Design and Evaluation of Scheduling Policies for Two Server Fork/Join Queueing Systems

Don Towsley, Shenze Chen
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

COINS Technical Report 91-39
April 1991

Design and Evaluation of Scheduling Policies for Two Server Fork/Join Queueing Systems *

Don Towsley Shenze Chen

Department of Computer & Information Science
University of Massachusetts
Amherst, MA 01003

Abstract

We consider a two server system that processes a mixture of regular customers and fork/join customers. A regular customer is one that requires service at one server whereas a fork/join customer requires service at both servers. We consider two classes of policies for scheduling regular customers to the servers. The first class consists of *distributed policies*, where each policy therein maintains two queues, one for each server. Policies in this class differ from each other according to the rules by which regular customers are routed to one or both of the two queues. The second class consists of *centralized policies*. Here a centralized policy maintains a common queue which feeds customers to both servers. We develop Markovian models for each of the policies that provides upper and lower bounds on their performance. These bounds are evaluated using matrix geometric techniques and can be made as tight as desired at the cost of additional computation. Using these models, we find the best policy to be a distributed policy called DP-MR, which allocates copies of a regular customer to each queue and as soon as one completes removes the other. However, in the case that the system cannot support a common queue or provide status information regarding each server, an appropriate policy is the distributed DP-RJ. Last, the method of evaluation is of independent interest as it may be applicable to a number of problems regarding the analysis of computer systems.

*This work is supported, in part, by the Office of Naval Research under contract N00014-87-K-0796.

1 Introduction

We consider a queueing system consisting of two servers that process two classes of customers. The first class consists of *regular* customers that can be processed at either of the two servers. The second class consists of *fork/join* customers that require service from both servers. A fork/join customer generates (forks) two tasks, one for each server. After completion of service, (i.e., completion of both tasks), the customer departs.

Our interest in this queueing system is motivated primarily by the following analysis problem in computer systems. Numerous fault tolerant disk I/O systems (see [3] for an example) require that two copies of each data item be maintained. As a consequence, whenever the data item is updated, both copies must be modified. On the other hand, a request to read a data item can be satisfied by either copy. This system maps into the model of interest to us where the update requests correspond to fork/join customers and read requests correspond to regular customers. Other applications are found in the area of parallel processing where two processors serve a mix of serial and parallel programs.

We propose a number of different policies for scheduling regular customers in this system. These policies fall into two classes.

- *Distributed Policies (DP).* A distributed policy maintains two separate queues, one for each server. Distributed policies differ from each other according to the rules used to route regular customers to these two queues.
- *Centralized Policies (CP).* A centralized policy maintains a common queue for both servers. Although centralized policies appear to require only one queue, we shall observe that a second queue is required for the server that lags behind while processing fork/join tasks. This second queue called an *auxiliary queue* contains only fork/join tasks except for possibly the request in service.

In all cases, queues are served in a first come first serve (FCFS) manner. The performance metrics of most interest are the stationary response times for regular and fork/join customers.

The primary contribution of this paper is the analysis and comparison of a variety of distributed and centralized policies. We develop tight bounds on the response time distribution of fork/join and regular customers under these policies for the assumptions of Poisson arrivals and exponential service times. This is accomplished by judiciously truncating one of the state variables in the

Markov chain that underlies each of these policies and applying the matrix geometric approach, [12] to obtain the bounds. The presence of these bounds provides an analyst the capability of approximating the statistics of the response times of fork/join and regular customers to the degree of accuracy required by changing the truncation parameter. For all these policies an increase in accuracy is obtained with an increase in computational cost.

Using this approach we compare the performance of these different policies and find that centralized policies outperform most of distributed policies, but are outperformed by the distributed DP-MR policy, which allocates copies of a regular customer to each queue and as soon as one completes removes the other, given the service time is exponential. Clearly, if regular customers are given a higher priority than fork/join customers, the overall system mean response time can be further improved [1]. In this study, however, we assume both regular and fork/join customers to have the same priority. Our method of obtaining the performance bounds may also be of independent interest. It is likely to be applicable to a large variety of analysis problems such as those considered in [6, 10].

Queueing systems with fork/join customers have only recently received attention. Flatto and Hahn [5] performed an exact analysis of a system with two exponential servers that process only fork/join customers. Fork/join customers arrive according to a Poisson process and the complete system is modeled by a Markov chain containing two state variables each of which is unbounded. They obtain the stationary probability distribution for this chain by transforming the problem into a boundary value problem. Rao and Posner, [13], performed an approximate analysis of this model by truncating one of the two state variables and solving the resulting model using the matrix-geometric methodology developed by Neuts [12]. We will show in the course of this paper that their approximation to the response time distribution for fork/join customers provides a lower bound on the correct distribution. In [8], Kim and Agrawala presented an approach to obtain approximate solutions for a K-server fork/join queueing system, in which response time of fork/join customers is obtained by tracking the *virtual waiting time* of each queue, i.e., time to empty a server when no more arrivals occur. With the restriction of service time being Erlangian distribution, their formulation results in a form of series, the worth of which is in actually computing numerical results.

Using a different approach, Nelson and Tantawi [11] have developed an accurate approximation for the case of K identical servers. Baccelli, Makowski, and Shwartz [4] developed simple (but loose) computational bounds for a K server system that processes several classes of customers that differ from each other according to the subset of servers required. This approach can be used to model one of the distributed policies, termed the DP-RJ policy, where regular customers are probabilistically routed to the two servers. However, with the exception of the approach used by Rao and Posner [13], none of the approaches appear to generalize easily to policies other than the aforementioned DP-RJ policy.

We point out that, although the truncated Markov chain for the DP-RJ policy is similar to that studied by Rao and Posner [13], they were apparently unaware that their approximation could be used to provide lower bound on the response time distribution in the system using that policy.

Other work relevant to one of the distributed policies, DP-SQ, can be found in [14], where Rao and Posner presented an approximate analysis of the shortest queue model. However, only regular customers are included in their discussions.

The remainder of the paper is structured in the following way. Section 2 contains the description of the various policies discussed in this paper. The analytic models for these policies are given in Section 3. Section 4 compares performance of these different policies. A summary of the paper is contained in Section 5.

2 Description of Policies

In this section, we describe various scheduling policies for the two server fork/join queueing system. We divide all of these policies into two classes, *distributed policies (DP)* and *centralized policies (CP)*. In all cases, customers are served in a first come first service manner within each queue.

2.1 Distributed Policies

In the class of distributed policies, two queues are maintained by the system, one for each server. At the time of arrival, a fork/join customer generates two tasks that enter each of the two queues. The various policies differ from each other by the way that they decide which server to send the regular customer.

- **The Random Join (DP-RJ) Policy**
  With this policy, a regular customer, at the time of its arrival, randomly chooses the queue associated with server i to enter with probability α_i, $i = 1, 2$ $(\alpha_1 + \alpha_2 = 1)$.
- **The Shortest Queue (DP-SQ) Policy**
  Under the DP-SQ policy, a regular customer selects the server with the shortest queue at the

time of its arrival.

- **The Shortest Queue with Minimum Service (DP-SQ-MS) Policy**
  The DP-SQ-MS policy behaves the same as the DP-SQ policy, except when a regular customer arrives to an empty system. In this case, under the DP-SQ policy, the regular customer randomly chooses a server for service, whereas under the DP-SQ-MS policy, it generates two tasks, one for each server, which begin service immediately. The regular customer completes as soon as the first task finishes, and the other task is aborted. However, if there are any new arrivals before the regular customer completes, one of the tasks (randomly chosen) is aborted and the freed server starts to service the new arrival. Obviously, the DP-SQ-MS policy can take advantage of the minimum service time for those regular customers who find both servers idle. The performance improvement by adopting this strategy is expected to be noticeable when the system is lightly loaded.

- **The Minimum Waiting (DP-MW) policy**
  Under the DP-MW policy, each regular customer generates two tasks, one for each queue. Whenever any one of these tasks begins service, its counterpart is immediately removed from the other queue.

- **The Minimum Waiting with Minimum Service (DP-MW-MS) Policy**
  The DP-MW-MS policy is defined in a similar way as the DP-SQ-MS policy to achieve a better performance for a light system. Namely, if a regular customer arrives to find both servers idle, it begins service on both servers. It departs the system as soon as the first of the two servers completes. If additional customers arrive before either task of the regular customer completes, one of its tasks is aborted.

- **The Minimum Response (DP-MR) Policy**
  The DP-MR policy is similar to the DP-MW policy which allows regular customers to enter both queues upon arrival. However, the DP-MR policy aborts one of the tasks associated with a regular customer once its peer completes its service, whereas the DP-MW policy aborts a task once its peer begins its service.

Remark: Among all these policies, the DP-RJ policy is suitable for systems where the two servers are physically located at two distant sites, since it requires no information interchange between the two servers. The other policies, however, are more suitable for systems where the two servers are located close to each other because those policies have to know the status of each queue in order to scheduling regular customers.

2.2 Centralized Policies

Policies in this class maintain a common queue for the two servers. Both regular and fork/join customers enter the common queue at the time of their arrival.

- **The Auxiliary Queue (CP-AQ) Policy**
  Under the CP-AQ policy, a customer at the front of the common queue begins service as soon as a server becomes available. If the customer at the head of the common queue is a fork/join customer, it also places the second task in an auxiliary queue associated with the other server. Thus the system requires an auxiliary queue associated with the server which lags behind. This auxiliary queue contains fork/join customers only (not including the customer in service). Both the common queue and the auxiliary queue are served in a FCFS manner.

- **The Auxiliary Queue with Minimum Service (CP-AQ-MS) Policy**
  The CP-AQ-MS policy is similarly defined as the CP-AQ policy except that a regular customer is allowed to begin service at both servers whenever they are both idle. This policy behaves in the same manner as the DP-SQ-MS policy regarding these two tasks.

