Available online at www.sciencedirect.com
SCIENCE DIRECT®
Performance Evaluation 62 (2005) 475–492
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

Optimal state-free, size-aware dispatching for heterogeneous $M/G/$-type systems

Hanhua Feng[a,*], Vishal Misra[a], Dan Rubenstein[b]

[a] *Department of Computer Science, Columbia University, New York, NY 10027, United States*
[b] *Department of Electric Engineering, Columbia University, New York, NY 10027, United States*

Available online 18 August 2005

Abstract

We consider a cluster of heterogeneous servers, modeled as $M/G/1$ first-come first-serve queues with different processing speeds. A dispatcher that assigns jobs to the servers takes as input only the size of the arriving job and the overall job-size distribution. This general model captures the behavior of a variety of real systems, such as web server clusters. Our goal is to identify assignment strategies that the dispatcher can perform to minimize expected completion time and waiting time. We show that there exist optimal strategies that are deterministic, fixing the server to which jobs of particular sizes are always sent. We prove that the optimal strategy for systems with identical servers assigns a non-overlapping interval range of job sizes to each server. We then prove that when server processing speeds differ, it is necessary to assign each server a distinct set of intervals of job sizes in order to minimize expected waiting or response times.

Keywords: Scheduling; Queueing systems; Load balancing; $M/G/1$; Parallel servers; Heterogeneous servers

1. Introduction

Many systems that can process multiple jobs in parallel fall into a class of systems that are commonly referred to as *dispatching systems*. A simple illustration of a model for these systems is depicted in Fig. 1. In a dispatching system, jobs arrive at a *dispatcher*, which must immediately decide the server to which

* Corresponding author.
E-mail address: hanhua@cs.columbia.edu (H. Feng).

0166-5316/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2005.07.031

the job is assigned. Each server has a separate queue in which it stores jobs that it was assigned by the dispatcher, and a separate processor that processes its assigned jobs. We refer to the decision process used by the dispatcher to assign jobs to various servers as the *dispatching strategy*, or simply *strategy* for short. The choice of strategy depends upon the information available to the dispatcher when it makes a selection (e.g., size of the received job, number of jobs waiting in each of the servers) and upon the optimization goal of the system.

In this paper, we investigate *state-free, size-aware* dispatching strategies for *homogeneous* and *heterogeneous* systems that minimize expected waiting time and completion time of arriving jobs. *State-free* means that the dispatcher uses neither the current status of the queues, nor the past history of assignments when deciding where to place an arriving job. This assumption is traditionally called *static* in the literature. *Size-aware* means that the dispatcher uses the size of the arriving job (i.e. the *processing time* that this job requires from the server) and the overall job-size distribution as input to select the server to which the job is sent. A *homogeneous* system is one in which all servers are configured identically, otherwise the system is *heterogeneous*. A rich class of systems are best modeled as state-free, size-aware dispatching systems. Some examples are:

- *Web delivery systems.* Web providers commonly cluster a set of servers together to serve requests. A simple way to assign jobs to servers is to distribute contents to multiple servers (e.g., put HTML files on one server and image files on another), using different host names in corresponding URLs. The system is naturally state-free and there is no explicit dispatcher. Since the size of each file is known in advance, it is possible to find a better way to distribute files to servers so that the expecting response time of the entire system is reduced.
- *Routing in content distribution networks*. Load-balanced routing allows an intermediate node to assign an arriving data chunk to any of several outgoing paths. A controlling "pushback" mechanism that informs upstream nodes of current downstream load, while useful, is difficult to implement efficiently in practice. However, a local, static dispatching strategy based on long-term statistics is very useful and easily implemented for such systems.
- *Database systems with multiple servers*. There are usually many different access paths to retrieve data in a database system. Modern database management systems estimate the processing time for each access path to make a good choice. With multiple servers, this decision can be performed by the dispatcher where the dispatcher uses the estimated processing time to choose a server for the actual data retrieval.

State-free, size-aware dispatchers are interesting and important because of the ease with which the dispatcher can be implemented. Dispatchers that utilize additional state (history of arriving jobs, or the current states of the queues) can likely outperform their static counterparts as long as the information used (such as the current state of the queues) is not significantly delayed [10]. However, it is unclear whether the expected gains in performance are worth the added complexity in implementation.

The paper proves several important results about the optimality of a variety of strategies that attempt to minimize job waiting and completion times for first-come first-serve (FCFS) queues. An important class of deterministic strategies we consider are *interval-based* strategies, where each server is assigned all jobs whose processing times fall within a distinct, continuous interval of processing times.

We show the following results in this paper: (1) When the dispatch system is comprised of a set of homogeneous servers, then there is in fact an interval-based strategy that is optimal amongst all static

strategies. (2) When the system is comprised of a set of heterogeneous servers, then there need not exist an interval-based strategy that is optimal. (3) A simple generalization of the class of interval-based strategies that uses *nested intervals* produces a class of strategies that does in fact contain an optimal static strategy for a system comprised of a set of heterogeneous servers. In addition, we also focus on identifying the best or optimal strategy within the class of interval-based strategies. In heterogeneous settings, the optimal strategy must select not only the boundaries of the intervals, but must also determine the mapping of intervals to the (heterogeneous) servers. While we have not yet identified an optimal strategy, we show through experimentation using a variety of traditional job size distributions that an optimal strategy depends heavily on the job size distribution.

1.1. Related work

If a shared queue is used and the dispatcher follows the first-come first-serve order and assigns jobs to servers whenever they are idle, then the system is a traditional $M/M/s$ or $M/G/s$ queueing system. The homogeneous $M/M/s$ system is well studied and can be found in many textbooks, e.g. [4]. On the other hand, there are only approximations of the mean response time for homogeneous $M/G/s$ systems [7]. With separate queues and exponential service times, Winston [15] and Ephremides et al. [6] analyze optimal dynamic strategies. Exponential service times are often assumed in the study of the heterogeneous servers. With separate heterogeneous queues and exponential service times, Chow and Kohler [3] compare three dynamic strategies with the random strategy. With the random static strategy, Buzen and Chen [2] give the optimal load partitioning for general services. Ni and Hwang [11] analyze the optimal load partitioning for multiple classes of exponential service times. Borst [1] considers the generalized case in which the weighted sum of the mean waiting times for multiple classes of generally distributed jobs is minimized. Hajek [8] considers two heterogeneous queues where the jobs departing from one queue may be sent to the other with a certain probability. Tantawi and Towsley [14] study a static decentralized probabilistic strategy where each queue can transfer some jobs to other queues with a communication delay. The size-interval strategy is proposed and studied by Harchol-Balter et al. [9,5]. Oida et al. numerically study the size-interval strategy with a finite set of jobs and show that the performance of size-interval strategy is close to the solution of the corresponding optimal deterministic problem under heavy traffic for homogeneous [12] and two-queue heterogeneous [13] systems.

The rest of the paper is organized as follows. In Section 2 we formulate the problem. In Section 3 we analyze the optimal size-aware strategies. In Section 4 we show some numerical results of the mean waiting times for three different classes of job-size distributions, and study the best mapping of size intervals. In Section 5 we give proofs for optimality. Finally we conclude in Section 6.

2. Preliminaries and problem formulation

To initiate our study of *state-free, size-aware* strategies for parallel *heterogeneous* servers with *separate* queues, we first introduce the notation we will use throughout the remainder of the paper.

The *capacity* of a queue is the maximal amount of processing that can be performed in a unit of time by the server associated with that queue. This is a formal measure of the processing speed. We denote by the random variable X the size of a job (the service time in a queue of unit capacity), and by $F(x)$ its

cumulative distribution function (CDF). For a queue of capacity c, the service time of a job of size X is X/c. The arrival rate of jobs to the entire system (of all queues) is denoted by λ. The waiting and response times are denoted by random variables W and T, respectively. The latter is also known as the completion time.

