this way, having state-dependent arrival rates captures some of the influence of the other queues on the designated queue.

The SQA approximation method is not limited to the *M/G/K/JSQ/PS* model we are primarily considering. We can consider other routing policies $\mathcal{R}$ (see e.g., Definition 3.2) for the K-server model and other scheduling rules $\mathcal{S}$ at this single queue. We can also accommodate heterogeneous servers. SQA can also be defined for a more general arrival process, but its performance under general arrival processes remains to be investigated. We now specify a class of routing policies for which SQA works well.

Definition 3.2. A *stationary queue-length-dependent routing policy* is a time-stationary routing policy that uses only information about queue lengths at the servers at the instant of an arrival. The decisions may be made probabilistically, and may be biased in favor of certain servers (allowing the modeling of heterogeneous servers).

We now show that SQA produces the exact stationary queue-length distribution for Markovian models (when the actual arrival process is Poisson and the job sizes are independent exponential random variables with a common distribution) and the routing and scheduling rules satisfy certain regularity conditions.

Theorem 3.1. *Consider an $M/M/K/\mathcal{R}/\mathcal{S}$ model, where $\mathcal{R}$ is any stationary queue-length-dependent routing policy, e.g., JSQ, and $\mathcal{S}$ is any stationary, size-independent, work-conserving scheduling policy, e.g., PS. Assume that this model has a unique proper steady-state distribution. Let Q be any particular server in the $M/M/K/\mathcal{R}/\mathcal{S}$ model. Then SQA with the exact conditional arrival rates $\lambda(n)$ yields the same steady-state queue-length distribution as in the original $M/M/K/\mathcal{R}/\mathcal{S}$ model.*

Proof. For simplicity, we will assume that $K = 2$, but it is easy to see that the proof can be extended to any number of servers, allowing unequal service rates. From the assumptions about the job-size distribution, the routing policy and the scheduling rule, the 2-dimensional vector of queue lengths evolves as a continuous-time Markov chain (CTMC) with stationary transition probabilities. By assumption, this CTMC has a unique steady-state distribution, where $\pi_{n,j}$ is the steady-state probability that there are n jobs at queue 1 and j jobs at queue 2 (including the jobs receiving service, if any). We will concentrate on queue 1. Let μ denote the service rate at queue 1. Let Π_n denote the steady-state probability that there are n jobs at queue 1. Clearly,

$$\Pi_n = \sum_{j=0}^{\infty} \pi_{n,j}.$$

As above, let $\lambda(n)$ denote the conditional arrival rate at queue 1 in the $M/M/K/\mathcal{R}/\mathcal{S}$ model. Let $M_n/M/1/\mathcal{S}$ denote the SQA model for queue 1, which has state-dependent arrival rates $\lambda(n)$. Let x_n denote the limiting probability that there are n jobs at the single $M_n/M/1/\mathcal{S}$ queue with state-dependent arrival rates $\lambda(n)$. Since the queue-length process in the $M_n/M/1/\mathcal{S}$ model is a birth-and-death process, x_n is the unique solution (after normalization) to the following balance equations:

$$x_n\lambda(n) = x_{n+1}\mu, \quad n \geq 0. \tag{2}$$

Our goal is to prove that $\Pi_n = x_n$, $n \geq 0$. To do this, we will show that Π_n is a solution to (2).

We need only two observations to show this: Our first observation is that we can rewrite $\lambda(n)$ as follows: Let $q_{n,j}$ denote the probability that the (time-stationary) routing policy $\mathcal{R}$ in the $M/M/K/\mathcal{R}/\mathcal{S}$ model routes an incoming job to queue 1 when n and j are the number of jobs at queue 1 and 2, respectively. By definition of $\lambda(n)$, we must have

$$\lambda(n) = \sum_{j=0}^{\infty} \left(\frac{\pi_{n,j}}{\Pi_n}\right) \cdot \lambda q_{n,j}. \tag{3}$$

Our second observation is that we can balance the rate of transitions between the set of states $S_n = \{(n, j) : j = 0, 1, \ldots\}$ and $S_{n+1} = \{(n+1, j) : j = 0, 1, \ldots\}$ in the $M/M/K/\mathcal{R}/\mathcal{S}$ model as follows:

$$\sum_{j=0}^{\infty} \pi_{n,j} \cdot \lambda q_{n,j} = \sum_{j=0}^{\infty} \pi_{n+1,j}\mu. \tag{4}$$

Results of this type using conditional arrival and departure rates have been obtained previously using sample path arguments, see [12] (Theorem 1.9, Section 1.4.2, page 21). We can now easily show that Π_n is a solution to (2), because

$$\begin{aligned} \Pi_n\lambda(n) &= \Pi_n \sum_{j=0}^{\infty} \left(\frac{\pi_{n,j}}{\Pi_n}\right) \cdot \lambda q_{n,j} && \text{by (3)} \\ &= \sum_{j=0}^{\infty} \pi_{n,j} \cdot \lambda q_{n,j} = \sum_{j=0}^{\infty} \pi_{n+1,j}\mu = \Pi_{n+1}\mu. && \text{by (4)} \quad \square \end{aligned}$$

4. Insensitivity

In Section 3 we showed that SQA can exactly model the $M/M/K/JSQ/PS$ system (exponential job-size distribution). We now look at general job-size distributions. We start in Section 4.1 with the case of a degenerate hyperexponential job-size distribution, and then consider more general distributions in Section 4.2. We provide some further support for SQA in Section 4.3 by observing that the insensitivity property of the $M/G/1/PS$ extends to the $M_n/G/1/PS$.

4.1. Insensitivity with the degenerate hyperexponential distribution

In this subsection we introduce a special two-parameter family of job-size distributions for which the $M/G/K/JSQ/PS$ model has the insensitivity property. This family is a subset of the hyperexponential (H_2) distributions (mixtures of two exponentials), which we refer to as *degenerate hyperexponential* distributions and denote by H_2^*. (The H_2^* distribution has been used previously to approximately capture the variability of job sizes in multi-server systems, e.g., [27,28].)

Definition 4.1. A random variable X distributed according to the H_2^* distribution with mean $1/\mu$ and squared coefficient of variation (variance divided by the square of the mean) C^2, denoted by $H_2^*(\mu^*, p)$, is given by

$$X \sim \begin{cases} 0 & w.p.\ p \\ \exp(\mu^*) & w.p.\ 1-p, \end{cases}$$

where $p = (C^2 - 1)/(C^2 + 1)$ and $\mu^* = \mu(1-p)$.

The degenerate hyperexponential distribution is a relatively minor modification of the exponential distribution, but the modification provides an additional parameter, so that it can be used to represent a full range of variability in the job-size distribution, with a squared coefficient of variation C^2 ranging from 1 to ∞. The next result shows that if the job sizes are drawn from an H_2^* distribution, then the steady-state queue-length distribution and the mean response time in the resulting $M/H_2^*/K/JSQ/PS$ model depend only on the mean job size, and not on the remaining free parameter; i.e., we have insensitivity within this H_2^* class.

