Approximate Analysis of Parallel Processor Allocation Policies

Rajesh K. Mansharamani
Mary K. Vernon

Technical Report #1187

November 1993

Approximate Analysis of Parallel Processor Allocation Policies *

Rajesh K. Mansharamani and Mary K. Vernon
(mansha@cs.wisc.edu) (vernon@cs.wisc.edu)

Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706.

November 15, 1993

Abstract

The complexity of parallel applications and parallel processor scheduling policies makes both exact analysis and simulation difficult, if not intractable, for large systems. In this paper we propose a new approach to performance modeling of multiprogrammed processor scheduling policies, that of *interpolation approximations*. We first define a workload model that contains parameters for the essential properties of parallel applications with respect to scheduling discipline performance, yet lends itself to mathematical analysis. Key features of the workload model include general distribution of total job processing time, general distribution of available job parallelism, and a simple characterization of parallelism overheads. We then show that one can find specific values of the system parameters for which the parallel system under a given scheduling policy reduces to a queueing system with a known (closed-form) solution. Finally, interpolation between the points with known solutions is used to arrive at mean response time estimates that hold over the entire system parameter space. The interpolation approximations readily yield insight into policy behavior and are easy to evaluate for systems with hundreds of processors.

We illustrate the approach by developing and validating models of three scheduling policies, under the assumptions of linear job execution rates and independence between job parallelism and processing time. We discuss several insights and results obtained from the analysis of the three policies under the assumed workloads. One result clarifies and generalizes observations in two previous simulation studies of how policy performance varies with the coefficient of variation in job processing requirement. Another result of the interpolation models yields new insight into how policy performance varies with job parallelism. We also comment on the generalizations of these insights for workloads with less restrictive assumptions.

*This research was partially supported by the National Science Foundation under grants CCR-9024144 and CDA-9024618.

1 Introduction

The algorithm for scheduling jobs on the processors of a multiprogrammed parallel computer can have a significant impact on system performance. Parallel processor scheduling disciplines have been studied using system measurement [10, 23, 44, 46], simulation [9, 17, 18, 24, 50], and analytic modeling methods [8, 16, 19, 25, 27, 28, 34, 35, 42]. While these studies have yielded various specific insights, the general performance characteristics of parallel scheduling policies still remains poorly understood. Questions such as which policies dominate over various regions of the system design space and which workload characteristics are the key *determinants* of policy performance remain largely unanswered.

Measurement studies are necessarily limited to specific mixes of applications, whereas simulation studies are limited to specific workload assumptions (e.g., particular distributions of job processing requirement). Analytic models have the potential to be efficient, broadly applicable, and readily yield insight, but to date analytic models of parallel processor scheduling disciplines have been limited in three respects. First, the models involve numerical solution of sets of simultaneous equations that yield no direct insight into the functional relationship between system performance and particular workload parameters. Second, the sets of simultaneous equations typically grow superlinearly in the number of processors thus limiting their solution to small system sizes, such as 20 or fewer processors. Third, all but one of the previous analytic models either assume exponential distributions of job service time [19, 35], or assume independent and identically distributed (i.i.d.) task execution times (implying a specific degree of correlation between total job demand and the number of tasks in a job) [16, 25, 27, 28, 34, 42]. Regarding the former assumption, experience from uniprocessor systems suggests that *relative* policy performance can be highly sensitive to the (second moment of) job service demand distribution, and that total job demand can have significantly higher variability than the exponential [33, 43]. Relative scheduling policy performance for parallel systems may also be sensitive to the degree of correlation between available job parallelism and total job service requirement. Thus, the particular assumptions in previous analytic models potentially limit the applicability of the results, as well as the ability to study policy behavior. The model in [8] allows general job processing requirement, but to apply their analysis one needs to know the probability that a job is allocated a given number of processors as a function of job type and system utilization. These probabilities may be difficult to obtain for many scheduling policies.

The above limitations suggest a need for alternative models of parallel processor scheduling disciplines.