3 Analytic Models

In order to analyze the policies described in the previous section, we assume that regular and fork/join customers arrive according to Poisson processes with rates λ and γ, respectively. Service times at server i are assumed to be exponentially distributed random variables with mean $1/\mu_i, i = 1, 2$. In most cases we will assume that $\mu_1 = \mu_2$. We will explicitly state when this is not so.

Under these assumptions, all of the policies can be modeled by multi-dimensional Markov chains. We will describe how these Markov chains can be manipulated in order to obtain computable bounds on the performance of each policy. In all cases, the resulting Markov chain can be solved through the use of Neut's matrix-geometric method [12]. In this section we will describe the analysis of the DP-RJ and CP-AQ policies in some detail. The analyses of the other policies are then briefly described.

3.1 An Analysis of the DP-RJ Policy

We allow μ_1 to differ from μ_2. Under this policy regular customers select queue i with probability α_i, $i = 1, 2$ with $\alpha_1 + \alpha_2 = 1$.

Let $W_i^{(r)}$ and $W_i^{(f)}$ denote the response times of the i-th regular and fork/join customers respectively. Let $\mathbf{T}_i = (T_{i,1}, T_{i,2})$ where $T_{i,j}$ denotes the response time of the task generated by the i-th fork/join customer that enters queue j, $j = 1, 2$. Here $T_{i,j}$ can be expressed as

$$T_{i,j} = U_{i,j} + X_{i,j} \tag{1}$$

where $U_{i,j}$ is the unfinished work in the queue at the time that the i-th fork/join customer arrives and $X_{i,j}$ is the service time for the task that it generates, $j = 1, 2$. Last, let $\hat{\mathbf{T}}_i = (\hat{T}_{i,1}, \hat{T}_{i,2})$ where $\hat{T}_{i,1} = \min\{T_{i,1}, T_{i,2}\}$ and $\hat{T}_{i,2} = \max\{T_{i,1}, T_{i,2}\}$. The response time of the i-th fork/join customer can be expressed as $W_i^{(f)} = \hat{T}_{i,2}$.

We are interested in the limiting random variables for the above defined random variables when they exist. We shall drop the subscript i when referring to these limiting random variables, i.e., $W^{(r)} = \lim_{i\to\infty} W_i^{(r)}$. We are also interested in the random variables $N^{(r)}$ and $N^{(f)}$ that respectively denote the stationary number of regular customers and fork/join customers in the system.

We first observe that each queue and server can be separately modeled as an M/M/1 system. As a consequence, the system exhibits stationary behavior so long as $\alpha_i\lambda + \gamma < \mu_i$, $i = 1, 2$. Since the response time of a regular customer is affected only by the queue that it enters, the distribution of $W^{(r)}$ is given by a weighted sum of the distributions of two independent M/M/1 systems

$$P[W^{(r)} > w] = \alpha_1 e^{-(\mu_1 - \alpha_1\lambda - \gamma)w} + \alpha_2 e^{-(\mu_2 - \alpha_2\lambda - \gamma)w}$$

with mean

$$E[W^{(r)}] = \alpha_1/(\mu_1 - \alpha_1\lambda - \gamma) + \alpha_2/(\mu_2 - \alpha_2\lambda - \gamma).$$

The expected number of regular customers in the system, $E[N^{(r)}]$, can be obtained through an application of Little's rule [9]. Consequently we focus only on the behavior of fork/join customers.

As a further consequence of the fact that each queue behaves as an M/M/1 system, we can write the following expressions for the *marginal* distributions of the response times of the two tasks associated with a fork/join customer

$$P[T_j > t] = e^{-(\mu_j - \alpha_j\lambda - \gamma)t}, \; j = 1, 2.$$

with means

$$E[T_j] = 1/(\mu_j - \alpha_j\lambda - \gamma), \; j = 1, 2.$$

Let us now conduct the following experiment; select a random fork/join customer. Select one of the two tasks associated with this customer with equal probability. Denote the response time of this task by T. Then T has the following distribution,

$$P[T > t] = (P[T_1 > t] + P[T_2 > t])/2.$$

This randomly chosen task is equally likely to be the first or the last of the tasks associated with the customer to complete. Consequently we also have the following identity,

$$P[T > t] = (F_{min}(t) + F_{max}(t))/2 \tag{2}$$

where $F_{min}(t) = P[\hat{T}_1 > t]$, and $F_{max}(t) = P[\hat{T}_2 > t]$. If we are able to obtain the marginal distribution for either $\hat{T}_1$ or $\hat{T}_2$, the above identity allows us to obtain the marginal distribution for the other random variable.

Equation (2) is also useful for obtaining bounds. For example, let us assume that we have upper bounds $F^{(ub)}_{min}(t)$ and $F^{(ub)}_{max}(t)$ to $F_{min}(t)$ and $F_{max}(t)$, i.e.,

$$\begin{aligned} F^{(ub)}_{min}(t) &\geq F_{min}(t), \ t \geq 0, \\ F^{(ub)}_{max}(t) &\geq F_{max}(t), \ t \geq 0 \end{aligned}$$

Then equation (2) can be used to obtain the following lower bounds for $F_{max}(t)$ and $F_{min}(t)$

$$\begin{aligned} F_{max}(t) &\geq 2P[T > t] - F^{(ub)}_{min}(t), \\ F_{min}(t) &\geq 2P[T > t] - F^{(ub)}_{max}(t), \ t \geq 0 \end{aligned} \tag{3}$$

In a similar manner, if we have expressions $F^{(lb)}_{min}(t)$ and $F^{(lb)}_{max}(t)$ that bound $F_{min}(t)$ and $F_{max}(t)$ from below, then equation (2) allows us to obtain the following upper bounds on the last two distributions

$$\begin{aligned} F_{min}(t) &\leq 2P[T > t] - F^{(lb)}_{max}(t), \\ F_{max}(t) &\leq 2P[T > t] - F^{(lb)}_{min}(t), \ t \geq 0 \end{aligned} \tag{4}$$

We shall make use of these relationships in order to obtain bounds on the statistics of the response time of a fork/join customer.

The DP-RJ policy can be modeled as a Markov chain with state $\mathrm{N}(t) = (N_1(t), N_2(t))$ where $N_1(t)$ and $N_2(t)$ are the number of tasks in the queues associated with servers 1 and 2 respectively at time t. Let $q(i,j) = \lim_{t\to\infty} P[N_1(t) = i, \ N_2(t) = j]$. The stationary probabilities satisfy the following equations,

$$\begin{array}{rcll}
(\gamma+\lambda)q(0,0) & = & \mu_1 q(1,0)+\mu_2 q(0,1), & \\
(\gamma+\lambda+\mu_1))q(i,0) & = & \lambda\alpha_1 q(i-1,0)+\mu_1 q(i+1,0)+\mu_2 q(i,1), & i=1,\cdots, \\
(\gamma+\lambda+\mu_2)q(0,j) & = & \lambda\alpha_2 q(0,j-1)+\mu_2 q(0,j+1)+\mu_1 q(1,j), & j=1,\cdots, \\
(\gamma+\lambda+\mu_1+\mu_2)q(i,j) & = & \gamma q(i-1,j-1)+\lambda\alpha_1 q(i-1,j) & \\
 & & +\lambda\alpha_2 q(i,j-1)+\mu_1 q(i+1,j)+\mu_2 q(i,j+1), & i=1,\cdots;\ j=1,\cdots.
\end{array}$$

Unfortunately, this model is not amenable to a simple analysis. We focus instead on a modified system in which the second queue (associated with server 2) can hold no more than B tasks. Whenever a fork/join customer arrives to the system at time t and finds $N_2(t)=B$, he generates a *single* task that enters the first queue. The customer completes when this task completes. Similarly, a regular customer that arrives to a full queue at the second server passes through without delay.

This modified system can be modeled as a Markov chain with the same state definition. In order to distinguish the modified system from the true system, we shall use the superscript (lb), i.e., $\mathbf{N}^{(lb)}(t)$ instead of $\mathbf{N}(t)$. We define $\mathbf{T}_i^{(lb)}$ according to equation (1) even though this does not produce the correct response time at the second queue.[1] We shall describe an ordering relationship between $\mathbf{T}$ and $\mathbf{T}^{(lb)}$. We first introduce the following definition [15].

Definition 1 *Let* $\mathbf{X}=(X_1,X_2,\cdots,X_n)$ *and* $\mathbf{Y}=(Y_1,Y_2,\cdots,Y_n)$ *be two real valued vector random variables. We say that* $\mathbf{X}$ *is stochastically larger than* $\mathbf{Y}$ $(\mathbf{X}\geq_{st}\mathbf{Y})$ *if for all increasing functions* f

$$E[f(\mathbf{X})]\geq E[f(\mathbf{Y})].$$

In the case that $n=1$, this is equivalent to

$$P[X>a]\geq P[Y>a], \forall\, a.$$

We will make use of the following property of the above stochastic ordering.

Property 1 *Let* $\mathbf{X}=(X_1,X_2,\cdots,X_n)$, $\mathbf{Y}=(Y_1,Y_2,\cdots,Y_n)$, *and* $\mathbf{X}\leq_{st}\mathbf{Y}$. *Then*

$$g(X_1,\cdots,X_n)\leq_{st} g(Y_1,\cdots,Y_n),\ \forall\ \textit{ increasing }\ g.$$

In particular, $\max(X_1,\cdots,X_n)\leq_{st}\max(Y_1,\cdots,Y_n)$.

We now state and prove the following theorem.

[1] This is because we have defined the response time at queue 2 to be 0 when the queue length is B at the time a customer arrives. Equation (1) produces a non-zero response time for that event.