The *load* is defined to be $\rho = \lambda E[X] = \lambda/\mu$, where $\mu = 1/E[X]$ is the average departure rate of jobs from a queue with unit capacity. Load measures the average amount of processing that can be done in a unit of time. We assume that the load for a queue in the steady state does not exceed its capacity, i.e., $\rho < c$. We let $\omega = \lambda E[X^2]$. This quantity is important and heavily used in the rest of the paper – it measures the performance of a first-come first-serve queue. We call it the *second-order load*, in the sense that $\rho = \lambda E[X]$ is the (first-order) load, and the arrival rate $\lambda = \lambda E[X^0]$ can be considered as the *zeroth-order load*.

2.1. Dispatching system model

Let us now define the class of dispatching strategies upon which we focus in this paper. We assume that there are n parallel queues in the system and the capacity of the ith queue is c_i. Without loss of generality, we assume that the sum of the capacities is 1, i.e., $\sum_{i=1}^{n} c_i = 1$. If $c_i = 1/n$ for all $i = 1, \ldots, n$, the system is *homogeneous*, otherwise it is *heterogeneous*. A *stochastic, size-aware, static* strategy, or simply a *static strategy*, only uses the size of a job to select the queue that processes that job. The queue to which a job of a given size is assigned may be selected either deterministically (using the size of the job) or at random (without considering the size of the job). Such strategies are *state-free*: neither history records, current states of the queues, nor sequence numbers of the arrivals are used by these types of strategies.

We assume that jobs in a single queue are serviced in the first-come first-serve (FCFS) order. We also assume the arrivals of jobs are Poisson and job sizes are independent, identically-distributed (IID). Note that each of the queues is modeled as an $M/G/1$-FCFS queue, although the job size distribution for each queue can differ from the distribution of the aggregate queueing system.

We denote by the random variable X_i the size of a job assigned to the ith queue, and its CDF by $F_i(x)$. Note that $\lambda = \sum_{i=1}^{n} \lambda_i$, where λ_i is the arrival rate of the ith queue. We then have $\sum_{i=1}^{n} \lambda_i F_i(x) = \lambda F(x)$. Note that function $\lambda F(\cdot)$ sufficiently describes both the Poisson arrival process and the job-size distribution: $\lambda F(x)$ is the arrival rate of jobs whose sizes are shorter than or equal to x. Hence its derivative $\lambda f(x)$ is the *density function of the arrival rate* for job size x. A static strategy, therefore, is equivalent to an algorithm that divides function $\lambda F(x)$ (or $\lambda f(x)$) to a sum of n parts, i.e., $\sum_{i=1}^{n} \lambda_i F_i(x)$ (or $\sum_{i=1}^{n} \lambda_i f(x)$, respectively), each of which is assigned to a queue. The performance of the system is evaluated by the *mean response time* or *main waiting time* on the per-job basis.

We denote the service time of the ith queue by $\hat{X}_i$ that is X_i/c_i. Also, we let random variable $\hat{X}$ be the overall service time of the entire system. Taking expectations per job, we get

$$E[\hat{X}] = \sum_{i=1}^{n} \frac{\lambda_i E[\hat{X}_i]}{\lambda} = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{\lambda_i E[X_i]}{c_i} = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{\rho_i}{c_i}, \tag{1}$$

where $\rho_i = \lambda_i E[X_i]$ is the load of the ith queue. We refer to the vector $[\rho_i]_{i=1}^{n}$ (such that $\rho = \sum_{i=1}^{n} \rho_i$) as *load partitioning*, which indicates the portion of the total load that each queue is assigned. The overall

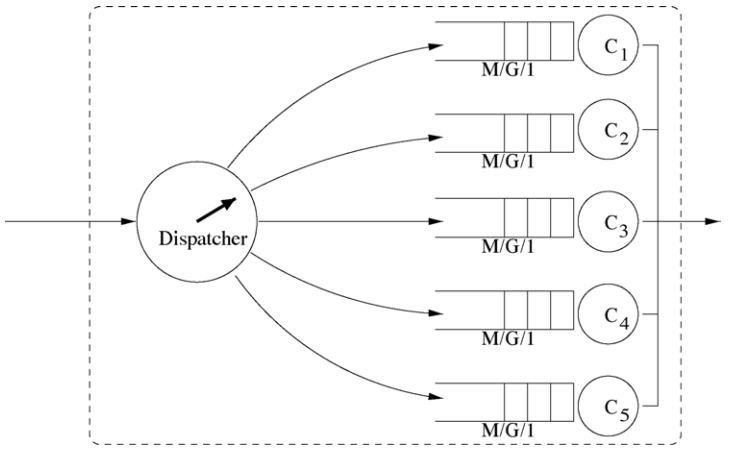

Fig. 1. A dispatcher assigns jobs to multiple queues with different processing speeds.

mean waiting time and the overall mean response time are, respectively,

$$E[W] = \frac{1}{\lambda}\sum_{i=1}^{n}\lambda_i E[W_i], \tag{2}$$

$$E[T] = \frac{1}{\lambda}\sum_{i=1}^{n}\lambda_i E[T_i] = E[W] + E[\hat{X}], \tag{3}$$

where $E[W_i]$ and $E[T_i]$ are the mean waiting time and the mean response time of the ith queue, respectively. For a single $M/G/1$-FCFS queue, the mean waiting time is (scaled from the Pollaczek-Khintchine formula [4])

$$E[W] = \frac{\lambda E[X^2]}{2c(c-\rho)} = \frac{\omega}{2c(c-\rho)}, \tag{4}$$

where $\omega = \lambda E[X^2]$. From (2) and (4), the mean waiting time for a heterogeneous system using a static strategy is

$$E[W^s] = \sum_{i=1}^{n}\frac{\lambda_i}{\lambda}\left[\frac{\lambda_i E[X_i^2]}{2c_i(c_i-\rho_i)}\right] = \frac{1}{2\lambda}\sum_{i=1}^{n}\left[\frac{\lambda_i\omega_i}{c_i(c_i-\rho_i)}\right], \tag{5}$$

where $\omega_i = \lambda_i E[X_i^2]$ is the second-order load of the ith queue.

2.2. Specific static strategies

Several particular static strategies are discussed throughout the rest of this paper. Here, we classify these strategies:

- *Random strategy*. Under the random strategy, jobs are assigned to the ith queue with a fixed probability p_i, independent of job size. The random strategy is a static strategy such that $F_i(x) = F(x)$ and $\lambda_i = p_i\lambda$.

- *Size-interval* (*SI*) *strategy*. With an SI strategy, job sizes are divided into n distinct, contiguous intervals, separated by $n-1$ thresholds. Jobs in each size interval are assigned to a single queue. We denote by ξ_j, $j=1,\ldots,n-1$ these thresholds, and let $\xi_0=0$ and $\xi_n=\infty$. The ith queue gets all the jobs that have sizes between $\xi_{m(i)-1}$ and $\xi_{m(i)}$, where $m(\cdot)$ is a one-to-one mapping from n queues onto n size-intervals, i.e., $(m(1), m(2), \ldots, m(n))$ is a permutation of $(1, 2, \ldots, n)$. The mapping $m(\cdot)$ indicates how to map size intervals to queues.

  Assuming continuous job-size distributions, with the SI strategy, we have

$$\lambda_i = \lambda[F(\xi_{m(i)}) - F(\xi_{m(i)-1})], \quad \rho_i = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} x \,\mathrm{d}F(x), \quad \omega_i = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} x^2 \,\mathrm{d}F(x).$$

  The SI strategy was proposed by Harchol-Balter et al. [9,5] (called as SITA-V in [5] and SITA-E for the case that each queue receives equal amount of load [9]). The optimality of the SI strategy for homogeneous systems is discussed in Section 3.
- *Nested size-interval* (*NSI*) *strategy*. We propose this strategy in order to generalize the SI strategy.

Definition 1. Suppose the minimum and maximum job sizes assigned to the ith queue are denoted by $\xi_i^{(0)}$ and $\xi_i^{(1)}$, respectively. A static strategy is a nested-size-interval (NSI) strategy if, for each pair of queues, say the ith and jth queues, either of the following is satisfied:

(i) $\xi_i^{(0)} < \xi_j^{(0)} \le \xi_j^{(1)} < \xi_i^{(1)}$ (the range of the jth queue is nested in the range of the ith queue),
(ii) $\xi_j^{(0)} < \xi_i^{(0)} \le \xi_i^{(1)} < \xi_j^{(1)}$ (the range of the ith queue is nested in the range of the jth queue),
(iii) $\xi_j^{(1)} \le \xi_i^{(0)}$ *or* $\xi_j^{(0)} \ge \xi_i^{(1)}$ (non-overlapping ranges).

