

Partitioning of Servers in Queueing Systems During Rush Hour

Bin Hu

Department of Operations and Management Science, Ross School of Business, University of Michigan,
Ann Arbor, Michigan 48109, hub@umich.edu

Saif Benjaafar

Division of Industrial and Systems Engineering, Department of Mechanical Engineering, University of Minnesota,
Minneapolis, Minnesota 55455, saif@umn.edu

This paper is motivated by two phenomena observed in many queueing systems in practice. The first is the partitioning of server capacity among different customers based on their service time requirements. The second is rush hour demand where a large number of customers arrive over a short period of time followed by few or no arrivals for an extended period thereafter. We study a system with multiple parallel servers and multiple customer classes. The servers can be partitioned into server groups, each dedicated to a single customer class. The system operates under a rush hour regime with a large number of customers arriving at the beginning of the rush hour period. We show that this allows us to reduce the problem to one that is deterministic and for which closed-form solutions can be obtained. We compare the performance of the system with and without server partitioning during rush hour and address three basic questions. (1) Is partitioning beneficial to the system? (2) Is it equally beneficial to all customer classes? (3) If it is implemented, what is an optimal partition? We evaluate the applicability of our results to systems where customers arrive over time using (1) deterministic fluid models and (2) simulation models for systems with stochastic interarrival times.

Key words: server partitioning; multiserver queueing systems; multiple demand classes; rush hour demand
History: Received: October 26, 2006; accepted: April 11, 2008. Published online in *Articles in Advance* September 24, 2008.

1. Introduction

This paper is motivated by two phenomena observed in many queueing systems in practice. The first is the partitioning of server capacity among different customers based on their service requirements. For example, in systems with parallel servers, it is not uncommon to dedicate a subset of the servers to customers with the shortest service times. The classic example is of course the express lane(s) found in grocery stores, but it is also observed elsewhere including banks, airports, government offices, and call centers. The second phenomenon is the arrival of a large number of customers over a short period of time followed by few or no arrivals for an extended period thereafter. This situation is observed in systems subject to *rush hour* demand, such as toll booths on highways during peak hours, fast-food restaurants during lunch time, concession stands in stadiums during intermissions, customs, and immigration controls at airports following an international flight, and many others.

The common feature to these examples is the almost simultaneous arrivals of a large number of customers followed by few or no additional arrivals until the next rush hour. In such systems, the primary concern is the waiting time that customers experience during this period because there is typically ample capacity at other times.

Despite the prevalence of both phenomena in practice, they appear not to have been sufficiently studied in the literature. In particular, there are few results regarding how to best partition servers among customer classes and whether or not this partitioning is beneficial. Similarly, there are relatively few results for systems subject to rush hour demand, particularly with respect to how capacity should be managed during this period.

In this paper, we address several of these limitations. First, we show that partitioning can indeed be beneficial to the system as a whole. Second, we show that there does not exist a partitioning of servers

among customers under which all customer classes are better off (although there exists a partitioning under which all customers are indifferent between the partitioned and unpartitioned system). Third, we show that there exists a unique partitioning that optimizes the performance of the system as a whole and we provide a closed-form expression for the number of servers allocated to each class under this optimal partition.

In our analysis, we make the assumption that all customers arrive instantaneously at the beginning of the rush hour period. This assumption allows us to reduce the problem to one that is deterministic and for which closed-form solutions can be obtained. Using simulation, we examine the effectiveness of our optimal server partitioning solution when customers arrive over time with stochastic interarrival times. We also compare our results with results obtained using a fluid model where both arrivals and service occur with fixed rates. In both cases, we find that our optimal allocation is effective when customer arrival rates are sufficiently higher than service rates. Our approach is not appropriate in systems where times between consecutive arrivals are much longer than service times. Obviously, those settings do not correspond to rush hour regimes.

There is a rich literature in queueing theory that compares partitioned versus unpartitioned systems. The partitioned system is often in the form of multiple single server queues with each queue serving an independent stream of customers who arrive continuously over time with stochastic interarrival times. The unpartitioned system, however, is in the form of a single multiserver queue from which all customers are served on a first-come, first-served basis; see, for example, Kleinrock (1976), Rothkopf and Rech (1987), Smith and Whitt (1981), Whitt (1992, 1999), Benjaafar (1995), and Benjaafar et al. (2005). An important insight from this literature is that the unpartitioned system is superior as long as all customers have identical service time distributions, but not necessarily so when customer streams have different service time requirements. The setting for this literature is different from ours in that it assumes steady state operation over an infinite horizon and continuous customer arrivals. Most of this literature also assumes that the set of servers associated with each

demand stream in the partitioned system either consists of a single server or is exogenously determined. A notable exception is Whitt (1999), who presents a heuristic procedure for assigning servers to each customer stream.

One of the benefits of partitioning, both in the queueing literature and in our model, is protecting customers with short processing times from experiencing delays due to customers with long processing times. This, of course, can be achieved without resorting to partitioning but by assigning different priorities to different customer streams or classes. However, having customers wait in a single queue and be sequenced based on a priority scheme can be unpractical in some settings or not acceptable to certain customers in others; see Rothkopf and Rech (1987) for an illuminating discussion on this topic. Partitioning can be viewed as an alternative to priority sequencing in such settings.

Another approach that has been used to differentiate between customer classes is to reserve servers (or buffer spaces in the server queues) for certain customer classes based on the current state of the system. This approach is useful when the number of buffer spaces is limited and customers are rejected at a cost when buffers are full (a special case is the so-called *loss system*, where no queueing is allowed). When the rejection costs for different customers are different, it becomes desirable to reserve capacity for customer classes with higher costs; see for example Ross and Yao (1990), Altman et al. (2001), and Savin and Wang (2006), and the references therein. Our setting is different because we do not place limits on the number of customers in the queue.

Our work is also related to the vast literature on deterministic scheduling with parallel machines (see, for example, Chapter 5 of Pinedo 2002 and the references therein). The focus of this literature is on the assignment and sequencing of jobs at each machine. The partitioning of customers we consider in this paper is not common in that literature. Partitioning is applicable in settings such as the one we consider in this paper when such sequencing is not possible and when jobs can be categorized into classes (e.g., product families) based on their processing times.

Finally, let us note that using deterministic analysis to model queueing systems is not new. There

is a well-established literature that uses deterministic fluid models to approximate the behavior of queueing systems; see, for example, Newell (1982), Hall (1991), Avram et al. (1995), and Whitt (2006), among others. In a fluid model, the arrival and service processes are approximated using deterministic and continuous rates. Discrete state variables, such as number of customers in the queue, are approximated using continuous variables (amount of fluid). In §7.1, we show that, using a fluid model, we can recover most of the results we obtain with our discrete model, including the same optimal server allocation.