Theorem 4.1. *The queueing systems $M/H_2^*((1-p)\mu, p)/K/JSQ/PS$ and $M/M(\mu)/K/JSQ/PS$ (both with mean job size $1/\mu$) have identical steady-state queue-length distributions and mean steady-state response times. Moreover, the* *response-time distribution of the $M/H_2^*((1-p)\mu, p)/K/JSQ/PS$ system is a mixture of a unit point mass at* 0*, with probability p, and the response-time distribution of the $M/M(\mu)/K/JSQ/PS$ system multiplied by* $1/(1-p)$*, with probability* $1-p$*.*

Proof. The jobs with size 0 do not have to wait, since the servers are doing processor sharing. Therefore, with respect to the queue-length distribution, we have that:

$$\Pi_n^{M(\lambda)/H_2^*(\mu^*,p)/K/JSQ/PS} = \Pi_n^{M(\lambda(1-p))/M(\mu(1-p))/K/JSQ/PS} \tag{5}$$

$$= \Pi_n^{M(\lambda)/M(\mu)/K/JSQ/PS}. \tag{6}$$

From the perspective of response time, the response time of the p-proportion of zero-sized jobs is the deterministic distribution with mean 0, while the remaining $(1-p)$-proportion of non-zero-sized jobs experience an $M(\lambda(1-p))/M(\mu(1-p))/K/JSQ/PS$ system. But the $M(\lambda(1-p))/M(\mu(1-p))/K/JSQ/PS$ system is the same as the $M(\lambda)/M(\mu)/K/JSQ/PS$ system seen on a slower time scale, slowed by a factor of $1/(1-p)$. Thus the $(1-p)$-proportion of non-zero-sized jobs experience a response time $1/(1-p)$ times higher than that in an $M(\lambda)/M(\mu)/K/JSQ/PS$ system. □

4.2. Near-insensitivity for all job-size distributions

The insensitivity of Section 4.1 is for very special job-size distributions. We will show that this insensitivity property does not extend exactly to other job-size distributions, but that it does for all practical purposes; i.e., we have *near-insensitivity*. To establish those conclusions, we simulated an $M/G/K/JSQ/PS$ system with the following job-size distributions (all with mean 2, in increasing order of C^2):

1. `Deterministic`: point mass at 2 (variance = 0)
2. `Erlang2`: sum of two exponential random variables with mean 1 (variance = 2)
3. `Exponential`: exponential distribution with mean 2 (variance = 4)
4. `Bimodal-1`: (mean = 2, variance = 9)

$$X = \begin{cases} 1 & w.p.\ 0.9 \\ 11 & w.p.\ 0.1 \end{cases}$$

5. `Weibull-1`: Weibull with shape parameter = 0.5 and scale parameter = 1 (heavy-tailed, mean = 2, variance = 20)
6. `Weibull-2`: Weibull with shape parameter = $\frac{1}{3}$ and scale parameter = $\frac{1}{3}$ (heavy-tailed, mean = 2, variance = 76)
7. `Bimodal-2`: (mean = 2, variance = 99)

$$X = \begin{cases} 1 & w.p.\ 0.99 \\ 101 & w.p.\ 0.01. \end{cases}$$

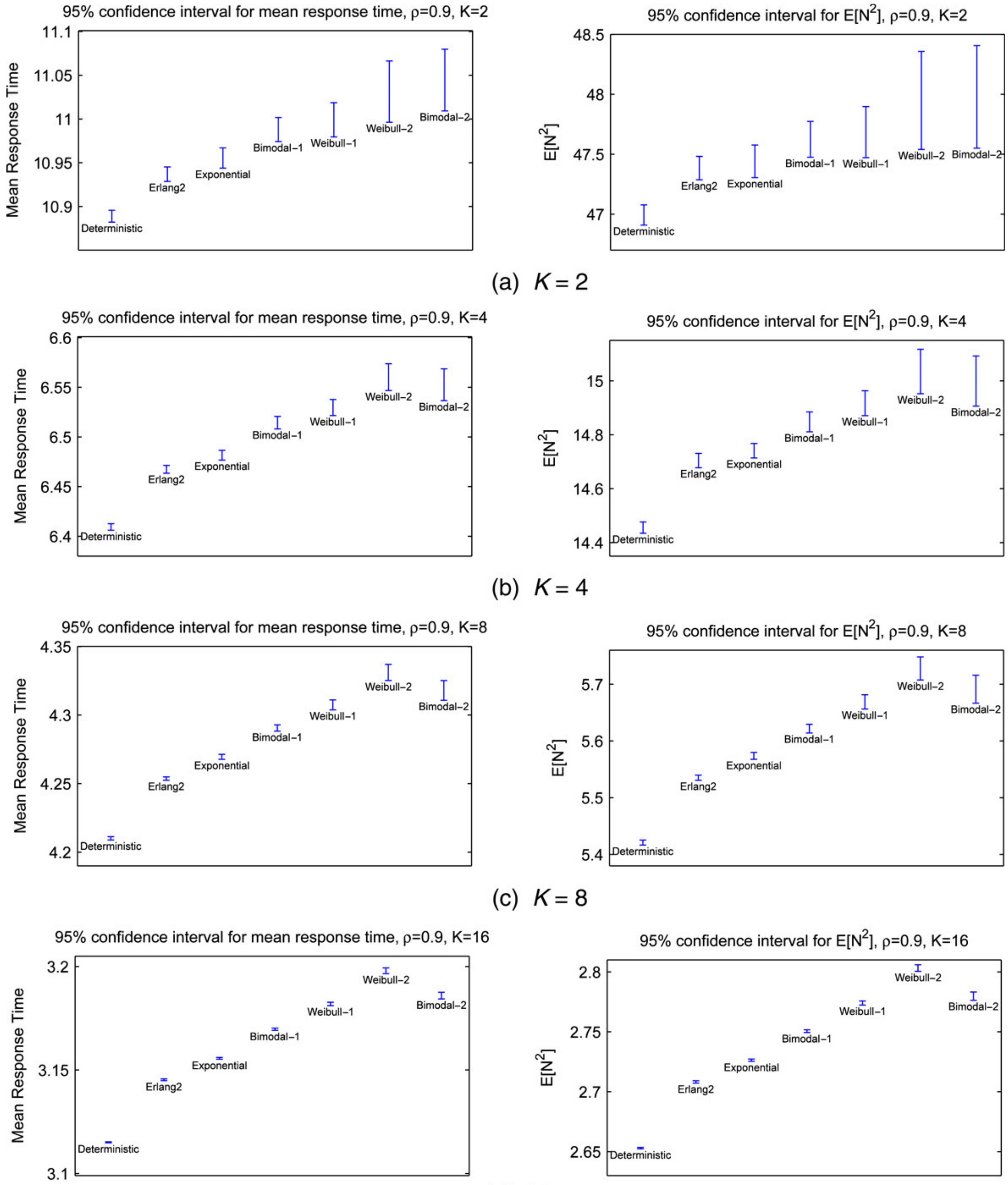
The load was set at $\rho = 0.9$ and simulations were run for $K = 2, 4, 8$ and 16 servers. For each value of K and each distribution, the simulation was run 50 times, each run consisting of $K \times 10^7$ departures. (These are long runs!) Statistics for completed requests were considered. Fig. 3 shows the 95% confidence intervals for the mean response time and second moment of queue length, for each service distribution and number of servers, K. The mean response time in Fig. 3 never deviates by more than 2% from the exponential case, regardless of the job-size distribution, and the deviation for the second moment of queue length is barely over 3%.

4.3. Intuition behind insensitivity results

In this section we provide additional support for the near-insensitivity of the $M/G/K/JSQ/PS$ model, and for the SQA technique.

On first thought, one might assume that the near-insensitivity of the $M/G/K/JSQ/PS$ model stems directly from the well-known insensitivity of the $M/G/1/PS$ queue. This does not explain everything, since, the arrival process into an individual queue under JSQ is *not* Poisson.

A more relevant piece of intuition is that insensitivity of the $M/G/1/PS$ also extends to the (state-dependent) $M_n/G/1/PS$. While this fact is not well-known, a proof of this fact follows directly from general results on symmetric queues (processor sharing is a symmetric discipline); see Theorems 3.10 and 3.14 on pp. 78, 90 in Kelly [19].



(a) $K = 2$

(b) $K = 4$

(c) $K = 8$

(d) $K = 16$

Fig. 3. 95% confidence intervals for mean response time (left column) and second moment of queue length (right column) in the $M/G/K/JSQ/PS$ model with $\rho = 0.9$ and mean job size 2 for different job-size distributions based on simulations.

Theorem 4.2. *Consider the $M_n/G/1/PS$ model. The (time-stationary) steady-state distribution of the number in system is insensitive to the service-time distribution G beyond its mean. Moreover, conditional on there being n jobs in the system in stationarity, the n (unordered) remaining service times are i.i.d. with the equilibrium density.*

Although we have thus far always thought of SQA as being applied to the $M/M/K/JSQ/PS$ queue, Theorem 4.2 provides justification for viewing SQA as being applied directly to an $M/G/K/JSQ/PS$ server farm, reducing it to an

Table 1
Conditional arrival rates for $M/H_2/K/JSQ/PS$ with $K = 4$ and $\rho = 0.9$, where the hyperexponential (H_2) distribution has parameters C^2 and r with mean 1, and the variability of H_2 ranges from $C^2 = 1$ to $C^2 = 64$. Results from simulation. (Conditional arrival rates for $M/D/K/JSQ/PS$ are also shown for reference in the top line.)