In other words, the intervals of job sizes assigned to one queue either fall inside the intervals assigned to another queue, or all sizes assigned to one queue are shorter than the sizes assigned to the other queue. The optimality of the NSI strategy for heterogeneous systems is discussed in Section 3.

We can see that, if case (iii) is satisfied for all pairs of queues and neither case (i) nor case (ii) happens, the static strategy is an SI strategy, hence an SI strategy is a special case of the NSI strategies. Sometimes only one of cases (i) and (ii) never happens. The following definition imposes additional rules to prohibit either case (i) or case (ii), or both, for each pair of queues, in order to make an NSI strategy more restricted and specific.

Definition 2. We say the jth queue can be nested in the ith queue if cases (i) and (iii) in Definition 1 are the only allowable cases. We call this asymmetric and transitive relation a nesting relation, denoted by $j \prec i$. By symmetry, relation $i \prec j$ is defined if cases (ii) and (iii) in Definition 1 are the only allowable cases. If neither $j \prec i$ nor $i \prec j$ is defined for the NSI strategy, case (iii) in Definition 1 is the only allowable case.

Fig. 2 shows a set of nesting relations: $\{A \prec B, A \prec C, B \prec D, C \prec D\}$ and depicts three valid NSI strategies that assign interval ranges of job sizes to queues. Note that relation $A \prec D$ is implied due to transitivity. In the topmost assignment, jobs assigned to queue A are nested within jobs assigned to queue D. Jobs assigned to queue B and C are also nested within jobs assigned to queue D. In the middle example, jobs assigned to A are nested within B, whose jobs are nested within D. Jobs assigned to queue C are also nested within D. The bottom example shows an SI strategy.

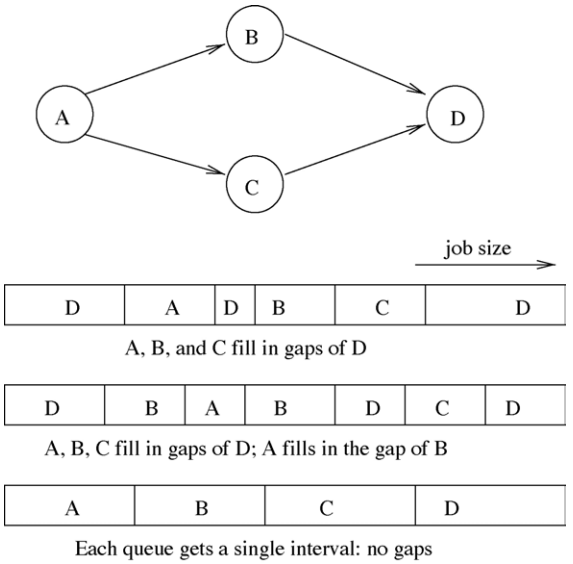

Fig. 2. Three examples of size interval allocations given by an NSI strategy restricted by a set of relations shown in the graph.

3. Optimal static strategies

We optimize within the class of static strategies. Theorem 3 states that the optimal static strategy is always an SI strategy for homogeneous systems.

Theorem 3. *The optimal static strategy for a homogeneous system of FCFS queues is an SI strategy, with respect to both the mean waiting time and the mean response time.*

Harchol-Balter et al. [9] provide an intuitive explanation for why under the SI strategy the mean response time is small: this strategy drastically reduces the variability of job sizes for each queue. Similarly, the optimality of the SI strategy stated in Theorem 3 might be intuitively explained as follows. Within all static strategies, strict thresholding might be the best way to divide the original arrival process $\lambda F(x)$ into n arrival processes with smallest possible job-size variability for each queue. Thus, with an optimal load partitioning, the SI strategy might be optimal in the entire class of static strategies. Unfortunately, this explanation is unsatisfactory, since the result does not hold for heterogeneous systems. Here is a counter example.

Example 4. Consider the scenario that jobs have only three different possible sizes: $x_1 = \theta$, $x_2 = 1$ and $x_3 = 1/\theta$, where $0 < \theta < 1$, and the loads of the three kinds of jobs are respectively ρ_ϵ, $\rho - 2\rho_\epsilon$, and ρ_ϵ, where ρ_ϵ is a small amount of load such that $\rho_\epsilon < \rho - c_2$, where c_2 is the capacity of the faster one of two queues.

We have not defined SI strategy for discrete distributions. However, A discrete distribution can be approximated by a series of continuous distributions, for example the one illustrated in Fig. 3(a). The limit of such series is illustrated in Fig. 3(b) – we consider this limit strategy is still an SI strategy.

There are two cases (mappings) for the SI strategy: the slower queue gets either (i) all jobs of size θ or (ii) all jobs of size $1/\theta$ (since we always have $\rho_\epsilon < \rho - c_2 < \rho_1$ and $\rho_\epsilon < \rho - c_2 \leq \rho - c_1 < \rho_2$).

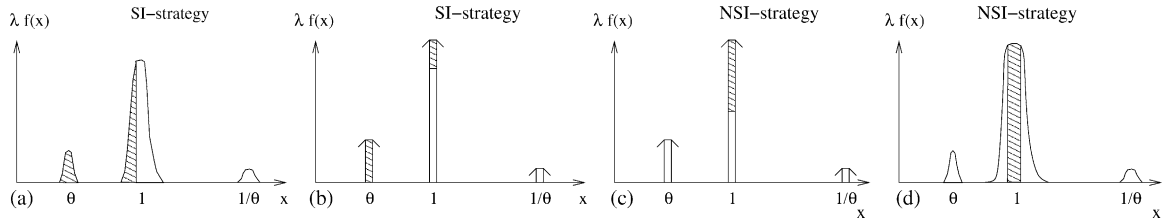

Fig. 3. The job-size assignment of an SI strategy for (a) continuous and (b) discrete distributions. The job-size assignment of an NSI strategy (it is not an SI strategy), for (c) discrete and (d) continuous distributions. Shaded areas are assigned to the slower queue.

Table 1
Optimal waiting times for the job-size distribution in Example 4

c_1	$E[W^{\mathrm{SI}}]^*$	$E[W^{\mathrm{NSI}}]^*$	$E[W^{\mathrm{R}}]^*$
0.5	9.189 (0.45)	9.25 (0.4512)	9.25 (0.45)
0.4	9.081 (0.3549)	9.101 (0.3561)	9.137 (0.3551)
0.3	8.743 (0.2601)	8.728 (0.2614)	8.784 (0.2606)
0.2	8.111 (0.1662)	8.077 (0.1676)	8.136 (0.1669)

Fig. 3(b) shows the case (i). In either case, a portion of jobs of size 1 are also assigned to this slower queue by random splitting.

Consider an alternative static strategy that assigns only a portion of jobs of size 1 to the slower queue, as illustrated in Fig. 3(c). This is in fact an NSI strategy but not an SI strategy, since it can be approximated by a series of continuous distributions for example the one illustrated in Fig. 3(d).

Let $\theta = 0.5$, $\rho_\epsilon = 0.05$, $\rho = 0.9$. The following Table 1 compares the optimal $E[W]$'s for the SI strategy, the alternate NSI strategy and the random strategy. Values in parentheses are optimal load partitionings, namely ρ_1^*. Note that we assume $c_1 + c_2 = 1$ and $\rho_2^* + \rho_1^* = \rho$.

As we can see in Table 1, the alternative NSI strategy is better if c_1 is small (for $c_1 = 0.2$ and 0.3). This shows that, with certain job size distributions and arrival rates, the optimal static strategy will never be an SI strategy for some heterogeneous systems.

Generalizing SI strategies to restricted NSI strategies, we can find a set of static strategies that contains the optimal static strategy.

Theorem 5. *For a heterogeneous system of FCFS queues, the optimal static strategies* (*with respect to the mean waiting time*) *is an NSI strategy where a slower queue can be nested in a faster queue* (*cf. Definition* 2).