The rest of this paper is organized as follows. In §2, we describe our model for systems with and without partitioning. In §§3 and 4, we describe our main theoretical results. In §5, we provide numerical results illustrating the benefit of partitioning. In §6, we examine the impact of server pooling within each class. In §7, we investigate the impact of phased customer arrivals using deterministic fluid approximations and Monte Carlo simulation. In §8, we offer some concluding comments.

2. Problem Description and Preliminary Results

We consider a queueing system consisting of n identical servers and l customer classes. Service times within each class are assumed to be independent and identically distributed with mean $E(S_i)$ for customer class i , $i = 1, \dots, l$, where S_i is a random variable denoting service time for customer class i . Without loss of generality, we assume that $E(S_1) \leq E(S_2) \leq \dots \leq E(S_l)$. We consider two scenarios, one in which the servers are partitioned among the customer classes with each class i , $i = 1, \dots, l$, assigned k_i servers where k_i is a positive integer (i.e., $k_i > 0$) and $k_1 + \dots + k_l = n$. We refer to this system as the partitioned system. The other scenario is one in which all servers are accessible to all customers. We refer to this system as the unpartitioned system. Upon arrival, a customer chooses a server among those assigned to his class and waits in the queue of that server until the server becomes available. We assume that no jockeying is allowed so that customers do not switch queues once they have joined one (see §6 for a discussion of the impact of pooling servers

within each class). Within each queue, we assume that customers are served on a first-come, first-served basis. Once a customer receives service, the customer leaves the system. Of course, in the unpartitioned system, a customer can choose any server.

For both the partitioned and unpartitioned systems, we assume that customers choose to join the queue with the fewest customers. This implies that in the absence of partitioning a customer does not know the type of other customers who are already in the system. We also assume that an arriving customer does not know the remaining service time for customers currently in service. For customers that seek to minimize expected delay, choosing to join the queue with the fewest customers is therefore plausible (and perhaps consistent with behavior in practice). This assumption is further justified given our rush hour setting as we discuss below.

We are concerned with a rush hour regime of operation whose beginning is marked by the simultaneous arrival of m customers to the system, with m being much greater than the number of servers ($m \gg n$). We assume that no further arrivals occur until these m customers have cleared the system. This is obviously an approximation of rush hour phenomena in practice. Arrivals are typically phased over time and continue to occur beyond the initial arrival rush (in §7, we provide results based on fluid approximations and simulation to test the impact of having customers arrive over time). We assume that the fraction of customers that are of type i is p_i , where $0 < p_i < 1$ and $p_1 + \dots + p_l = 1$, so that the number of customers of type i is $m_i = p_i m$, and also assume that $m_i \gg n$. Our main results in Theorems 1–3 regarding server allocations do not depend on the specific values of m and m_i but only on the ratio m_i/m .

Without loss of generality, we assign an arbitrary but unique index from 1 to m to each customer, where customer j ($j = 1, \dots, m$) chooses which queue to join before customer $j + 1$ does. Because we assume that customers join the server with the shortest queue immediately, the length of each of the queues dedicated to class i would be m_i/k_i customers (including the one in service) once all m_i customers have joined a queue. We shall treat m_i/k_i as an integer because $m_i \gg k_i$. In a system without server partitioning, the length of all queues is equal to m/n , which we also

treat as an integer. We assume that customers of different types are perfectly mixed so that the probability of a customer, selected at random from any queue in the unpartitioned system, to be of type i is p_i .

We use expected time customers spend in the system as our measure of performance. For the unpartitioned system, the expected time a customer spends in the system (regardless of his class) is given by

$$\begin{aligned} E(W^u) &= \sum_{i=1}^l p_i E(S_i) \left(\frac{1+2+\dots+m/n}{m/n} \right) \\ &= \left(\frac{1}{2} + \frac{m}{2n} \right) \sum_{i=1}^l p_i E(S_i). \end{aligned} \quad (1)$$

Similarly, for the partitioned system, the expected time in the system for customers of type i is given by

$$E(W_i^p(k_i)) = \left(\frac{1}{2} + \frac{p_i m}{2k_i} \right) E(S_i), \quad (2)$$

and the expected time in the system for an arbitrary customer is

$$E(W^p(\mathbf{k})) = \sum_{i=1}^l p_i E(W_i^p(k_i)), \quad (3)$$

where $\mathbf{k} = (k_1, \dots, k_l)$ is a vector specifying the number of servers dedicated to each class. Although the above performance measures depend on the parameter m , we show in the next two sections that our main results in Theorems 1–3 are in fact independent of m .

Note that expectation in the above expressions is taken with respect to service times. However, because the expected value of a linear combination of random variables is a linear combination of the expected values of the random variables, the above expressions are a function of only the means of service times. Hence, the formulations are essentially the same as those for a deterministic system (and the above expressions can be viewed as averages). This is of course due to our assumption of instantaneous customer arrivals at the beginning of the rush hour period. A consequence of this assumption is that each server works continuously until its queue has been emptied. The time a customer spends in the system depends only on how many customers are ahead in its queue and the expected service time for these customers, which are known with certainty. This is not

the case in a system where arrivals are phased over time. In that case, the number of customers that are ahead in the queue of an arriving customer is stochastic and depends on the distribution both arrival times and service times; see §7 for a discussion of phased customer arrivals.

Finally, we should note that throughout the paper we assume that the values of m_i and m are known prior to server partitioning, including the optimal partitioning described in §4. This is, of course, a reasonable assumption when server partitioning can be quickly carried out (or adjusted) upon observing customer arrivals. This is also a reasonable assumption when there is advance notice on the number of customers from each type. For example, in a sporting or entertainment event, information about number and mix of customers could be obtained from the number and type of tickets sold. However, there are settings where the number and mix of customers are not known a priori, although information about their distribution may be available, and partitioning must be carried out prior to observing actual arrivals (or the same partitioning must be maintained over multiple rush hour occurrences). In those settings, our approach and the results of Theorems 1–3 would still continue to apply as long as the fraction of customers p_i from each class i is known a priori and remains constant from one rush hour occurrence to the next. For example, this would be the case when the total number of customers m is very large (strictly speaking, when $m \rightarrow \infty$) and p_i corresponds to the known probability of a customer being of type i , independently of other customers. In that case, by virtue of the law of large numbers, the fraction of customers of type i converges to p_i as m becomes very large. There may be settings where the fraction of customers p_i varies stochastically from one rush hour occurrence to the next but the same partitioning must be maintained over multiple occurrences. Determining the optimal server partitioning in this case is more difficult and we leave it as a topic for future research.

3. Fairness of Partitioning

In this section, we address the question of fairness. Namely, is it possible to find a server partitioning scheme under which all customer classes are better off than in the unpartitioned system?

