Optimal State-Free, Size-aware Dispatching for Heterogeneous $M/G/$-type systems

Hanhua Feng, Vishal Misra, Dan Rubenstein
Columbia University

April 2005

Abstract

We consider a cluster of heterogeneous servers, modeled as $M/G/1$ queues with different processing speeds. The scheduling policies for these servers can be either processor-sharing or first-come first-serve. Furthermore, 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. We explore some of the practical challenges of identifying the optimal strategy, and also study a related problem that uses our model of how to provision server processing speeds to minimize waiting and completion time given a job size distribution and fixed aggregate processing power.

Keywords - Scheduling, Queueing systems, Load balancing, M/G/1, Processor sharing, First-come first-serve, Parallel 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 Figure 1. In a dispatch system, jobs arrive at a *dispatcher*, which must immediately decide the server to which 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 the 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 (e.g., in [22]). *Size-aware* means that the dispatcher uses the size of the arriving job and the overall job-size distribution as input for its choice of 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 content. Many systems [4] dispatch a request to a server by hashing on the URL of the request, sending identical requests to be served by the same server, ignoring instantaneous loads on the servers. If the response to a particular request is a constant size that is known in advance by the serving system, it becomes possible to design a hashing strategy that reduces the expecting serving time of the entire system. Furthermore, since over time, new server components are brought in and old components slowly phased out, a heterogeneous mix of servers is likely to exist.

- *Load-balanced routing*. Load-balancing routing allows a router to assign an arriving packet to any of several outgoing links. A controlling "pushback" mechanism that informs upstream routers of current downstream load, while useful, is difficult to implement efficiently in practice. Hence, a local, static dispatching strategy that keeps expected delays small is very useful for such systems.

- *Distributed database systems*. Database transactions are often very short, lasting less than a second. In contrast, the feedback on processing loads from the servers is returned at a low period, making it difficult to provide load information to the dispatcher.

In general, state-free, size-aware dispatching strategies are useful wherever it is not feasible to track the exact state of the various servers' queues. Even if feedback from the servers themselves is possible, unless this information is received instantaneously, it is of little use [16].

The paper proves several important results about the optimality of a variety of strategies that attempt to minimize job waiting and completion times. In particular, we compare *random* strategies, where the dispatcher randomly chooses a server to handle a given job, to *deterministic* strategies, where the job's size is used to deterministically selects the server. An important class of deterministic strategies we consider are *interval-based* strategies, where each server is assigned all jobs whose sizes fall within a distinct, continuous interval of sizes. Our main results are:

1. When the servers employ processor sharing, a strategy that assigns jobs randomly to servers is optimal, i.e., there is no benefit in utilizing job size information (Proposition 3.1).

2. We prove that when the dispatch system is comprised of a set of homogeneous FCFS servers, then there is in fact an interval-based strategy that is optimal (Theorem 3.3).

3. We show that when the system is comprised of a set of heterogeneous FCFS servers, then there need not exist an interval-based strategy that is optimal (Example 3.4).

4. We prove that 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 strategy for a system comprised of a set of heterogeneous servers (Theorem 3.5).

5. When the system is FCFS, we explore how the distribution of job size affects how jobs should be partitioned to servers. A high-level summary of our results is that if aggregate capacity is fixed, then as job-size distributions become more heavy-tailed, splitting off a small portion of the aggregate server capacity into a separate queue can yield significant performance benefits.

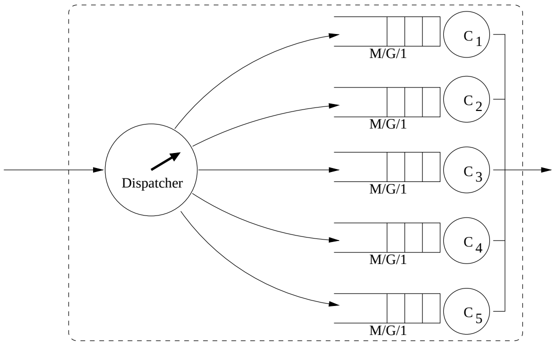
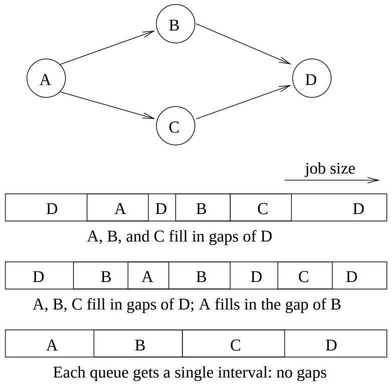


Figure 1: A dispatcher assigns jobs to multiple queues with different processing speeds.



A, B, and C fill in gaps of D

A, B, C fill in gaps of D; A fills in the gap of B

Each queue gets a single interval: no gaps

Figure 2: Three examples of size interval allocations given by an NSI strategy restricted by a set of four relations shown in the graph: $A \prec B$, $A \prec C$, $B \prec D$, and $C \prec D$. The relation $A \prec D$ is also implied due to transitivity but is omitted in the graph.

In addition to the above results that prove or disprove the existence of an interval-based strategy, we also focus on identifying the best or optimal strategy within the class of interval-based strategies proposed by [10]. For homogeneous systems, since all servers are identical, we must simply determine the boundary points of these intervals. However, 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, through experimentation using a variety of traditional job size distributions such as the Bounded Pareto, Log-normal and Weibull, we show that an optimal strategy depends heavily on the job size distribution.

1.1 Related work

Systems with parallel servers have been studied for decades. 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. [6]. On the other hand, there are only approximations of the mean response time for homogeneous $M/G/s$ systems [8]. With separate queues and exponential service times, Winston [23] shows an optimal dynamic strategy. Ephremides et al. [7] analyze two dynamic strategies.

Exponential service times are often assumed in the studying of the heterogeneous servers. With shared queue and exponential service times, Agrawala et al. [1] give an optimal strategy using thresholds for system without arrivals. This strategy is shown to be optimal for two queues with Poisson arrivals by Lin and Kumar [14], for multiple queues with a small arrival rate by Rosberg and Makowski [21], and finally for arbitrary arrival rate by Luh and Viniotis [15]. With separate heterogeneous queues and exponential service times, Chow and Kohler [5] compare three dynamic strategies (strategies that know the current states of the queues) with the random static strategy. With the random static strategy, Buzen and Chen [3] give the optimal load partitioning for general services. Ni and Hwang [18] analyze the optimal load partitioning for multiple classes of exponential service times. Li [13] considers the case that services on different queues have different coefficients of variation. Hajek [9] considers two heterogeneous queues that the jobs departing from one queue may be sent to the other with a certain probability. Tantawi and Towsley [22] study a static decentralized probabilistic strategy that each queue can transfer some jobs to other queues with a communication delay.

For homogeneous queues, if the job sizes are available to the strategy, the dynamic strategy that assigns jobs to the shortest queue is suggested optimal for exponential service times [17]. The size-interval strategy is proposed and studied by Harchol-Balter [10]. This strategy is later extended to the TAGS strategy [11]. Using TAGS, long jobs are first assigned to queues designated to shortest jobs. If the time quota of a job is used up, it gets restarted at a queue for longer jobs. These two strategies are shown to perform well for heavy-tailed distributions. 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 [20] and two-queue heterogeneous [19] 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 time for three different classes of job-size distributions, and study the best mapping of size intervals and the capacity planning problem. 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 *separated* queues, we first introduce the notation we will use throughout the remainder of the paper.

2.1 Notation

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 *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., $\rho < c$. The load, similar to the job size, is an invariant attribute of the arrival process, independent of any particular queue. For a particular queue, the *utilization rate* is defined as ρ/c. Clearly the load of a queue is the same as the utilization rate if the server has unit capacity, and therefore they are usually used interchangeably in literature, but must be distinguished for heterogeneous queueing systems.

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 to be the *zeroth-order load*.

2.2 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 i-th 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 also assume the arrivals of jobs are Poisson and job sizes are independent, identically-distributed (IID). Since independent mixes or Bernoulli selections from Poisson arrivals is still Poisson, each queue's arrival process remains Poisson, although the job size distribution for each queue can differ from the distribution of the aggregate queueing system. In other words, each of the queues is still modeled as an $M/G/1$ queue.

We denote by the random variable X_i the size of a job assigned to the i-th 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 i-th 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 on the i-th queue by $\hat{X}_i$ which equals to X_i/c_i. Also, we let random variable $\hat{X}$ be the overall service time on 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 i-th queue. Finally we denote the second-order load of the i-th queue by $\omega_i = \lambda E[X_i^2]$. Clearly we have $\rho = \sum_{i=1}^n \rho_i$ and $\omega = \sum_{i=1}^n \omega_i$. The overall mean waiting time and the overall 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 i-th queue, respectively.