		$\lambda(0)$	$\lambda(1)$	$\lambda(2)$	$\lambda(3)$	$\lambda(4)$	$\lambda(5)$	$\lambda(6)$	$\lambda(7)$	$\lambda(8)$	$\lambda(9)$
$C^2 = 0$		2.2379	0.9865	0.6931	0.6575	0.6605	0.6645	0.6678	0.6696	0.6722	0.6727
$C^2 = 1$	$r = 0.1$	2.2136	0.9966	0.7098	0.6622	0.6562	0.6543	0.6557	0.6572	0.6583	0.6578
	$r = 0.5$	2.2125	0.9962	0.7098	0.6631	0.6573	0.6550	0.6543	0.6551	0.6549	0.6538
	$r = 0.9$	2.2136	0.9956	0.7084	0.6622	0.6562	0.6535	0.6522	0.6555	0.6564	0.6588
$C^2 = 2$	$r = 0.1$	2.2080	1.0000	0.7123	0.6629	0.6541	0.6516	0.6542	0.6514	0.6537	0.6474
	$r = 0.5$	2.2074	0.9975	0.7119	0.6609	0.6520	0.6522	0.6525	0.6494	0.6504	0.6532
	$r = 0.9$	2.2077	0.9947	0.7114	0.6649	0.6560	0.6554	0.6557	0.6553	0.6518	0.6527
$C^2 = 4$	$r = 0.1$	2.2068	1.0041	0.7144	0.6611	0.6513	0.6531	0.6522	0.6487	0.6486	0.6542
	$r = 0.5$	2.2018	0.9992	0.7150	0.6653	0.6585	0.6553	0.6520	0.6511	0.6494	0.6547
	$r = 0.9$	2.2075	0.9971	0.7110	0.6630	0.6572	0.6560	0.6549	0.6526	0.6535	0.6536
$C^2 = 16$	$r = 0.1$	2.2032	1.0092	0.7201	0.6641	0.6544	0.6521	0.6536	0.6495	0.6515	0.6418
	$r = 0.5$	2.1957	0.9982	0.7181	0.6649	0.6534	0.6510	0.6559	0.6537	0.6488	0.6506
	$r = 0.9$	2.2091	0.9965	0.7146	0.6672	0.6598	0.6567	0.6572	0.6550	0.6537	0.6477
$C^2 = 64$	$r = 0.1$	2.2061	1.0104	0.7157	0.6572	0.6515	0.6497	0.6597	0.6715	0.6671	0.6710
	$r = 0.5$	2.1893	0.9959	0.7233	0.6702	0.6569	0.6526	0.6529	0.6533	0.6521	0.6486
	$r = 0.9$	2.2072	0.9964	0.7136	0.6668	0.6583	0.6573	0.6554	0.6539	0.6554	0.6564

$M_n/G/1/PS$ queue, which we now know is insensitive in G. That is still not the whole story however, because, as we will see in Section 7, other common routing policies for PS server farms, like Least-Work-Left (sending the job to the host with the least total work), or Round-Robin, do *not* exhibit near-insensitivity, although one might think that a similar argument could be applied to them.

What seems to be unique about JSQ is that the conditional arrival rates, the $\lambda(n)$'s, derived from the server farm, are nearly insensitive to G, as we will see in Table 1. This fact allows us to write:

$$M/G/K/JSQ/PS \overset{SQA}{\approx} M_n^{(G)}/G/1/PS \approx M_n^{(M)}/G/1/PS = M_n/M/1/PS$$

where $M_n^{(G)}$ denotes the state-dependent arrival process in the case of general service times and $M_n^{(M)}$ denotes the state-dependent arrival process for the exponential service times. This insensitivity of the conditional arrival rates seems related to the fact that the JSQ policy uses the *number* of jobs in queue for making decisions, as compared with the Least-Work-Left policy, for example.

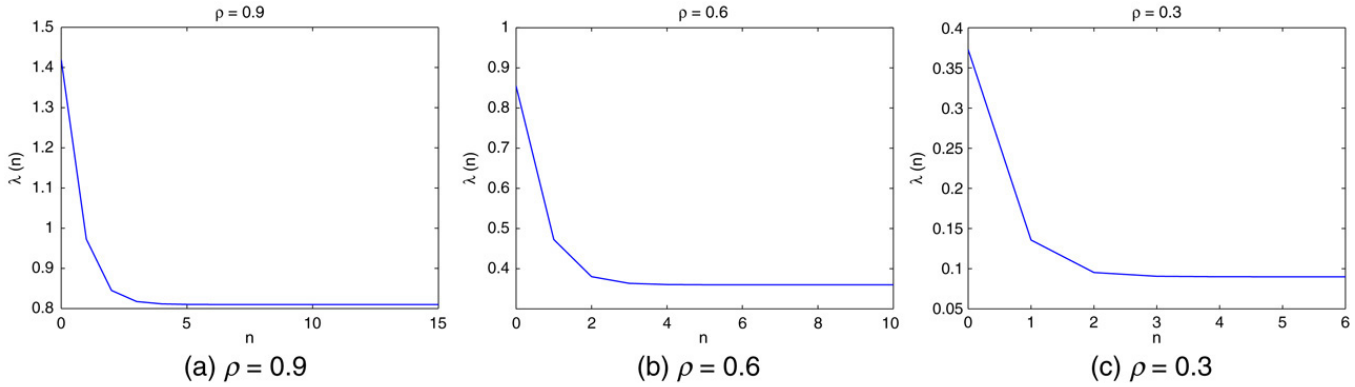
5. The conditional arrival rates

The feasibility of the SQA method hinges on obtaining the conditional arrival rates $\lambda(n)$, $n \geq 0$, defined in (1). In this section we will derive closed-form approximations for these conditional arrival rates. Our results here draw on extensive simulation experiments in which we estimated these conditional arrival rates for a range of job-size distributions and other model parameters. Fortunately, we found remarkable regularity, greatly simplifying our task.

First, we observed that the conditional arrival rates rapidly converge to a limiting value as n (the number of jobs at the queue) increases. Indeed, we found that

$$\frac{\lambda(n)}{\mu} \approx \rho^K \quad \text{for all } n \geq 3, \tag{7}$$

for $\rho \leq 0.95$. Simulations of the $M/M/K/JSQ/FCFS$ model showed this approximation to be consistently within 2% of the actual values (provided that ρ is not too extreme, i.e., for $0.3 \leq \rho \leq 0.95$). This fact is illustrated in Fig. 4 for the case of $K = 2$. We also prove this convergence in the limit for the case $K = 2$ in Theorem 5.1 below.



(a) $\rho = 0.9$ (b) $\rho = 0.6$ (c) $\rho = 0.3$

Fig. 4. Illustrating the convergence of conditional arrival rates, $\lambda(n)$, for a given queue of an $M/M/K/JSQ/PS$, with mean job size 1, where $K = 2$.

Theorem 5.1. *For the* $M(\lambda)/M(\mu)/2/JSQ/PS$ *system,*

$$\lim_{n\to\infty} \frac{\lambda(n)}{\mu} = \rho^2. \tag{8}$$

Proof. The proof is presented in the Appendix. It relies closely on the paper of Adan et al. [2]. We believe that we can generalize the proof to any finite K, however we state it only for $K = 2$. □

Observe that it makes intuitive sense that $\lambda(n)$, the average arrival rate into a designated queue conditioned on that queue having n jobs, should decrease as n is increased, because, if the designated queue has many jobs then it is likely that other queues have fewer jobs than itself. What is interesting is that the limit is reached so quickly.