Theorem 1 *The following relationships hold between the real system and the modified system.*

1. $\mathbf{N} \geq_{st} \mathbf{N}^{(lb)}$, .

2. $\mathbf{T} \geq_{st} \mathbf{T}^{(lb)}$.

Proof. In order to prove 1), it is useful to study the queue lengths of the system prior to the arrival of a customer of either class. Let $\mathbf{M}_i^{(lb)} = (M_{i,1}^{(lb)}, M_{i,2}^{(lb)})$ and $\mathbf{M}_i = (M_{i,1}, M_{i,2})$ where $M_{i,j}$ and $M_{i,j}^{(lb)}$ are the numbers in queue j ($j = 1, 2$) prior to the arrival of customer i in the real system and the modified system respectively. These r.v.'s satisfy the following recurrences,

$$\begin{aligned} M_{i+1,j} &= (M_{i,j} + I_{i,j} - D_{i,j})^+, \quad j = 1, 2, \\ M_{i+1,1}^{(lb)} &= M_{i+1,1}, \\ M_{i+1,2}^{(lb)} &= (M_{i,2}^{(lb)} + A_i I_{i,2} - D_{i,2})^+ \end{aligned}$$

where

$$I_{i,j} = \begin{cases} 0 & i\text{-th customer does not generate a task for } j\text{-th queue} \\ 1 & \text{otherwise} \end{cases}$$

$$A_i = \begin{cases} 0 & \text{queue 2 is full at time of arrival of customer } i \\ 1 & \text{otherwise} \end{cases}$$

and $D_{i,j}$ is the number of departures from queue j between the arrivals of the i-th and $i+1$-th customer.

If the initial state vectors of the two systems satisfy $\mathbf{M}_0 \geq \mathbf{M}_0^{(lb)}$, then an induction argument can be used to show $\mathbf{M}_i \geq \mathbf{M}_i^{(lb)}$ for $i = 0, 1, 2, \cdots$.[2] Consequently we conclude that $\mathbf{M}_i \geq_{st} \mathbf{M}_i^{(lb)}$ for $i = 0, 1, 2, \cdots$. Whenever the real system is ergodic, i.e., the r.v.'s $\mathbf{M}_i$ and $\mathbf{M}_i^{(lb)}$ converge to the limiting r.v.'s $\mathbf{M}$ and $\mathbf{M}^{(lb)}$, then $\mathbf{M} \geq_{st} \mathbf{M}^{(lb)}$. Finally, since arrivals from a Poisson process see time averages[7, sec. 11-2], $\mathbf{M}$ and $\mathbf{M}^{(lb)}$ have the same joint distribution as $\mathbf{N}$ and $\mathbf{N}^{(lb)}$ and we conclude that $\mathbf{N} \geq_{st} \mathbf{N}^{(lb)}$.

The second part of the theorem is shown in a similar manner by focusing on the Lindley equations that must be satisfied by the unfinished work in the system at the time of customer arrivals. ∎

[2] Here two vectors $\mathbf{V} = (v_1, \cdots, v_n)$ and $\mathbf{V}' = (v_1', \cdots, v_n')$ satisfy the relation $\mathbf{V} \geq \mathbf{V}'$ iff $v_i \geq v_i'$, $1 \leq i \leq n$.

Remark. It is possible to show that $\mathbf{N}(t) \geq_{st} \mathbf{N}^{(lb)}(t)$ for $t \geq 0$ where $\mathbf{N}(t)$ and $\mathbf{N}^{(lb)}(t)$ are the queue lengths of the two systems at time $t \geq 0$, and that $\mathbf{T}_i \geq_{st} \mathbf{T}_i^{(lb)}$ for $i = 1, 2, \cdots$ for any arrival process and i.i.d sequences of service times at each queue.

The modified lower bound system is solved by using the matrix-geometric method (see Appendix A), which parallels the analysis given by Rao and Posner [13]. They derive the following expression for the joint distribution of $\mathbf{T}^{(lb)}$

$$\begin{aligned} T^{(lb)}(w_1, w_2) &\equiv P[T_1^{(lb)} \leq w_1, T_2^{(lb)} \leq w_2], \\ &= \pi[I - \exp(-\mu_1(I - R)w_1)]B(w_2) \end{aligned}$$

where I is the identity matrix, π is a $B+1$ element vector containing the stationary queue length distribution for the M/M/1/B queue with arrival rate $\gamma + \alpha_2\lambda$ and service rate μ_2, i.e., the i-th element of π is $(1-u)u^i/(1-u^{(B+1)})$ where $u = (\alpha_2\lambda + \gamma)/\mu_2$, $B(w_2)$ is a $(B+1)$ column vector with i-th component $\left[1 - \sum_{r=0}^{i} (\mu_2 w_2)^r / r! \, \exp(-\mu_2 w_2)\right]$, and R is the solution of a quadratic matrix equation given in Appendix A. An application of Theorem 1 along with Property 1 of stochastic dominance yields the following bound on $F_{max}(w)$,

$$\begin{aligned} F_{max}(w) &\geq F_{max}^{(lb)}(w), \\ &= 1 - T^{(lb)}(w, w), \\ &= 1 - \pi[I - \exp(-\mu_1(I - R)w))]B(w). \end{aligned}$$

Similar arguments can be used to obtain the following bound on the distribution of the time until the first of the two tasks associated with a fork/join customer complete,

$$\begin{aligned} F_{min}(w) &\geq F_{min}^{(lb)}(w), \\ &= \pi \exp(-\mu_1(I - R)w - \mu_2 w)C(w) \end{aligned}$$

where $C(w)$ is a $(B+1)$ element column vector with ordered components $\sum_{j=0}^{i} (\mu_2 w)^j / j!$, $i = 0, 1, \cdots, B$. Substitution of $F_{min}^{(lb)}(w)$ into equation (4) yields the following upper bound on $F_{max}(w)$,

$$\begin{aligned} F_{max}(w) \leq\ & \exp(-(\mu_1 - \alpha_1\lambda - \gamma)w) + \exp(-(\mu_2 - \alpha_2\lambda - \gamma)w) \\ & - \pi \exp(-\mu_1(I - R)w - \mu_2 w)C(w). \end{aligned}$$

These can be used to obtain bounds on the moments of the response time.

Table 1 gives bounds on the average response time of a fork/join customer for different values of B. In this example, no regular customers enter the system and both servers are identical. If we take

ρ	$B = 4$		$B = 8$		$B = 16$		$B = 32$	
	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.
.1	1.653	1.653	-	-	-	-	-	-
.2	1.843	1.844	1.844	1.844	-	-	-	-
.3	2.082	2.092	2.089	2.089	-	-	-	-
.4	2.380	2.431	2.415	2.417	2.417	2.417	-	-
.5	2.764	2.925	2.861	2.879	2.875	2.875	-	-
.6	3.287	3.709	3.494	3.586	3.560	3.563	3.562	3.562
.7	4.093	5.103	4.444	4.823	4.676	4.716	4.708	4.708
.8	5.651	8.088	6.122	7.518	6.711	7.102	6.982	7.003
.9	10.37	17.63	10.81	16.55	11.73	15.19	13.05	14.17

Table 1: Bounds for DP-RJ under different values of B, $\mu_1 = \mu_2 = 1$.

ρ	$\mu_1/\mu_2 = 1$	$\mu_1/\mu_2 = 2/3$	$\mu_1/\mu_2 = 1/2$	$\mu_1/\mu_2 = 1/3$
0.1	1.65	1.38	1.28	1.19
0.2	1.84	1.53	1.41	1.32
0.3	2.09	1.71	1.58	1.50
0.4	2.42	1.95	1.81	1.73
0.5	2.88	2.28	2.14	2.06
0.6	3.56	2.77	2.62	2.55
0.7	4.71	3.58	3.44	3.37
0.8	6.99±.01	5.21	5.08	5.03
0.9	13.61±.56	10.12	10.02	10.01

Table 2: Bounds for DP-RJ under different values of μ_1/μ_2.

the average of the bounds for an approximation, the error is less than 3% for $\rho \leq .8$ when $B = 16$. An error of less than 5% can be achieved for $\rho = 0.9$ by taking $B = 32$. Table 3.1 gives bounds on the average response time of a fork/join customer for different values of μ_1/μ_2 when $\mu_1 = 1$. Here, the value $B = 32$ was used. For values of $\mu_2 > 1.5\mu_1$, the resulting bounds are identical to at least three decimal places. As expected the average response time is a decreasing function of μ_2. Table 3 presents bounds for the average response time of a fork/join customer in a system with identical servers that also serves regular customers. Here we observe that for a fixed server load, the average response time of a fork/join customer increases as the fraction of regular customers increases. This is because the coupling of the arrival processes to the two queues decreases as there are fewer fork/join customers. Again B was chosen to be 32.

ρ	$p_r = 0$	$p_r = 1/4$	$p_r = 1/2$	$p_r = 3/4$	$p_r = 1$
0.1	1.65	1.65	1.66	1.66	1.67
0.2	1.84	1.85	1.85	1.86	1.88
0.3	2.08	2.10	2.11	2.12	2.14
0.4	2.42	2.43	2.45	2.47	2.50
0.5	2.88	2.89	2.92	2.95	3.00
0.6	3.56	3.59	3.63	3.68	3.75
0.7	4.71	4.76	4.82	4.89	5.00
0.8	6.99±.01	7.08±.01	7.18±.01	7.31±.01	7.50
0.9	13.61±.56	13.77±.55	13.98±.55	14.24±.55	15.00

Table 3: Bounds for DP-RJ under different p_r, $\mu_1 = \mu_2 = 1$.

3.2 An Analysis of the CP-AQ Policy

In this model, we assume the two servers are identical with $\mu_1 = \mu_2 = \mu$.