2.3 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 i-th 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. In the 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_i, $i = 1, 2, \ldots, n-1$ these thresholds, and let $\xi_0 = 0$ and $\xi_n = \infty$. The i-th 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

$$\lambda_i = \lambda \left[F(\xi_{m(i)}) - F(\xi_{m(i)-1}) \right], \quad \rho_{m(i)} = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} x dF(x), \quad \omega_i = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} x^2 dF(x),$$

and

$$F_i(x) = \begin{cases} \frac{F(x)-F(\xi_{m(i)-1})}{F(\xi_{m(i)})-F(\xi_{m(i)-1})}, & \xi_{m(i)-1} \le x < \xi_{m(i)} \\ 0, & \text{otherwise} \end{cases} .$$

The SI strategy was proposed by Harchol-Balter, Crovella and Murta [10] (called as SITA in [10]). 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 2.1. *Suppose the minimum and maximum job sizes assigned to the i-th 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 i-th and j-th 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 j-th queue is nested in the range of the i-th queue)*

(ii) $\xi_j^{(0)} < \xi_i^{(0)} \le \xi_i^{(1)} < \xi_j^{(1)}$ *(the range of the i-th queue is nested in the range of the j-th 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 intervals assigned to one queue are smaller 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, the static strategy is an SI strategy, hence an SI strategy is a special case of the NSI strategies. We may say that the j-th queue takes the inner range in case **(i)** and the outer range in case **(ii)**. The following definition adds some restrictions on choosing NSI strategies regarding which of a pair of queues can take the inner range.

Definition 2.2. *We say the j-th queue* ***can be nested in*** *the i-th queue if cases* ***(i)*** *and* ***(iii)*** *in Definition 2.1 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 2.1 are the only allowable cases. If neither $j \prec i$ nor $i \prec j$ is defined for the NSI strategy, case* ***(iii)*** *in Definition 2.1 is the only allowable case.*

Figure 2 depicts three possible NSI strategy assignment of interval ranges of job sizes to queues. In the topmost assignment, job sizes 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.

3 Optimal Static Strategies

We optimize within the class of static strategies. In Section 3.1 we study the scenario that all queues use processor-sharing. In Section 3.2 we study the scenario that all queues use first-come first-serve. In Section 3.3 we show that, with proportional partitioning, the SI strategy is always better than the random strategy in a homogeneous system. This result is generalized in Section 3.4: the optimal optimal static strategy is always an SI strategy for homogeneous systems whereas it is not so for heterogeneous systems.

3.1 PS systems using the static strategy

We now consider the case that the scheduling policies for all queues are processor-sharing (PS). We shall show that no static strategy can perform better than the optimal random strategy. The closed forms of the steady-state mean response times for a single $M/G/1$ queue under these policies are well known for a queue with unit capacity. Scaled to a queue of capacity c, the mean response time of an $M/G/1$-PS queue for a job of size x is [12]

$$T(x)_{\text{PS}} = \frac{x/c}{1-\rho/c} = \frac{x}{c-\rho}, \tag{4}$$

and the overall mean response time is then

$$E[T_{\text{PS}}] = \frac{1}{\mu(c-\rho)}. \tag{5}$$

For each queue, applying (4) to (3) with c and ρ replaced by c_i and ρ_i, respectively, we get the mean response time of the entire system

$$E[T^{\text{S}}_{\text{PS}}] = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} \int_0^\infty \frac{x}{c_i - \rho_i} dF_i(x) = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{\rho_i}{c_i - \rho_i}. \tag{6}$$

We can see that the mean response time of a system with PS queues under static strategies depends only on ρ_i, We refer to the vector $[\rho_i]_{i=1}^n$ as *load partitioning.* Then the following proposition immediately follows.

Proposition 3.1. *The mean response time of a heterogeneous multi-queue PS system using an arbitrary* static *strategy is the same as that under a random strategy with the same load partitioning. In other words, with performance measured by the mean response time, there is no advantage for a* static *strategy to use size information in a heterogeneous PS system.*

The optimal load partitioning for a PS system is identical to an FCFS system where job sizes are exponentially distributed with the same mean, as we see in Section 3.2. Note that Proposition 3.1 also applies to the mean service time $E[\hat{X}]$; as seen in (1), the mean service time depends only on the load partitioning.

3.2 FCFS systems using the static strategy

We now derive formulae for the mean waiting time when the queues are FCFS (c.f. (11)). For a single $M/G/1$-FCFS queue, the mean waiting time is (scaled from the Pollaczek-Khintchine formula [6])

$$E\left[W_{\text{FCFS}}\right] = \frac{\lambda E[X^2]}{2c(c-\rho)} = \frac{\omega}{2c(c-\rho)}, \tag{7}$$

and the mean response time is

$$E[T_{\text{FCFS}}] = E\left[W_{\text{FCFS}}\right] + E\left[\frac{X}{c}\right] = \frac{\omega}{2c(c-\rho)} + \frac{1}{c\mu}, \tag{8}$$

where $\omega = \lambda E[X^2]$. Note that for exponential service times, since $E[X^2] = 2/\mu^2$ (hence $\omega = 2\rho/\mu$), we have

$$E[T_{\text{FCFS}}] = \frac{\rho}{c\mu(c-\rho)} + \frac{1}{c\mu} = \frac{1}{\mu(c-\rho)}, \tag{9}$$

which is identical to (5), the mean response time of a PS queue for any job size distribution with the same expectation.

Unlike the PS policy, the mean response time under the FCFS policy can be significantly improved if size information is used. From (7) and (8), the mean response time for a heterogeneous FCFS system using a static strategy is

$$E[T^{\mathrm{S}}_{\mathrm{FCFS}}] = E[W^{\mathrm{S}}_{\mathrm{FCFS}}] + E[\hat{X}] \tag{10}$$

where

$$E[W^{\mathrm{S}}_{\mathrm{FCFS}}] = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} \left[\frac{\lambda_i E[X_i^2]}{2c_i\,(c_i - \rho_i)} \right] = \frac{1}{2\lambda} \sum_{i=1}^{n} \left[\frac{\lambda_i \omega_i}{c_i\,(c_i - \rho_i)} \right] \tag{11}$$

is the mean waiting time. In fact, the quantity $(\lambda_i \omega_i)/[2c_i(c_i - \rho_i)] = \lambda_i E[W_i]$ is the average number of jobs in the waiting list of the i-th queue, by Little's Law [6]. Then the right-hand side of (11) is equal to $\left(\sum_{i=1}^{n} \lambda_i E[W_i]\right)/(2\lambda)$. So, minimizing $E[W^{\mathrm{S}}_{\mathrm{FCFS}}]$ is actually equivalent to minimizing the total number of jobs in the waiting lists of all queues. This is not surprising, since this is an immediate result of applying Little's law to the waiting list of the entire system.

Now we consider the special case that the random strategy is used. Using the random strategy we have $\lambda_i = \lambda\rho_i/\rho$ and $\omega_i = \omega\rho_i/\rho$. From (11), the mean waiting time is

$$E[W^{\mathrm{R}}_{\mathrm{FCFS}}] = \frac{\omega}{2\rho^2} \sum_{i=1}^{n} \frac{\rho_i^2}{c_i(c_i - \rho_i)} = \frac{C_X^2 + 1}{2\lambda} \sum_{i=1}^{n} \frac{\rho_i^2}{c_i(c_i - \rho_i)}, \tag{12}$$

where $C_X^2 = E[X^2]/E^2[X] - 1 = \lambda\omega/\rho^2 - 1$ is the square of the coefficient of variation for the random variable X. Then the mean response time is

$$E[T^{\mathrm{R}}_{\mathrm{FCFS}}] \quad = \quad \frac{C_X^2 + 1}{2\lambda} \sum_{i=1}^{n} \frac{\rho_i^2}{c_i(c_i - \rho_i)} + \frac{1}{\lambda} \sum_{i=1}^{n} \frac{\rho_i}{c_i}. \tag{13}$$

For the random strategy, Buzen and Chen [3] gave the optimal load partitioning:

$$(\rho_i^*)^{\mathrm{R}}_{\mathrm{FCFS}} = c_i - c_i \sqrt{\frac{C_X^2 + 1}{2\beta c_i + C_X^2 - 1}}, \tag{14}$$

where β is a parameter satisfying the total load constraint that $\rho = \sum_{i=1}^{n} \rho_i^*$.

Later, Ni and Hwang [18] give the solution that $\sqrt{\beta} = \left(\sum_{i=1}^{n} \sqrt{c_i}\right)/(1 - \rho)$ for the special case of *exponential* service times (so it also applies for all distributions where $C_X = 1$). In this case, the optimal load partitioning is

$$(\rho_i^*)^{\mathrm{R}}_{\mathrm{FCFS/exp}} = c_i - \frac{\sqrt{c_i}}{\sum_{i=1}^{n} \sqrt{c_i}}(1 - \rho) = (\rho_i^*)^{\mathrm{S}}_{\mathrm{PS}}\,. \tag{15}$$

Note that, since (5) and (9) are same, $E[T^{\mathrm{S}}_{\mathrm{PS}}]$ is also the same as the mean response time for a heterogeneous FCFS system using the random strategy, $E[T^{\mathrm{R}}_{\mathrm{FCFS}}]$, with *exponential* job sizes, as mentioned in Section 3.1.

By now we have formularized the mean waiting time for the heterogeneous FCFS systems under static strategies. The following comparison and proofs of optimality are based on these formulas.