Next, consistent with the other near-insensitivity results, we observed that these conditional arrival rates also exhibit near-insensitivity; there is almost no dependence on the variability of the job-size distribution. This fact is illustrated in Table 1 for the case of $K = 4$, with hyperexponential job-size distributions having squared coefficient of variation ranging from 1 to 64, where r denotes the fraction of load made up by one branch of the hyperexponential (hence $r = 0.5$ denotes a hyperexponential with balanced load on its branches). The near-insensitivity of the $\lambda(n)$'s provides further justification for focusing on the special case of an exponential job-size distribution.

Based on the key observation in (7), our task has been reduced to obtaining approximations for the first 3 conditional arrival rates: $\lambda(0)$, $\lambda(1)$ and $\lambda(2)$. The following lemma, allows us to reduce our task further to just deriving two conditional arrival rates, $\lambda(0)$ and $\lambda(2)$, since $\lambda(1)$ can be estimated from these, assuming the relation in (7).

Lemma 5.1. *Under the approximating approximation of Eq.* (7) *for the* $M/M/K/JSQ/PS$ *model, we obtain*

$$\lambda(1) = \mu \frac{\left[\frac{\mu}{\lambda(0)} \frac{\rho - \rho^{K+1}}{(1-\rho)} + \rho^K - 1\right]}{1 + \lambda(2)/\mu - \rho^K}. \tag{9}$$

Proof. Since all the servers are homogeneous, the time-average arrival rate into any one queue is $\lambda/K = \mu\rho$. By Theorem 3.1, SQA is exact given the conditional arrival rates. Therefore, we can write the time-average arrival rate into any server as

$$\mu\rho = \sum_{n=0}^{\infty} \Pi_n \lambda(n).$$

By Little's law (focusing on the servers), $1 - \Pi_0 = \rho$. Using that with (7), we obtain

$$\begin{aligned} \mu\rho = {} & (1-\rho)\lambda(0) + (1-\rho)\frac{\lambda(0)}{\mu}\lambda(1) + (1-\rho)\frac{\lambda(0)\lambda(1)}{\mu^2}\lambda(2) \\ & + \left(\rho - (1-\rho)\frac{\lambda(0)}{\mu} - (1-\rho)\frac{\lambda(0)\lambda(1)}{\mu^2}\right)\rho^K. \end{aligned} \tag{10}$$

This gives the desired approximation for $\lambda(1)$. □

The approximations for $\lambda(2)$ and $\lambda(0)$ were obtained empirically using MATLAB's curve fitting toolbox (version 1.1.5), which uses a trust-region method for a nonlinear least-squares fit. For each value of load, ρ, we approximate $\lambda(2)$ as a function of K by a simple exponential function of the form

$$\lambda(2) \approx \mu(u_\rho v_\rho^K). \tag{11}$$

Empirical fit yields the following functions of ρ:

$$u_\rho = c_3\rho^3 + c_2\rho^2 + c_1\rho + c_0 \quad \text{and} \quad v_\rho = c_2'\rho^2 + c_1'\rho + c_0',$$

where $c_3 = -0.29$, $c_2 = 0.8822$, $c_1 = -0.5349$, and $c_0 = 1.0112$, while $c_2' = -0.1864$, $c_1' = 1.195$, and $c_0' = -0.016$.

For $\lambda(0)$, we used a function with two exponential terms, namely,

$$\lambda(0) \approx \mu(a_\rho - b_\rho c_\rho^K - d_\rho e_\rho^K) \tag{12}$$

where $c_\rho, e_\rho < 1$. The constant a_ρ in (12) is clearly the limit as $K \to \infty$. The following lemma gives the value of this limit.

Lemma 5.2.

$$\lim_{K\to\infty} \frac{\lambda(0)}{\mu} = \frac{\rho}{1-\rho}. \tag{13}$$

Proof. For any value of $\rho < 1$, as the number of servers becomes large enough, there is a high probability that any arrival will find at least one server idle. Therefore, $\lambda(i) \approx 0$ for $i \geq 1$. Equating the expressions for time-average arrival rates into any queue,

$$(1-\rho)\lambda(0) = \mu\rho \quad \text{or} \quad \frac{\lambda(0)}{\mu} = \frac{\rho}{1-\rho}. \quad \square$$

The remaining functions b_ρ, c_ρ, d_ρ, and e_ρ were determined empirically for $0.3 \leq \rho \leq 0.95$; we did not have accurate enough simulations outside this range. The final functions are

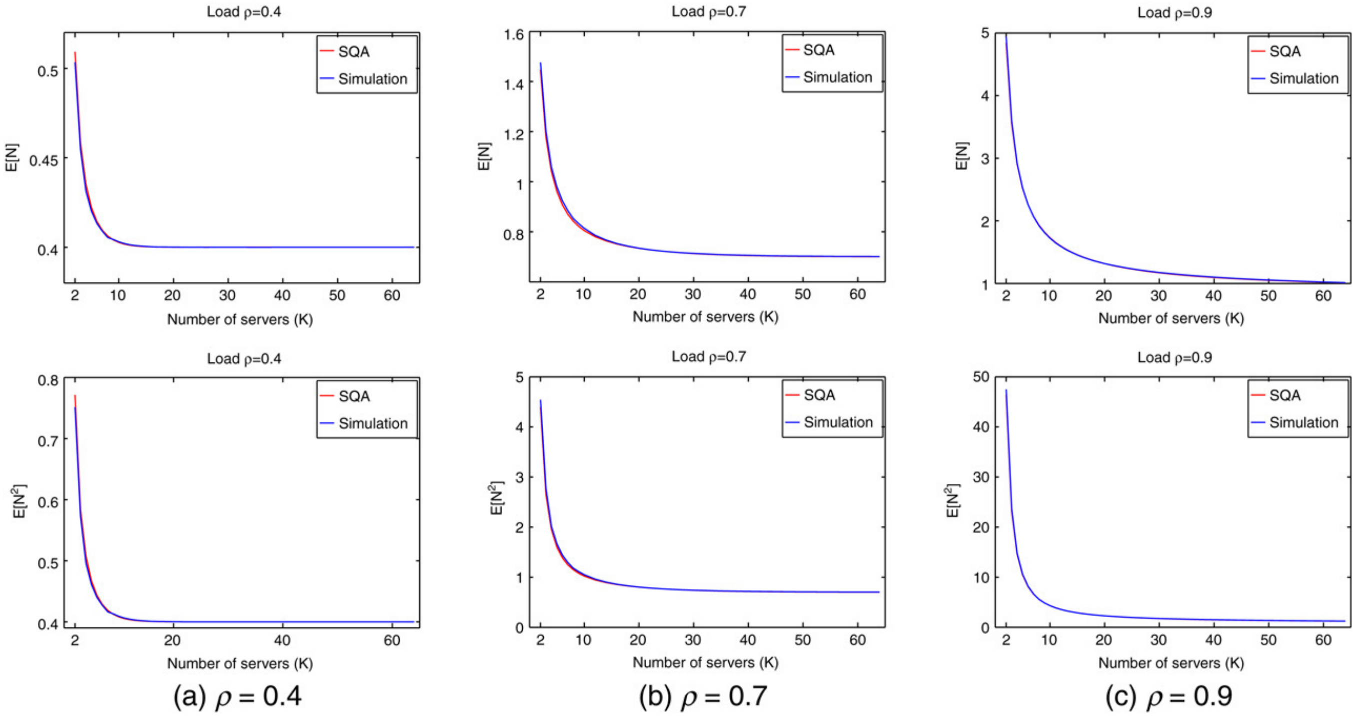
$$\begin{aligned}
b_\rho &= \frac{-0.0263\rho^2 + 0.0054\rho + 0.1155}{\rho^2 - 1.939\rho + 0.9534}\\
c_\rho &= -6.2973\rho^4 + 14.3382\rho^3 - 12.3532\rho^2 + 6.2557\rho - 1.005\\
d_\rho &= \frac{-226.1839\rho^2 + 342.3814\rho + 10.2851}{\rho^3 - 146.2751\rho^2 - 481.1256\rho + 599.9166}\\
e_\rho &= 0.4462\rho^3 - 1.8317\rho^2 + 2.4376\rho - 0.0512.
\end{aligned}$$

6. Evaluating the approximation

In this section we evaluate our SQA approximation for the *M*/*G*/*K*/*JSQ*/*PS* model, where the conditional arrival rates used in the SQA are the approximate ones derived in Section 5. Our approach is not exact even for the case of an exponential job-size distribution, because the conditional arrival rates are approximate. Therefore, we first evaluate our method for exponential job-size distributions in Section 6.1. Afterwards, we consider general job-size distributions in Section 6.2.