THEOREM 1. Let

$$\begin{aligned} \mathbf{k}^w &= (k_1^w, \dots, k_l^w) \\ &= \left(\frac{p_1 E(S_1)}{\sum_{i=1}^l p_i E(S_i)} n, \dots, \frac{p_l E(S_l)}{\sum_{i=1}^l p_i E(S_i)} n \right). \end{aligned} \quad (4)$$

Then, the following equivalence holds for all i :

$$E(W_i^p(k_i)) \leq E(W^u) \text{ if and only if } k_i \geq k_i^w.$$

Therefore, the unique solution to the system of inequalities

$$\begin{cases} E(W_1^p(k_1)) \leq E(W^u) \\ \dots \\ E(W_l^p(k_l)) \leq E(W^u) \end{cases}$$

is

$$\begin{cases} k_1 = k_1^w \\ \dots \\ k_l = k_l^w \end{cases}$$

for which $E(W_i^p(k_i^w)) = E(W^u)$ for all i .

The proof for Theorem 1 is straightforward and we omit it. Theorem 1 shows that a partitioned system can never simultaneously improve the performance of all customer classes. Hence, an improvement achieved by any customer class comes only at the expense of other classes. It is possible for a partitioned system to provide the same level of performance for each class as an unpartitioned system. However, this is the case if and only if each class is allocated a number of servers proportional to its workload, that is, $k_i = k_i^w$. We refer to such an allocation as the *workload-proportional* allocation. Note that a workload-proportional allocation is feasible only if the k_i^w are integer valued. If not, then no allocation exists under which the partitioned and unpartitioned systems are equivalent.

Although the workload-proportional allocation (if feasible) is the only allocation that guarantees that each class is not worse off with partitioning, it is not the only allocation that could provide the same overall expected time in the system as the unpartitioned system.

THEOREM 2. Let

$$\mathbf{k}^m = (k_1^m, \dots, k_l^m) = (p_1 n, \dots, p_l n). \quad (5)$$

Then, $E(W^p(\mathbf{k}^m)) = E(W^u)$. Furthermore, for any $1 \leq j \leq l$ such that $E(S_j) \leq \sum_{i=1}^l p_i E(S_i)$, we have $k_j^m \geq k_j^w$. Otherwise, if $E(S_j) \geq \sum_{i=1}^l p_i E(S_i)$, we have $k_j^m \leq k_j^w$.

The proof is also straightforward and we again omit it. Theorem 2 shows that an allocation proportional to the population of each class leads to the same overall system performance as the unpartitioned system. This allocation is of course feasible only if it yields integer-valued k_i^m 's. We refer to this allocation as the *mix-proportional* allocation. As we can see, a mix-proportional allocation favors customer classes with relatively short service times (classes whose mean service time is smaller than the overall mean service time); these customers receive more servers than they would under the workload-proportional allocation. Consequently the performance of other customer classes suffers relative to their performance in an unpartitioned system. We will use this property in the next section to further characterize the optimal allocation.

4. Optimal Partitioning

In this section, we address the question of whether or not partitioning can improve the overall performance of the system, even though it may not improve the performance of all customers, and, if so, what is the optimal way to partition servers among different classes. Our criterion for system optimality is the expected time in the system for an arbitrary customer, as defined in (3). This is consistent with treatments in the literature for a variety of queueing systems. Our analysis can be easily extended to systems where different weights or costs are associated with delay for different customer classes. We briefly discuss this extension at the end of this section. There may of course be other criteria for optimality; we discuss a few of these in §8.

THEOREM 3. Let

$$\begin{aligned} \mathbf{k}^* &= (k_1^*, \dots, k_l^*) \\ &= \left(\frac{p_1 \sqrt{E(S_1)}}{\sum_{i=1}^l p_i \sqrt{E(S_i)}} n, \dots, \frac{p_l \sqrt{E(S_l)}}{\sum_{i=1}^l p_i \sqrt{E(S_i)}} n \right). \end{aligned} \quad (6)$$

1. The minimum

$$\min_{\sum_{i=1}^l k_i = n} E(W^p(\mathbf{k})) = \frac{1}{2} \sum_{i=1}^l p_i E(S_i) + \frac{m}{2n} \left(\sum_{i=1}^l p_i \sqrt{E(S_i)} \right)^2 \quad (7)$$

is attained with the unique vector \mathbf{k}^* .

2. For each $1 \leq j \leq l$, if $E(S_j) \leq \sum_{i=1}^l p_i E(S_i)$, then $k_j^* \geq k_j^w$; otherwise, $k_j^* \geq k_j^m$. In general, $k_j^* \geq \min(k_j^m, k_j^w)$.

3. There exists at least some j such that $E(S_j) \leq \sum_{i=1}^l p_i E(S_i)$ and $k_j^w \leq k_j^* \leq k_j^m$. Similarly, there exists at least some j' such that $E(S_{j'}) \geq \sum_{i=1}^l p_i E(S_i)$ and $k_{j'}^w \geq k_{j'}^* \geq k_{j'}^m$.

PROOF.

1. We need to show that

$$\begin{aligned} E(W^p(\mathbf{k})) &= \frac{1}{2} \sum_{i=1}^l p_i E(S_i) + \frac{m}{2n} \sum_{i=1}^l \frac{p_i^2 E(S_i) n}{k_i} \\ &\geq \frac{1}{2} \sum_{i=1}^l p_i E(S_i) + \frac{m}{2n} \left(\sum_{i=1}^l p_i \sqrt{E(S_i)} \right)^2. \end{aligned}$$

Using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \left(\sum_{i=1}^l p_i \sqrt{E(S_i)} \right)^2 &= \left[\sum_{i=1}^l \left(p_i \sqrt{E(S_i)} \sqrt{\frac{n}{k_i}} \right) \sqrt{\frac{k_i}{n}} \right]^2 \\ &\leq \left(\sum_{i=1}^l \frac{p_i^2 E(S_i) n}{k_i} \right) \left(\sum_{i=1}^l \frac{k_i}{n} \right). \end{aligned}$$

The equality holds if and only if $p_i \sqrt{E(S_i)} \sqrt{n/k_i} = c \sqrt{k_i/n}$ for all $1 \leq i \leq l$ with a common $c \neq 0$, which, together with $\sum_{i=1}^l k_i = n$, leads to $c = \sum_{i=1}^l p_i \sqrt{E(S_i)}$ and $\mathbf{k} = \mathbf{k}^*$.