3.3 Two strategies with proportional partitioning

Let us now compare the random strategy with the SI strategy under proportional partitioning for the homogeneous queues. We shall show that the SI strategy achieves lower mean waiting and response times for all job size distributions. Earlier, we define *load partitioning* to be the vector $[\rho_i]_{i=1}^{n}$; the proportional load partitioning, or simply *proportional partitioning*, is a particular load partitioning such that the amount of load assigned to each queue is proportional to its capacity, i.e., $\rho_i =$

$c_i\rho$. Proportional partitioning yields more specific static strategies: e.g., random strategy with proportional partitioning (**RP strategy)**, and size-interval strategy with proportional partitioning (**SIP strategy**).

It can be seen from (12) that, if the load partitioning $[\rho_i]_{i=0}^n$ is determined, the mean waiting time under random strategies can be determined. For a homogeneous system, since the capacities of all queues are the same, the mapping $m(i)$ is irrelevant to the mean waiting time. Without loss of generality, we can just set $m(i) = i$, and then λ_i and ω_i can be determined by the vector $[\rho_i]_{i=1}^n$. Therefore, we can see from (11) that the mean waiting time of the SI strategy is also determined. In short, for homogeneous systems, the load partitioning uniquely determines the mean waiting times under both random and SI strategies.

Consider a homogeneous system using the SI strategy. The following lemma, which is used to prove our main results, states that, under the SI strategy, the queue that receives shorter jobs has more zeroth-order load (i.e., arrival rate) and less second-order load, if queues are given the same first-order load.

Lemma 3.2. *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 $F_j(\xi) = 0$. Since

$$\frac{\lambda_i}{\rho_i} = \frac{\lambda \int_0^\xi dF_i(x)}{\lambda \int_0^\xi x dF_i(x)} \ge \frac{\int_0^\xi dF_i(x)}{\xi \int_0^\xi dF_i(x)} = \frac{1}{\xi}, \text{ and } \frac{\lambda_j}{\rho_j} = \frac{\lambda \int_\xi^\infty dF_i(x)}{\lambda \int_\xi^\infty x dF_i(x)} \le \frac{\int_\xi^\infty dF_i(x)}{\xi \int_\xi^\infty dF_i(x)} = \frac{1}{\xi}, \tag{16}$$

we have $\lambda_i/\rho_i \ge \lambda_j/\rho_j$. Similarly,

$$\frac{\omega_i}{\rho_i} \le \frac{\xi \int_0^\xi x dF_i(x)}{\int_0^\xi x dF_i(x)} = \xi = \frac{\xi \int_\xi^\infty x dF_i(x)}{\int_\xi^\infty x dF_i(x)} \le \frac{\omega_j}{\rho_j}. \tag{17}$$

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 (16) strictly holds. Same for (17). □

Let us first study a simple scenario: comparing the random strategy and the SI strategy under the *proportional partitioning* (i.e., $\rho_i = \rho/n$) for *homogeneous* (i.e., $c_i = 1/n$) queues: that is, RP strategy versus SIP strategy. From (12), we get the mean waiting time of the RP strategy

$$E[W_{\text{FCFS}}^{\text{RP}}] = \frac{n\omega}{2(1-\rho)}.$$

From (11), we get the mean waiting time of the SIP-strategy

$$E[W_{\text{FCFS}}^{\text{SIP}}] = \frac{n^2}{2\lambda(1-\rho)} \sum_{i=1}^n \lambda_i \omega_i = E[W_{\text{FCFS}}^{\text{RP}}] \left(\frac{n}{\lambda\omega} \sum_{i=1}^n \lambda_i \omega_i \right), \tag{18}$$

where $\lambda_i = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} dF(x)$ and $\omega_i = \lambda \int_{\xi_{m(i)-1}}^{\xi_{m(i)}} x^2 dF(x)$ are respectively the arrival rate and the second-order load for the i-th queue. Without loss of generality, we assume $m(i) = i$ for homogeneous queues. (Note that we cannot have this assumption for heterogeneous cases; the mapping of queues matters – we come back to this in a later section.) Since $\rho_i = \rho_{i+1} = 1/n$ (proportional partitioning), by Lemma 3.2, we get $\lambda_i \ge \lambda_{i+1}$ and $\omega_i \le \omega_{i+1}$ for all $i = 1, \dots, n-1$. So, λ_i is a decreasing series and ω_i is an increasing series. By applying the discretized version of Chebyshev Integral

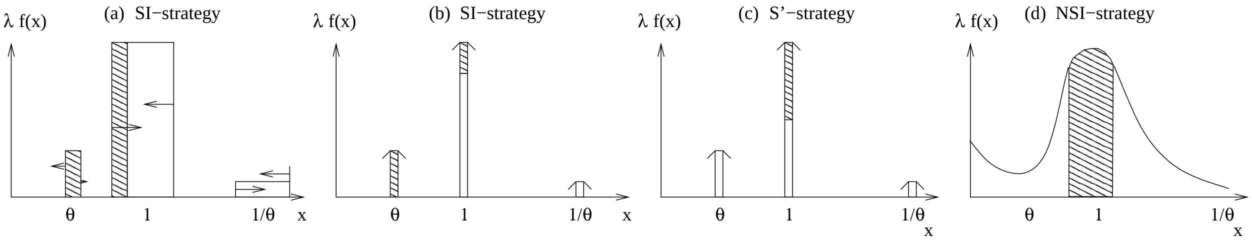


Figure 3: (a) Illustration of a continuous job-size distribution containing three uniform parts at around θ, 1, and $1/\theta$. The SI strategy with ascending mapping assigns all jobs of size around θ and portion of jobs of size around 1 to the slower queue. (b) As the width of each uniform part goes to zero, the distribution approximates a discrete distribution, and the SI strategy probabilistically assigns jobs of size 1. (c) The alternate static strategy that assigns only portion of jobs of size 1 to the slower queue. This static strategy is indeed an NSI strategy for a discrete job size distribution. (d) Illustration of an NSI strategy that optimizes the mean waiting time for this particular continuous distribution. The jobs of shaded area is assigned to the slower queue.

Inequality (see [2] for details), we have

$$\frac{n}{\lambda\omega}\sum_{i=1}^{n}\lambda_i\omega_i \le \frac{1}{\lambda\omega}\left(\sum_{i=1}^{n}\lambda_i\right)\left(\sum_{i=1}^{n}\omega_i\right) = 1.$$

Therefore, in a homogeneous system, the SIP-strategy always attains a smaller mean waiting time than the RP-strategy for any job-size distribution. It also applies to the mean response time since the mean service time in a homogeneous system is constant.

3.4 Optimal static strategies for homogeneous FCFS systems

The simple deduction above shows, with proportional partitioning, the SI strategy is always doing a better job than the random strategy in a homogeneous system, disregarding the job size distribution. Could the SI strategy generally beat other strategies? We shall see that the answer is always yes for homogeneous systems but not always yes for heterogeneous systems, respectively implied by the Theorem 3.3 and Example 3.4.

Theorem 3.3. *The optimal* static *strategies for a homogeneous FCFS system is an SI strategy, with respect to both the mean waiting time and the mean response time.*

We delay the proof of Theorem 3.3 to Section 5. Harchol-Balter et al. [10] provides 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 3.3 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 3.4. *Consider the scenario that jobs that have only three different possible sizes: $x_1 = \theta$, $x_2 = 1$ and $x_3 = 1/\theta$, where $0 < \theta < 1$, such that 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. Then, the arrival rates of the three kinds of jobs are respectively $\frac{\rho_\epsilon}{\theta}$, $\rho - 2\rho_\epsilon$, and $\theta\rho_\epsilon$, whereas the second-order loads of the three kinds of jobs are respectively $\theta\rho_\epsilon$, $\rho - 2\rho_\epsilon$ and $\frac{\rho_\epsilon}{\theta}$. The total arrival rate and second-order* *load are the same in quantity:* $\lambda = \omega = \rho + \left(\frac{1}{\theta} + \theta - 2\right)\rho_\epsilon$*. The mean job size is* $E[X] = \lambda/\rho = 1 + \left(\frac{1}{\theta} + \theta - 2\right)\frac{\rho_\epsilon}{\rho}$*, a little more than* 1*.*

We have not defined SI strategy for discrete distributions. However, we can approximate it by a series of continuous distributions, as illustrated in Figure 3(a). The limit of the series is illustrated in Figure 3(b) – we consider this limit strategy is still an SI strategy. (See also the remark at the end of Section 3.4.)