6.1. Exponential job sizes

Theorem 3.1 implies that SQA is exact if the conditional arrival rates are correct. In this section, we apply SQA with our approximate conditional arrival rates to determine the first two moments of queue lengths for exponential service requirements. The results are shown in Fig. 5, where N represents the queue length of a single queue in the server farm.



(a) $\rho = 0.4$ (b) $\rho = 0.7$ (c) $\rho = 0.9$

Fig. 5. The top row shows the effectiveness of SQA in predicting mean queue length, and the bottom row shows the effectiveness of SQA in predicting the second moment of queue length. Results are shown for three values of load: $\rho = 0.4$, $\rho = 0.7$, and $\rho = 0.9$, K up to 64 servers.

From Fig. 5, it is difficult to see that the SQA method with our derived approximate conditional arrival rates exhibits any error at all, when compared with simulations. However, the error is actually <2% for mean queue length and <2.4% for the second moment of queue length, when the number of servers is up to $K = 64$ and $\rho = 0.9$. Given that we have exponential job sizes, this error is solely due to error in the approximation of the conditional arrival rates.

Looking at Fig. 5, we see an interesting convergence in performance as K increases. If we denote by N the number of jobs at any designated queue in steady state, then we see that:

$$\lim_{K\to\infty} \mathbf{E}[N] = \rho.$$

This regularity occurs because, when $\rho < 1$ and the number of servers is allowed to increase, $\lambda(0) \to \mu\rho/(1-\rho)$ and $\lambda(i) \to 0$ for any $i > 0$; see Lemma 5.2.

6.2. *General job sizes*

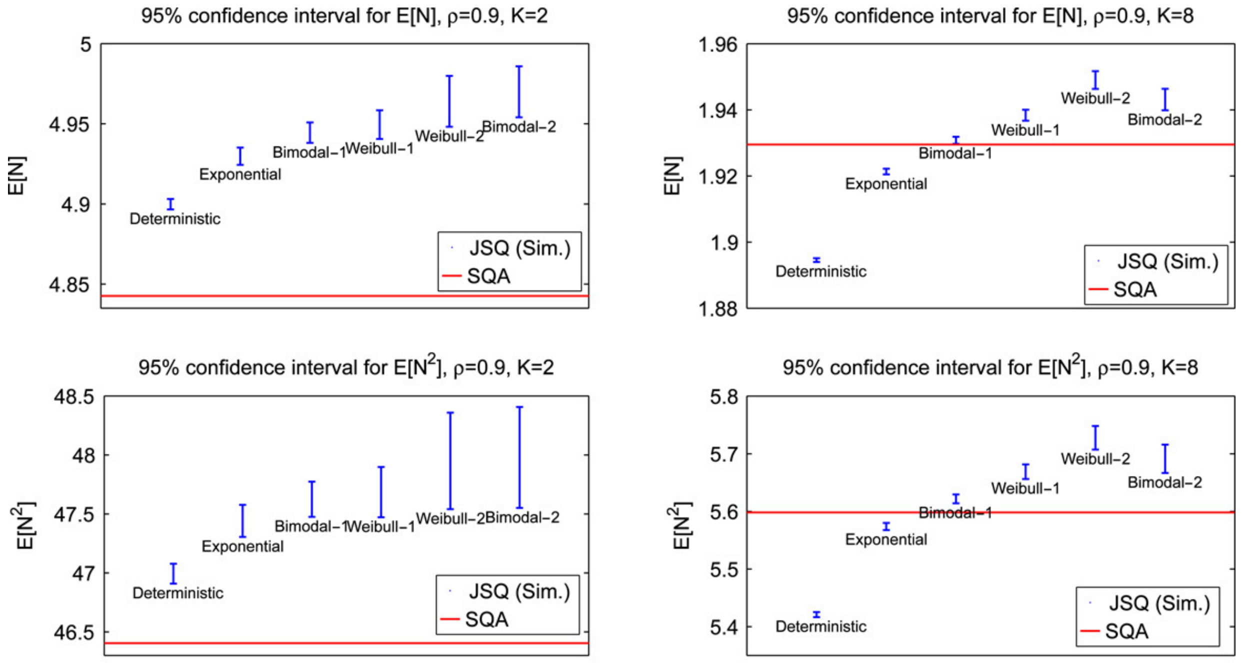
We now move on to the case of general job-size distributions. Fig. 6 shows the 95% confidence intervals for the first and second moment of queue length obtained from simulations of the original $M/G/K/JSQ/PS$ server farm for the distributions mentioned in Section 4.2. Each plot also shows the results of the SQA approximation: the analysis of the $M_n/G/1/PS$ system with the conditional arrival rates derived in Section 5. The results are also summarized in Tables 2 and 3.

The error is at most 2.6% for mean queue length, and at most 3.3% for the second moment of queue length when $\rho = 0.9$.

7. Comparison of JSQ with other routing policies

So far, we have only considered the commonly used JSQ routing policy. However, it is natural to wonder how good a routing policy JSQ is for PS server farms. In this section we show, via simulation, that it is unlikely that there is a routing policy which outperforms JSQ by more than about 10%. We also pose many interesting open problems regarding the optimality of JSQ.

Fig. 7 compares the performance of JSQ for a PS server farm with that of several other policies, via simulation, on a range of job-size distributions, defined in Section 4.2. The policies shown are:



(a) $K = 2$ (b) $K = 8$

Fig. 6. Comparison of the first and second moments of queue length at a single queue in the JSQ/PS server farm with those obtained using SQA for various service distributions with load $\rho = 0.9$ and number of servers $K = 2$ and 8. The top row shows $E[N]$ and the bottom row shows $E[N^2]$.

Table 2
Evaluation of SQA: First moment of queue length, obtained via simulation versus SQA, evaluated on distributions mentioned in Section 4.2