2. By virtue of the Cauchy-Schwarz inequality,

$$\sum_{i=1}^l p_i \sqrt{E(S_i)} = \sum_{i=1}^l \sqrt{p_i} \sqrt{p_i E(S_i)} \leq \sqrt{\sum_{i=1}^l p_i E(S_i)}.$$

If

$$E(S_j) \leq \sum_{i=1}^l p_i E(S_i),$$

we have

$$\begin{aligned} k_j^* &= \frac{p_j \sqrt{E(S_j)}}{\sum_{i=1}^l p_i \sqrt{E(S_i)}} n \geq \frac{p_j \sqrt{E(S_j)}}{\sqrt{\sum_{i=1}^l p_i E(S_i)}} n \\ &\geq \frac{p_j E(S_j)}{\sum_{i=1}^l p_i E(S_i)} n = k_j^w \end{aligned}$$

and, by Theorem 2,

$$k_j^m \geq k_j^w.$$

Similarly, if

$$E(S_j) \geq \sum_{i=1}^l p_i E(S_i),$$

we have

$$k_j^* = \frac{p_j \sqrt{E(S_j)}}{\sum_{i=1}^l p_i \sqrt{E(S_i)}} n \geq \frac{p_j \sqrt{\sum_{i=1}^l p_i E(S_i)}}{\sqrt{\sum_{i=1}^l p_i E(S_i)}} n = k_j^m$$

and

$$k_j^m \leq k_j^w.$$

In general, we have $k_j^* \geq \min(k_j^m, k_j^w)$.

3. One side of the inequalities were already shown in Result 2 of the theorem. For the remaining inequalities, consider the fact

$$\sum_{j=1}^l k_j^* = \sum_{j=1}^l k_j^w = \sum_{j=1}^l k_j^m = n,$$

which would lead to a contradiction if the inequalities did not hold. \square

We should note that the optimal allocation \mathbf{k}^* is feasible only if it leads to integer-valued k_i^* 's. If not, the optimal allocation would have to be obtained via a search over the discrete space of feasible values for the variables k_i . Note that the function $E(W^p(\mathbf{k}))$ is jointly convex in the variables k_i , which would facilitate such a search.

COROLLARY 4. *The allocation defined by \mathbf{k}^* , if feasible, provides an overall expected time in the system that is at least as small as the one provided by the unpartitioned system.*

PROOF. The result follows from the fact that

$$E(W^p(\mathbf{k}^*)) = \min_{\sum_{i=1}^l k_i = n} E(W^p(\mathbf{k})) \leq E(W^p(\mathbf{k}^w)) = E(W^u),$$

where the last equality is due to Theorem 1. \square

The expression of the optimal allocation \mathbf{k}^* is reminiscent of the workload-proportional allocation, except that the effect of service times is attenuated by the square root factor. This, of course, favors customer classes with relatively short service times and, as we can see, they receive a higher allocation of servers than they would under the workload-proportional allocation. This comes at the expense of customer with relatively long service times who receive a smaller allocation of servers; nevertheless, these customers are guaranteed to receive an allocation that is

at least greater than or equal to the one they would receive under a mix-proportional allocation. In general, regardless of customer class, $k_j^* \geq \min(k_j^m, k_j^w)$ is always satisfied. The square root effect observed under the optimal allocation is also reminiscent of similar results obtained in other settings where the optimal capacity for a demand stream is found to be a function of the square root of the workload associated with that stream (e.g., demand rate); see, for example, Kleinrock (2002).

The workload- and mix-proportional allocations can be viewed as two extreme forms of allocation, one accounting for service times, the other ignoring service time differences and accounting only for the relative population size of each class. The optimal allocation accounts for service times but not to the extent that the workload-proportional allocation does. Because of this, one might assume that an ordering among the three allocations always holds. However, this is not true in general, although it is so for the important case of a system with two classes (this case is important because in practice customers are often partitioned into two classes). For this case, as a consequence of Result 3 in Theorem 3, we have $k_1^w \leq k_1^* \leq k_1^m$ and $k_2^w \geq k_2^* \geq k_2^m$.

In the above analysis, we have implicitly assumed that customer classes are equally important. However, there may be settings where some customer classes are more important than others. In that case, a greater weight would need to be placed on the time in the system experienced by the more important classes. This could be achieved by assigning quantitative weights (which may correspond to delay penalties or costs) to each customer class, say $w_i > 0$ for class i , and using weighted expected time in the system

$$E(\widehat{W}^p(\mathbf{k})) = \sum_{i=1}^l p_i w_i E(W_i^p(k_i)) \tag{8}$$

as the measure of system optimality. This changes little to the analysis except that the optimal allocation is now given by

$$\hat{k}_i^* = \frac{p_i \sqrt{w_i E(S_i)}}{\sum_{j=1}^l p_j \sqrt{w_j E(S_j)}} n \tag{9}$$

for $i = 1, \dots, n$.

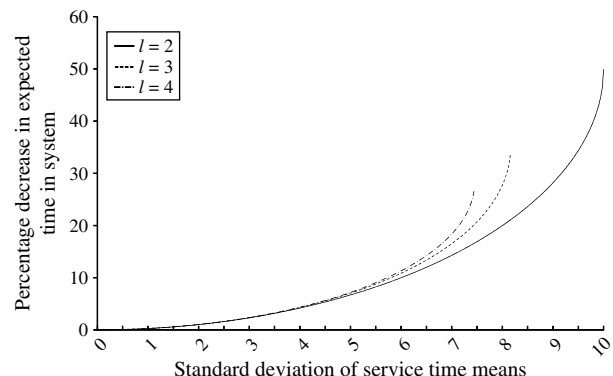
5. Benefit of Partitioning

In this section, we illustrate using a simple example the relative benefit that could be realized from partitioning. We consider a system with l classes with equal populations so that $p_i = 1/l$ for $i = 1, \dots, l$. The means of the service times of different classes are uniformly distributed over the range $[M - x, M + x]$, where $x < M$ and $M > 0$. This implies that $E(S) = \sum_{i=1}^l p_i E(S_i) = M$ and $E(S_i) = M - x + 2x(i - 1)/(l - 1)$. For example, if $M = 10$, $x = 3$, and $l = 4$, then $E(S_1) = 7$, $E(S_2) = 9$, $E(S_3) = 11$, and $E(S_4) = 13$. Using this construction, we can compare systems with the same overall mean M but different variability in mean service times by varying x and l as shown in Figure 1.

Figure 1 shows the percentage reduction in the expected time in the system that results from optimal partitioning relative to no partitioning, as a function of the standard deviation of service time means. We do so for different values of l and for different values of standard deviation in the service time means (obtained by varying the parameter x). Other parameters are $M = 10$, $n = 10$, and $m = 500$. The following observations can be made:

1. The percentage decrease in expected time in system due to partitioning can be significant, up to 50% in the cases shown.
 2. The improvement is increasing in the variability among the means of service times of the different demand classes.
 3. For a given level of variability, the improvement is increasing in the number of demand classes.
- The first observation is not surprising because the improvement that could be achieved with partitioning is

Figure 1 Percentage Decrease in Expected Time in System Due to Partitioning



not theoretically bounded but rather depends on the differences among service time means of the different demand classes. The fact that the improvement is not bounded can be seen from the limit

$$\delta = \lim_{m \rightarrow \infty} \frac{E(W^p(\mathbf{k}^*))}{E(W^u)} = \frac{(\sum_{i=1}^l p_i \sqrt{E(S_i)})^2}{E(S)}.$$

It is easy to construct examples where δ is arbitrarily small and even approaches zero. The second observation is also consistent with intuition. The more variability in mean service times there is among different classes, the more valuable it is to separate these classes (by protecting those with short service times from those with long service times). This observation is consistent with observations made by Whitt (1999) in the context of a standard queueing system with Poisson arrivals. The third observation is due to the fact that, everything else being equal, dividing customers in more classes is always desirable because we can always choose to treat two or more demand classes in the same way in terms of server assignments (which would be equivalent to merging them in a single class).