Suppose we have $\rho_\epsilon < \rho - c_2 < \rho_1 < c_1 < 1 - 2\rho_\epsilon$*, where* ρ_1 *and* c_1 *corresponds to the slower queue. There are two cases (mappings) for the SI strategy: the first (slower) queue gets either (i) all jobs of size* θ *or (ii) all jobs of size* $1/\theta$*. In either case, a fixed portion of job of size* 1 *is also assigned to this queue. Assuming the case (i), i.e., short jobs are assigned to the first queue. Then we have*

$$\lambda_1 = \frac{\rho_\epsilon}{\theta} + (\rho_1 - \rho_\epsilon), \qquad \omega_1 = \theta\rho_\epsilon + (\rho_1 - \rho_\epsilon),$$
$$\lambda_2 = \theta\rho_\epsilon + (\rho_2 - \rho_\epsilon), \qquad \omega_2 = \frac{\rho_\epsilon}{\theta} + (\rho_2 - \rho_\epsilon).$$

If we assume the case (ii), the difference is that λ_i *and* ω_i *is swapped in quantity for* $i = 1, 2$*. Hence the two mappings of the SI strategies result in the same mean waiting time.*

Consider an alternative static strategy that assigns only jobs of size 1 *to the first (slower) queue (This is in fact an NSI strategy but not an SI strategy). Then we have*

$$\lambda_1 = \omega_1 = \rho_1, \qquad \lambda_2 = \omega_2 = \rho_2 + \left(\frac{1}{\theta} + \theta - 2\right)\rho_\epsilon.$$

Figure 3(b) illustrates the SI strategy and Figure 3(c) illustrates the alternative NSI strategy.

Let $\theta = 0.5$*,* $\rho_\epsilon = 0.05$*,* $\rho = 0.9$*. The following table compares the optimal* $E[W]$*'s for the SI strategy, the alternate NSI strategy and the random strategy as in (12). Values in parentheses is the optimal load partitioning, namely* ρ_1^**. Note that we assume* $c_1 + c_2 = 1$ *and* $\rho_2^* + \rho_1^* = \rho$*.*

c_1	$E[W^{\text{SI}}_{\text{FCFS}}]^*$	$E[W^{\text{NSI}}_{\text{FCFS}}]^*$	$E[W^{\text{R}}_{\text{FCFS}}]^*$
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)*

As we can see in the table, the alternative NSI strategy is better if c_1 *is small (for* $c_1 = 0.2$ *and* $c_1 = 0.3$*). This shows that sometimes the optimal static strategy will never be an SI strategy for heterogeneous systems.*

3.5 Optimal static strategy for heterogeneous FCFS systems

In Section 3.4, we give an example to show that the SI strategy sometimes cannot be optimal. By generalizing the SI strategy, we can find a set of static strategy that contains the optimal static strategy, as implied by the following theorem.

Theorem 3.5. *For a heterogeneous FCFS system, 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 (c.f. Definition 2.2).*

We delay the proof of Theorem 3.5 to Section 5.

Let us provide an intuitive explanation on 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 variability 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 queues, a slower queue has a stronger desire for less job variability than the faster queue, since the job variability has a stronger effect in deteriorating the performance on a slower queue than on a faster queue. So the slower queue gets some priority over a faster queue in choosing job sizes. As shown in Figure 3(b) and Figure 3(c), the alternative NSI strategy offers the slower queue a size variability of zero, whereas the SI strategy does not. However the slow queue does not always have a higher priority; in Example 3.4, the SI strategy is still better for $c_1 = 0.4 < c_2 = 0.6$. As the difference between capacities of two queues increases, the desire for a small job variability from the slower queue gets stronger. Therefore, the optimal static strategy is a strategy that discriminates against the faster queue by assigning a better interval to the slower queue, as shown Figure 3(d).

Under an NSI strategy, if the size of a job is known upon arrival, the queue, to which the job is assigned, is fixed. Therefore the NSI strategy is deterministic. Generally, a deterministic strategy can be considered as a special case of stochastic strategies, where a queue is chosen for a job with probability one.

Remarks on deterministic strategies for discrete job-size distributions. So far, we assume the job size distributions are continuous, and argue that a discrete distribution can be approximated by a series of continuous distributions. However, the word *deterministic* needs a refinement for discrete job-size distributions or distributions containing both discrete and continuous portions. For these distributions, the strategies must still satisfy certain load partitioning requirement. Therefore a deterministic strategy may assign jobs of those discrete sizes (where the CDF has a jump) into different queues probabilistically. Here we abuse the word deterministic a little (we may instead say that the deterministic strategy is actually deterministic for *almost every* size); this generalization is necessary to extend the analysis to multiple $M/D/1$ queues. For SI strategies, if $\Pr\{X = \xi_i\} > 0$, i.e., the CDF has a jump on ξ_i, we may allocate part of jobs of size ξ_i to class i with a certain probability and the rest to class $i + 1$, in order to satisfy the requirement of load partitioning. If two or more thresholds coincide, the jobs whose size equals to these thresholds are split into three or more classes.

By now, the optimal static strategies for homogeneous and heterogeneous systems are identified but not proved – the proofs are in Section 5.

4 Mapping of size intervals and capacity planning

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 on the SI strategy 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 [10]. 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 and discuss the capacity planning problem.

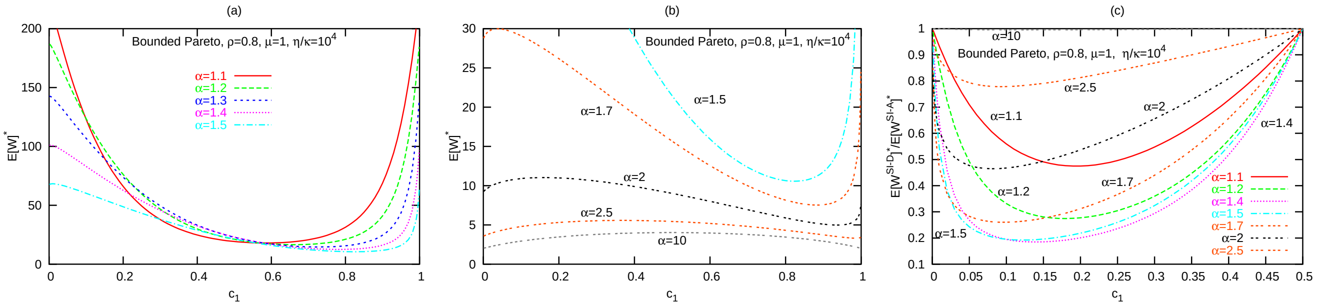


Figure 4: Two heterogeneous FCFS queues using SI strategies of two different mappings, with bounded-Pareto distributed job sizes. (a,b) The optimal mean waiting time as functions of the capacity of the first queue. (c) The ratio of optimal mean waiting times between the ascending and descending mappings,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

4.1 Three kinds of distributions

Three kinds of distributions are used in this section: bounded Pareto, log-normal and Weibull. Each of them has one parameter that controls its variability: from heavy-tailed distributions to approximately deterministic. Also, the coefficient of variation, C_X, is a monotonic function of the control parameter for each kind of the distributions.

Bounded Pareto distribution. A bounded-Pareto distribution has a power-law tail. The CDF is

$$F(x) = \frac{1}{1-\left(\frac{\eta}{\kappa}\right)^{-\alpha}}\left[1-\left(\frac{x}{\kappa}\right)^{-\alpha}\right] \quad \text{for } \kappa \le x \le \eta,$$

where κ is the lower bound of the random variable and η is the upper bound. The variability is controlled by the parameter $\alpha \in (0,\infty)$. We set the ratio η/κ to a fixed value (10^4 by default) and then choose a κ such that $E[X] = 1/\mu$ ($= 1$ by default). For $\alpha \in (0,2]$, the bounded Pareto is a classical heavy-tailed distribution [2]. As $\alpha \to \infty$, this distribution approaches a deterministic value $1/\mu$; this can be easily seen because, as $\alpha \to \infty$, we get $\kappa \to 1/\mu$ and the probability $\Pr[X > \kappa + \epsilon]$ vanishes quickly even for a small ϵ.

Log-normal distribution. A random variable X has a log-normal distribution if $\log X$ has a normal (Gaussian) distribution. Its CDF is $(1/2)\left[1 + \text{erf}\left((\ln x - m)/(s\sqrt{2})\right)\right]$, where m and s are the mean and deviation of the Gaussian $\log X$, respectively, and $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-\tau^2} d\tau$. The variability is controlled by parameter s, and parameter m is chosen such that $E[X] = 1/\mu$. As $s \to 0$, the Gaussian, $\log X$, is close to a deterministic value, so X is close to a deterministic value as well. As $s \to \infty$, a large variation is shown in both $\log X$ and X.

Weibull distribution. A random variable X has a Weibull distribution if X^α is exponential. Its CDF is $1 - \exp(-\beta x^\alpha)$, where $\beta = 1/E[X^\alpha]$ and $\alpha \ge 0$. The Weibull distribution degenerates to an exponential distribution when $\alpha = 1$. The variability is controlled by parameter α, and parameter β is chosen such that $E[X] = 1/\mu$. For $\alpha < 1$ the distribution has greater variability and a heavier tail than the exponential distribution, and as $\alpha \to \infty$ it is close to a deterministic value.