Let us provide an intuitive explanation for why the optimal SI strategy is probably not an optimal static strategy for heterogeneous systems. To improve performance, all queues desire variability of jobs as small as possible. However, the variabilities of jobs for different queues are correlated to each other, so we have to balance their desires. In a homogeneous system, their desires are equally strong, whereas in a heterogeneous system, a slower queue has a stronger desire for less job variability than a faster queue, since the job variability has a stronger effect in deteriorating the performance on a slower queue than on a faster queue. As the difference between capacities of two queues increases, the desire for a small job variability from the slower queue gets stronger. When the capacity difference becomes large enough, the

slower queue gets priority over a faster queue in choosing job sizes. As shown in Fig. 3(b) and (c), the alternative NSI strategy offers the slower queue a size variability of zero, whereas the SI strategy does not. In short, the optimal static strategy is a strategy that discriminates against the faster queue by assigning a better interval to the slower queue.

By now, the optimal static strategies for homogeneous and heterogeneous systems are identified but not proved – we delay the proof of Theorems 3 and 5 to Section 5.

4. Mapping of size intervals

Although the optimal SI strategy is not always an optimal static strategy, it is simpler than the general case of the NSI strategy. Therefore in this section we restrict our study within the set of SI strategies in a *heterogeneous* system. The SI strategy has been shown to outperform the random strategy by several orders of magnitude when the job-size distribution is a heavy-tailed distribution, bounded Pareto in particular, for homogeneous systems [9]. In this section, we first show some numerical results of SI strategies for heterogeneous systems, with some particular job-size distributions. These results show that the mapping of size intervals to the queues, or simply the *mapping*, significantly affects the mean waiting time, and demonstrate that the problem of finding the best mapping is probably very difficult for general distributions. Then we present a class of distributions that are mapping-invariant for two-queue systems.

Three kinds of distributions are used in this section:

Bounded Pareto distribution. A bounded-Pareto distribution has a power-law tail. The CDF is $F(x) = [\kappa^{-\alpha} - x^{-\alpha}]/[\kappa^{-\alpha} - \eta^{-\alpha}]$, where κ is the lower bound of the random variable and η is the upper bound. We set the ratio η/κ to a fixed value (10^4 by default) and then choose a κ such that $E[X] = 1/\mu$.

Log-normal distribution. A random variable X has a log-normal distribution if $\log X$ is Gaussian distributed. Its CDF is $(1/2)[1 + \mathrm{erf}((\ln x - m)/(s\sqrt{2}))]$, where m and s are the mean and deviation of the Gaussian $\log X$, respectively, and $\mathrm{erf}(x) = (2/\sqrt{\pi}) \int_0^x \mathrm{e}^{-\tau^2}\, \mathrm{d}\tau$. We choose an m such that $E[X] = 1/\mu$.

Weibull distribution. A random variable X has a Weibull distribution if X^α is exponential. Its CDF is $1 - \exp(-\beta x^\alpha)$, where $\alpha \geq 0$. It degenerates to an exponential distribution when $\alpha = 1$. We choose a β such that $E[X] = 1/\mu$.

Each of the above has one parameter (α, s, and α respectively) that controls its variability: from heavy-tailed distributions to approximately deterministic. Also, the coefficient of variation is a monotonic function of the control parameter for each kind of the distributions.

4.1. Numerical results

We show numerical results of heterogeneous systems under the SI strategy for the above-mentioned three classes of distributions. We shall see that, for heterogeneous queues, the mapping of size intervals to queues, namely $m(\cdot)$, greatly affects the waiting time of the SI strategy. Two particular mappings are of special interest: the ascending mapping and the descending mapping. With the ascending (descending) mapping, the queues are mapped in the ascending (descending) order of their capacities: the slowest (fastest) queue gets the first size interval containing shortest jobs. In other words, with $c_1 \leq c_2 \leq \cdots \leq c_n$, the ascending mapping means $m(i) = i$ and the descending mapping means $m(i) = n - i + 1$, where $i = 1, 2, \ldots, n$. In a system of two queues, the ascending and descending mappings are the only two mappings.

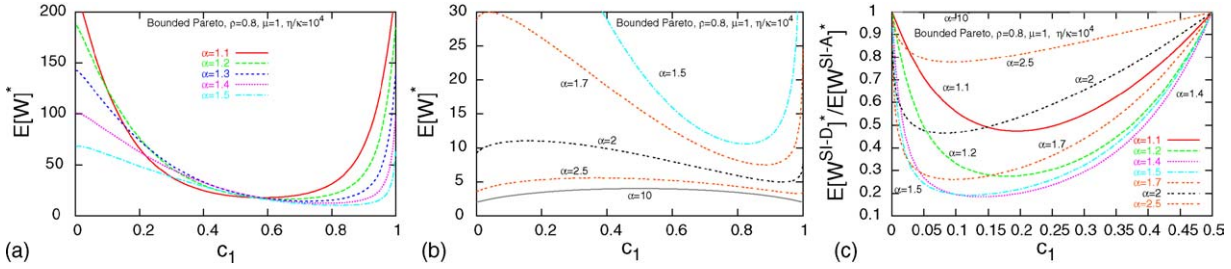

Fig. 4. Two heterogeneous queues using SI strategies of two different mappings, with bounded-Pareto distributed job sizes. (a, b) The optimal mean waiting times as functions of the capacity of the first queue. (c) The ratio of optimal mean waiting times between the ascending and descending mappings, as functions of the capacity of the slower queue.

Without loss of generality, let $E[X] = 1/\mu = 1$. The numerical results in Figs. 4–6 show how the optimal mean waiting time $E[W^{\mathrm{SI}}]^*$ changes as the capacity of the first queue changes, for a two-queue system at $\rho = 0.8$. In Figs. 4(a) and (b), 5(a) and (b) and 6, the X-dimensions are the capacity of the first queue, and the Y-dimensions are the mean waiting time. Since we require a unit total capacity, i.e., $c_1 + c_2 = 1$, the difference of mean waiting times between the two mappings can be seen by comparing each curve with its reflex over $c_1 = 0.5$ (clearly the smaller waiting time is better). In order to show this difference, in Figs. 4(c) and 5(c) we plot the ratio of the optimal mean waiting time under the descending mapping, $E[W^{\mathrm{SI\text{-}D}}]^*$, to that under the ascending mapping, $E[W^{\mathrm{SI\text{-}A}}]^*$, as functions of the capacity of the slower queue (so the range of the X-axis is $[0, 0.5]$). Figs. 4–6 are for bounded Pareto job-size distributions, Weibull job-size distributions, and log-normal job-size distributions, respectively.

As we can see in these figures, the descending mapping is better for bounded-Pareto distributions, whereas the ascending mapping is better for Weibull distributions (and for the exponential job-size distribution since it is a special case of Weibull distributions). Similar results can be observed with other load values. For the log-normal distributions, interestingly, the two mappings are equally good. The difference between two mappings is on the magnitude of computational errors.

If the variability of job sizes becomes large (corresponding to a small α) for bounded-Pareto and Weibull, the difference between optimal mean waiting times of the two mappings becomes very sensitive to c_1. So in reality if we have a heterogeneous system under the SI strategy, the mapping of size intervals must be taken into account.

Fig. 7 shows the best mapping of size intervals for the system with three heterogeneous queues and exponential service times, at $\rho = 0.8$. In Fig. 7, the X-dimension is the capacity of the slowest queue, with

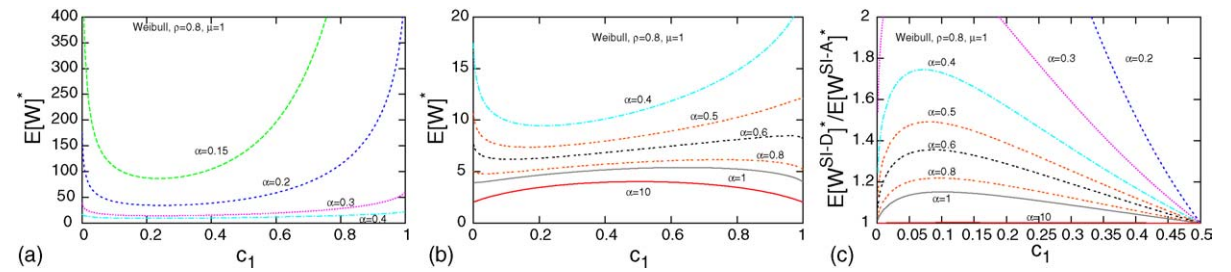

Fig. 5. With Weibull distributed job-sizes (cf. Fig. 4).