Let $N_1(t)$ $(0 \leq N_1(t))$ be the number of requests in the common queue, $N_2(t)$ $(0 \leq N_2(t))$ be the number of fork/join requests to the server that lags behind, and $N_3(t)$ $(N_3(t) = 0, 1)$ denote whether the other server is processing a request or not at time t. The state $\mathrm{N}(t) = (N_1(t), N_2(t), N_3(t))$ forms a Markov chain. If the system is stationary, $(\mathrm{N}(t) \to \mathrm{N}$ as $t \to \infty)$ and we let $p(m,n,l) = \lim_{t\to\infty} P[\mathrm{N}(t) = (m,n,l)]$, then these probabilities satisfy the following equations,

$$\begin{aligned}
(\lambda+\gamma)p(0,0,0) &= \mu p(0,1,0), \\
(\lambda+\gamma+\mu)p(0,1,0) &= \mu p(0,2,0) + \lambda p(0,0,0) + 2\mu p(0,1,1), \\
(\lambda+\gamma+\mu)p(0,n,0) &= \mu p(0,n+1,0) + \mu p(0,n,1), && n = 2, \cdots, \\
(\lambda+\gamma+2\mu)p(0,1,1) &= \mu p(0,2,1) + 2p_r\mu p(1,1,1) + \gamma p(0,0,0) + \lambda p(0,1,0), \\
(\lambda+\gamma+2\mu)p(0,2,1) &= \mu p(0,3,1) + 2p_r\mu p(1,1,1) + p_r\mu p(1,2,1) \\
&\quad + \gamma p(0,1,0) + \lambda p(0,2,0), \\
(\lambda+\gamma+2\mu)p(0,n,1) &= \mu p(0,n+1,1) + p_f\mu p(1,n-1,1) \\
&\quad + p_r\mu p(1,n,1) + \gamma p(0,n-1,0) + \lambda p(0,n,0), && n = 3, \cdots \\
(\lambda+\gamma+2\mu)p(i,1,1) &= \mu p(i,2,1) + 2p_r\mu p(i+1,1,1) + (\lambda+\gamma)p(i-1,1,1), && i = 1, \cdots \\
(\lambda+\gamma+2\mu)p(i,2,1) &= \mu p(i,3,1) + 2p_f\mu p(i+1,1,1) + p_r\mu p(i+1,2,1) \\
&\quad + (\lambda+\gamma)p(i-1,2,1), && i = 1, \cdots \\
(\lambda+\gamma+2\mu)p(i,n,1) &= \mu p(i,n+1,1) + p_f\mu p(i+1,n-1,1) \\
&\quad + p_r\mu p(i+1,n,1) + (\lambda+\gamma)p(i-1,n,1), && i = 1, \cdots; n = 3, \cdots
\end{aligned}$$

We develop bounds on the average response times for regular and fork/join customers by truncating one of the first two state variables, suitably modifying the infinitesimal generator and applying matrix-geometric techniques to the resulting model. Our computational experience indicates that we achieve greater accuracy for the same amount of computation by truncating $N_2(t)$. Consequently we report on this approach. Unfortunately there is insufficient symmetry in this system to allow us to obtain both optimistic and pessimistic bounds from a single model as we did for the DP-RJ policy. We describe and analyze separate models for each bound.

3.2.1 Optimistic Bound

In order to obtain optimistic bounds we impose the constraint $N_2(t) \leq B$. Whenever the second queue contains B fork/join requests and a new fork/join request arrives, it passes through without incurring any delay. Consequently, the fork/join customer associated with this request completes as soon as its other task completes at the other server.

This modified system can be modeled as a Markov chain with the same state description, $\mathbf{N}^{(lb)}(t) = (N_1^{(lb)}(t),\ N_2^{(lb)}(t),\ N_3^{(lb)}(t))$. Let $\mathbf{N}^{(lb)} = \lim_{nt\to\infty} \mathbf{N}^{(lb)}(t)$.

Lower bounds on the response times for the true system can be calculated as follows. We define the following workload vectors $\mathbf{U}_i = (U_{1,i}, U_{2,i})$ and $\mathbf{U}_i^{(lb)} = (U_{1,i}^{(lb)}, U_{2,i}^{(lb)})$ where $U_{j,i}$ and $U_{j,i}^{(lb)}$, $j = 1,2$, are the amounts of unfinished work associated with each of the two servers immediately before the arrival of the i-th customer in the true system and lower bound system respectively. We order the workload measures so that $U_{1,i} \leq U_{2,i}$ and $U_{1,i}^{(lb)} \leq U_{2,i}^{(lb)}$. Last, we define an ordering function $\psi(\mathbf{V})$ that takes an arbitrary finite element vector, $\mathbf{V}$, whose elements are real numbers and returns a vector with the elements ordered in increasing value.

Let $\{\tau_i\}_{i=1,\ldots}$ be a sequence of random variables denoting the time between customer arrivals, i.e., τ_i is the time between the arrival of the $(i-1)$th and ith customers. Let $\{R_i\}_{i=1,\ldots}$ be a sequence of binary random variables that denote whether the customers are regular or fork/join. Here $R_i = 0$ if the i-th customer is regular and $R_i = 1$ if it is fork/join. Let $\{X_{1,i}, X_{2,i}\}_{i=1,\ldots}$ be a sequence of random variables that denote the service times associated with the customers. If the i-th customer is fork/join, then its two tasks are assigned $X_{1,i}$ and $X_{2,i}$ as their service times; otherwise if it is regular it is assigned service time of $X_{1,i}$. Let $\{A_i\}_{i=1,\ldots}$ be a sequence of random variables denoting whether the auxiliary queue associated with the server that lags behind is full at the time that the customers are scheduled for service (i.e., the queue contains B customers at the time that the i-th customer is scheduled into service). Here $A_i = 0$ if that queue is full when the i-th customer is scheduled for service and $A_i = 1$ otherwise. The workload vectors evolve according to the following equations,

$$\mathbf{U}_{i+1} = (\psi(\mathbf{U}_i + (X_{1,i}, R_i X_{2,i})) - (\tau_i, \tau_i))^+, \qquad (5)$$

$$\mathbf{U}_{i+1}^{(lb)} = (\psi(\mathbf{U}_i^{(lb)} + (X_{1,i}, A_iR_iX_{2,i})) - (\tau_i, \tau_i))^+.$$

Here $((V_1, V_2)^+ = (\max\{V_1, 0\}, \max\{V_2, 0\})$.

Define $W_i^{(f)}$, $W_i^{(r)}$, $W_i^{(f)(lb)}$, and $W_i^{(r)(lb)}$ to be the response times of the i-th customer in the true and lower bound systems respectively for the case that they are either fork/join or regular. They are calculated from the work load vectors as follows:

$$\begin{aligned} W_i^{(f)} &= \max\{U_{1,i} + X_{1,i}, U_{2,i} + X_{2,i}\}, \\ W_i^{(r)} &= U_{1,i} + X_{1,i}, \\ W_i^{(f)(lb)} &= \max\{U_{1,i}^{(lb)} + X_{1,i}, U_{2,i}^{(lb)} + A_iX_{2,i}\}, \qquad (6) \\ W_i^{(r)(lb)} &= U_{1,i}^{(lb)} + X_{1,i}. \qquad (7) \end{aligned}$$

We further define another random variable W_i', which is useful in our following calculations, as,

$$W_i' = \max\{U_{1,i}^{(lb)} + X_{1,i}, U_{2,i}^{(lb)} + X_{2,i}\}$$

We have the following theorem. Here $W^{(r)}$, $W^{(f)}$, $W^{(r)(lb)}$, $W^{(f)(lb)}$, and W' are the stationary values of $W_i^{(r)}$, $W_i^{(f)}$, $W_i^{(r)(lb)}$, $W_i^{(f)(lb)}$, and W_i' respectively.

Theorem 2 *The true system and the modified system satisfy the following relationships,*

1. $W^{(f)} \geq_{st} W' \geq_{st} W^{(f)(lb)}$,
2. $W^{(r)} \geq_{st} W^{(r)(lb)}$.
3. $\mathbf{N} \geq_{st} \mathbf{N}^{(lb)}$

Proof. If the workload vectors initially satisfy $\mathbf{U}_0 \geq_{st} \mathbf{U}_0^{(lb)}$, then an induction argument can be used to show $\mathbf{U}_i \geq_{st} \mathbf{U}_i^{(lb)}$ for $i = 0, 1, \cdots$. Whenever the real system is ergodic (i.e., the r.v.'s $\mathbf{U}_i$ and $\mathbf{U}_i^{(lb)}$ converge to the limiting r.v.'s $\mathbf{U}$ and $\mathbf{U}^{(lb)}$), $\mathbf{U} \geq_{st} \mathbf{U}^{(lb)}$. This is certainly true under the assumptions of Poisson arrivals, exponential service times, and $\lambda + 2\gamma < 2\mu$. It then follows from the relation $\mathbf{U}_i \geq_{st} \mathbf{U}_i^{(lb)}$ that $W_i^{(f)} \geq_{st} W_i' \geq_{st} W_i^{(f)(lb)}$ and $W_i^{(r)} \geq_{st} W_i^{(r)(lb)}$, $i = 1, \cdots$. Again, if the real system is ergodic, then $W^{(f)} \geq_{st} W' \geq_{st} W^{(f)(lb)}$ and $W^{(r)} \geq_{st} W^{(r)(lb)}$.

The proof of the third relation is similar to the proof of theorem 1. ∎

Remark. The transient results hold for an arbitrary arrival process and arbitrary service times.