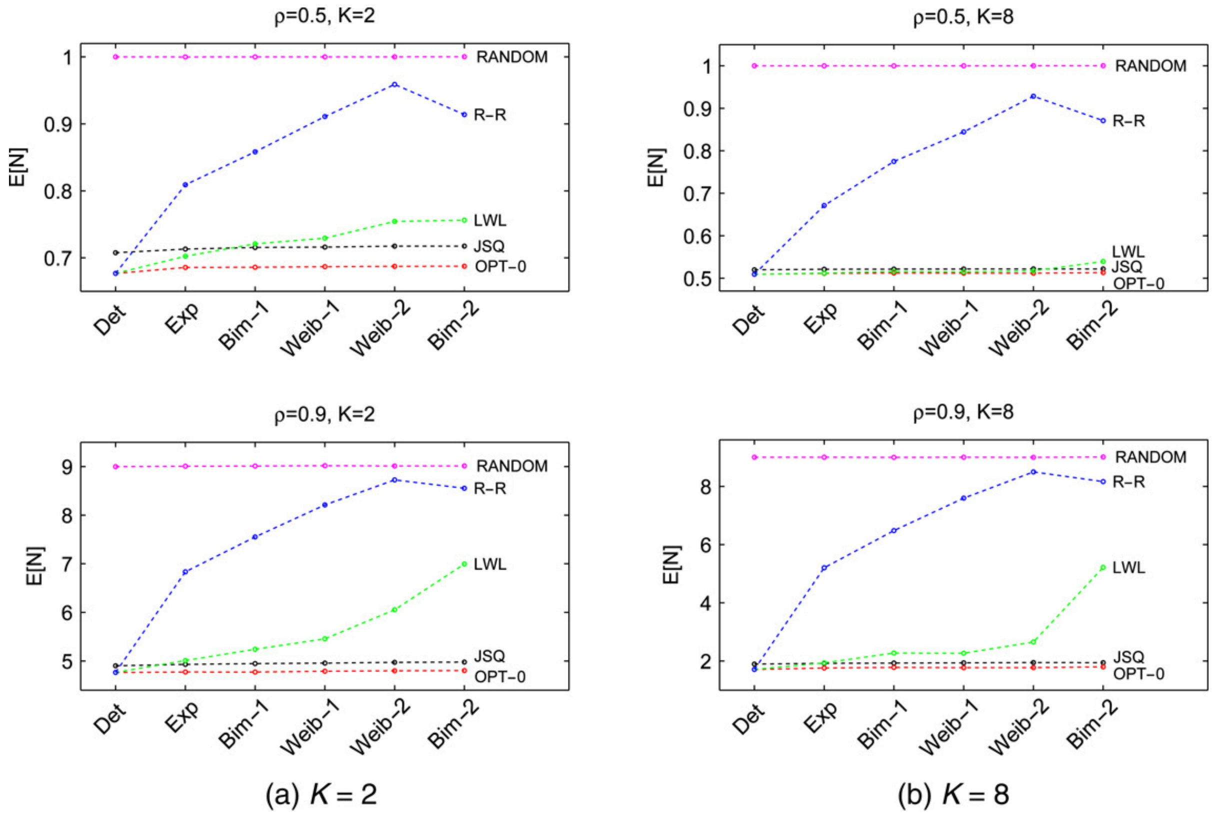
	$K = 2$			$K = 8$		
	$E[N]^{\text{JSQ}}$	$E[N]^{SQA}$	% error	$E[N]^{\text{JSQ}}$	$E[N]^{SQA}$	% error
`Deterministic`	4.8999	4.8426	1.1676	1.8946	1.9295	1.8449
`Erlang2`	4.9216	4.8426	1.6055	1.9142	1.9295	0.8015
`Exponential`	4.9298	4.8426	1.7678	1.9213	1.9295	0.4260
`Bimodal-1`	4.9445	4.8426	2.0592	1.9308	1.9295	0.0668
`Weibull-1`	4.9495	4.8426	2.1589	1.9384	1.9295	0.4573
`Weibull-2`	4.9640	4.8426	2.4456	1.9490	1.9295	1.0010
`Bimodal-2`	4.9700	4.8426	2.5618	1.9431	1.9295	0.7004

Table 3
Evaluation of SQA: Second moment of queue length, obtained via simulation versus SQA, evaluated for distributions mentioned in Section 4.2

	$K = 2$			$K = 8$		
	$E[N^2]^{\text{JSQ}}$	$E[N^2]^{SQA}$	% error	$E[N^2]^{\text{JSQ}}$	$E[N^2]^{SQA}$	% error
`Deterministic`	46.9934	46.4050	1.2523	5.4210	5.5982	3.2690
`Erlang2`	47.3844	46.4050	2.0669	5.5354	5.5982	1.1352
`Exponential`	47.4411	46.4050	2.1840	5.5738	5.5982	0.4375
`Bimodal-1`	47.6244	46.4050	2.5606	5.6217	5.5982	0.4187
`Weibull-1`	47.6847	46.4050	2.6837	5.6688	5.5982	1.2464
`Weibull-2`	47.9491	46.4050	3.2203	5.7277	5.5982	2.2616
`Bimodal-2`	47.9787	46.4050	3.2801	5.6912	5.5982	1.6343

Random—We flip a fair coin in deciding to which queue an incoming job should be assigned. Note that in this case, each queue looks like an $M/G/1/PS$ queue with arrival rate λ/K.

Round-Robin (RR)—Assign jobs in Round-Robin order, where if the previous job was assigned to queue $i \bmod K$, then the next job will be assigned to queue $(i + 1) \bmod K$.



(a) $K = 2$

(b) $K = 8$

Fig. 7. Comparison of the first moment of queue length for JSQ, Least-Work-Left (LWL), Round-Robin (R-R) and Random routing policies for $K = 2$ and $K = 8$ servers for a PS server farm with a range of job-size distributions.

Least-Work-Left (LWL)—Each job is assigned to the queue with the least total remaining work.

Join-Shortest-Queue (JSQ)—Each job is assigned to the queue with the fewest number of jobs. Ties are broken by flipping a fair coin.

OPT-0—Each incoming job is assigned so as to minimize the mean response time for all jobs currently in the system, *assuming that there are* 0 *future arrivals*. Note that we are not being greedy from the perspective of the incoming job, but rather trying to minimize across all the jobs in the system. This policy is followed for each successive incoming arrival. The OPT-0 policy was introduced by Bonomi [5].

Observe that policies OPT-0 and Least-Work-Left are both less practical than the other policies because they require knowledge of the job sizes.

There are many interesting things to see in Fig. 7. First, we note that OPT-0 is in fact the best routing policy across all job-size distributions of those policies shown. Also JSQ is very close to OPT-0, within no more than 10%. This is surprising because JSQ utilizes only the *number* of jobs at each queue, whereas OPT-0 uses the remaining sizes of all jobs and the size of the incoming job.

From an insensitivity perspective, we see that that there are some policies, e.g., OPT-0 and JSQ, that are nearly insensitive to the job-size distribution, whereas other policies, e.g., LWL and RR, are highly sensitive to the job-size distribution. It is an interesting question whether there is some detectable common characteristic among those routing policies that are nearly insensitive to the job-size distribution under PS server farms. This is an important question in light of the fact that the empirical workloads in Web server farms are very variable.

Turning to the question of optimality, note that the case of deterministic job sizes yields the lowest mean response times, as compared with other job-size distributions, and that all the three policies: RR, LWL, and OPT-0, yield the *same* performance for the case of deterministic job sizes—in fact, they behave identically on every sample path when the job-size distribution is deterministic. Conjecture 7.1 below hypothesizes that this value is the minimum response time possible across all policies and job-size distributions for PS farms.

Conjecture 7.1. *For an $M/G/K/\mathcal{R}/PS$ system, where the job-size distribution has mean μ^{-1}, we conjecture that setting $G \equiv \texttt{Deterministic}(\mu^{-1})$ and $\mathcal{R} \equiv RR$ results in the lowest possible mean response time, over all other pairs $(G, \mathcal{R})$.*

Conjecture 7.1 gives us a handle on evaluating the optimality of JSQ. Making use of the fact that JSQ is one of the policies that is nearly insensitive to the job-size distribution, by the above conjecture, it would suffice to compare the performance of JSQ under deterministic job sizes with RR under deterministic job sizes. Even under the narrowed scope of deterministic job sizes, the comparison between JSQ and RR is not obvious, because JSQ can differ from RR both in tie-breaks and non-tie-break situations. Hence there is much open work left to be done.

8. Conclusion

This paper has presented the first analysis of JSQ routing for PS server farms. Our analysis introduces many new ideas which we believe will be applicable in much more general settings. The first is the idea of a Single-Queue Approximation (SQA), whereby one designated queue in the farm can be analyzed *in isolation* from all the other queues, where a state-dependent arrival rate is used to, in some sense, capture the effect of the other queues. Understanding what these state-dependent arrival rates look like is also a very interesting topic that we introduce and study via analysis and simulation. Finally, and perhaps most interesting, is the notion of near-insensitivity, and the discovery that the *M/G/K/JSQ/PS* farm exhibits near-insensitivity to the job-size distribution, apart from the mean job size. This is particularly intriguing in light of the fact that so many other routing policies for PS server farms, like Least-Work-Left or Round-Robin, do not exhibit this near-insensitivity property. All of the above topics are studied carefully both via analysis and simulation across a wide range of job-size distributions. We end with a simulation study of different routing policies, leading us to pose several open questions regarding the near-optimality of JSQ routing. Additional experimental results will appear in an online supplement.

Acknowledgements

VG and MHB are supported by the NSF Division of Computer and Network Systems, Grant CNS-0615262. WW is supported in part by the NSF Grant DMI-0457095.