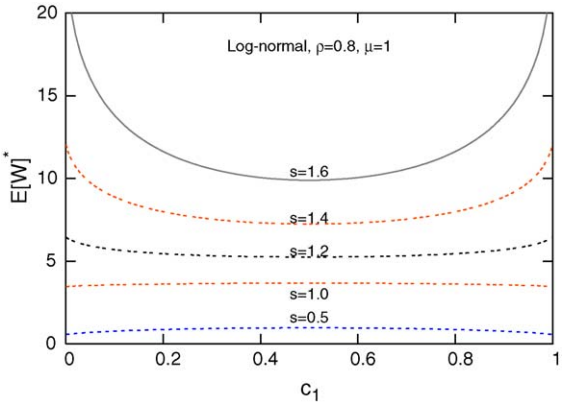

Fig. 6. The optimal mean waiting time as functions of the capacity of the first queue, for two heterogeneous queues using SI strategy. The job size distribution is log-normal.

a range of (0, 0.33), and the *Y*-dimension is the capacity of the second to the slowest queue, with a range of (0, 0.5). The coordinates of each point indicate a combination of three capacities (sum to one). The mark of a point represents the best mapping for the corresponding capacity combination. There are six different mappings but we can see only two kinds of marks in the figure: if the slowest queue has a small capacity (less than about 0.12), the ascending mapping (1–2–3) is best; if the slowest queue has a larger capacity (more than 0.16), the best mapping is (2–1–3), i.e., the slowest queue gets the middle-sized jobs whereas the fastest queue still gets the longest jobs.

From these figures we can see that the best mapping is distribution-dependent, either for heavy-tailed distributions or for distributions close to a deterministic value. From Fig. 7 we notice that, for exponential distributions, the best mapping depends on the capacity of the slowest queue. In fact, all these job size

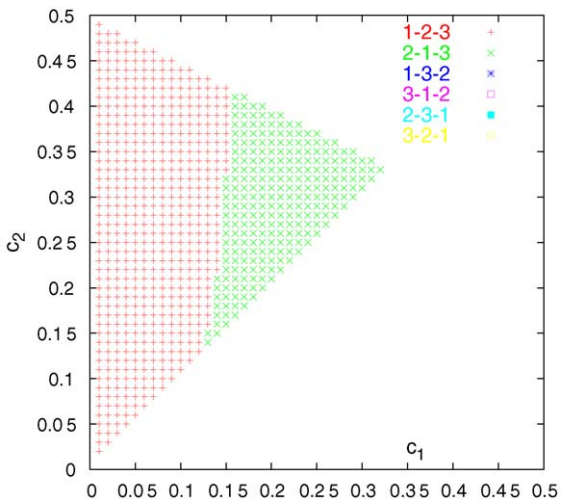

Fig. 7. The best mapping for different capacity combinations for three heterogeneous queues using SI strategy, with exponentially distributed job sizes.

distributions we studied here are somewhat "regular"; we believe the problem of finding the optimal mapping for a general distribution is a hard problem, analytically and computationally.

4.2. Mapping-invariant distributions

It can be observed in Fig. 6 that the mean waiting times for log-normal job-size distributions seem to be invariant to the mapping of size intervals in a two-queue system. In fact, this is not a coincidence; we have the following proposition.

Proposition 6. *For a heterogeneous system with two queues, if the partial load* $\rho(x) = \lambda \int_0^x t\,\mathrm{d}F(t)$ *satisfies* $\rho(x) = \rho - \rho(\psi/x)$ *for some positive* ψ*, then the optimal mean waiting time is independent of the mapping of queue. Only load partitioning affects the mean waiting time.*

Proof. Let ξ be the threshold used in the ascending mapping and $\xi' = \psi/\xi$ be the one used in descending mapping. Let λ_i, ρ_i, ω_i be the corresponding quantities under the ascending mapping and λ_i', ρ_i', ω_i' be those under the descending mapping. Clearly we have $\rho_1 = \rho(\xi) = \rho - \rho(\xi') = \rho_2'$ and similarly $\rho_2 = \rho_1'$. In other words, load partitioning is symmetric for two mappings of queues. Suppose the distribution is continuous. Taking the derivatives on both sides of $\rho(x) = \rho - \rho(\psi/x)$ (note that $\rho(x) = \lambda \int_0^x t\,\mathrm{d}F(t)$), we get $f(x) = (\psi^2/x^4) f(\psi/x)$, where $f(x) = \mathrm{d}F(x)/\mathrm{d}x$ is the probability density function (PDF). Then we have

$$\lambda_1 = \lambda \int_0^{\xi} \frac{\psi^2}{x^4} f\left(\frac{\psi}{x}\right) \mathrm{d}x = -\lambda \int_0^{\xi} \frac{\psi}{x^2} f\left(\frac{\psi}{x}\right) \mathrm{d}\left(\frac{\psi}{x}\right) = \frac{\lambda}{\psi} \int_{\xi'}^{\infty} y^2 f(y)\,\mathrm{d}y = \frac{\omega_1'}{\psi}.$$

Similarly we get $\lambda_2' = \omega_2'/\psi$. Due to symmetry, we get $\omega_1 = \psi\lambda_1'$ and $\omega_2 = \psi\lambda_2'$. Then

$$E[W^{\text{SI-A}}] = \frac{1}{2\lambda} \sum_{i=1}^{2} \frac{\lambda_i \omega_i}{c_i(c_i - \rho_i)} = \frac{1}{2\lambda} \sum_{i=1}^{2} \frac{\left(\frac{\omega_i'}{\psi}\right)(\psi\lambda_i')}{c_i(c_i - \rho_i')} = E[W^{\text{SI-D}}],$$

for any ξ and the corresponding ξ'. Therefore their optimums are also same. □

Both the log-normal distribution and the job-size distribution in Example 4 satisfy $\rho(x) = \rho - \rho(\psi/x)$ and hence is mapping-invariant in two-queue systems.

5. Proofs on the optimal static strategies

We have shown the main results in Section 3 but delayed the proofs to this section. In this section, we prove Theorems 3 and 5 and show another proposition that helps seek the optimal NSI strategy. For each of the proofs, first we show that the corresponding theorem holds for systems with two queues, and then extend the result to systems with multiple queues. We use notation $\alpha_i = 1/[c_i(c_i - \rho_i)]$ to simplify the equations in the rest of this section. Note that by (5) the mean waiting time of a static strategy is $E[W^{\text{S}}] = \left[\sum_{i=1}^{n} \alpha_i \lambda_i \omega_i\right]/(2\lambda)$. Job size distributions are assumed to be continuous. For discrete distributions, we can always argue by continuous approximations.

First we need the following lemma that shows an inequality between the three quantities, λ_i, ρ_i, and ω_i, if an SI strategy is used:

Lemma 7. *Let X_i and X_j be two job size distributions and λ_i, ρ_i, and ω_i (λ_j, ρ_j, and ω_j) be the corresponding arrival rate, load and second-order load of X_i (X_j), respectively. If $X_i \le \xi \le X_j$ holds, then $\lambda_i/\rho_i \ge \lambda_j/\rho_j$ and $\omega_i/\rho_i \le \omega_j/\rho_j$. If $\Pr[X_i < X_j] > 0$, then $\lambda_i/\rho_i > \lambda_j/\rho_j$ and $\omega_i/\rho_i < \omega_j/\rho_j$.*

Proof. Let $F_i(\cdot)$ and $F_j(\cdot)$ be the CDFs of X_i and X_j, respectively. Clearly we have $F_i(\xi) = 1$ and $F_j(\xi) = 0$. Then,

$$\frac{\lambda_i}{\rho_i} = \frac{\lambda \int_0^{\xi} \mathrm{d}F_i(x)}{\lambda \int_0^{\xi} x\,\mathrm{d}F_i(x)} \ge \frac{\int_0^{\xi} \mathrm{d}F_i(x)}{\xi \int_0^{\xi} \mathrm{d}F_i(x)} = \frac{1}{\xi} \ge \frac{\lambda \int_{\xi}^{\infty} \mathrm{d}F_i(x)}{\lambda \int_{\xi}^{\infty} x\,\mathrm{d}F_i(x)} = \frac{\lambda_j}{\rho_j}. \tag{6}$$

Similarly, $\omega_i/\rho_i \le \xi \le \omega_j/\rho_j$. If $\Pr[X_i < X_j] > 0$, i.e., either $\Pr[X_i < \xi] > 0$ or $\Pr[X_j > \xi] > 0$, or both, holds, then at least one of two inequalities in (6) strictly holds, i.e., $\lambda_i/\rho_i > \lambda_j/\rho_j$. Similarly, under the same condition we have $\omega_i/\rho_i < \omega_j/\rho_j$. □

For systems with two queues, consider two actions to improve the mean waiting time: transferring some load from one queue to the other, or swapping loads between two queues. (Here by saying load transferring or swapping, we actually mean to transfer or to swap the jobs that constitute the specified load.)