Corollary 1 *Let $N^{(r)}$, $N^{(f)}$, $N^{(r)(lb)}$, and $N^{(f)(lb)}$ denote the stationary queue lengths of regular and fork/join customers in the two systems. The following inequalities hold between their expectations, $E[N^{(r)}] \geq E[N^{(r)(lb)}]$ and $E[N^{(f)}] \geq \gamma E[W'] \geq E[N^{(f)(lb)}]$.*

Proof. The proof follows from Little's result and the fact that $E[X] \geq E[Y]$ whenever X and Y are r.v.'s such that $X \geq_{st} Y$. ∎

Again, the matrix-geometric method is used to solve the modified lower bound CP-AQ system. The detailed calculation is given in Appendix B.

In the remainder of this section, we will derive lower bounds on the expected number of regular and fork/join customers as well as the expected response time for regular and fork/join customers. In the case of fork/join customers, we will make use of Theorem 2 and its Corollary.

The expected length of the common queue, $E[N_1^{(lb)}]$, is

$$E[N_1^{(lb)}] = y(1)R[I - R]^{-2}e,$$

where the vector $y(1)$ and rate matrix R are solved for in Appendix B, and e is a vector with all elements 1. The average number of regular customers that are in service equals λ/μ. Consequently, by Theorem 2 and Corollary 1, a lower bound on the expected number of regular customers is

$$E[N^{(r)}] \geq E[N^{(r)(lb)}] \equiv \lambda E[N_1^{(lb)}]/(\lambda + \gamma) + \lambda/\mu.$$

Little's result yields

$$E[W^{(r)}] \geq E[W^{(r)(lb)}] = E[N_1^{(lb)}]/(\lambda + \gamma) + 1/\mu.$$

The expected number of fork/join customers in the common queue of the modified system is $\gamma E[N_1^{(lb)}]/(\lambda + \gamma)$. A lower bound on the expected number of fork/join customers that are in service is obtained by first determining the expected service delay incurred by a fork/join customer (i.e., time from beginning of processing of first task until completion of both tasks) in the modified system and then applying Little's result. Denote this expected delay by $d_1^{(lb)}$. The time required to service a fork/join customer depends on the length of the auxiliary queue and whether both servers are busy. If a fork/join customer begins service when the auxiliary queue contains $i-1$ tasks ahead of it, then the time to complete service is denoted by $h(i)$ which satisfies the recurrence

$$\begin{aligned}&h(1) = 3/(2\mu),\\ &h(i) = 1/(2\mu) + h(i-1)/2 + i/(2\mu), \quad i = 2, 3, \cdots.\end{aligned}$$

The first term in the recurrence for $h(i)$ corresponds to the average delay until the first of the two servers completes. If the server associated with the auxiliary queue completes, then the fork/join customer observes the system with one less customer in the auxiliary queue. The average delay in this case corresponds to the second term in the above recurrence. Last, when a fork/join customer begins service, one of his requests immediately begins service in the other server. Consequently, if this server completes, the average of the remaining delay of the fork/join corresponds to the average delay of the request still present in the auxiliary queue. This is the sum of average service times of his request and of the $i-1$ requests ahead of it in the auxiliary queue. This gives rise to the third term. This recurrence has the following solution

$$h(i) = (i + 2^{-i})/\mu, \quad i = 1, ...$$

There are three possible scenarios when a fork/join customer arrives to the system. First, both servers may be idle, second one of the servers may be idle, and third both servers may be busy. In the first two cases, the customer initiates service immediately. In the last case, the customer begins service only when he is at the head of the common queue. If at this moment an auxiliary queue has not built up at either server, the customer begins service as soon as either server completes service of its customer. Otherwise, the customer begins service only if the server without the auxiliary queue completes service.

The above observations yield the following expression for $d_1^{(lb)}$,

$$d_1^{(lb)} = q(0,0,0)h(1) + y(0)V_1 + \{y(1)[I-R]^{-1}e\}y(1)R[I-R]^{-1}V_2/\{y(1)R[I-R]^{-1}V_3\}$$

where

$$\begin{aligned}V_1 &= (h(2), h(3), \cdots, h(B), h(B+1))^T,\\ V_2 &= (2h(2), h(3), h(4), \cdots, h(B), h(B+1))^T,\\ V_3 &= (2, \underbrace{1, 1, \cdots, 1}_{B-1})^T.\end{aligned}$$

The coefficient 2 for the first element of V_2 and V_3 is a consequence of the observation that service of a new customer is initiated whenever a departure occurs from either server and there is no auxiliary queue.

The expected value of W', which bounds the $W^{(f)}$ from below, can now be computed by,

$$E[W'] = E[N_1^{(lb)}]/(\gamma + \lambda) + d_1^{(lb)}.$$

This can be used to obtain the following lower bound on the average number of fork/join customers (Corollary 1),

$$E[N^{(f)}] \geq \gamma E[N_1^{(lb)}]/(\gamma + \lambda) + \gamma d_1^{(lb)}.$$

Lower bounds on the expected response times are found in Tables 4 for different mixes of regular and fork/join customers (the entries in columns $p_r = 1/2$ and $p_r = 3/4$ are both upper and lower bounds accurate to three places). These values were calculated for $B = 16$.

3.3 Pessimistic Bound

A pessimistic bound on the performance of the CP-AQ policy is obtained in a similar manner. The auxiliary queue is allowed to have up to B requests. Whenever this queue is full and a request completes at the other server, the completed request is required to repeat its service. This avoids the possible event of a fork/join customer at the head of the common queue placing a request in the auxiliary queue and thus increasing its length above B. The resulting Markov chain is identical to the one described for the optimistic bound except for the following changes in the transition rates,

1. Remove state $(0, B, 0)$ and all transitions to and from it.

2. Remove the transition from state $(i, B, 1)$ to $(i-1, B, 1)$, $i = 2, \cdots$.

Let $\mathbf{N}^{(ub)}(t) = (N_1^{(ub)}(t), N_2^{(ub)}(t), N_3^{(ub)}(t))$ be the state of this new system. Let $W^{(r)(ub)}$ and $W^{(f)(ub)}$ denote the stationary response times for a regular customer and fork/join customer respectively. Let $\mathbf{N}^{(ub)}$ be the stationary queue length vector for the upper bound system. We state the following theorem.

Theorem 3 *The true system and the modified system satisfy the following relationships,*

1. $W^{(ub)(f)} \geq_{st} W^{(f)}$,

	$p_r = 0$	$p_r = 1/4$		$p_r = 1/2$		$p_r = 3/4$	
ρ	$E[W_f]$	$E[W_r]$	$E[W_f]$	$E[W_r]$	$E[W_f]$	$E[W_r]$	$E[W_f]$
0.1	1.65	1.03	1.65	1.02	1.65	1.02	1.65
0.2	1.84	1.07	1.84	1.07	1.83	1.06	1.81
0.3	2.09	1.14	2.06	1.14	2.03	1.12	1.99
0.4	2.42	1.25	2.35	1.24	2.28	1.23	2.21
0.5	2.88	1.42	2.74	1.41	2.61	1.39	2.48
0.6	3.56	1.70	3.28	1.69	3.05	1.65	2.84
0.7	4.70±.02	2.20	4.12	2.19	3.73	2.12	3.41
0.8	6.92±.19	3.26	5.66	3.24	4.97	3.09	4.47
0.9	13.23±2.11	6.68±.07	9.73±.09	6.51	8.45	6.06	7.52

Table 4: Bounds for CP-AQ under different p_r, $\mu = 1$.

2. $W^{(ub)(r)} \geq_{st} W^{(r)}$.

3. $\mathbf{N}^{(ub)} \geq_{st} \mathbf{N}$.

Proof. We observe that the workload vector for the upper bound system satisfies equation (5) except that the service times are no longer the same. Instead the service times are $X'_{i,j} \geq X_{i,j}$, $i = 1, 2$; $j = 1, \ldots$ The reason for the inequality is due to the fact that an occasional task is required to take an additional service time in the upper bound system. Consequently a simple proof by induction yields $\mathbf{U}_i \leq_{st} \mathbf{U}_i^{(ub)}$ for $i = 1, \ldots$ The rest of the theorem duplicates the arguments in the proof of theorem 2. ∎

The procedure for calculating the lower bounds on the average buffer occupancies and response times in the previous section applies with no change to the computation of the upper bounds. Numerical results for these bounds can be found in Tables 4 and for different mixes of regular and fork/join customers. Again B is taken to be 16. One observes that the bounds are tight for server utilizations less than 0.9 or when the fraction of regular customers exceeds 1/4.

3.4 Analysis of Other Policies

In this subsection, we simply describe analytic models for all other policies discussed in this paper. The basic techniques used to provide bounds for these systems are similar to those presented above.

3.4.1 The DP-SQ Policy

In this case we assume that the servers are identical, $\mu_1 = \mu_2 = \mu$. This policy can be modeled by a Markov chain $\mathrm{N}(t) = (N_{max}(t), N_{min}(t))$, where $N_{max}(t)$ is the number of tasks in the longest queue and $N_{min}(t)$ is the number of tasks in the shortest queue at time t. We are interested in the stationary behavior of this policy. We describe how tight bounds can be obtained for this system.

Optimistic Bound for the DP-SQ Policy: The optimistic bound for the DP-SQ can be obtained by truncating one of the state variables, say N_{min}, to a constant B, and then applying the matrix-geometric method. The modified system behaves as follows. If a fork/join customer arrives to find B tasks in the shortest queue, it generates only one task which enters the longest queue. In this case, the fork/join customer completes as soon as the task in the longest queue finishes. If an arriving regular customer finds B tasks in the shortest queue, then it exits the system immediately (and incurs zero delay).

We denote the state of the modified system as $\mathrm{N}^{(lb)} = (N^{(lb)}_{max}, N^{(lb)}_{min})$, and the stationary response time as $W^{(r)(lb)}$ and $W^{(f)(lb)}$.