In this paper we define (1) a workload model that captures the significant features of parallel applications in a few simple parameters, and (2) a new approach for deriving mean response time equations that are efficient to evaluate for kiloprocessor systems and that yield direct insight into which workload parameters are the key determinants of scheduling policy performance. The workload model allows general distribution of available job parallelism, sublinear as well as linear job execution rates (i.e. speedup curves), and general distribution of total job processing requirement. For the sake of space, we define the workload model for the case of no correlation between a job's available parallelism and processing requirement. Simple extensions that allow full correlation and arbitrary correlation are also possible [21]. The model lends itself to mathematical analysis, yet allows for broad applicability of the results and provides the basis for a fairly complete understanding of policy performance.

The approach we propose for obtaining mean job response time equations is that of interpolation approximations. That is, we first find points in the parameter space for which the parallel system, under a specific scheduling policy, reduces to a queueing system with a known (closed-form) solution for mean response time. We then interpolate among the points with known solution to arrive at mean job response time formulas over the entire parameter space. Although the approach is "ad hoc" in nature, extensive validations show that it can yield reasonably accurate results. Since the formulas are extremely efficient to evaluate and readily yield insight into policy behavior, the approach may offer the right trade-off between accuracy on the one hand, and efficiency, insight, and applicability on the other.

We illustrate the interpolation approximation approach and the insights that can be derived from it for three parallel scheduling policies and a particular set of workload assumptions. These assumptions include general distribution of total job processing requirement, linear job execution rates, and no correlation between job demand and job parallelism. The approach can be applied to cases of sublinear execution rates and correlated workloads [21], but focusing on the more restrictive assumptions simplifies the exposition of the technique, which is the primary purpose of this paper.

The interpolation approximation approach has previously been applied to multiserver and fork-join queues. In section 2, we review interpolation approximations that have appeared in the previous literature and define the three scheduling policies considered in this paper (FCFS, EQ, and PSAPF). Section 3 presents our workload model and system assumptions. In section 4 we show how the FCFS and EQ policies, under particular values of the system parameters, reduce to queues with known solutions. In section 5 we

present the interpolation approximations that yield the mean response time over the entire parameter space for each of the EQ and FCFS policies. We also present validations of the approximations in that section. In section 6 we derive and validate interpolation approximations for the mean response time under the PSAPF policy. Section 7 presents some results and insights obtained from the interpolation models and comments briefly on the relationship between these results and the results in previous papers. Finally, section 8 presents a summary of our conclusions.

2 Background

In this section we first outline the interpolation approximation approach and summarize such approximations that have appeared in previous literature. We then define three scheduling policies that have been proposed in previous literature, whose performance will be analyzed using interpolation approximations.

2.1 Interpolation Approximations

The underlying principle behind interpolation approximations is simple: *use the known to predict the unknown.* The first step is to derive efficient solutions at extreme values of system parameters, for example, light and heavy traffic limits of mean response time. The next step is to form a function that interpolates between the extreme points in a way that approximates system behavior. The interpolation is on the parameter for which exact results are derived under extreme values. In some cases it is necessary to normalize the measure of interest before forming an interpolation function, and then 'unnormalize' the function to obtain the desired approximation. For example, if the interpolation is on system utilization, ρ, the mean job response time is first multiplied by $1 - \rho$ so that the heavy traffic limit does not go to infinity.

The following is a summary of interpolation approximations that have appeared in previous literature. Cosmetatos interpolates between the mean waiting time in an M/D/c queue and in an M/M/c queue to obtain an approximation for the mean waiting time in an M/G/c queue when the coefficient of variation in service time $C_X \leq 1$ [4]. The parameter of interpolation is C_X^2. (The approximation can be used as an extrapolation for $C_X > 1$.) Burman and Smith perform a linear interpolation between light and heavy traffic limits of the ratio of the mean delay in a single server FCFS queue with non-homogeneous Poisson arrivals to the mean delay in an M/G/1 FCFS queue with the same mean arrival rate and service time distribution [2]. In [3] they use a similar approach to obtain estimates for the mean delay in single server