Load transferring. We transfer some jobs from the second queue to the first queue. Let the arrival rate, the load, and the second-order load of transferred jobs be $\Delta\lambda$, $\Delta\rho$, and $\Delta\omega$, respectively. They can be either all positive or all negative (the latter case means we are actually transferring jobs from the first queue to the second queue). If we assume that the number of transferred jobs are very small, the change of the mean waiting time due to transferring can be approximated by computing partial derivatives of $E[W^{\mathrm{S}}]$ in (5) with respect to λ, ρ, and ω, i.e.,

$$\Delta(E[W^{\mathrm{S}}]) = \frac{1}{2\lambda}\{[\alpha_1\lambda_1 - \alpha_2\lambda_2]\Delta\omega + [\alpha_1\omega_1 - \alpha_2\omega_2]\Delta\lambda + [c_1\alpha_1^2\lambda_1\omega_1 - c_2\alpha_2^2\lambda_2\omega_2]\Delta\rho\}. \tag{7}$$

Load swapping. We swap some load between two queues. Let λ_i^s, ρ_i^s, and ω_i^s, $i = 1, 2$, be the arrival rate, the load, and the second-order load that are *swapped* from the ith queue to the other, and let λ_i^r, ρ_i^r, and ω_i^r be the corresponding quantities that *remain* in the ith queue. Then we have

$$\begin{aligned} 2\lambda E[W^{\mathrm{S}}] &= \alpha_1\lambda_1\omega_1 + \alpha_2\lambda_2\omega_2 = \alpha_1(\lambda_1^r + \lambda_1^s)(\omega_1^r + \omega_1^s) + \alpha_2(\lambda_2^r + \lambda_2^s)(\omega_2^r + \omega_2^s) \\ &= 2\lambda E[\tilde{W}^{\mathrm{S}'}] - (\alpha_1 + \alpha_2)[\lambda_1^s - \lambda_2^s][\omega_1^s - \omega_2^s] + [\alpha_1\lambda_1 - \alpha_2\lambda_2][\omega_1^s - \omega_2^s] \\ &\quad + [\lambda_1^s - \lambda_2^s][\alpha_1\omega_1 - \alpha_2\omega_2], \end{aligned} \tag{8}$$

where $E[\tilde{W}^{\mathrm{S}'}] = [\alpha_1(\lambda_1^r + \lambda_2^s)(\omega_1^r + \omega_2^s) + \alpha_2(\lambda_1^s + \lambda_2^r)(\omega_1^s + \omega_2^r)]/(2\lambda)$ is the mean waiting time after the swap of loads.

Proof. (Theorem 3) For load swapping, we let ρ_1^s be the load of the jobs above ξ in the first queue, for some ξ, and let ρ_2^s be the load of jobs below ξ in the second queue, as illustrated by shaded areas in Fig. 8(a). At $\xi = 0$, $\rho_1^s = \rho_1 > 0 = \rho_2^s$, while at $\xi = \infty$, $\rho_1^s = 0 < \rho_2 = \rho_2^s$. Quantities ρ_1^s and ρ_2^s are continuous, monotonically decreasing and increasing functions of ξ, respectively, so they must meet somewhere. We can find a ξ such that $\rho_1^s = \rho_2^s$. Assuming so, by Lemma 7, we get $\lambda_1^s \le \lambda_2^s$ and $\omega_1^s \ge \omega_2^s$.

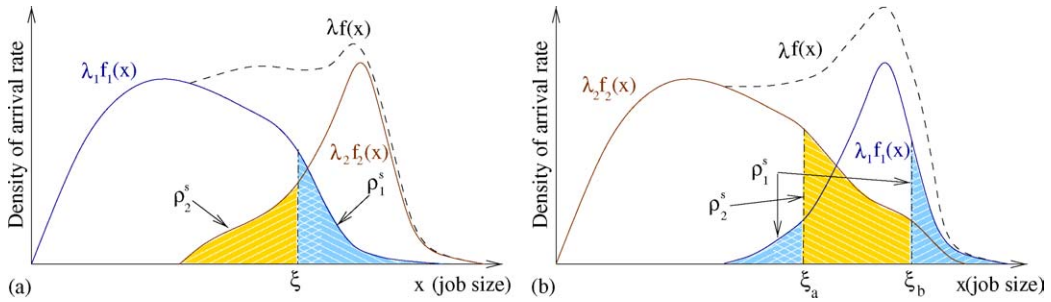

Fig. 8. The density function of arrivals $\lambda f(x)$ (dashed curve) is the sum of two functions, $\lambda_1 f_1(x)$ and $\lambda_2 f_2(x)$ (solid curves), assigned to two queues. (a) After swapping jobs with equal loads between two queues (differently shaded areas), an SI strategy is obtained. (b) After swapping jobs with equal arrival rates and loads, an NSI strategy is obtained.

Then we consider four conditions:

1. If both $\alpha_1\lambda_1 \geq \alpha_2\lambda_2$ and $\alpha_1\omega_1 \leq \alpha_2\omega_2$ are satisfied, from (8), we have
$$E[W^{\mathrm{S}}] \geq E[\tilde{W}^{\mathrm{S}'}] = E[\tilde{W}^{\mathrm{SI}}], \tag{9}$$
since $\lambda_1^s \leq \lambda_2^s$ and $\omega_1^s \geq \omega_2^s$. Note that, after load swapping, the static strategy becomes an SI strategy. If, before the swapping, the static strategy is not yet an SI strategy, the inequality in (9) strictly holds, since we have both $\lambda_1^s < \lambda_2^s$ and $\omega_1^s > \omega_2^s$ by Lemma 7.
2. If both $\alpha_1\lambda_1 \leq \alpha_2\lambda_2$ and $\alpha_1\omega_1 \geq \alpha_2\omega_2$ are satisfied, we can swap the first queue and the second queue, then case 1 is satisfied.
3. If both $\alpha_1\lambda_1 < \alpha_2\lambda_2$ and $\alpha_1\omega_1 < \alpha_2\omega_2$ are satisfied, by (7), as long as $c_1 \leq c_2$, we can continuously transfer jobs from the second queue to the first queue, with any job-size distribution, such that $\Delta(E[W^{\mathrm{S}}]) < 0$, i.e. the mean waiting time is strictly improving, until either $\alpha_1\lambda_1 \geq \alpha_2\lambda_2$ or $\alpha_1\omega_1 \geq \alpha_2\omega_2$ is satisfied. (Note that $a_1b_1 \leq a_2b_2$ if $0 \leq a_1 \leq a_2$ and $0 \leq b_1 \leq b_2$.)
4. If both $\alpha_1\lambda_1 > \alpha_2\lambda_2$ and $\alpha_1\omega_1 > \alpha_1\omega_1$ are satisfied, by (7), as long as $c_1 \geq c_2$, we can transfer jobs of any sizes from first queue to the second to improve the mean waiting time strictly, until either $\alpha_1\lambda_1 \leq \alpha_2\lambda_2$ or $\alpha_1\omega_1 \leq \alpha_2\omega_2$ is satisfied.