Appendix. Proof of Theorem 5.1

The proof follows directly from the work of Adan et al. [2] on using compensation approach to analyze the *M/M/2/JSQ/FCFS* queue and will use Lemmas A.1 and A.2 mentioned below. We begin by reviewing the notation. Let $\pi_{m,n}$ be the stationary probability that length of queue 1 is m and length of queue 2 is n. For $m \geq 0$ and $r \geq 0$, define $q_{m,r}$ as:

$$q_{m,r} = \pi_{m,m+r}. \tag{14}$$

That is, $q_{m,r}$ is the probability that queue 1 is the shorter queue and has m jobs and queue 2 has $m + r$ jobs.

Lemma A.1 (Adan et al. [2]). *The stationary probabilities $q_{m,r}$ for $m \geq 0$ and $r \geq 1$ are given by:*

$$q_{m,r} = Cx_{m,r}.$$

The normalization constant C is given by

$$C = \frac{2(1-\rho^2)(2-\rho)}{\rho(2+\rho)}$$

and

$$x_{m,r} = \sum_{i=0}^{\infty} d_i(\alpha_i^m + c_i\alpha_{i+1}^m)\beta_i^r \tag{15}$$

where α_i, β_i, c_i and d_i's are given by the following recursion scheme:

$$\begin{aligned}
&d_0 = 1\\
&\alpha_0 = \rho^2\\
&\beta_0 = \frac{\rho^2}{2+\rho}\\
&\alpha_i\alpha_{i+1} = 2\rho\beta_i^2\\
&\beta_i\beta_{i+1} = \alpha_{i+1}^2/(2\rho+\alpha_{i+1})\\
&c_i = -\frac{\alpha_{i+1}-\beta_i}{\alpha_i-\beta_i}\\
&d_{i+1} = -\frac{(\alpha_{i+1}+\rho)/\beta_{i+1}-(\rho+1)}{(\alpha_{i+1}+\rho)/\beta_i-(\rho+1)}c_i d_i.
\end{aligned}$$

We will use the following lemma to bound the infinite sum of Eq. (15) by a finite sum.

Lemma A.2. *The infinite sum of $x_{m,r}$ ($m \geq 0, r \geq 1$) in Eq. (15) can be bounded by the following finite sums:*

$$(\alpha_0^m + c_0\alpha_1^m)\beta_0^r + d_1(\alpha_1^m + c_1\alpha_2^m)\beta_1^r = \underline{x_{m,r}} < x_{m,r} < \overline{x_{m,r}} = (\alpha_0^m + c_0\alpha_1^m)\beta_0^r. \tag{16}$$

Proof. Let $s_i = |d_i(\alpha_i^m + c_i\alpha_{i+1}^m)\beta_i^r|$. In [2] (Lemma 8), the authors prove that:

$$s_{i+1} < Rs_i$$

where $R = 4/(4+2\rho+\rho^2) < 1$. Also as a consequence of Lemma 1 of [2], $d_{i+1}/d_i < 0$. That is d_i alternate signs, d_0 being defined to equal 1. Hence,

$$\begin{aligned}
x_{m,r} &= s_0 - s_1 + s_2 - s_3 + s_4 - \cdots\\
&< s_0 - s_1 + Rs_1 - s_3 + Rs_3 - \cdots\\
&= s_0 - (1-R)(s_1 + s_3 + \cdots)\\
&< s_0\\
&\overset{\text{def}}{=} \overline{x_{m,r}}
\end{aligned}$$

and,

$$\begin{aligned}
x_{m,r} &= s_0 - s_1 + s_2 - s_3 + s_4 - s_5 + \cdots\\
&> s_0 - s_1 + s_2 - Rs_2 + s_4 - Rs_4 + \cdots\\
&= s_0 - s_1 + (1-R)(s_2 + s_4 + \cdots)\\
&> s_0 - s_1\\
&\overset{\text{def}}{=} \underline{x_{m,r}}. \quad \square
\end{aligned}$$

Proof of Theorem 5.1. Let Π_n be the stationary probability that there are n jobs in queue 1. Since we know that SQA is exact, we can express the conditional arrival rates, $\lambda(n)$, as

$$\lambda(n) = \mu\frac{\Pi_{n+1}}{\Pi_n} = \mu\frac{\sum_{i=0}^{\infty}\pi_{n+1,i}}{\sum_{i=0}^{\infty}\pi_{n,i}}.$$

Let $x_{m,0} = C^{-1}q_{m,0}$. Since for $m > 0$,

$$q_{m,0} = \frac{1}{1+\rho}(2\rho q_{m-1,1} + q_{m,1}) \tag{17}$$

we also have the following bounds on $x_{m,0}$:

$$\frac{1}{1+\rho}(2\rho\underline{x_{m-1,1}} + \underline{x_{m,1}}) = \underline{x_{m,0}} < x_{m,0} < \overline{x_{m,0}} = \tfrac{1}{1+\rho}(2\rho\overline{x_{m-1,1}} + \overline{x_{m,1}}).$$

Expressing π's in terms of the x's gives us the following bounds on $\lambda(n)$:

$$\underline{\lambda(n)} < \lambda(n) < \overline{\lambda(n)} \tag{18}$$

where,

$$\underline{\lambda(n)} = \mu \frac{\underline{x_{n+1,0}} + \sum_{i=1}^{\infty} \underline{x_{n+1,i}} + \sum_{j=0}^{n} \underline{x_{j,n+1-j}}}{\overline{x_{n,0}} + \sum_{i=1}^{\infty} \overline{x_{n,i}} + \sum_{j=0}^{n-1} \overline{x_{j,n-j}}} \tag{19}$$

$$\overline{\lambda(n)} = \mu \frac{\overline{x_{n+1,0}} + \sum_{i=1}^{\infty} \overline{x_{n+1,i}} + \sum_{j=0}^{n} \overline{x_{j,n+1-j}}}{\underline{x_{n,0}} + \sum_{i=1}^{\infty} \underline{x_{n,i}} + \sum_{j=0}^{n-1} \underline{x_{j,n-j}}}. \tag{20}$$

The expression for $\overline{\lambda(n)}$ in (20) is obtained by upper bounding the numerator, Π_{n+1}, and lower bounding the denominator, Π_n. Doing the opposite gives $\underline{\lambda(n)}$ (19).

To prove the convergence of $\lambda(n)$, we will prove that

$$\lim_{n\to\infty} \underline{\lambda(n)} = \lim_{n\to\infty} \overline{\lambda(n)} = \mu\rho^2.$$

We will first show the convergence of $\overline{\lambda(n)}$. Proof for $\underline{\lambda(n)}$ is similar. Now,

$$\overline{\lambda(n)} = \mu \frac{\overline{x_{n+1,0}} + \sum_{i=1}^{\infty} \overline{x_{n+1,i}} + \sum_{j=0}^{n} \overline{x_{j,n+1-j}}}{\underline{x_{n,0}} + \sum_{i=1}^{\infty} \underline{x_{n,i}} + \sum_{j=0}^{n-1} \underline{x_{j,n-j}}} = \mu \frac{S_{n+1}}{S_n + T_n} \tag{21}$$

where,

$$S_i = \frac{\beta_0}{1+\rho}[2\rho(\alpha_0^{i-1} + c_0\alpha_1^{i-1}) + (\alpha_0^i + c_0\alpha_1^i)] + (\alpha_0^i + c_0\alpha_1^i)\frac{\beta_0}{1-\beta_0} + \beta_0\left(\frac{\alpha_0^i - \beta_0^i}{\alpha_0 - \beta_0} + c_0\frac{\beta_0^i - \alpha_1^i}{\beta_0 - \alpha_1}\right)$$

$$T_i = d_1\left[\frac{\beta_1}{1+\rho}[2\rho(\alpha_1^{i-1} + c_1\alpha_2^{i-1}) + (\alpha_1^i + c_1\alpha_2^i)] + (\alpha_1^i + c_1\alpha_2^i)\frac{\beta_1}{1-\beta_1} + \beta_1\left(\frac{\alpha_1^i - \beta_1^i}{\alpha_1 - \beta_1} + c_1\frac{\beta_1^i - \alpha_2^i}{\beta_1 - \alpha_2}\right)\right].$$