6. Impact of Server Pooling

In our analysis so far, we have assumed that customer jockeying is not allowed. That is, customers from a class cannot switch queues if one of the queues available to this class becomes empty. The need for jockeying would of course be mitigated if the servers within each class were pooled and customers wait in a single queue where they are processed by the first available server on a first-come, first-served basis. Unfortunately, the analysis of systems with pooled queues is difficult. To get insights into the value of pooling, let us consider instead a system where both queues and servers are pooled, so that instead of k_i servers being available to class i , there is a single server that is k_i times faster. Therefore, the expected service time of customers of class i is $E(S_i)/k_i$. In the unpartitioned system, a single server that is n times faster would be available to all classes so that expected service time is $\sum_{i=1}^l p_i E(S_i)/n$. This system is clearly more efficient (i.e., yields lower expected time in the system) than a system where only the queues are pooled but the servers remain distinct. Therefore, if we are able

to show that the difference in performance between this system and the system with separate servers and queues is small, then this would also show that the difference between the system where only queues are pooled and the system with separate servers and queues is small.

For an unpartitioned system with both queue and server pooling, the expected time in the system for a customer regardless of his class is given by

$$\begin{aligned} E(W_{pooled}^u) &= \frac{1}{m} \sum_{i=1}^l p_i \frac{E(S_i)}{n} (1 + 2 + \dots + m) \\ &= \left(\frac{1}{2n} + \frac{m}{2n} \right) \sum_{i=1}^l p_i E(S_i), \end{aligned} \quad (10)$$

whereas for the partitioned system, the expected time in the system for a class i customer is given by

$$E(W_{i,pooled}^p(k_i)) = \left(\frac{1}{2k_i} + \frac{p_i m}{2k_i} \right) E(S_i). \quad (11)$$

It is easy to verify that the difference $\Delta^u \equiv E(W^u) - E(W_{pooled}^u) = E(S)/2(1 - 1/n) < E(S)/2$ where $E(S) = \sum_{i=1}^l p_i E(S_i)$, and that $\Delta^p \equiv E(W_i^p(k_i)) - E(W_{i,pooled}^p(k_i)) = E(S_i)/2(1 - 1/k_i) < E(S_i)/2$. Hence, for both the unpartitioned and partitioned systems, the reduction in expected time in the system due to pooling is less than half the expected service time. Given that the number of customers is large, this difference is rather insignificant. This can be further verified by noting that the ratios $E(W^u)/E(W_{pooled}^u)$ and $E(W_i^p(k_i))/E(W_{i,pooled}^p(k_i))$ are small when m is large and both approach 1 as m approaches infinity. These results provide support for using Expressions (1) and (2) as approximations for systems with pooling, whether this involves the pooling of queues only or both queues and servers.

It is important to note that server pooling in our context does not have the same impact observed in other queueing systems. This is because in the rush hour regime all customers arrive at once. In a pooled system, the *average* customer will have more customers ahead of him in the queue, but these customers will be processed at a faster rate. In a system without pooling, this same customer will have fewer customers ahead of him, but they will be processed at a slower rate. Because the number of customers in the queue and the service rate are scaled by the same factor, the net effect

is that there is practically no difference between the two systems when the total number of customers is large. This is not the case in a system where customers arrive over time and interarrival times are stochastic. The pooled system has a distinct advantage in that case, because it eliminates the inefficiency of having one queue with a large number of customers and one with very few or no customers.

7. Impact of Phased Customer Arrivals

We have so far assumed that customers arrive to the system all at once at the start of the rush hour period. However, in practice, even in rush hour, customers tend to arrive one at a time or in small groups with random interarrival times. What defines rush hour is the relatively short interarrival time between consecutive arrivals. In this section, we examine the impact of having customers arrive over time. We are interested in determining the extent to which the server allocation given by the vector \mathbf{k}^* continues to be effective in systems where customers arrive over time. Unfortunately, exact analysis for systems with phased arrivals and stochastic arrival interarrival times is difficult. Therefore, to obtain some insights, we present results based on (1) deterministic fluid approximations and (2) stochastic computer simulation. In both cases, we associate an arrival rate $\lambda_i = p_i \lambda$ with customers of type i , where $p_i = m_i/m$ and $\lambda > 0$ is the overall arrival rate to the system. This means that the expected time between consecutive customers of type i is $1/\lambda_i$ and that it takes, on average, an amount of time $m_i/\lambda_i = m/\lambda$ for all customers from each class to arrive. Note that the system is stable regardless of the intensity of the arrival rates because the number of arrivals is finite.

7.1. Fluid Approximation

Under a fluid approximation, we treat the arrival of customers of type i as the arrival of a fluid that occurs continuously with a constant rate λ_i . For the unpartitioned system, we assume that the arrivals from different customer classes are perfectly mixed so that each unit of arriving fluid contains a fraction p_i of fluid of type i . This arriving fluid is split among the queues of the n servers so that each queue receives fluid at rate λ/n . The fluid is pumped out of each

queue at a continuous rate $\mu = 1/\sum_{i=1}^n p_i E(S_i)$. Similarly, for the partitioned system, each server of type i receives fluid at a rate $p_i \lambda/k_i$, where k_i is the number of servers allocated to class i . In this case, the fluid at each queue is pumped out at a rate $\mu_i = 1/E(S_i)$. To avoid the trivial case where there is no fluid accumulation (and therefore no congestion) we assume that for both systems the input rate to each queue is higher than its output rate (i.e., $\mu < \lambda/n$ and $\mu_i < p_i \lambda/k_i$ for $i = 1, \dots, l$).