and multiple server FCFS queues (sequential jobs) with more general arrival processes. Fleming interpolates between light and heavy traffic limits of the moments of the waiting time distribution in an M/G/1 Round Robin queue [6]. Simon and Willie estimate response time characteristics in priority queueing networks using interpolation approximations based on simulation and heavy traffic limits [37]. Reiman and Simon [30], and Reiman et al. [31] provide interpolation approximations for the moments of response time and queue lengths in a variety of single server queueing systems using light and heavy traffic limits as well as derivatives of the computed measure at light traffic. Fleming and Simon derive interpolation approximations for response time distributions in several single server queues, based on a similar approach [7]. Whitt [47], Fendick and Whitt [5] interpolate between light and heavy traffic limits to obtain approximations for a measure they call *mean steady-state workload* (or virtual waiting time) in a GI/G/1 queue and in general single server queues without independence conditions. Varma and Makowski [45] propose interpolation approximations for the mean response times of a symmetric fork-join queue with general inter-arrival and service time distributions.

Although interpolation approximations have been used for the analysis of single server, multiserver, and fork-join queues, we have not encountered the use of this technique for the analysis of parallel processor scheduling policies.

2.2 Processor Allocation Policies

Three parallel processor scheduling policies in the previous literature are considered in this paper: FCFS, EQ, and PSAPF. The FCFS policy is very simple and has been shown to have high performance for specific workloads [42]. The EQ policy is an idealization of the class of scheduling policies that allocate an equal fraction of processing power to each job in the system, subject to the constraint that a job is never allocated more processors than its available parallelism. An example of such a policy is the default CM-5 scheduler for jobs that fit in the memory of a single partition. Previous studies have shown that variants of the EQ policy have high performance under a variety of workloads [44, 17, 23, 10, 16, 35, 24]. Finally, we examine the PSAPF policy proposed in [18] because of its potential for high performance for workloads where job processing time is correlated with parallelism [18, 16]. Furthermore, this policy allows us to illustrate an interesting aspect of the interpolation approximation approach.

Each of the policies is defined in the context of a global or central job queue. The three policies are defined in terms of the processing power that they allocate to jobs in the queue, and not in terms of the allocation

of processors to individual tasks within a job. All three policies make use of the *available parallelism* in the jobs to decide how many processors to allocate to each job in the system. We define the available parallelism of a job to be the number of processors the scheduler believes the job can make productive use of.

(i) FCFS[1]: The FCFS policy allocates processors to jobs on a first-come-first-serve basis. Each job is allocated processors as they become available up to a maximum of its available parallelism. Processors released by a departing job are first allocated to the job in service (if any) whose allocation is less than its available parallelism and then to jobs waiting for service. For example, if there are five jobs in a 100-processor system and the available parallelism per job is (50, 25, 100, 10, 10), then the allocation of processing power is (50, 25, 25, 0, 0). This policy has been studied under different workload assumptions in previous literature [28, 18, 25, 17, 42, 16].

(ii) EQ: The dynamic EQuiallocation policy allocates an equal fraction of processing power to each job in the system unless a job has smaller available parallelism than the equiallocation value, in which case each such job is allocated as many processors as its available parallelism, and the equiallocation value is recursively recomputed for the remaining jobs. Allocation of processing power for the above example is (27.5, 25, 27.5, 10, 10). Reallocation of power can occur on job arrivals, job departures, and changes in a job's available parallelism.

Partitioning of processing power can be spatial, temporal, or some combination of the two [44, 17, 23, 10, 16]. The analysis in this paper holds for any of these cases. Both simulation and analytic models for the EQ policy, based on various workload assumptions, have appeared in the previous literature [17, 16, 35, 24].

(iii) PSAPF: Preemptive Smallest Available Parallelism First. The central job queue is a preemptive queue that is ordered in ascending order of available job parallelism. Jobs with the same available parallelism are served in first-come-first-serve order. As in the FCFS policy each job is allocated processors as they become available (or preempted) up to a maximum of its available parallelism, and processors released by a departing job are first allocated to the job in service (if any) whose allocation is less than its available parallelism and then to the jobs waiting for service. Processor allocation for the above

[1]The FCFS policy is defined for the case that available parallelism is fixed throughout the life of a job, as assumed in the workload model in section 3. There exist extensions to the policy for the case where the available parallelism of the job changes during its lifetime and the system scheduler can detect and react to the changes.

example is (50, 25, 5, 10, 10). Processor allocation to jobs can change upon job arrivals, job departures, and changes in job parallelism. This policy was proposed in [18] and also studied in [16, 17] under specific workloads.