Note that for homogeneous queues we have $c_1 = c_2$. Then one of the four conditions is satisfied. If either Conditions 3 or 4 is satisfied, we can improve the mean waiting time by continuously transferring some jobs from one queue to the other until either Conditions 1 or 2 is satisfied. If Condition 2 is satisfied we can swap two queues so that Condition 1 is satisfied. If Condition 1 is satisfied, we can swap portions of loads of two queues as illustrated in Fig. 8(a). After load swapping, the static strategy becomes an SI strategy, and the mean waiting time is decreased (strictly if it is not an SI strategy before swapping). In short, for any non-SI static strategy, we can always find an SI strategy with a smaller mean waiting time. Hence the optimal static strategy for two equal queues is an SI strategy.

The result can by extended to n queues by pairwise using the process described above to improve the mean waiting time, until the static strategy converges to an SI strategy. Since for a homogeneous system $E[\hat{X}]$ is invariant, this result also applies to mean response time. □

Now we look at Theorem 5. We need to prove a series of lemmas as follows:

Lemma 8. *Suppose $g(t)$ is a monotonic positive function in interval $[a, b]$. For interval (a_1, b_1) such that $a \le a_1 < b_1 \le b$ and $b - a = 2(b_1 - a_1)$, suppose $g(t)$ satisfies $\int_{a_1}^{b_1} g(t)\,\mathrm{d}t = \int_a^{a_1} g(t)\,\mathrm{d}t + \int_{b_1}^{b} g(t)\,\mathrm{d}t$. Then $\int_{a_1}^{b_1} \mathrm{d}t/g(t) \le \int_a^{a_1} \mathrm{d}t/g(t) + \int_{b_1}^{b} \mathrm{d}t/g(t)$.*

Proof. The proof uses the convexity of function $1/x$ for $x > 0$. Let

$$h(x) = \alpha x + \beta \equiv \frac{1}{g(b_1) - g(a_1)} \left[\frac{g(b_1) - x}{g(a_1)} + \frac{x - g(a_1)}{g(b_1)} \right],$$

be a linear function such that $h(g(a_1)) = 1/g(a_1)$ and $h(g(b_1)) = 1/g(b_1)$. By the convexity of $1/x$ and monotonicity of $g(\cdot)$, we have $h(g(t)) \ge 1/g(t)$ for $t \in [a_1, b_1]$ and $h(g(t)) \le 1/g(t)$ for $t \in [a, a_1] \cup [b_1, b]$. Then

$$\int_{a_1}^{b_1} \frac{\mathrm{d}t}{g(t)} \le \int_{a_1}^{b_1} h(g(t))\,\mathrm{d}t = \alpha \left(\int_{a_1}^{b_1} g(t)\,\mathrm{d}t \right) + \beta(b_1 - a_1)$$

because of the linearity of $h(\cdot)$. Similarly (note that $b - a = 2(b_1 - a_1)$)

$$\int_a^{a_1} \frac{\mathrm{d}t}{g(t)} + \int_{b_1}^{b} \frac{\mathrm{d}t}{g(t)} \ge \int_a^{a_1} h(g(t))\,\mathrm{d}t + \int_{b_1}^{b} h(g(t))\,\mathrm{d}t = \alpha \left(\int_{a_1}^{b_1} g(t)\,\mathrm{d}t \right) + \beta(b_1 - a_1).$$

With two inequalities above, we complete the proof of Lemma 8. □

Lemma 9. *Suppose two thresholds, ξ_1 and ξ_2, $0 \le \xi_1 < \xi_2$, divide the job sizes into three size intervals, such that $\rho_2 = \rho_1 + \rho_3 = \rho/2$, where ρ_i is load of the ith interval. Then, $\lambda_2 \le \lambda_1 + \lambda_3$ if $\omega_2 = \omega_1 + \omega_3$, whereas $\omega_2 \le \omega_1 + \omega_3$ if $\lambda_2 = \lambda_1 + \lambda_3$, where λ_i and ω_i are the arrival rate and the second-order load of the ith interval, respectively, for $i = 1, 2, 3$.*

Proof. Let $\rho(x) := \lambda \int_0^x t\,\mathrm{d}F(t)$, and let $x(r) := \rho^{-1}(r)$ be the inverse function of $\rho(x)$. Assume $\xi_0 = 0$ and $\xi_3 = \infty$. Then we have, for $i = 1, 2, 3$,

$$\lambda_i = \lambda \int_{\xi_{i-1}}^{\xi_i} \mathrm{d}F(x) = \int_{\xi_{i-1}}^{\xi_i} \frac{1}{x}\,\mathrm{d}\rho(x) = \int_{r_{i-1}}^{r_i} \frac{\mathrm{d}r}{x(r)}, \qquad \omega_i = \int_{\xi_{i-1}}^{\xi_i} x\,\mathrm{d}\rho(x) = \int_{r_{i-1}}^{r_i} x(r)\,\mathrm{d}r,$$

where $r_i = \rho(\xi_i)$, in particular, $r_0 = 0$ and $r_3 = \rho$.

Suppose $\omega_2 = \omega_1 + \omega_3$. Let $a = r_0 = 0$, $a_1 = r_1$, $b_1 = r_2$, $b = r_3 = \infty$, and $g(\cdot) = x(\cdot)$, which is an increasing function. Using Lemma 8 we get $\lambda_2 \le \lambda_1 + \lambda_3$. Supposing $\lambda_2 = \lambda_1 + \lambda_3$ and using Lemma 8 again with $g(\cdot) = 1/x(\cdot)$, we get $\omega_2 \le \omega_1 + \omega_3$. □

Proof. (Theorem 5) Again, first we consider two queues. Without loss of generality, we assume $c_1 < c_2$. Hence the $c_1 \ge c_2$ part of Condition 4 in the proof of Theorem 3 no longer holds. However, we show that the following replacement of the Condition 4 holds:

4′. If both $\alpha_1\lambda_1 > \alpha_2\lambda_2$ and $\alpha_1\omega_1 > \alpha_2\omega_2$ are satisfied, we can find a NSI strategy, where the first queue can be nested in the second queue, i.e., $1 \prec 2$, provides a lower mean waiting time than the original

static strategy.

Suppose there are two thresholds ξ_a and ξ_b such that $\xi_a < \xi_b$. Now we swap load between two queues. Let ρ_1^s be the load of the jobs below ξ_a and above ξ_b in the first queue, and ρ_2^s be the load of jobs between ξ_a and ξ_b in the second queue, as illustrated by the shaded areas in Fig. 8(b). We swap two loads, if both $\rho_1^s = \rho_2^s$ and $\lambda_1^s = \lambda_2^s$ are satisfied. Then, by Lemma 9, we have $\omega_1^s \geq \omega_2^s$. Then, from (8) we get $E[W^S] \geq E[W^{S'}] = E[\tilde{W}^{NSI}]$. Note that after swapping, the static strategy becomes an NSI strategy where the first (slower) queue can be nested in the second (faster) queue, i.e., the slower queue gets the inner range.

For completing the claim in Condition 4′, it remains to show that there are actually such ξ_a and ξ_b satisfying $\rho_1^s = \rho_2^s$ and $\lambda_1^s = \lambda_2^s$. First let $\xi_a = 0$ and find ξ_b such that $\rho_1^s = \rho_2^s$. This can be done in the same way as in the proof of Theorem 3. At this time, $\lambda_1^s \leq \lambda_2^s$ due to Lemma 7. Now we shift ξ_a to the right on the real axis and also shift ξ_b to the right accordingly such that $\rho_1^s = \rho_2^s$. This can also be done until ξ_b goes to infinity. At $\xi_b = \infty$ we have $\lambda_1^s \geq \lambda_1^s$ because of once again Lemma 7. Then before ξ_b approaches infinity, there must be a value of ξ_a and the corresponding ξ_b such that both $\rho_1^s = \rho_2^s$ and $\lambda_1^s = \lambda_2^s$ are satisfied, due to continuity of all these quantities. Hence the claim in Condition 4′ is true.