Given the above assumptions, it is relatively straightforward to show that the expected numbers of customers (the amount of fluid) at each queue in the partitioned and unpartitioned systems are given, respectively, by

$$E(\bar{N}^u) = \frac{m}{2n} \left(1 - \frac{\mu}{\lambda/n} \right) \quad (12)$$

and

$$E(\bar{N}_i^p(k_i)) = \frac{p_i m}{2k_i} \left(1 - \frac{\mu_i}{p_i \lambda/k_i} \right). \quad (13)$$

The expected time in the system for an arbitrary customer can then be obtained as

$$\begin{aligned} E(\bar{W}^u) &= \frac{m}{2n} \left(1 - \frac{\mu}{\lambda/n} \right) \sum_{i=1}^n p_i E(S_i) \\ &= \frac{m}{2n} \sum_{i=1}^n p_i E(S_i) - \frac{m}{2\lambda} \end{aligned} \quad (14)$$

for the unpartitioned system and

$$E(\bar{W}^p(\mathbf{k})) = \sum_{i=1}^l p_i E(\bar{W}_i^p(k_i)), \quad (15)$$

where

$$E(\bar{W}_i^p(k_i)) = \frac{p_i m}{2k_i} E(S_i) - \frac{m}{2\lambda}, \quad (16)$$

for the partitioned system.

We can see that the expected time in the system is increasing in the arrival rate λ . The limit case of $\lambda \rightarrow \infty$ corresponds to the instantaneous arrivals of all the customers. In that case, the expressions for the expected time in the system reduce to $E_\infty(\bar{W}^u) = (m/2n) \sum_{i=1}^n p_i E(S_i)$ and $E_\infty(\bar{W}_i^p(k_i)) = (p_i m/2k_i) E(S_i)$, respectively. These expressions converge asymptotically to those we obtained in Equations (1) and (2) as $m \rightarrow \infty$, with $\lim_{m \rightarrow \infty} E(W^u)/E_\infty(\bar{W}^u) = 1$ and $\lim_{m \rightarrow \infty} E(W_i^p(k_i))/E_\infty(\bar{W}_i^p(k_i)) = 1$. The limit of the

difference between the fluid approximation and the exact expression is also small, with $E(W^u) - E_\infty(\bar{W}^u) = 1/2\mu$ and $E(W_i^p(k_i)) - E_\infty(\bar{W}_i^p(k_i)) = 1/2\mu_i$. The fact that the difference is strictly positive is due to the fact that, in our original model, customers leave the system at discrete points in time, and not continuously as in the fluid model.

In general, the expressions for the expected time in the system in (2) and (15) differ only by a constant that is independent of the number of servers allocated to each class. Consequently, the allocation vector \mathbf{k}^* minimizes also $E(\bar{W}^p(\mathbf{k}))$. This allocation is feasible if the k_i^* 's are integer valued and satisfy the constraints $p_i\lambda > k_i^*/E(S_i)$ for $i = 1, \dots, l$. These constraints are equivalent to requiring that λ be sufficiently large, namely $\lambda > (n/\sqrt{E(S_i)})/\sum_{i=1}^l p_i\sqrt{E(S_i)}$ for $i = 1, \dots, l$.

These results provide support for the robustness of the allocation given by \mathbf{k}^* . They show that, although assuming instantaneous arrivals would lead to inaccurate estimates for the expected time in the system, the server allocation \mathbf{k}^* continues, nevertheless, to be optimal. These results are not entirely surprising given the similarities between our original setting and the setting described by the fluid model. In particular, in both cases, servers are continuously busy until all customers have been cleared.

7.2. Monte Carlo Simulation

In the simulation model, customers arrive over time with stochastic interarrival times that are independent and identically distributed. Upon arrival, a customer is routed to the shortest queue within his class. Customers are processed by the corresponding server when it becomes available with stochastic service times that are also independently and identically distributed. The results we present here are for systems with two customer classes. We denote by $\rho = \lambda(p_1E(S_1) + p_2E(S_2))/n$ the relative load that is placed on the system. As λ increases, so does ρ . When $\lambda \rightarrow \infty$, or, equivalently, $\rho \rightarrow \infty$, the system approaches the idealized rush hour regime where all arrivals occur at once.

We carry out simulation experiments over a wide range of system parameters. For each combination of parameters (a scenario), we first search for the optimal allocation, denoted by the vector \mathbf{k}^s , by simulating the system under all possible allocations and choosing the one that leads to the lowest expected time

in the system. We then compare the expected time in the system under the optimal allocation $E(\bar{W}^p(\mathbf{k}^s))$ to the expected time in the system under allocation \mathbf{k}^* $E(\bar{W}^p(\mathbf{k}^*))$. Note that because \mathbf{k}^* is not guaranteed to be an integer, we evaluate both the integer floor and integer ceiling of \mathbf{k}^* and choose the one that leads to the lower expected time in the system. For each scenario that we test, we carry out enough replications so that our estimate of the expected time in the system is within $\pm 2\%$ of the true expected value with 95% confidence. To measure the impact of using allocation \mathbf{k}^* instead of allocation \mathbf{k}^s , we use the percentage difference in performance defined as follows

$$\gamma = 100\% \frac{E(\bar{W}^p(\mathbf{k}^*)) - E(\bar{W}^p(\mathbf{k}^s))}{E(\bar{W}^p(\mathbf{k}^*))}$$

For each scenario, we simulate the system for different values of ρ (by progressively increasing λ from 0.1 in increments of 0.1 while keeping everything else fixed) until we reach a value of ρ for which $\gamma \leq 1\%$. In other words, we identify the minimum relative load under which using allocation vector \mathbf{k}^* leads to a performance difference γ that is less than 1%. We refer to this minimum relative load as ρ_{\min} .

Representative results are provided in Tables 1 and 2 (additional numerical results are available from the authors upon request). The values shown in the tables correspond to ρ_{\min} for different combinations of system parameters. The results are shown for a system with 25 servers, two customer classes, number of customers ranging from $m = 100$ to 5,000, and fraction of customers from each class ranging from $p_i = 0.1$ to 0.9. Service and interarrival times are drawn from a gamma distribution. The gamma distribution (of which the exponential distribution is a special case) allows us to vary the mean and variance independently. We consider scenarios where the mean of service times is varied from $E(S_1) = 0.2$ to 1.4 for class 1 and from $E(S_2) = 1.6$ to 2.8 for class 2 and the coefficient of variation (the ratio of the standard deviation to the mean) of service times for both classes from $C_{S_i} = 0.2$ to 2 for $i = 1, 2$. The coefficient of variation for interarrival times is also varied from $C_A = 0.2$ to 2. In varying the coefficients of variation, we change the variance but keep the mean constant. Note that in several cases the value of ρ_{\min} shown in the tables is 0.00. This means that a γ less than 1% is obtained

Table 1 Effect of Number of Customers, Customer Mix, and Service Time Difference on Minimum Relative Load, ρ_{\min} ($C_A = 1, C_{S_1} = C_{S_2} = 1$)