3 Model

A goal of this work is to develop a system model that is broadly applicable, characterizes the essential features of parallel workloads with a small number of parameters, and is easy to analyze. Below we define a system and workload model that we believe achieves these goals and comment on the trade-offs between realism and tractability that were made when constructing the model. Finally, we give a brief review of the model notation, which will be used throughout the remainder of the paper.

3.1 System Model

We consider an open system model with P identical processors, a central job queue, and a scheduling policy denoted by Ψ. Note that the central job queue is a conceptual model; the actual implementation of the queue might allow for distributed access. We assume zero scheduling and preemption overhead, with the understanding that the actual implementation of a particular scheduling policy will include limits on preemption rates (i.e., delayed preemptions) so as to reduce overhead to a small fraction of the productive execution on the processors. We next describe our workload model.

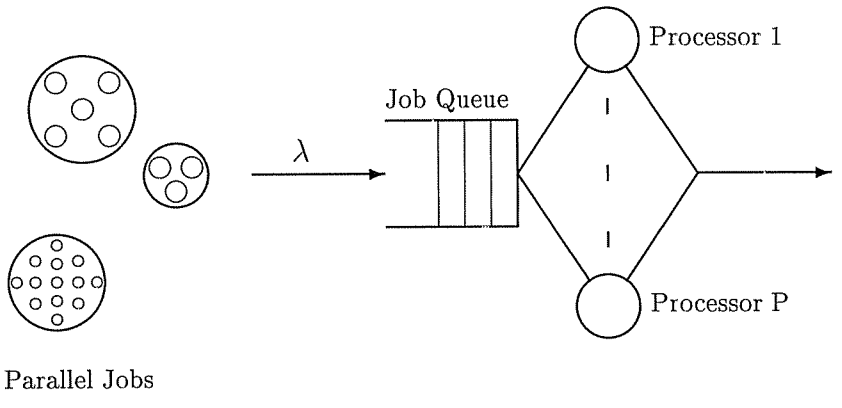


Figure 1: Open System Model

3.2 Workload Model

Jobs arrive at the system according to a Poisson process with rate λ as shown in figure 1. All jobs are considered to be statistically identical. Each job is characterized by the following random variables:

(1) Total service demand (execution time on one processor) D,

(2) Available parallelism $N \in \{1, 2, \ldots, P\}$,

(3) Execution rate function $E : [0, P] \rightarrow [0, N]$, which is nondecreasing and has the following properties:

$$E(x) \begin{cases} \leq x, & 1 < x \leq N, \\ = E(N), & N < x \leq P. \end{cases}$$

The system operates as follows. Upon arrival each job joins the central job queue. At each time $t \geq 0$, the P processors are allocated to jobs present in the queue according to the processor allocation policy Ψ. If $a(t)$ processors are allocated to a job at time t, then its demand is satisfied at rate $E(a(t))$. The job leaves the system upon completion of its total demand, D. The available parallelism, N, of a job is the number of processors the system scheduler believes the job can productively use. The workload model assumes that this value is fixed throughout the lifetime of the job. The workload model also assumes that the job actually can't use use more than N processors productively (i.e., $E(x) = E(N)$ for $N < x \leq P$).

We make the following assumptions:

- The available job parallelism, N, has a general (bounded) distribution, $\mathcal{F}_N$, with mean $\overline{N}$ and coefficient of variation[2] C_N. The probability mass function of N is specified by $\underline{p} = (p_1, p_2, \ldots, p_P)$, where $p_i = \Pr[N = i]$, $i = 1, \ldots, P$.
- E is derived from a *deterministic* function γ, that is nondecreasing and is such that $\gamma(x) = x$ for $0 \leq x \leq 1$ and $\gamma(x) \leq x$ for $1 < x \leq P$. We refer to γ as the execution rate function (ERF) of the workload. The ERF γ is said to be *linear* if $\gamma(x) = x$, for all $0 \leq x \leq P$.