Dividing the numerator and denominator of (21) by α_0^{n-1}, taking $\lim_{n\to\infty}$ and noting that $\frac{\alpha_1}{\alpha_0} < 1$, $\frac{\alpha_2}{\alpha_0} < 1$, $\frac{\beta_0}{\alpha_0} < 1$ and $\frac{\beta_1}{\alpha_0} < 1$:

$$\lim_{n\to\infty} \overline{\lambda(n)} = \mu\alpha_0 \frac{\frac{\beta_0}{1+\rho}[2\rho + \alpha_0] + \alpha_0\frac{\beta_0}{1-\beta_0} + \beta_0\left(\frac{\alpha_0}{\alpha_0-\beta_0}\right)}{\frac{\beta_0}{1+\rho}[2\rho + \alpha_0] + \alpha_0\frac{\beta_0}{1-\beta_0} + \beta_0\left(\frac{\alpha_0}{\alpha_0-\beta_0}\right)} \tag{22}$$

$$= \mu\alpha_0$$

$$= \mu\rho^2. \tag{23}$$

Similarly,

$$\lim_{n\to\infty} \underline{\lambda(n)} = \lim_{n\to\infty} \mu\frac{S_{n+1} + T_{n+1}}{S_n} = \mu\rho^2$$

and hence convergence of $\lambda(n)$ follows by convergence of its upper and lower bounds. □

References

[1] I.J.B.F. Adan, G.J. van Houtum, J. van der Wal, Upper and lower bounds for the waiting time in the symmetric shortest queue system, Ann. Oper. Res. 48 (1994) 197–217.

[2] I.J.B.F. Adan, J. Wessels, W.H.M. Zijm, Analysis of the symmetric shortest queue problem, Stoch. Models 6 (1990) 691–713.

[3] I.J.B.F. Adan, J. Wessels, W.H.M. Zijm, Matrix-geometric analysis of the shortest queue problem with threshold jockeying, Oper. Res. Lett. 13 (1993) 107–112.

[4] P. Barford, M.E. Crovella, Generating representative Web workloads for network and server performance evaluation, in: Proceedings of Performance '98/SIGMETRICS'98, July 1998, pp. 151–160. Software for Surge is available from Mark Crovella's home page.

[5] F. Bonomi, On job assignment for a parallel system of processor sharing queues, IEEE Trans. Comput. 39 (7) (1990) 858–869.

[6] O.J. Boxma, J.W. Cohen, Boundary Value Problems in Queueing System Analysis, North Holland, Amsterdam, 1983.

[7] P. Brémaud, Point Processes and Queues, Springer-Verlag, New York, 1981.

[8] B.W. Conolly, The autostrada queueing problem, J. Appl. Probab. 21 (1984) 394–403.

[9] M. Crovella, M. Harchol-Balter, C. Murta, On choosing a task assignment policy for a distributed server system, J. Parallel Distrib. Comput. 59 (2) (1999) 204–228.

[10] M.E. Crovella, A. Bestavros, Self-similarity in World Wide Web traffic: Evidence and possible causes, in: Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, May 1996, pp. 160–169.

[11] D. Daley, D. Vere-Jones, An Introduction to the Theory of Point Processes, Springer, New York, 1988.

[12] M. El-Taha, Shaler Stidham, Sample-Path Analysis of Queueing System, Kluwer, Boston, 1999.

[13] A. Ephremides, P. Varaiya, J. Walrand, A simple dynamic routing problem, IEEE Trans. Automat. Control AC-25 (4) (1980) 690–693.

[14] L. Flatto, H.P. McKean, Two queues in parallel, Comm. Pure Appl. Math. 30 (1977) 255–263.

[15] G. Foschini, J. Salz, A basic dynamic routing problem and diffusion, IEEE Trans. Commun. 26 (3) (1978) 320–328.

[16] W.K. Grassmann, Transient and steady state results for two parallel queues, Omega 8 (1980) 105–112.

[17] M. Harchol-Balter, Task assignment with unknown duration, J. ACM 49 (2) (2002) 260–288.

[18] M. Harchol-Balter, B. Schroeder, N. Bansal, M. Agrawal, Size-based scheduling to improve web performance, ACM Trans. Comput. Syst. 21 (2) (2003) 207–233.

[19] F.P. Kelly, Reversibility and Stochastic Networks, Wiley, Chichester, 1979.

[20] J.F.C. Kingman, Two similar queues in parallel, Biometrika 48 (1961) 1316–1323.

[21] C. Knessl, B.J. Matkowsky, Z. Schuss, C. Tier, Two parallel $M/G/1$ queues where arrivals join the system with the smaller buffer content, IEEE Trans. Commun. 35 (11) (1987) 1153–1158.

[22] H.C. Lin, C.S. Raghavendra, An analysis of the join the shortest queue (JSQ) policy, in: Proc. 12th Int'l Conf. Distributed Computing Systems, 1992, pp. 362–366.

[23] J.C.S. Lui, R.R. Muntz, D.F. Towsley, Bounding the mean response time of the minimum expected delay routing policy: An algorithmic approach, IEEE Trans. Comput. 44 (12) (1995) 1371–1382.

[24] R.D. Nelson, T.K. Philips, An approximation to the response time for shortest queue routing, ACM Perform. Eval. Review 17 (1989) 181–189.

[25] B.M. Rao, M.J.M. Posner, Algorithmic and approximation analyses of the shorter queue model, Naval Res. Logist. 34 (1987) 381–398.

[26] R.W. Weber, On optimal assignment of customers to parallel servers, J. Appl. Probab. 15 (1978) 406–413.

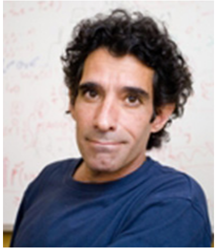
[27] W. Whitt, Comparison conjectures about the $M/G/s$ queue, Oper. Res. Lett. 2 (5) (1983) 203–209.

[28] W. Whitt, Heavy-traffic limits for the G/H$_2$*/n/m queue, Math. Oper. Res. 30 (1) (2005) 1–27.

[29] W. Winston, Optimality of the shortest line discipline, J. Appl. Probab. 14 (1977) 181–189.

Varun Gupta is a Ph.D. candidate at the Department of Computer Science, Carnegie Mellon University. He received his B.Tech. degree in computer science and engineering from Indian Institute of Technology, Delhi in 2004 and was awarded the prestigious President's Gold Medal. His research interests are in all aspects of design and analysis of algorithms. Currently, he focuses on performance modeling and analysis of scheduling policies for multiserver systems from a queueing theoretic perspective.

Mor Harchol-Balter is the McCandless Associate Professor of Computer Science at Carnegie Mellon University. She received her doctorate from the Computer Science department at the University of California at Berkeley under the direction of Manuel Blum. She is a recipient of the NSF CAREER award, the NSF Postdoctoral Fellowship in the Mathematical Sciences, multiple best paper awards, and several teaching awards, including the Herbert A. Simon Award for Teaching Excellence. She currently serves as Treasurer of the SIGMETRICS board, and Program Chair for SIGMETRICS 2007 and for QEST 2007. Professor Harchol-Balter is heavily involved in the ACM SIGMETRICS research community. Her work focuses on designing new scheduling/resource allocation policies for various distributed computer systems including Web servers, distributed supercomputing servers, networks of workstations, and database systems. Her work spans both queueing analysis and implementation and emphasizes integrating measured workload distributions into the problem solution.



Karl Sigman is a full professor in the Department of Industrial Engineering and Operations Research at Columbia University. He received his MS and Ph.D. degrees from the University of California, Berkeley, and has been at Columbia University as a faculty member for 20 years. His research is in stochastic modeling.

Ward Whitt is the Wai T. Chang Professor of Industrial Engineering and Operations Research at Columbia University. He received his doctorate from the School of Operations Research and Information Engineering at Cornell University in 1969. After teaching at Stanford University and Yale University, he joined Bell Laboratories in 1977. He spent twenty-five years in research at Bell Labs and AT&T Labs before joining Columbia University in 2002. His research focuses on stochastic models and their applications.