With Conditions 1, 2, 3, and 4′, in the same way as the argument in the proof of Theorem 3, it is proved that, for any static strategy, there is an NSI strategy that improves the mean waiting time, for two heterogeneous queues. Note again that an SI strategy is a special case of NSI strategies. With this NSI strategy, the slower queue can be nested in the faster queue, by the claim of Condition 4′. By doing pairwise load transferring and swapping, this results can be extended to multiple queues. Hence we have Theorem 5. □

We can see that the key elements of these proofs are the two measures, namely $\Lambda := \alpha_i\lambda_i$ and $\Omega := \alpha_i\omega_i$, for the ith queue, as shown in Conditions 1–4 and 4′. For *each* pair of queues, generally we have two scenarios:

(i) One queue has a greater Λ whereas the other has a greater Ω (Conditions 1 and 2). In this case, pairwise swapping of equal amounts of loads, as illustrated in Fig. 8(a), improves the mean waiting time.
(ii) One queue, say X, gets both a greater Λ and a greater Ω than the other queue, say Y (Conditions 3 and Condition 4 or 4′). In this case,

(ii-a) if the capacity of X is greater than or equal to that of Y, we can transfer some load from X to Y in order to improve the mean waiting time;
(ii-b) if the capacity of X is strictly less than that of Y, load transferring cannot guarantee an improvement in the mean waiting time. The NSI strategy can be used to improve the mean waiting time: we add a nesting relation between X and Y, i.e., $X \prec Y$, and do load swapping as illustrated in Fig. 8(b).

From the proof of Theorem 5, we can observe that Condition 4′ does not actually assume either $c_1 \geq c_2$ or $c_1 < c_2$ (in Condition 4 we do have such assumption). Hence if Condition 3 is satisfied, alternatively, we can swap two queues so that Condition 4′ is satisfied, i.e., we can find a better NSI strategy totally without load transferring. In other words, we can merge case (ii-a) to case (ii-b) and then replace the case (ii) above with a different operation:

(ii′) One queue, say X, gets both greater Λ and greater Ω than the other queue, say Y. The NSI strategy can be used to improve the mean waiting time: we add a nesting relation between X and Y, i.e., $X \prec Y$, and do load swapping as illustrated in Fig. 8(b).

Clearly, due to Condition 3, the mean waiting time of this NSI strategy cannot be optimal if X is not slower than Y. However, it can be optimal given that the load assigned to each queue cannot be changed. We summarize this observation with the following proposition. (Note that the mean service time is fixed if the load partitioning is fixed: cf. (1), and therefore the result also applies to mean response time.)

Proposition 10. *For a heterogeneous system with FCFS queues, if the load partitioning is fixed, the optimal static strategy (for mean waiting time and mean response time) is an NSI strategy with a set of nesting relations. A relation $X \prec Y$ is added to this set if Λ and Ω of X are both greater than those of Y.*

The difference between Theorem 5 and Proposition 10 is that they use a different set of relations. In Theorem 5, a slower queue can be nested in a faster queue whereas, in Proposition 10, a queue with both greater Λ and greater Ω can be nested in the other. Moreover, Proposition 10 requires fixed load partitioning. The implication of Proposition 10 is two-fold. First, for fixed load partitioning, in particular the proportional load partitioning (where $\rho_i = \rho c_i$), we can still find an optimal NSI strategy to minimize mean waiting and response times. Note that the proportional load partitioning is safe (it does not overload any of the queues as long as $\rho < 1$) in the case that the load of the entire system, ρ, is hard to estimate. Second, for an NSI strategy, fewer the number of nesting relations is, more simple and approachable the NSI strategy would be. In Theorem 5, we assume there is a nesting relation for each pair of queues of unequal capacities. However, by load transferring and Proposition 10, we can get an NSI strategy where a nesting relation exists only if the slower queue has both greater Λ and Ω. In other words, some nesting relations can be eliminated so that it becomes easier to search for the optimal static strategy. It is not always possible to remove all the relations, though, as in the case of Example 4; but if one manages to do so, the optimal static strategy degenerates to an SI strategy, and the problem is then simplified to finding the best mapping and the optimal load partitioning $[\rho_i]_{i=1}^n$.

6. Conclusion

In this paper, we investigate parallel queueing systems with separate heterogeneous queues, using stochastic, size-aware, static strategies. For first-come first-serve (FCFS) queues, we prove that there is a size-interval strategy that optimizes mean response and waiting times within all *static* strategies, if the system is homogeneous, whereas a counter-example is found for a heterogeneous system. Then we prove that there is a nested size-interval based strategy that optimizes a heterogeneous system. We also study the effects of the mapping of size intervals on the mean waiting time with three kinds of job-size distributions, and show that the best mapping is hard to determine.

References

[1] S.C. Borst, Optimal probabilistic allocation of customer types to servers, in: ACM SIGMETRICS, 1995, pp. 116–125.
[2] J.P. Buzen, P.P.-S. Chen, Optimal load balancing in memory hierarchies, IFIP 74 (1974) 271–275.

[3] Y.-C. Chow, W.H. Kohler, Models for dynamic load balancing in a heterogeneous multiple processor system, IEEE Trans. Comput. 28 (5) (1979) 354–361.
[4] R.B. Cooper, Introduction to Queueing Theory, 2nd ed., Elsevier/North-Holland, 1981.
[5] M.E. Crovella, M. Harchol-Balter, C.D. Murta, Task assignment in a distributed system: improving performance by unbalancing load, in: ACM SIGMETRICS, 1998, pp. 268–269.
[6] A. Ephremides, P. Varaiya, J. Walrand, A simple dynamic routing problem, IEEE Trans. Autom. Control 25 (4) (1980) 690–693.
[7] M. Escobar, A.R. Odoni, E. Roth, Approximate solution for multi-server queueing systems with Erlangian service times, Comput. Oper. Res. 29 (10) (2002) 1353–1374.
[8] B. Hajek, Optimal control of two interacting service stations, IEEE Trans. Autom. Control 29 (6) (1984) 491–499.
[9] M. Harchol-Balter, M.E. Crovella, C.D. Murta, On choosing a task assignment policy for a distributed server system, J. Parallel Distribut. Comput. 59 (1999) 204–228.
[10] M. Mitzenmacher, How useful is old information? IEEE Trans. Parallel Distribut. Syst. 11 (1) (2000) 6–20.
[11] L.M. Ni, K. Hwang, Optimal load balancing in a multiple processor with many job classes, IEEE Trans. Software Eng. 11 (5) (1985) 491–496.
[12] K. Oida, S. Saito, A packet-size aware adaptive routing algorithm for parallel transmission server system, J. Parallel Distribut. Comput. 64 (2004) 36–47.
[13] K. Oida, K. Shinjo, Characteristics of deterministic optimal routing for two heterogeneous parallel servers, Int. J. Found. Comput. Sci. 12 (6) (2001) 775–790.
[14] A.N. Tantawi, D. Towsley, Optimal static load balancing in distributed computer systems, J. ACM 32 (2) (1985) 445–465.
[15] W. Winston, Optimality of the shortest line discipline, J. Appl. Prob. 13 (1977) 826–834.

Hanhua Feng has been a PhD student of Computer Science at Columbia University since 2002. He received a BSE degree from Shanghai Jiaotong University and an MS from Columbia University. His research interests are in modeling and performance analysis of network systems, queueing theory, and algorithms.

Vishal Misra has been an Assistant Professor of Computer Science and Electrical Engineering at Columbia University since 2001. He received a BTech degree from IIT Bombay, and an MS and PhD from the University of Massachusetts, Amherst, all in Electrical Engineering. His interests lie in the modeling, analysis and design of various aspects of communication networks, with particular emphasis on network traffic and congestion control mechanisms. He is currently also interested in the robustness of network architectures and protocols. He has received an NSF Career Award, an IBM Faculty Award and a DoE Career Award.

Dan Rubenstein is an Associate Professor of Electrical Engineering and Computer Science at Columbia University. He received a BS degree in mathematics from MIT, an MA in math from UCLA, and a PhD in computer science from University of Massachusetts, Amherst. His research interests are in network technologies, applications, and performance analysis, with a recent emphasis on resilient and secure networking, distributed communication algorithms, and overlay technologies. He has received an NSF Career Award, the Best Student Paper Award from the ACM SIGMETRICS 2000 Conference, and a Best Paper Award from the IEEE ICNP 2003 Conference.