4.2 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

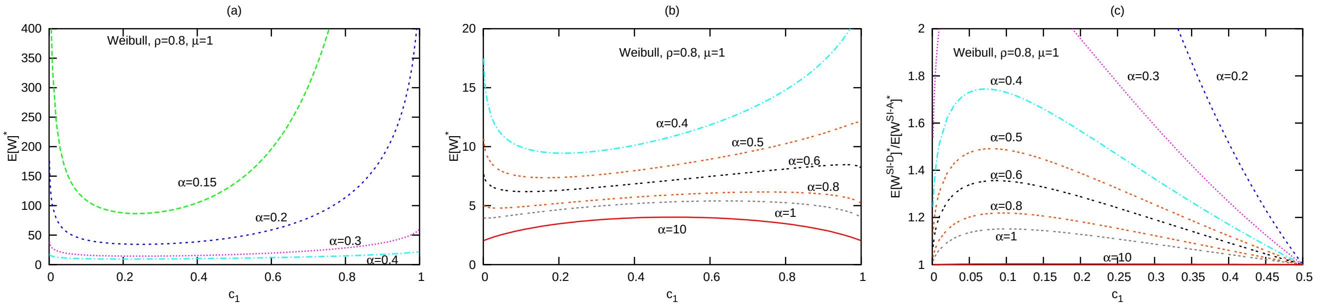


Figure 5: Two heterogeneous FCFS queues using SI strategy, with Weibull distributed job sizes. (a,b)The optimal mean waiting time as functions of the capacity of the first queue. (c) The ratio of optimal mean waiting times between the ascending and descending mappings, as functions of the capacity of the slower queue.

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, the ascending mapping means $m(i) = i$ and the descending mapping means $m(i) = n - i + 1$, where $i = 1, 2 \ldots, m$. In a system of two queues, the ascending and descending mappings are the only two mappings. We let $E[X] = 1/\mu = 1$, without loss of generality.

We consider only the SI strategy in this section. The numerical results in Figures 4, 5 and 6 show how the optimal mean waiting time $E[W^{\text{SI}}_{\text{FCFS}}]^*$ changes as the capacity of the first queue changes, for a two-queue system at $\rho = 0.8$. In Figures 4(a,b), 5(a,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 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 make this difference clear, in Figures 4(c) and 5(c) we plot the ratio of the optimal mean waiting time under the descending mapping, $E[W^{\text{SI-D}}_{\text{FCFS}}]^*$, to that under the ascending mapping, $E[W^{\text{SI-A}}_{\text{FCFS}}]^*$, as functions of the capacity of the slower queue (so the range of the X-axis is $[0, 0.5]$). Figures 4, 5 and 6 are for bounded Pareto job-size distributions, the Weibull job-size distributions, and the 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. Note that ascending mapping is also better 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.

The differences of the optimal $E[W^{\text{SI}}_{\text{FCFS}}]$'s (with respect to different c_1's) become significant if the variation of the distribution become larger, in particular for bounded Pareto distributions. So in reality if we have a heterogeneous FCFS system under the SI strategy, the mapping of size intervals should be taken into account. For example, in supermarkets, there are often some express check-out lines with different thresholds; this is an actual application of the SI strategy. Meanwhile, this system is heterogeneous – senior cashiers and new employees have different processing speeds. The decision of whether or not to put a senior cashier at an express line, depends on the distribution of service times for a customer (by an average cashier).

Figure 7 shows the best mapping of size intervals for the system with three heterogeneous queues and exponential services, at $\rho = 0.8$. In Figure 7, the X-dimension is the capacity of the slowest queue, with a range of $(0, 0.33)$, and the Y-dimension is the capacity of the second to the slowest queue, with a range of $(0, 0.5)$. So the coordinates of each

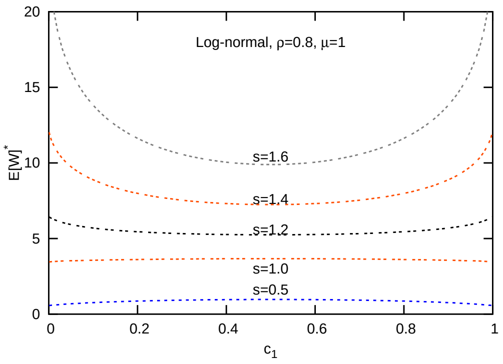


Figure 6: The optimal mean waiting time as functions of the capacity of the first queue, for two heterogeneous FCFS queues using SI-strategy. The job size distribution is log-normal.

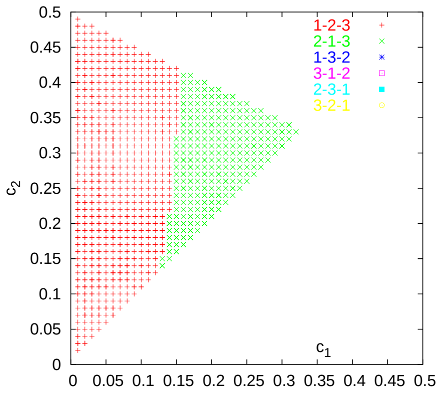


Figure 7: The best mapping for different capacity combinations for three heterogeneous FCFS queues using SI-strategy, with exponentially distributed job sizes. Note that 1-2-3 means the ascending mapping and 3-2-1 means the descending mapping.

point indicates 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 Figure 7 we notice that, for exponential distributions, the best mapping depends on the quantity of the capacity of the slowest queue. In fact, all these job size distributions we studied here are somewhat “regular”; we believe the problem of finding the optimal mapping for a general distribution is a hard problem, both analytically and computationally.

4.3 Mapping-invariant distributions

It can be observed in Figure 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 4.1. *For a heterogeneous FCFS system with two queues, if the partial load $\rho(x) = \lambda \int_0^x t dF(t)$ satisfies $\rho(x) = \rho - \rho(\psi/x)$ for some positive ψ, then $E[W_{\text{FCFS}}^{\text{SI}}]$ 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 $\rho(x) = \lambda \int_0^x tdF(t)$), we get $f(x) = \left(\psi^2/x^4\right) f\left(\psi/x\right)$, where $f(x) = dF(x)/dx$ is the probability density function (PDF). Then we have

$$\lambda_1 = \lambda \int_0^{\xi} \frac{\psi^2}{x^4} f\left(\frac{\psi}{x}\right) dx = -\lambda \int_0^{\xi} \frac{\psi}{x^2} f\left(\frac{\psi}{x}\right) d\left(\frac{\psi}{x}\right) = \frac{\lambda}{\psi} \int_{\xi'}^{\infty} y^2 f(y) dy = \frac{\omega_2'}{\psi}.$$

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

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

Hence the distribution is mapping-invariant in two-queue systems. □

The log-normal distribution and the job-size distribution in Example 3.4 satisfy $\rho(x) = \rho - \rho(\psi/x)$ and hence is mapping-invariant in two-queue systems.

4.4 Capacity planning

Given a total capacity of queues, the capacity planning problem is to find the best way to allocate an amount of capacity to each queue. First let us consider the case that the random strategy is used. It can be derived from (14) that proportional partitioning is the optimal load partitioning under random strategy for homogeneous systems. From (12) and (13) it is easy to see that the mean waiting time or the mean response time of a homogeneous system ($c_i = 1/n$) is n times worse than those of a single queue of unit capacity, as in (7) and (8), respectively. Then what if the capacity is divided unevenly? Let us assume that the coefficient of variation $C_X = 1$ so that $E[T_{\text{FCFS}}^{\text{R}}]^* = E[T_{\text{PS}}^{\text{S}}]^*$. Letting $n = 2$ and plugging $\rho_1 = (\rho_1^*)_{\text{PS}}^{\text{R}}$ from (15) to (6), we finally obtain

$$E[T_{\text{PS}}^{\text{S}}]^* = \frac{2}{\mu(1-\rho)} - \frac{1 - 2\sqrt{c_1(1-c_1)}}{\lambda(1-\rho)}. \tag{19}$$

Since the function $\sqrt{x(1-x)}$ is a concave function, $E[T_{\text{PS}}^{\text{S}}]$ is also a concave function of c_1. It is maximized at $c_1 = 0.5$ and minimized at $c_1 = 0$ or $c_1 = 1$. So, if we are going to split a unit capacity into two PS queues, the best way is to allocate all the capacity of one of them, and the worst way is to allocate the capacity homogeneously to each queue. Clearly it can be extended to n queues. Our numerical results show that, this claim is generally true for an FCFS system using the random strategy, even if $C_X \neq 1$.

Similar results hold for all static strategies if jobs sizes are close to deterministic: since $\rho_i \approx \lambda_i/\mu$ and $\omega_i \approx \lambda_i/\mu^2$, we get $\lambda_i\omega_i \approx \rho_i^2$, hence (11) is close to (12).

Generally for static strategies it is no longer true that allocating all capacity to a single queue is the best solution. With the SI strategy, if jobs have large variability, it is wise to use more queues. We can see from Figures 4(a) and 5(a) that, for bounded Pareto and Weibull distributions with a small α, the mean waiting time for two heterogeneous queues with any capacity combination is smaller than that for a single queue (shown in the figure at $c_1 = 0$ or $c_1 = 1$). In other words, with large job-size variability, the performance is improved by splitting a small amount of capacity to an independent queue. In this case, the mapping of size intervals is very important; for example, as shown in 4(b), for the bounded Pareto distribution with $\alpha = 1.7$, the mean waiting time improves three times as much even if we just split out 5% of the capacity. This very slow queue must service longest jobs, however, otherwise we get almost no improvement.