The stationary distribution for this modified system satisfies the following equations. The calculations leading to their solution are based on the matrix geometric approach and are omitted.

$$
\begin{array}{rcll}
(\lambda+\gamma)p(0,0) & = & \mu p(1,0), & \\
(\lambda+\gamma+\mu)p(i,0) & = & \mu p(i,1)+\mu p(i+1,0), & i=1,\cdots \\
(\lambda+\gamma+2\mu)p(i,i) & = & \lambda p(i,i-1)+\mu p(i+1,i)+\gamma p(i-1,i-1), & i=1,\cdots,B-1. \\
(\lambda+\gamma+2\mu)p(i+1,i) & = & \gamma p(i,i-1)+\lambda p(i,i)+\lambda p(i+1,i-1) & \\
 & & \qquad +2\mu p(i+1,i+1), & i=1,\cdots,B-1. \\
(\lambda+\gamma+2\mu)p(i,j) & = & \gamma p(i-1,j-1)+\lambda p(i,j-1)+\mu p(i,j+1) & \\
 & & \qquad +\mu p(i+1,j), & j=1,\cdots,B-1; i=j+2,\cdots \\
(\lambda+\gamma+2\mu)p(B,B) & = & \gamma p(B-1,B-1)+\lambda p(B,B-1) & \\
 & & +\mu p(B+1,B), & \\
(\gamma+2\mu)p(B+1,B) & = & (\lambda+\mu)p(B,B)+\gamma p(B,B-1) & \\
 & & +\lambda p(B+1,B-1)+\mu p(B+2,B), & \\
(\gamma+2\mu)p(i,B) & = & \gamma p(i-1,B)+\gamma p(i-1,B-1)+\lambda p(i,B-1) & \\
 & & \qquad +\mu p(i+1,B) & i=B+2,\cdots
\end{array}
$$

Note: $p(i,j)=0$ for $i<j$.

Pessimistic Bound for the DP-SQ Policy: To obtain a pessimistic bound, we make a slight change to the system state description. The state is now defined as $\mathrm{N}(t) = (N_{min}(t), \Delta(t))$, where

$N_{min}(t)$ is the number of tasks in the shortest queue at time t, and $\Delta(t)$ is the difference between the shortest queue and the longest queue at time t, i.e., $N_{max}(t) = N_{min}(t) + \Delta(t)$. An upper bound on performance is obtained by truncating the state variable Δ to B. Whenever Δ equals B and there is a departure from the shortest queue, while the true system will transit from (N_{min}, B) to $(N_{min} - 1, B + 1)$, the modified system generates a fictitious task occupying the server, so that the system state is unchanged.

Let $\mathbf{N}^{(ub)}(t) = (N_{min}^{(ub)}(t), \Delta^{(ub)}(t))$, and the stationary response time be $W^{(r)(ub)}$ and $W^{(f)(ub)}$. We omit the equations that describe the behavior of the stationary distribution for this upper bound system along with the calculations leading to their solution.

We state the following theorem without proof.

Theorem 4 *The true system and the two modified systems satisfy the following relationships,*

1. $N_{max}^{(lb)} \leq_{st} N_{max} \leq_{st} N_{max}^{(ub)}$,
2. $N_{min}^{(lb)} \leq_{st} N_{min} \leq_{st} N_{min}^{(ub)}$,
3. $W^{(r)(lb)} \leq_{st} W^{(r)} \leq_{st} W^{(r)(ub)}$,
4. $W^{(f)(lb)} \leq_{st} W^{(f)} \leq_{st} W^{(f)(ub)}$.

3.4.2 The DP-SQ-MS Policy

Observe that the only difference in the behaviors of the DP-SQ-MS and the DP-SQ policies occurs when a regular customer arrives to a idle system. While under the DP-SQ policy, the regular customer randomly selects a server from which to obtain service, under the DP-SQ-MS policy, it generates two tasks and commences service at the both servers simultaneously. The Markov chain for the DP-SQ-MS policy is obtained from that of the DP-SQ policy by adding an additional state, say $(1, 1^*)$, that corresponds to the state where a regular customer has a task executing on each server. The Markov chain for this policy is partially illustrated in Figure 1.

A modification for producing an upper bound can be made in a similar manner.

3.4.3 The DP-MW Policy

Consider the behavior of the DP-MW policy. Suppose at some time t, there are n tasks waiting in the first queue, and m tasks waiting in the second queue, w.l.o.g., $n \geq m$. Then the last m tasks in

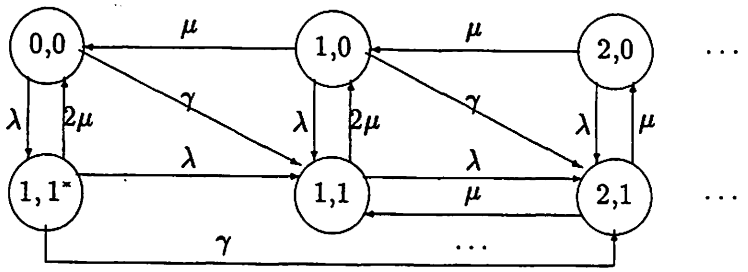


Figure 1: Partial Transition Diagram for DP-SQ-MS.

both queues are associated with the m most recently arrived customers who have not received any service yet. These m customers correspond to those customers waiting in the common queue under the CP-AQ policy. On the other hand, the first $n - m$ tasks waiting in the first queue must be fork/join tasks, which correspond to those fork/join tasks waiting in the auxiliary queue under the CP-AQ policy. This observation leads to the conclusion that the DP-MW policy behaves identically to the CP-AQ policy.

3.4.4 The CP-AQ-MS and the DP-MW-MS Policies

The same observation as that made in the last subsection leads to the equivalence of the DP-MW-MS and CP-AQ-MS policies. As was done in analyzing the DP-SQ-MS policy, the Markov chain describing the behavior of the CP-AQ-MS policy differs from that of the CP-AQ policy only when a regular customer arrives to an empty system. This is accounted for by introducing a new state that represents a regular customer receiving service at both servers.

3.4.5 The DP-MR Policy

Similar to the DP-SQ policy, this system can also be modeled by a Markov chain $\mathrm{N}(t) = (N_{max}(t), N_{min}(t))$, where $N_{max}(t)$ is the number of tasks in the longest queue at time t, and the $N_{min}(t)$ is the number of tasks in the shortest queue at time t. However, the infinitesimal generator matrix is different from that of DP-SQ. Lower bounds on the stationary behavior of this policy is obtained by truncating the state N_{min} and applying the matrix-geometric method. Upper bounds are obtained by truncating and solving for the stationary probabilities of the Markov chain $\mathrm{N}(t) = (N_{min}(t), \Delta(t))$ where $\Delta(t)$ denotes the difference between the longest and shortest queues. Similar relationships as stated in Theorem 4, i.e., the stochastic ordering between N and $\mathrm{N}^{(lb)}, \mathrm{N}^{(ub)}$, $W^{(r)}$ and $W^{(lb)(r)}, W^{(ub)(r)}$, and $W^{(f)}$ and $W^{(lb)(f)}, W^{(ub)(f)}$, also exist here. The detailed proof and calculations are omitted.

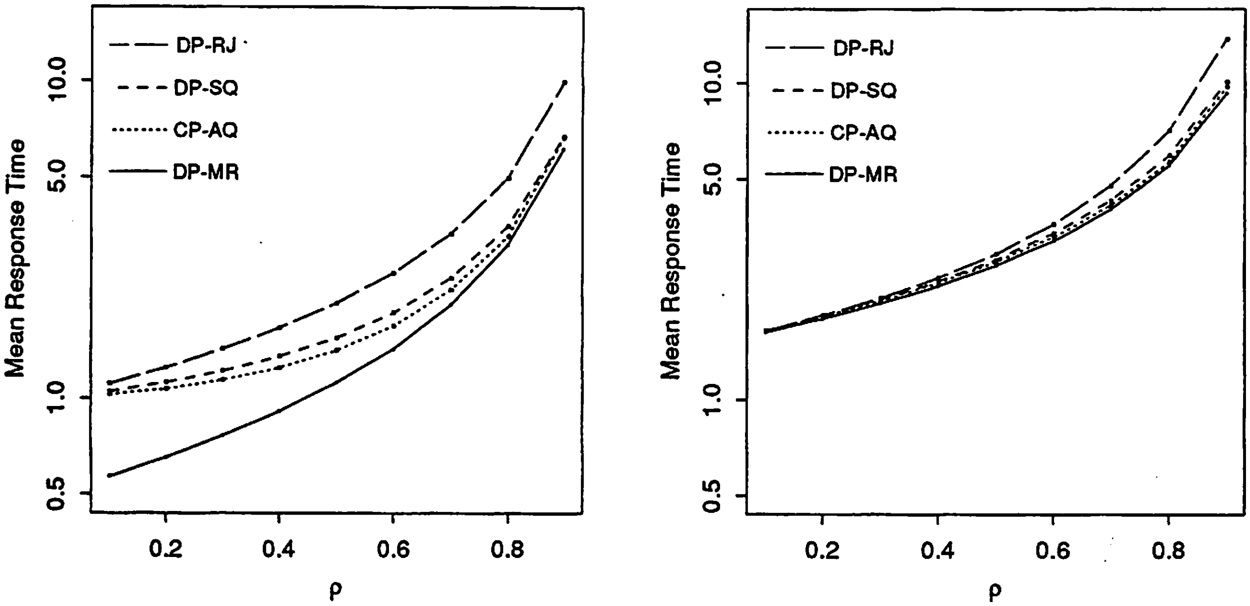
4 Performance Comparisons of Different Policies

In this section, we compare the performance of the policies described and analyzed in the previous two sections. The performance metrics of most interest are the mean sojourn times of regular and fork/join customers.