$E(S_1)/E(S_2)$	ρ_1/ρ_2								
	0.1/0.9	0.2/0.8	0.3/0.7	0.4/0.6	0.5/0.5	0.6/0.4	0.7/0.3	0.8/0.2	0.9/0.1
$m = 100$									
1.4/1.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.2/1.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.89
1.0/2.0	0.00	0.00	0.00	0.97	0.00	0.00	0.87	0.79	0.57
0.8/2.2	0.00	0.00	0.00	0.72	0.71	0.00	0.91	0.84	0.95
0.6/2.4	0.00	0.00	0.00	1.22	1.27	0.68	1.24	1.08	0.99
0.4/2.6	0.00	0.00	0.00	1.66	1.96	1.60	1.32	1.49	1.15
0.2/2.8	0.00	0.00	0.00	0.91	2.13	2.20	2.02	2.21	1.65
$m = 1,000$									
1.4/1.6	0.00	0.00	0.00	0.00	0.00	0.00	0.76	0.00	0.77
1.2/1.8	0.00	0.00	0.00	0.00	1.09	0.00	1.00	1.01	0.98
1.0/2.0	0.00	0.00	0.00	1.31	1.18	1.07	1.10	1.06	0.92
0.8/2.2	0.00	1.04	1.40	1.41	1.33	1.20	1.20	1.12	1.11
0.6/2.4	0.00	0.00	1.42	1.62	1.57	1.43	1.40	1.27	1.14
0.4/2.6	0.00	1.25	1.44	2.03	2.09	1.93	1.65	1.61	1.33
0.2/2.8	0.00	0.00	1.48	1.94	2.73	2.84	2.48	2.11	1.75
$m = 5,000$									
1.4/1.6	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.77
1.2/1.8	0.00	0.00	0.00	0.00	1.09	0.85	1.03	1.03	0.99
1.0/2.0	0.00	0.00	0.00	1.29	1.18	1.09	1.10	1.07	0.97
0.8/2.2	0.00	1.15	1.38	1.38	1.33	1.20	1.20	1.13	1.09
0.6/2.4	0.00	0.00	1.42	1.57	1.54	1.42	1.41	1.26	1.14
0.4/2.6	0.00	1.31	1.48	2.00	2.02	1.88	1.60	1.58	1.30
0.2/2.8	0.00	0.00	1.54	1.99	2.71	2.77	2.41	2.06	1.70

for all the values of λ tested. It also means that the value of ρ that corresponds to the smallest value of λ tested ($\lambda = 0.1$) is smaller than 0.005 and is therefore rounded down to 0.00.

Based on these results, the following observations can be made:

- In all the scenarios tested, the minimum relative load does not exceed 3.0. This means that, for systems where the arrival rate is sufficiently high so that the relative load is at least 3, the performance difference γ is less than 1%. For many of the cases observed, ρ_{\min} is significantly lower than 3.

- The value of ρ_{\min} is highest for systems where the difference in the means of service times is the most significant. It is also highest when the customers with the shortest service time represent a large fraction of the total number of customers. This is perhaps not surprising. We expect performance to be particularly sensitive to how servers are allocated when the customer classes are sufficiently different and when there are enough customers with short processing times.

- The value of ρ_{\min} is relatively invariant to the number of customers m . This seems consistent with our results for the rush hour setting in which the optimal allocation is independent of m .

- The value of ρ_{\min} is somewhat insensitive to changes in the coefficients of variation in service times and interarrival times. This may be due to the fact that variability has a second-order effect on performance compared with the effect of the arrival rate or the effect of differences in the mean processing times among the customer classes (the scenarios shown in Table 2 are for systems where these differences are significant).

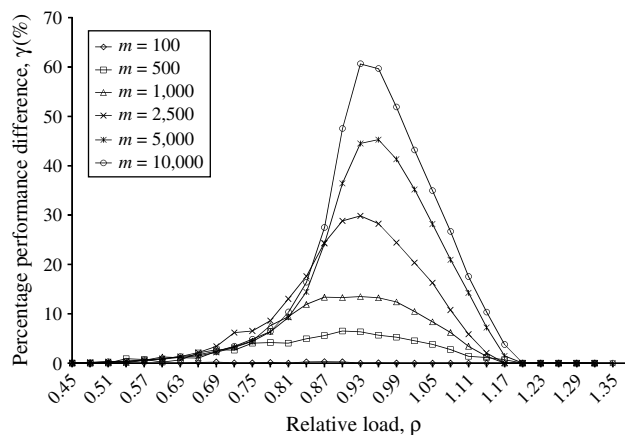
The results in Tables 1 and 2 show that for sufficiently high arrival rates, the allocation given by k^* can be quite effective. However, the results do not provide information on how effective k^* is for lower values of arrival rates. In Figure 2, we show representative results that illustrate how the percentage performance difference γ varies with ρ (results for other combinations of system parameters show

Table 2 Effect of Coefficients of Variation on Minimum Relative Load,
 $\rho_{\min} (C_{S_1} = C_{S_2} = C_S, \rho_1 = \rho_2 = 0.5, E(S_1) = 0.3,$
 $E(S_2) = 2.7)$

C_A	C_S									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
<i>m</i> = 100										
0.2	2.20	2.28	2.12	1.72	1.76	1.91	1.97	2.02	1.73	0.00
0.4	2.22	2.46	2.32	1.90	1.79	1.95	1.78	1.97	0.00	0.00
0.6	2.27	2.39	2.21	1.88	1.81	1.87	1.91	1.95	1.64	0.00
0.8	2.09	2.33	2.40	1.80	1.79	1.77	1.87	1.66	1.40	0.00
1.0	2.24	2.36	2.15	1.79	1.78	1.94	1.83	1.70	1.73	0.00
1.2	2.09	1.97	1.96	1.89	1.94	1.87	1.90	0.88	1.21	1.87
1.4	2.09	2.22	2.24	1.81	1.91	1.88	1.85	1.81	0.00	0.00
1.6	2.16	2.32	1.78	2.14	2.0	1.68	1.76	1.83	1.59	0.00
1.8	1.89	2.14	2.17	1.55	1.60	1.78	1.10	1.13	0.00	0.00
2.0	1.79	2.12	1.55	1.82	1.78	1.76	1.29	1.39	0.00	0.00
<i>m</i> = 1,000										
0.2	2.24	2.24	2.24	2.25	2.27	2.24	2.26	2.30	2.28	2.23
0.4	2.24	2.26	2.27	2.25	2.24	2.25	2.32	2.27	2.29	2.28
0.6	2.23	2.23	2.23	2.21	2.27	2.26	2.30	2.27	2.32	2.33
0.8	2.23	2.29	2.27	2.24	2.29	2.25	2.26	2.29	2.33	2.25
1.0	2.27	2.30	2.21	2.26	2.29	2.29	2.29	2.29	2.30	2.27
1.2	2.24	2.26	2.25	2.24	2.27	2.26	2.29	2.30	2.34	2.26
1.4	2.24	2.29	2.21	2.27	2.23	2.22	2.28	2.28	2.28	2.31
1.6	2.23	2.27	2.26	2.29	2.28	2.26	2.24	2.25	2.27	2.33
1.8	2.26	2.22	2.32	2.26	2.30	2.20	2.24	2.26	2.26	2.35
2.0	2.27	2.20	2.29	2.31	2.20	2.23	2.30	2.27	2.35	2.30
<i>m</i> = 5,000										
0.2	2.23	2.23	2.24	2.24	2.23	2.23	2.23	2.24	2.21	2.21
0.4	2.23	2.21	2.23	2.23	2.23	2.24	2.24	2.24	2.27	2.26
0.6	2.22	2.23	2.24	2.23	2.21	2.24	2.24	2.23	2.24	2.25
0.8	2.22	2.24	2.23	2.23	2.24	2.21	2.24	2.21	2.27	2.20
1.0	2.22	2.24	2.24	2.24	2.24	2.23	2.24	2.27	2.23	2.25
1.2	2.21	2.22	2.23	2.25	2.23	2.22	2.23	2.25	2.19	2.21
1.4	2.24	2.22	2.22	2.24	2.22	2.24	2.22	2.26	2.20	2.24
1.6	2.23	2.22	2.21	2.26	2.23	2.24	2.25	2.23	2.21	2.22
1.8	2.25	2.23	2.26	2.23	2.21	2.25	2.26	2.26	2.23	2.24
2.0	2.25	2.23	2.24	2.23	2.23	2.26	2.26	2.24	2.28	2.25