For bounded-Pareto and Weibull job-size distributions, the homogeneous capacity allocation ($c_1 = c_2 = 0.5$) is usually not the optimal. As seen in 4(a) and 5(a), with the best heterogeneous capacity allocation, we can improve the mean waiting time by 10-50% over the homogeneous allocation.

In this section we have discussed the capacity planning problem assuming that the expense of a server is a linear function of the server capacity. In other words, in the above capacity planning problem we assume $\sum_{i=1}^{n} c_i = 1$. In reality, this is usually not accurate. The price of a server is usually a non-linear function, say $\$(\cdot)$, of the server capacity. In future work, we may use an alternative constraint, e.g., $\sum_{i=1}^{n} \$(c_i) \leq \$_{\text{TOTAL}}$.

5 Proofs on the optimal static strategies

We show the main results in Section 3 but delay the proofs to this section. In this section, we prove theorems 3.3 and 3.5 and show a proposition that helps seeking 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 (11) the mean waiting time of an a static strategy is $E[W_{\text{FCFS}}^{\text{s}}] = \left[\sum_{i=1}^{n} \alpha_i \lambda_i \omega_i\right]/(2\lambda)$. Job size distributions are assumed to be continuous. For discrete distributions, we can always argue by continuous approximations.

For systems with two queues, consider two actions to improve the mean waiting time: transfer 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 $E[W_{\text{FCFS}}^{\text{s}}]$ in (11) with respect to λ, ρ, and ω, i.e.,

$$\Delta\left(E[W_{\text{FCFS}}^{\text{s}}]\right) = \frac{1}{2\lambda}\left[\alpha_1\lambda_1 - \alpha_2\lambda_2\right]\Delta\omega + \frac{1}{2\lambda}\left[\alpha_1\omega_1 - \alpha_2\omega_2\right]\Delta\lambda + \frac{1}{2\lambda}\left[c_1\alpha_1^2\lambda_1\omega_1 - c_2\alpha_2^2\lambda_2\omega_2\right]\Delta\rho. \tag{20}$$

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 i-th queue to the other, and let λ_i^r, ρ_i^r, and ω_i^r be the corresponding quantities that *remain* in the i-th queue. Then we have

$$\begin{aligned} 2\lambda E[W_{\text{FCFS}}^{\text{s}}] &= \alpha_1\lambda_1\omega_1 + \alpha_2\lambda_2\omega_2 = \alpha_1\left(\lambda_1^r + \lambda_1^s\right)\left(\omega_1^r + \omega_1^s\right) + \alpha_2\left(\lambda_2^r + \lambda_2^s\right)\left(\omega_2^r + \omega_2^s\right) \\ &= 2\lambda E[\tilde{W}_{\text{FCFS}}^{\text{s'}}] - (\alpha_1 + \alpha_2)\left[\lambda_1^s - \lambda_2^s\right]\left[\omega_1^s - \omega_2^s\right] + \left[\alpha_1\lambda_1 - \alpha_2\lambda_2\right]\left[\omega_1^s - \omega_2^s\right] + \left[\lambda_1^s - \lambda_2^s\right]\left[\alpha_1\omega_1 - \alpha_2\omega_2\right] \end{aligned} \tag{21}$$

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

Proof. (Theorem 3.3) For load swapping, we denote by ρ_1^s the load of the jobs above ξ in the first queue, for some ξ, and by ρ_2^s the load of jobs below ξ in the second queue, as illustrated by shaded areas in Figure 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 such a ξ such that $\rho_1^s = \rho_2^s$. Assuming $\rho_1^s = \rho_2^s$, by Lemma 3.2, we get $\lambda_1^s \leq \lambda_2^s$ and $\omega_1^s \geq \omega_2^s$.

Then we consider four conditions:

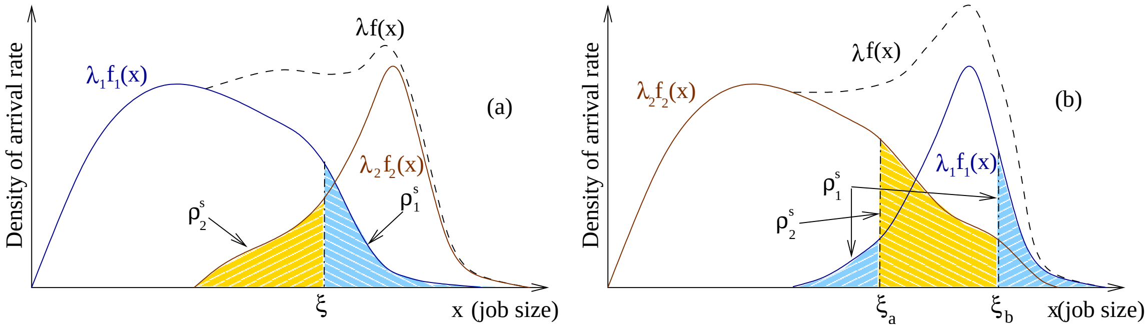


Figure 8: The density function of arrivals $\lambda f(x)$ (dashed curve) is the sum of two functions, $\lambda_1 f_1(x)$ and $\lambda_2 f_2(x)$ (solid curves), assigned to two queues. (a) After swapping jobs with equal loads between two queues (differently shaded areas), an SI-strategy is obtained. (b)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

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 (21), we have

$$E[W^{\text{S}}_{\text{FCFS}}] \geq E[\tilde{W}^{\text{S'}}_{\text{FCFS}}] = E[\tilde{W}^{\text{SI}}_{\text{FCFS}}], \tag{22}$$

since $\lambda^s_1 \leq \lambda^s_2$ and $\omega^s_1 \geq \omega^s_2$. 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 (22) strictly holds, since we have both $\lambda^s_1 < \lambda^s_2$ and $\omega^s_1 > \omega^s_2$ by Lemma 3.2.

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 (20), 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 $\Delta\left(E[W^{\text{s}}_{\text{FCFS}}]\right) \leq 0$, 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 $a_1 b_1 \leq a_2 b_2$ 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 (20), 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, 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 Condition 3 or Condition 4 is satisfied, we can improve the mean waiting time by continuously transferring some jobs from one queue to the other until either Condition 1 or Condition 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 Figure 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[1]. Hence we have Theorem 3.3. □

Now we look at Theorem 3.5. We need to prove a series of lemmas as follows.

[1] To be rigorous, we show that the convergence is achieved in finite steps. Please see Appendix A for details.

Lemma 5.1. *Suppose function $g(t)$ is a monotonic positive function in interval $[a, b]$. For (a_1, b_1) such that $a \leq a_1 < b_1 \leq b$ and $b - a = 2(b_1 - a_1)$, suppose $g(t)$ satisfies $\int_{a_1}^{b_1} g(t)dt = \int_a^{a_1} g(t)dt + \int_{b_1}^{b} g(t)dt$. Then*

$$\int_{a_1}^{b_1} \frac{dt}{g(t)} \leq \int_a^{a_1} \frac{dt}{g(t)} + \int_{b_1}^{b} \frac{dt}{g(t)}.$$

Proof. The proof is an application of 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 $h(g(t)) \geq 1/g(t)$ for $t \in [a_1, b_1]$ and $h(g(t)) \leq 1/g(t)$ for $t \in [a, a_1] \cup [b_1, b]$. Then

$$\int_{a_1}^{b_1} \frac{dt}{g(t)} \leq \int_{a_1}^{b_1} h(g(t))dt = \alpha \left(\int_{a_1}^{b_1} g(t)dt \right) + \beta(b_1 - a_1)$$

because of the linearity of $h(\cdot)$. Similarly

$$\begin{aligned} &\int_a^{a_1} \frac{dt}{g(t)} + \int_{b_1}^{b} \frac{dt}{g(t)} \geq \int_a^{a_1} h(g(t))dt + \int_{b_1}^{b} h(g(t))dt \\ &= \alpha \left(\int_a^{a_1} g(t)dt + \int_{b_1}^{b} g(t)dt \right) + \beta(a_1 - a + b - b_1) = \alpha \left(\int_{a_1}^{b_1} g(t)dt \right) + \beta(b_1 - a_1). \end{aligned}$$

With two inequalities above, we complete the proof of Theorem 5.1. □

Lemma 5.2. *Suppose two thresholds, ξ_1 and ξ_2, $0 \leq \xi_1 < \xi_2$, 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 i-th interval. Then, $\lambda_2 \leq \lambda_1 + \lambda_3$ if $\omega_2 = \omega_1 + \omega_3$, whereas $\omega_2 \leq \omega_1 + \omega_3$ if $\lambda_2 = \lambda_1 + \lambda_3$, where λ_i and ω_i are the arrival rate and the second-order load of the i-th interval, respectively, for $i = 1, 2, 3$.*

Proof. Let $\rho(x) := \lambda \int_0^x t dF(t)$, 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 $i = 1, 2, 3$,

$$\lambda_i = \lambda \int_{\xi_{i-1}}^{\xi_i} dF(x) = \int_{\xi_{i-1}}^{\xi_i} \frac{1}{x} d\rho(x) = \int_{r_{i-1}}^{r_i} \frac{1}{x(r)} dr, \quad \text{and}$$

$$\omega_i = \lambda \int_{\xi_{i-1}}^{\xi_i} x^2 dF(x) = \int_{\xi_{i-1}}^{\xi_i} x d\rho(x) = \int_{r_{i-1}}^{r_i} x(r) dr,$$