In our experiments, we assume that the two servers are homogeneous with rate $\mu = 1$. The arrival rates of regular customers and fork/join jobs are λ and γ respectively. The probability of a randomly chosen customer being a regular customer is $p_r = \lambda/(\lambda + \gamma)$, and the probability that a randomly chosen customer is a fork/join customer is $p_f = 1 - p_r$. We assume that under the DP-RJ policy, regular customers choose either server with equal probability, $\alpha_1 = \alpha_2 = 1/2$. We estimate the mean sojourn times of the two classes under different policies by averaging the lower and the upper bounds obtained from the models introduced in last section. In fact, as shown in previous tables, the bounds are very tight. The difference between the lower and the upper bounds appears only when the system is highly loaded (server utilization around or over 90%).

In Figure 2 and 3, we show the mean sojourn times for regular and fork/join customers as a function of the server utilization. In Figure 2, most customers are fork/join customers ($p_r = 0.25$), and in Figure 3, most customers are regular customers ($p_r = 0.75$). From these results, we observed that the DP-SQ policy performs better than the DP-RJ policy. This is because the DP-SQ policy can achieve a better balance among the queue lengths. The CP-AQ policy shows a higher performance than the DP-SQ policy because, instead of balancing the queue lengths, the CP-AQ tries to balance the unfinished work among the two servers. It is interesting to note however that the DP-MR policy provides the best performance among all the policies. This is in spite of the fact that the servers are given higher loads, i.e., both servers may simultaneously process tasks belonging to the same regular customer. However, the benefit is probably explained by the fact in this case, each server will not be required to work for a time that exceeds the minimum of two i.i.d. exponentially distributed service times which has mean that is one half that of a single service time. In addition, the DP-MR policy accrues a definite advantage when the system is empty at the arrival of a regular customer.

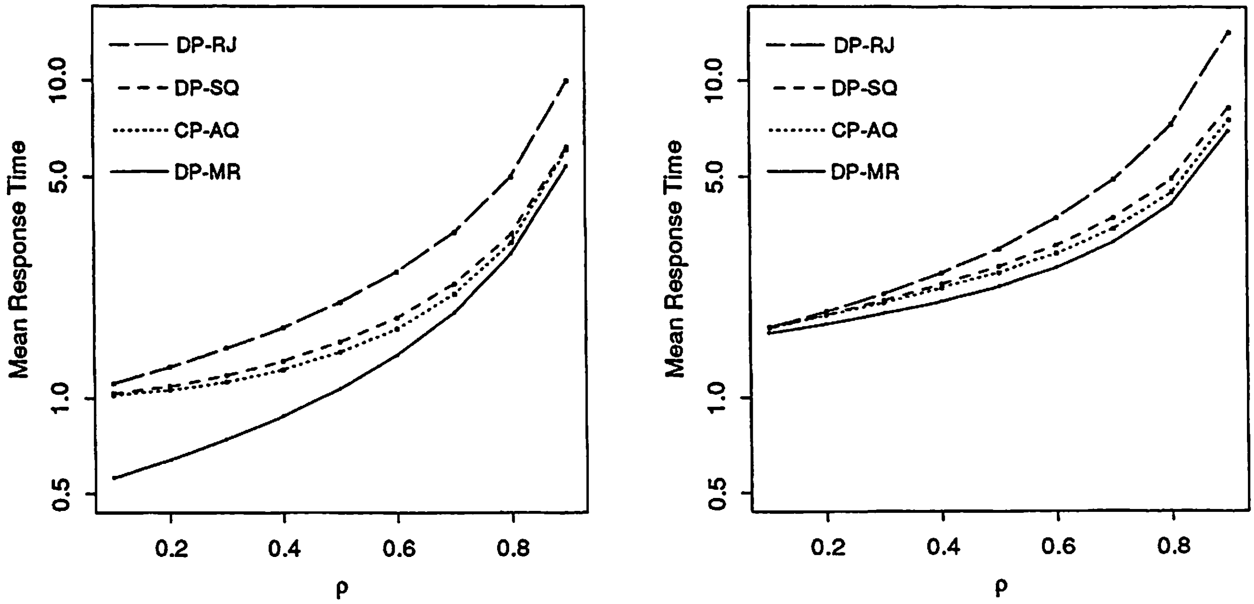
Figure 4 shows the impact of processing a regular customer on both servers when the system is empty at arrival. The results are obtained by comparing the DP-SQ-MS policy against the DP-SQ policy. As we expected, the performance improvement is noticeable when the server utilization decreases. In addition, the improvement is greater when the fraction of regular customers is large than when the fraction is small. In this case, fork/join customers can also benefit a little from the MS-modification. The improvement of the DP-MW-MS(CP-AQ-MS) over the DP-MW(CP-AQ) is similar and is omitted here.



(a). Regular Customers.

(b). Fork/Join Customers.

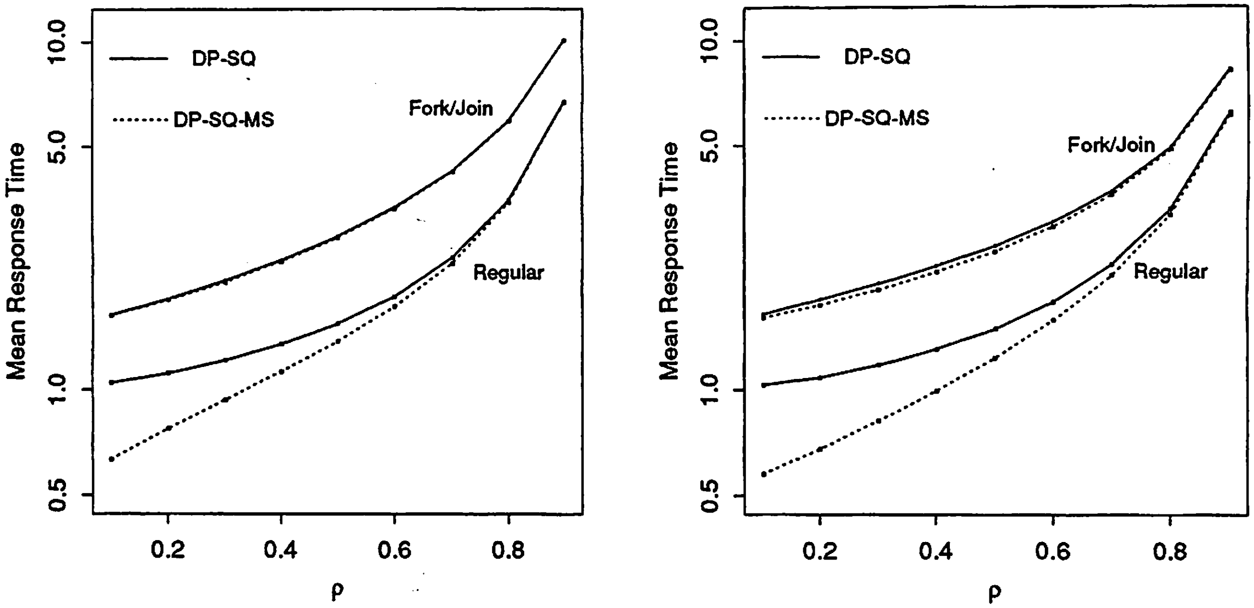
Figure 2: Performance of Different Policies ($p_r = 0.25$)



(a). Regular Customers.

(b). Fork/Join Customers.

Figure 3: Performance of Different Policies ($p_r = 0.75$)



(a). DP-SQ vs. DP-SQ-MS Policies (Pr=0.25).

(a). DP-SQ vs. DP-SQ-MS Policies (Pr=0.75).

Figure 4: The Impact of MS-Discipline

5 Summary

In this paper we have described several policies suitable for two server fork/join queueing systems which serve both regular and fork/join customers. We developed analytic models that can be used to bound the performance of these policies. These bounds are obtained by appropriately approximating a Markov chain with two unbounded state variables by a chain with only one unbounded state variable. These chains are solved using the Matrix geometric methodology and lead to either upper or lower bounds on the performance metrics of interest. Numerical results show that these bounds can be quite accurate. Consequently, the performance of the two categories of fork/join queuing systems can be studied for different workloads, server speeds, etc.

Using these models, we observed that the performance of *centralized* policy, CP-AQ, to be better than most of *distributed* policies, since it tries to balance the unfinished workload between the two servers. Allowing a regular customer who arrives to a idle system to start service at the two servers simultaneously may lead to a slight performance improvement, especially when the system workload is light or most customers are regular. Finally, assigning each regular customer to both servers and as soon as one completes removing the other leads yet to another additional performance improvement, given the service times are exponentially distributed. In a previous work

[16], where performance of a mirrored disk is examined, we also observed that the DP-MR policy outperforms other policies while the disk service time is non-exponential. However, we point out that when service times are constant, the DP-MR policy is worse than the DP-SQ, since serving a regular customer at two servers simultaneously may not reduce any service time but will block other customers from attaining service.