similar effects). As we can see, the allocation \mathbf{k}^* performs well when ρ is small (less than 0.5 in the examples shown) but can perform poorly for values of ρ in the middle range (from approximately 0.5 to 1 in the examples shown). These results can be explained as follows. When ρ is small, the manner in which servers are allocated does not matter significantly because there is excess capacity in the system and queue sizes are always small. In contrast, when ρ is in the middle range, there is queueing but the system is far from the rush regime of instantaneous arrivals. This is particularly so when m is large, in which case

Figure 2 Percentage Performance Difference Due to Using Optimal Allocation \mathbf{k}^* , Varying m



the system behaves more like a traditional queueing system.

In summary, the results from the simulation suggest that the server allocation obtained for systems under the rush hour assumption can be used effectively in settings where customers arrive over time with stochastic interarrival times. However, the arrival rate has to be sufficiently high. Otherwise, this allocation could lead to poor performance.

8. Conclusions

We presented a model for studying the partitioning of servers during a rush hour demand regime. We set out to answer three basic questions. (1) Is partitioning beneficial to the system? (2) Is it equally beneficial to all customer classes? (3) If it is implemented, what is an optimal partition? Using the expected time in the system as our performance criterion, we found that partitioning can indeed be beneficial to the system and this benefit can be significant. However, we also found that this benefit is realized only at the expense of one or more customer classes. In fact, we showed that it is impossible for all customer classes to benefit from partitioning. We showed that there is an optimal way to partition servers and provided, via simple closed-form expressions, a characterization of the optimal partitioning. We found that this partitioning can also be useful in settings where customers arrive over time if the effective arrival rate to each queue is sufficiently high.

In this paper, we used the expected time in the system as our criterion for performance evaluation. There may be settings where other criteria are more appropriate. For example, the time until all customers are cleared (the maximum completion time) is a useful measure in situations where the cost of operating the service facility depends on when the last customer leaves. Minimizing the variance of time in the system may be appropriate in settings where fairness in treating the different customer classes is a concern. Alternatively, there may be situations where multiple criteria are applicable or there are certain service-level requirements with respect to certain criteria that must be guaranteed. In general, we expect different criteria to lead to different server allocations. For instance, the optimal partitioning discussed in this paper tends to favor customers with short service times. Therefore, there may be significant variance in the time in the system across different customer classes. A useful future research direction would be to investigate when optimizing with respect to one criterion may be particularly detrimental with respect to other criteria.

In this paper, we assumed that customer classes are exogenously determined. A potential future extension of our model is to consider jointly the partitioning of servers among customer classes as well as the classification of customers into classes. This would entail jointly determining the number of customer classes, the range of service times associated with each customer class, and then the allocation of servers to classes. For example, in a supermarket environment, this would mean determining whether or not to have express lanes (i.e., dedicated lanes based on the number of items purchased by customers), the range

of number of items to allow in each type of lane, and the number of lanes to have of each type.

References

- Altman, E., T. Jimenez, G. Koole. 2001. Optimal call admission control in a resource-sharing system. *IEEE Trans. Comm.* **49** 1659–1668.
- Avram, F., D. Bertsimas, M. Ricard. 1995. Fluid models of sequencing problems in open queueing networks: An optimal control approach. F. Kelly, R. Williams, eds. *Stochastic Networks, Proceedings of the IMA*, Vol. 71. Springer, New York, 199–234.
- Benjaafar, S. 1995. Performance bounds for the effectiveness of pooling in multi-processing systems. *Eur. J. Oper. Res.* **87** 375–388.
- Benjaafar, S., W. L. Cooper, J. S. Kim. 2005. On the benefits of pooling in production-inventory systems. *Management Sci.* **51** 548–565.
- Hall, R. W. 1991. *Queueing Methods for Services and Manufacturing*. Prentice Hall, Upper Saddle River, NJ.
- Kleinrock, L. 1976. *Queueing Systems, Volume 2: Computer Applications*. John Wiley & Sons, New York.
- Kleinrock, L. 2002. Creating a mathematical theory of computer networks. *Oper. Res.* **50** 125–131.
- Newell, G. F. 1982. *Applications of Queueing Theory*. Chapman-Hall, London.
- Pinedo, M. 2002. *Scheduling: Theory, Algorithms, and Systems*, 2nd ed. Prentice Hall, Englewood Cliffs, NJ.
- Ross, K. W., D. D. Yao. 1990. Monotonicity properties for the stochastic knapsack. *IEEE Trans. Inform. Theory* **36** 1173–1179.
- Rothkopf, M. H., P. Rech. 1987. Perspectives on queues: Combining queues is not always beneficial. *Oper. Res.* **35** 906–909.
- Savin, S., B. Wang. 2006. Capacity allocation in rental businesses with reservations. Working paper, Columbia University, New York.
- Smith, D. R., W. Whitt. 1981. Resource sharing for efficiency in traffic systems. *Bell System Tech. J.* **60** 39–55.
- Whitt, W. 1992. Understanding the efficiency of multi-server service systems. *Management Sci.* **38** 708–723.
- Whitt, W. 1999. Partitioning customers into service groups. *Management Sci.* **45** 1579–1592.
- Whitt, W. 2006. Fluid models for multiserver queues with abandonments. *Oper. Res.* **54** 37–54.