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 5.1 we get $\lambda_2 \leq \lambda_1 + \lambda_3$. Supposing $\lambda_2 = \lambda_1 + \lambda_3$ and using Lemma 5.1 again with $g(\cdot) = 1/x(\cdot)$, we get $\omega_2 \leq \omega_1 + \omega_3$. □

Proof. (Theorem 3.5) Again, first we consider two queues. Without loss of generality, we assume $c_1 < c_2$. Hence the $c_1 \geq c_2$ part of Condition 4 in the proof of Theorem 3.3 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 Figure 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 5.2, we have $\omega_1^s \geq \omega_2^s$. Then, from (21) we get $E[W^{\mathrm{S}}_{\mathrm{FCFS}}] \geq E[W^{\mathrm{S'}}_{\mathrm{FCFS}}] = E[\tilde{W}^{\mathrm{NSI}}_{\mathrm{FCFS}}]$. 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 $\omega_1^s = \omega_2^s$. 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 3.3. At this time, $\lambda_1^s \leq \lambda_2^s$ due to Lemma 3.2. 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 $\lambda_1^s \geq \lambda_1^s$ once again because of Lemma 3.2. 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 3.3, 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[2]. Hence we have Theorem 3.5. □

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 i-th 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 Figure 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 Figure 8(b).

From the proof of Theorem 3.5, 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 without load transferring. In other words, we can merge case **(ii-a)** to case **(ii-b)** and then replace the case **(ii)** above with

(ii') One queue, say X, gets both a greater Λ and a 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 Figure 8(b).

[2] To be rigorous, we need to show that the convergence is achieved in finite steps. Please see Appendix A for details.

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: c.f. (1)).

Proposition 5.3. *For a heterogeneous FCFS system, if the load partitioning is fixed, the optimal static strategy (for both mean waiting time and mean response time) is an NSI strategy with a set of nesting relations. A relation X≺Y is added to this set if Λ and Ω of X are both greater than those of Y.*

The difference between Theorem 3.5 and Proposition 5.3 is that they uses different set of relations. In Theorem 3.5, a slower queue can be nested in a faster queue whereas, in Proposition 5.3, a queue with both a greater Λ and a greater Ω can be nested in the other. Moreover, Proposition 5.3 requires fixed load partitioning. The implication of Proposition 5.3 has two-fold. First, for fixed load partitioning, in particular the proportional partitioning that is fair to each queue, we can still find an optimal NSI strategy to minimize mean waiting and response times. Note that the propotional 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 3.5, we assume there is a nesting relation for each pair of queues of unequal capacities. However, by load transferring and Proposition 5.3, 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 3.4; 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. We showed that, if the scheduling policy of the queues is processor-sharing, the mean response time of the entire system is determined solely by the load partitioning, i.e., the amount of load received by each queue, and therefore size information of each job does not help improving the mean response time. For first-come first-serve 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. Finally we turn to the capacity planning problem and show that splitting capacity is important for systems with jobs whose sizes have large variability.

It remains to study the optimal dispatching strategy in the scenario that other scheduling policies are used by the queues, e.g., the shortest-job-first (SJF). In contrast to the FCFS policy, the SJF policy favors jobs with large variability. Hence, intuitively we conjecture that the optimal dispatching strategy is not deterministic, i.e., the size of a job does not determine which queue for the job to go, and some randomness mechanism should be used. We also conjecture that, in contrast to the FCFS queues, the random strategy has a good performance for SJF queues, much better than using the size-interval strategy.

References

[1] Ashok K. Agrawala, Edward G. Coffman, Michael R. Garey, and Satish K. Tripathi. A stochastic optimization algorithm minimizing expected flow times on uniform processors. *IEEE Transactions on Computers*, 33(4):351–356, Apr. 1984.

[2] Nikhil Bansal and Mor Harchol-Balter. Analysis of SRPT scheduling: Investigating unfairness. In *ACM SIGMETRICS*, pages 279–290, Jun. 1 2001. Cambridge, MA, USA.

[3] Jeffrey P. Buzen and Peter P.-S. Chen. Optimal load balancing in memory hierchies. *International Federation on Information Processing*, 74:271–275, 1974.

[4] J.S. Chase. Server switching: Yesterday and tomorrow. In *IEEE Workshop on Internet Applications (WIAPP)*, 2001.

[5] Yuan-Chieh Chow and Walter H. Kohler. Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Transactions on Computers*, 28(5):354–361, May 1979.

[6] Robert B. Cooper. *Introduction To Queueing Theory*. Elsevier North Holland, 2nd edition, 1981.

[7] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, 25(4):690–693, Aug. 1980.

[8] Marcos Escobar, Amedeo R. Odoni, and Emily Roth. Approximate solution for multi-server queueing systems with erlangian service times. *Computers and Operations Research*, 29(10):1353–1374, Sep. 2002.

[9] Bruce Hajek. Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control*, 29(6):491–499, Jun. 1984.

[10] Mor Harchol-Balter. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, 59:204–228, 1999.

[11] Mor Harchol-Balter. Task assignment with unknown duration. *Journal of the Association for Computing Machinary*, 49(2):260–288, Mar. 2002.

[12] Leonard Kleinrock. *Queueing Systems Volume II: Computer Applications*. John Wiley & Sons, 1976.

[13] Keqin Li. Optimizing average job response time via decentralized probabilitic job dispatching in heterogeneous multiple computer systems. *The Computer Journal*, 41(4):223–230, 1998.

[14] Woei Lin and P. R. Kumar. Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*, 29(8):696–703, Aug. 1984.

[15] Hsing Paul Luh and Ioannis Viniotis. Threshold control policies for heterogeneous server systems. *Mathematical Methods of Operations Research*, 55:121–142, 2002.

[16] Michael Mitzenmacher. How useful is old information? *IEEE Transactions on Parallel and Distributed Systems*, 11(1):6–20, Jan. 2000.

[17] Randolph D. Nelson and Thomas K. Philips. An approximation for the mean response time for shortest queue routing with general interarrival and service times. *Performance Evaluation*, 17(2):123–139, Mar. 1993.

[18] Lionel M. Ni and Kai Hwang. Optimal load balancing in a multiple processor with many job classes. *IEEE Transactions on Software Engineering*, 11(5):491–496, May 1985.

[19] K. Oida and K. Shinjo. Characteristics of deterministic optimal routing for two heterogeneous parallel servers. *International Journal of Foundations of Computer Science*, 12(6):775–790, 2001.

[20] Kazumasa Oida and Shigeru Saito. A packet-size aware adaptive routing algorithm for parallel transmission server system. *Journal of Parallel and Distributed Computing*, 64:36–47, 2004.

[21] Zvi Rosberg and Armand M. Makowski. Optimal routing to parallel heterogeneous servers – small arrival rates. *IEEE Transactions on Automatic Control*, 35(7):789–796, Jul. 1990.

[22] Asser N. Tantawi and Don Towsley. Optimal static load balancing in distributed computer systems. *Journal of the Association for Computing Machinary*, 32(2):445–465, Apr. 1985.

[23] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 13:826–834, 1977.

A Extend the proof in Section 5 to multiple queues.

In order to extend the proofs for Theorems 3.3 and 3.5 from two queues to multiple queues, we need to show that we can get an SI strategy in finite steps by pairwise operations. There are many different sequences of doing so; we provide one of the algorithms that terminates in finite steps.

Note that we define two measures in the paper that $\Lambda := \alpha_i\lambda_i$ and $\Omega := \alpha_i\omega_i$ for the i-th queue. We say the i-th queue is *unfairly loaded over* the j-th queue if the i-th queue has a greater Λ and a greater Ω than the j-th queue, i.e.,$\alpha_i\lambda_i > \alpha_j\lambda_j$ and $\alpha_i\omega_i > \alpha_j\omega_j$. Also, let us refer to the real interval $[\xi_i^{(0)}, \xi_i^{(1)}]$ as *span* of the i-th queue where, as in Definition 2.1, $\xi_i^{(0)}$ is the size of the shortest job and $\xi_i^{(1)}$ is the size of the longest job that assigned to the i-th queue.