  For all jobs with available parallelism N, $E(N) = \gamma(N)$. This is the only assumption needed for the reductions in this paper that pertain to the FCFS, PSAPF, and temporal EQ policies. The reductions for spatial EQ and the interpolation approximations in this paper require a specification of the execution rate $E(j)$ on $j < N$ processors. Several alternatives are possible. The assumption in this paper is that the work for a job can be dynamically redistributed across the number of processors allocated to it such that it executes as if it has available parallelism equal to the processor allocation, i.e., $E(j) = \gamma(j)$, for $1 \leq j < N$. This could be appropriate, for example, for applications based on the work queue model, or in certain cases where the processes of a job are timeshared on the allocated processors. Another alternative is that parallelism overhead is roughly the same on fewer processors as on N processors, i.e., $E(j) = \frac{j}{N}\gamma(N)$ for $1 \leq j < N$. This case might be appropriate, for example, for a system with

[2]The coefficient of variation of a random variable is defined as the ratio of the standard deviation to the mean.

jobs that have fixed parallelism in which overhead is primarily due to message passing software and processing load is balanced across the processors, e.g., through judicious cyclic rotation of processes. The two cases reduce to the same $E(j)$ when γ is linear, which is assumed in the reductions and interpolation approximations in this paper.

In cases where the allocated processing power, x, is nonintegral we use a linear interpolation between $\gamma(\lfloor x \rfloor)$ and $\gamma(\lceil x \rceil)$ to compute $E(x)$.

- The total job processing demand, D, is independent of N and E.
- D has a general distribution, $\mathcal{F}_D$, with mean $\overline{D}$ and coefficient of variation C_D.

The service time of a job when executing on N processors is denoted by the random variable $S = D/\gamma(N)$ with mean $\overline{S}$. Under the assumption of linear execution rates, $S = D/N$.

Important generalizations that improve both the flexibility and the potential applicability of the above workload model, compared with previous models of parallel scheduling policy performance, include the general distribution of available job parallelism, the general distribution of job demand, and the general nondecreasing execution rate function. All but one previous study have assumed specific distributions of demand and/or parallelism. Furthermore, there is a fairly simple extension to the above model to allow controlled correlation between job demand and available parallelism [21]. However, as stated in section 1 this is beyond the scope of this paper.

The workload model defined above contains three simplifying assumptions, each of which represents a trade-off between analytic tractability and the simplicity of the parameter space on the one hand, and generality of the model on the other hand. The first is the assumption of constant available parallelism per job, the second is the assumption of a fixed execution rate, $E(k)$, whenever the job is allocated k processors, and the third is the assumption of the same deterministic function γ for all jobs. The first assumption is realistic for static processor allocation policies, in which a job runs to completion on whatever number of processors is initially allocated to it. The assumption is also realistic for certain systems and/or workloads where processor allocation is dynamic. For example, if the job is based on a work queue model and can continuously adapt to any given number of processors up to a maximum value of N throughout (most of) its lifetime, or if the system scheduler assumes the job's parallelism is fixed (as in the CM-5). Similarly, the second assumption is realistic for static scheduling policies and for certain cases of dynamic scheduling (i.e., when execution rates are nearly linear and/or when parallelism overheads including load imbalance are relatively evenly distributed throughout the execution of the program, on any number of processors). Furthermore, since the purpose of the model is to analyze *scheduling policy* behavior and performance, as

opposed to obtaining precise mean response times for the applications, assumptions that approximately represent key workload characteristics while keeping the model tractable and the parameter space simple, are acceptable even when they don't precisely describe the behavior of individual applications. For example, if jobs have varying available parallelism, one can view the model with constant available parallelism as capturing the contention that occurs between phases of different jobs, where a phase is a portion of the job in which available parallelism is constant. Similarly, although jobs actually have differing degrees of sublinearity, one can view the model as representing how policy generally performs as execution rates are more or less sublinear. Extensions that would further increase the applicability of the model yet preserve its tractability and parameter simplicity would be desirable, but appear to be quite difficult to obtain.