Appendix A: Calculations of the DP-RJ Policy

In this Appendix, we obtain the stationary probabilities for the model that produces a lower bound on the performance of the DP-RJ policy, $P[\mathrm{N}^{(lb)} = (i,j)] = q(i,j), i = 0, 1, \cdots; \ j = 0, 1, \cdots, B$. We define the steady state probability vector $Y = (y(0), y(1), y(2), \cdots)$ where $y(i)$ is a $(B+1)$ element vector, $y(i) = (q(i,0), q(i,1), \cdots, q(i,B))$, $i = 0, 1, \cdots$. The infinitesimal generator Q, satisfying $YQ = 0$, is listed below

$$Q = \begin{bmatrix} B_1 & A_0 & 0 & 0 & \cdots \\ B_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where the matrices B_1, B_2, A_0, A_1, and A_2 are defined as $(B+1) \times (B+1)$ matrices,

$$A_0 = \begin{bmatrix} \alpha_1\lambda & \gamma & & \\ 0 & \alpha_1\lambda & \gamma & \\ & & \ddots & \\ & & & \alpha_1\lambda + \gamma \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -(\gamma+\lambda) & \lambda\alpha_2 & 0 & & & \\ \mu_2 & -(\gamma+\mu_2+\lambda) & \alpha_2\lambda & & & \\ 0 & \mu_2 & -(\gamma+\mu_2+\lambda) & & & \\ & & \ddots & \ddots & & \\ & & & \mu_2 & -(\gamma+\mu_2+\lambda) & \alpha_2\lambda \\ & & & & \mu_2 & -(\gamma+\mu_2+\alpha_1\lambda) \end{bmatrix}$$

$$B_2 = A_2 = \mu_1 I_{(B+1)\times(B+1)}$$

$$A_1 = \begin{bmatrix} c_0 & \alpha_2\lambda & & & & & \\ \mu_2 & c_1 & \alpha_2\lambda & & & & \\ 0 & \mu_2 & c_1 & \alpha_2\lambda & & & \\ & \cdot & \ddots & \ddots & \ddots & & \\ & & & & \mu_2 & c_1 & \alpha_2\lambda \\ & & & & & \mu_2 & c_2 \end{bmatrix} \qquad \begin{aligned} c_0 &= -(\gamma + \mu_1 + \lambda) \\ c_1 &= -(\gamma + \mu_1 + \mu_2 + \lambda) \\ c_2 &= -(\gamma + \mu_1 + \mu_2 + \alpha_1\lambda) \end{aligned}$$

Define the matrix $A = A_0 + A_1 + A_2$. Neuts [12] showed that Q is positive recurrent if $\pi A_0 e < \pi A_2 e$, where π is the unique solution to $\pi A = 0$, $\pi e = 1$. Here e is a column vector containing $B + 1$ ones and π is a $B + 1$ element vector containing the stationary queue length distribution for the M/M/1/B queue with arrival rate $\gamma + \alpha_2\lambda$ and service rate μ_2, i.e., the i-th element *of* π *is* $(1 - u)u^i/(1 - u^{(B+1)})$ where $u = (\alpha_2\lambda + \gamma)/\mu_2$. In the modified model, the condition for positive recurrence is $\gamma + \alpha_1\lambda < \mu_1$. Whenever Q is positive recurrent, the stationary probability vector Y can be expressed as

$$y(i) = y(0)R^i, \quad i = 0, 1, \cdots.$$

where R is the minimal solution of $A_0 + RA_1 + R^2A_2 = 0$. The matrix R can be obtained iteratively by the following approach. Let $R(n)$ denote the value of R after n iterations.

$$\begin{aligned} R(0) &= 0, \\ R(n+1) &= -A_0A_1^{-1} - R^2(n)A_2A_1^{-1}, \quad n > 0. \end{aligned}$$

Rao and Posner [13] have shown that the vector $y(0)$ takes the form

$$y(0) = \pi(I - R).$$

The preceding analysis differs from that presented by Rao and Posner only in the definition of the submatrices B_1, B_2, A_0, A_1, A_2.

Appendix B: Calculations of the CP-AQ Policy

The stationary probability distribution for the model that provides a lower bound on the performance of the CP-AQ policy, $P[\mathrm{N}^{(lb)} = (m, n, l)] = q(m, n, l)$, $m = 0, 1, \cdots$; $n = 0, 1, \cdots, B$; $l = 0, 1$, is obtained here. We define the steady state probability vector $Y = (x, y(0), y(1), y(2), \cdots)$ where $x = [q(0, 0, 0)]$, $y(0)$ is the B element vector $[q(0, 1, 0), q(0, 2, 0), \cdots, q(0, B, 0)]$, and $y(i)$ is a B element vector $y(i) = (q(i-1, 1, 1), \cdots, q(i-1, B, 1))$, $i = 1, 2, \cdots$. The infinitesimal generator

Q satisfying $YQ = 0$ is listed below,

$$Q = \begin{bmatrix} D_2 & C_1 & B_0 & 0 & 0 & \cdots \\ D_3 & C_2 & B_1 & 0 & 0 & \cdots \\ 0 & C_3 & A_2 & A_1 & 0 & \cdots \\ 0 & 0 & A_3 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where D_2 is a one element vector $D_2 = -[\lambda+\gamma]$, D_3 is a B element column vector $D_3 = [\mu, 0, \cdots, 0]^T$, B_0 and C_1 are B element vectors $B_0 = [\mu, 0, \cdots, 0]$, $C_1 = [\lambda, 0, \cdots, 0]$, and C_2, C_3, B_1, A_1, A_2, A_3 are $B \times B$ matrices

$$C_2 = \begin{bmatrix} -(\gamma+\lambda+\mu) & 0 & 0 & & \\ \mu & -(\gamma+\lambda+\mu) & 0 & & \\ 0 & \mu & -(\gamma+\lambda+\mu) & & \\ & & \ddots & \ddots & \\ & & & \mu & -(\gamma+\lambda+\mu) \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 2\mu & 0 & & \\ 0 & \mu & & \\ & \ddots & \ddots & \\ & & 0 & \mu \end{bmatrix}$$

$$B_1 = \begin{bmatrix} \lambda & \gamma & & & \\ & \lambda & \gamma & & \\ & & \ddots & \ddots & \\ & & & \lambda & \gamma \\ & & & & \lambda+\gamma \end{bmatrix}$$

$$A_1 = (\gamma+\lambda)I_{B\times B},$$

$$A_2 = \begin{bmatrix} -(\gamma+\lambda+2\mu) & 0 & & \\ \mu & -(\gamma\lambda+2\mu) & & \\ & \ddots & \ddots & \\ & & \mu & -(\gamma+\lambda+2\mu) \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 2\mu\lambda/(\gamma+\lambda) & 2\mu\gamma/(\gamma+\lambda) & 0 & & \\ 0 & \mu\lambda/(\gamma+\lambda) & \mu\gamma/(\gamma+\lambda) & & \\ & \ddots & \ddots & & \\ & & & \mu\lambda/(\gamma+\lambda) & \mu\gamma/(\gamma+\lambda) \\ & & & & \mu \end{bmatrix}$$

Define the matrix $A = A_1 + A_2 + A_3$. The infinitesimal generator Q is positive recurrent if $\pi A_1 e < \pi A_3 e$, where π is the unique solution to $\pi A = 0$, $\pi e = 1$. The stationary probability vector Y satisfies the matrix-geometric form

$$y(i) = y(1)R^{i-1},\ i = 1, 2, \cdots$$

where R is the minimal solution of $A_1 + RA_2 + R^2 A_3 = 0$. The matrix R can be obtained in a similar manner as for the model of the DP-RJ policy. Finally, the vectors x, $y(0)$, and $y(1)$ are obtained by solving the following set of equations,

$$\begin{aligned} xD_2 + y(0)D_3 &= 0, \\ xC_1 + y(0)C_2 + y(1)C_3 &= 0, \\ xB_0 + y(0)B_1 + y(1)[A_2 + RA_3] &= 0, \\ x + [y(0) + y(1)[I - R]^{-1}]e &= 1. \end{aligned}$$

References

[1] Avi-Itzhak, B. and Halfin S. 1990. "Non-Preemptive Priorities in Simple Fork-Join Queue," *RUTCOR Research Report*, # 41-90, Aug.

[2] F. Baccelli. 1985. "Two parallel queues created by arrival with two demands: The M/G/2 symmetrical case". Report INRIA No. 426.

[3] J.F. Bartlett. 1981. "A Nonstop* Kernel," *Proc. Eighth Symp. on Operating System Principles*, pp. 22-29.

[4] F. Baccelli, A. Makowski, A. Shwartz. 1989. "Fork-join queues and related systems with synchronization constraints: Stochastic ordering, approximations and computable bounds", *Adv. Appl. Prob.* 21, pp.629-660.

[5] Flatto L. and S. Hahn. 1984. "Two parallel queues created by arrivals with two demands I," *SIAM J. Appl. Math.*, vol. 44, pp. 1041-1053.

[6] Green, L. 1985. "A queueing system with general-use and limited -use servers," *Operations Research*, Vol. 33, pp. 168-182.

[7] D.P. Heyman, M.J. Sobel, 1982. *Stochastic Models in Operations Research, Vol. I*, McGraw Hill.

[8] Kim, C. and Agrawala, A. K., 1989. "Analysis of the Fork-Join Queue," *IEEE Trans. on Comp.*, Vol.38, No.2, Feb., pp.250-255.

[9] Little, J.D.C. 1961. "A proof of the queueing formula $L = \lambda W$," *Operations Research*, Vol. 9, pp. 383-387.

[10] Nelson, R. and B.R. Iyer. 1985. "Analysis of a replicated data base," *Performance Evaluation*, Vol. 5, pp. 133-148.

[11] Nelson, R. and A.N. Tantawi. 1985. "Approximate analysis of fork/join synchronization in parallel queues," IBM Report RC11481.

[12] Neuts, M.F. 1981. *Matrix-Geometric solutions in stochastic models - an algorithmic approach*, John Hopkins University Press.

[13] Rao, B.M. and M.J.M. Posner. 1985. "Algorithmic and Approximation Analyses of the Split and Match Queue," *Stochastic Models*, Vol. 1, pp. 433-456.

[14] Rao, B.M. and M.J.M. Posner. 1987. "Algorithmic and Approximation Analyses of the Shortest Queue Model," *Naval Research Logistic*. Vol. 24, pp.381-398.

[15] Stoyan D. 1983. *Comparison methods for queues and other stochastic models*, John Wiley & Sons, Chichester England.

[16] Towsley, D., Chen, S. and Yu, S-P. 1990. "Performance Analysis of a Fault Tolerant Mirrored Disk System," *Proc. PERFORMANCE'90*, Sept. Edinburgh, Scotland, pp.239-253.