Homogeneous systems:

```
procedure SEARCH-A-BETTER-SI STRATEGY
  LOAD-BALANCE()  ; transferring load between queues
  LOAD-SWAP()    ; find an SI strategy by pairwise swapping load

procedure LOAD-SWAP()
  $\mathcal{S} \leftarrow \varnothing$     ; queues whose intervals are disjoint with each other
  $\mathcal{S}_1 \leftarrow \{1, 2, \ldots, n\}$               ; queues except for those in $\mathcal{S}$
  for $i \leftarrow 1$ to $n$
    $x \leftarrow \arg\max_{k \in \mathcal{S}_1} \alpha_k\lambda_k$          ; the queue maximizing $\Lambda$ in $\mathcal{S}_1$
    foreach $j \in \mathcal{S}_1 - \{x\}$
      LOAD-SWAP-PAIRWISE($x$,$j$)
    end
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$, $\mathcal{S}_1 \leftarrow \mathcal{S}_1 - \{x\}$ ; move $x$ from $\mathcal{S}_1$ to $\mathcal{S}$
  end

procedure LOAD-SWAP-PAIRWISE($i$,$j$)
  swap portions of loads between queue $i$ and $j$
      (with $\alpha_i\lambda_i \geq \alpha_j\lambda_j$ and $\alpha_i\omega_i \geq \alpha_j\omega_j$)
      according to Condition 1 in the proof of Theorem  3.3
      as shown in Figure 8(a).


procedure LOAD-BALANCE()
  $\mathcal{S} \leftarrow \{1\}$                     ; no unfairly loaded queues in $\mathcal{S}$
  for $i \leftarrow 2$ to $n$
    LOAD-BALANCE-INDUCTIVE($\mathcal{S}$,$i$)  ; balance the load between $i$ and $\mathcal{S}$
    $\mathcal{S} \leftarrow S \cup \{i\}$                    ; add queue $i$ to $\mathcal{S}$ one by one
  end

procedure LOAD-BALANCE-INDUCTIVE($\mathcal{S}$,$i$)
  $\mathcal{S}_1 \leftarrow \{x | x \in \mathcal{S} \wedge$UNFAIRLY-LOADED($x$,$i$)$\}$
  if $\mathcal{S}_1 \neq \varnothing$         ; $\mathcal{S}_1$: unfairly loaded queues in $\mathcal{S}$ over queue $i$
    while $\mathcal{S}_1 \neq \varnothing$
      LOAD-TRANSFER-TO($\mathcal{S}_1$,$i$)    ; transfer load from $\mathcal{S}_1$ to queue $i$
      $\mathcal{S}_1 \leftarrow \mathcal{S}_1 - \{x | x \in \mathcal{S}_1 \wedge \neg$UNFAIRLY-LOADED($x$,$i$)$\}$
         ; remove queues from $\mathcal{S}_1$ that are not unfairly loaded over $i$
    end
    return
  end
  $\mathcal{S}_1 \leftarrow \{x | x \in \mathcal{S} \wedge$UNFAIRLY-LOADED($i$,$x$)$\}$
  if $\mathcal{S}_1 \neq \varnothing$   ; $\mathcal{S}_1$: queues in $\mathcal{S}$ over which queue $i$ is unfairly loaded
    while $\mathcal{S}_1 \neq \varnothing$
      LOAD-TRANSFER-FROM($\mathcal{S}_1$,$i$)   ; transfer load from $i$ to queue $\mathcal{S}_1$
      $\mathcal{S}_1 \leftarrow \mathcal{S}_1 - \{x | x \in \mathcal{S}_1 \wedge \neg$UNFAIRLY-LOADED($i$,$x$)$\}$
```

```
            ; remove queues from $\mathcal{S}_1$ over which $i$ is not unfairly loaded
41     end
42   end
44
45 function Unfairly-Loaded(i,$j$)
46   return $\alpha_i\lambda_i > \alpha_j\lambda_j \wedge \alpha_i\omega_i > \alpha_j\omega_j$
47
48 procedure Load-Transfer-To($\mathcal{S}_1$,$i$)
49   transfer loads from queues in $\mathcal{S}_1$ to $i$
         while keep load balanced in $\mathcal{S}_1$
         until $\{x|x \in \mathcal{S}_1 \wedge \neg$Unfairly-Loaded($x$,$i$)$\} \neq \varnothing$
50
51 procedure Load-Transfer-From($\mathcal{S}_1$,$i$)
52   transfer loads from $i$ to queues in $\mathcal{S}_1$
         while keep load balanced in $\mathcal{S}_1$
         until $\{x|x \in \mathcal{S}_1 \wedge \neg$Unfairly-Loaded($i$,$x$)$\} \neq \varnothing$
53
```

Remarks:

- Procedures Load-Transfer-From and Load-Transfer-To transfer load between a queue, x, and a group of queues in $\mathcal{S}_1$ while keeping the orders of Λ and Ω in the group unchanged. This can be done by finding a feasible solution for the amount of load transfered from each queue, with a bunch of inequalities. Also, a queue in $\mathcal{S}_1 \cup \{x\}$ is not unfairly loaded over any queue in $\mathcal{S} - \mathcal{S}_1$, by the formation of $\mathcal{S}_1$, and vice versa. This property is not affected by transferring load between $\mathcal{S}_1$ and x because the load transferring would only shift their Λ's and Ω's more close to each other.
- The transferred jobs can have any distribution. The simplest way is to transferred jobs randomly without considering job sizes, i.e., $\Delta\lambda/\lambda_k = \Delta\rho/\rho_k = \Delta\omega/\omega_k$ where k is the index of the *source* queue.
- In Line 9, we always choose queue x to be the queue that has the greatest Λ in $\mathcal{S}_1$. After a pairwise load swapping between queue x and a queue that has a smaller Λ (and therefore a greater Ω), we can see from the proof of Theorem 3.3, queue x is always at the left-hand side of the real axis. Meanwhile, each load swapping makes Λ of queue x even greater so it is always the queue that has the greatest Λ in $\mathcal{S}_1$. During load swapping, we do not change the total amount of load for queue x while replacing long jobs with small jobs, so the size of the longest job for queue x will become shorter and shorter. This property is important for showing finite convergence: if we have done load swapping between queue x and a queue, say i, a later load swapping between queue x with another queue, say j, will not make the spans of queue x and queue i overlap. In other words, we need only one load swapping between each pair of queues.
- Line 28 finds the subset of queues in $\mathcal{S}$ that is unfairly loaded over queue i, where Line 36 finds the subset of queues in $\mathcal{S}$ over which queue i is unfairly loaded. Note that only one of the two subsets is non-empty because of the transitive property of $>$ and $<$.

Heterogeneous systems In order to extend the proof of Theorem 3.5 from two queues to multiple queues, we need to made a few changes to the previous algorithm:

- It is fine that a slower queue is unfairly loaded in the load balancing stage. So replace Line 36 with the following:

  ```
  36′   $\mathcal{S}_1 \leftarrow \{x|x \in \mathcal{S} \wedge c_i \geq c_x \wedge$Unfairly-loaded($i$,$x$)$\}$
  ```

- The load swapping operation is different from homogeneous systems. So replace Line 3 with the following:

  ```
  3′    Load-Swap-Hetero()  ; find an NSI strategy by pairwise load swapping
  ```

We instead use the following load swapping procedure for heterogeneous systems.

```
54  procedure Load-Swap-Hetero()
55    $\mathcal{S} \leftarrow \varnothing$     ; $\mathcal{S}$: queues whose intervals are disjoint with each other
56    $\mathcal{S}_1 \leftarrow \{1, 2, \ldots, n\}$    ; $\mathcal{S}_1$: all queues except for those in $\mathcal{S}$
57    for $i \leftarrow 1$ to $n$
58      $x \leftarrow \arg\max_{k \in \mathcal{S}_1} \alpha_k \lambda_k$      ; get the queue having the max $\Lambda$ in $\mathcal{S}_1$
59      foreach $j \in \mathcal{S}_1 - \{x\} - \{y | y \in \mathcal{S}_1 \wedge$Unfairly-Loaded($x$,$y$)$\}$
60        Load-Swap-Pairwise($x$,$j$)
61      end
62      foreach $j \in \{y | y \in \mathcal{S}_1 \wedge$Unfairly-Loaded($x$,$y$)$\}$
63        Load-Swap-Pairwise-Extended($x$,$j$)
64      end
65      $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$, $\mathcal{S}_1 \leftarrow \mathcal{S}_1 - \{x\}$    ; move $x$ from $\mathcal{S}_1$ to $\mathcal{S}$
66   end
67
68 procedure Load-Swap-Pairwise-Extended($i$,$j$)
69   swap portion of loads between queue $i$ and $j$
       (with $\alpha_i \lambda_i > \alpha_j \lambda_j$ and $\alpha_i \omega_i > \alpha_j \omega_j$)
       according to Condition 4′ in the proof of Theorem 3.5
       as shown in Figure 8(b).
70
```

Remarks:

- The new load swapping procedure still iterates in the descending order of Λ. For each queue, say i, we first do pairwise swapping with the queues over which queue i is not unfairly loaded, as illustrated in Figure 8(a) (Line 59-61). As stated earlier, the size of the longest job assigned to queue x will become smaller and smaller and its Λ remains the greatest in $\mathcal{S}_1$. Then we do pairwise swapping with the queues over which queue i *is* unfairly loaded, as illustrated in Figure 8(b) (Line 62-64). Since queue i is unfairly loaded, it corresponds to $\lambda_1 f_1(x)$ in Figure 8(b). Figure 8(b) shows the span of the unfairly loaded queue will become narrower and narrower and meanwhile Λ of queue x remains the greatest in $\mathcal{S}_1$ (since Λ is unchanged). For the same reason as stated earlier, we need only one load swapping between each pair of queues.