We will use the linear ERF to illustrate the reductions and interpolation approximation technique in the remainder of the paper. This greatly simplifies the explanation of the technique yet still allows for obtaining insights from the results pertaining to the sensitivity of policy performance to job demand and parallelism parameters. We comment in section 7 on how the key parallelism parameters obtained by assuming linear execution rates generalize to the case of sublinear job execution rates. We have also derived reductions and interpolation approximations for mean response time under sublinear execution rates [21].

3.3 Notation

Table 1 provides a summary of the notation for the system and workload model defined above. The following notation specifies a particular system, denoted by Γ, and its associated workload:

$$\Gamma = (\Psi, P, \lambda, \mathcal{F}_D, \mathcal{F}_N).$$

Implicit in the above notation is the assumption of Poisson job arrivals, independence between D and N, and linear execution rates. To indicate a general distribution of demand or available parallelism we simply leave the notation as $\mathcal{F}_D$ or $\mathcal{F}_N$, respectively. To indicate a system with specific distribution functions, we will use notation that should be widely understood, such as $N = 1$ or Uniform(1,P) for the available parallelism distribution, and exp(μ) or H_2 for the distribution of D.

Table 1: System Notation

Ψ	Processor allocation policy (e.g., EQ, FCFS, PSAPF)
P	Number of processors in the system
λ	Arrival rate of jobs
D	Total job demand
$\mathcal{F}_D$	Distribution of job demand
$\overline{D}$	Mean job demand
C_D	Coefficient of variation in job demand
ρ	System utilization
N	Available job parallelism
$\mathcal{F}_N$	Distribution of available parallelism
p_i	Probability$[N = i]$, $i = 1, \ldots, P$
$\underline{p}$	$(p_1, p_2, \ldots, p_P)$
$\overline{N}$	Mean available parallelism across all jobs
$\gamma()$	Execution rate function
S	Job service time on N processors
$\overline{S}$	Mean job service time
$\overline{R}_\Psi$	Mean response time for policy $\Psi \in \{EQ, FCFS, PSAPF\}$
$\hat{R}^x_\Psi$	Estimator for mean response time under policy Ψ, obtained using an interpolation approximation on parameter x
$M/G/1_P$	An $M/G/1$ system with a processor of capacity P.

4 Reductions to Queueing Systems with Known Solutions: FCFS, EQ

In this section we show how the parallel system model, under the FCFS or EQ scheduling policy, reduces to queueing systems with known solutions for particular extreme values of the model parameters. We first review the queueing systems with known solutions that are used in the reductions. We then present the reductions followed by a summary of the results obtained from these reductions.

4.1 Queueing Systems with Known Solutions

4.1.1 The M/G/c Queue

Consider an open multiserver queue with sequential work as shown in figure 2. We consider the special case of an M/G/c queue in which jobs arrive according to a Poisson process with rate λ, and have i.i.d. service times with mean $\overline{x}$ and coefficient of variation C_x. Server utilization is given by $\rho = \lambda\overline{x}/c$, c being the number of servers.

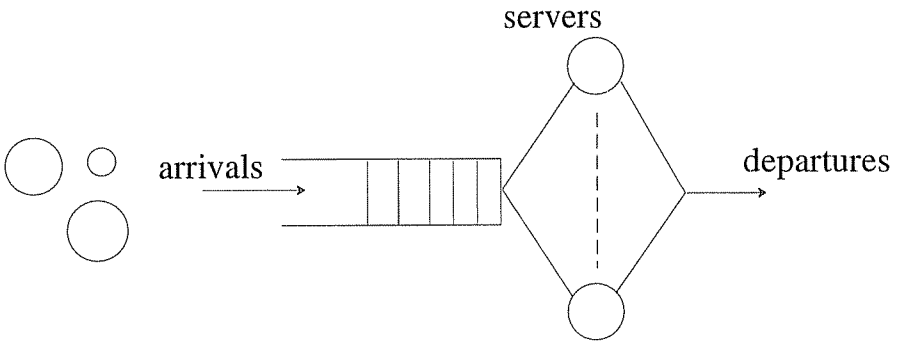


Figure 2: Multiserver queue with sequential work

There is no known exact solution of the mean response time of the M/G/c (FCFS) queue. As a result there have been a number of approximations for $\overline{R}_{M/G/c}$ in the literature [32, 39, 40, 48, 49]. Of particular interest to us is the simple approximation proposed in [32] for the mean number in a GI/G/c queue, which leads to the following approximate formula for $\overline{R}_{M/G/c}$:

$$\overline{R}_{M/G/c} \approx \overline{x} + \frac{\rho\sqrt{2(c+1)}(1+C_x^2)}{2\lambda(1-\rho)}. \tag{1}$$

Note that this approximation is exact for $c = 1$ and $c = \infty$. Using this approximation and the fact that $\overline{R}_{M/G/c\ PS} = \overline{R}_{M/M/c}$ [33], one can derive the following approximation:

$$\overline{R}_{M/G/c\ PS} \approx \overline{x} + \frac{\rho\sqrt{2(c+1)}}{\lambda(1-\rho)}. \tag{2}$$

We note that this approximation has a much simpler form than the exact expression for mean response time in the $M/G/c\ PS$ queue. It is also exact for $c = 1$ and very accurate as shown by validations in [32] for the M/M/c queue. We use the approximate expressions given by (1) and (2) for the reductions in section 4.2.

4.1.2 The Symmetric Queue

Kelly defines a queue to be *symmetric* if it operates in the following manner [11].

(i) The service requirement of a job is a random variable whose distribution may depend upon the class of the job.

(ii) A total service effort is supplied at the rate $\phi(j)$, where j is the total number of jobs in the queue.

(iii) A proportion $\alpha(l, j)$ of this effort is directed to the job in position l ($l = 1, 2, \ldots, j$); when this job leaves the queue, jobs in positions $l+1, l+2, \ldots, j$ move to positions $l, l+1, \ldots, j-1$, respectively.

(iv) When a job arrives at the queue it moves into position l ($l = 1, 2, \ldots, j+1$) with probability $\alpha(l, j+1)$; jobs previously in positions $l, l+1, \ldots, j$ move to positions $l+1, l+2, \ldots, j+1$, respectively, where j is the total number of jobs in the queue as seen by the arrival.

Note that ϕ and α are parameters of the symmetric queue.

Theorems 3.8 and 3.10 of [11] state that for a stationary symmetric queue with a Poisson arrival process with rate λ and an arbitrary distribution of job service time, S, the steady state probability of i jobs in the queue is given by

$$\pi_i = \frac{ba^i}{\prod_{l=1}^{i} \phi(l)}, \quad i = 0, 1, 2, \ldots \tag{3}$$

where

$$a = \lambda \overline{S}, \quad \text{and} \quad b = \left[\sum_{i=0}^{\infty} \frac{a^i}{\prod_{l=1}^{i} \phi(l)} \right]^{-1}.$$

The steady state probabilities can be used to obtain the mean number in the system and thereby the mean response time.

4.2 Reductions for EQ and FCFS

We consider two types of points in the parameter space for finding reductions. The first is when available parallelism, N, is constant across all jobs. The second is light and heavy traffic limits. For constant parallelism and light traffic, we find reductions for FCFS and EQ; the heavy traffic limit is derived for EQ and for a restricted case for FCFS.

4.2.1 Constant Available Parallelism ($N = k$)

By constant available parallelism we mean that $N = k$, $1 \leq k \leq P$, for all jobs in the system. The mean response time under EQ and FCFS for $N = k$, when k evenly divides P is given by the following proposition.

Proposition 4.1

$$\overline{R}_{EQ}(N=k) = \overline{R}_{M/G/c\ PS}, \qquad \textit{where } c = \frac{P}{k},\ \ P \bmod k = 0. \tag{4}$$

$$\overline{R}_{FCFS}(N=k) = \overline{R}_{M/G/c}, \qquad c = \frac{P}{k},\ \ P \bmod k = 0. \tag{5}$$

In particular,

$$\overline{R}_{EQ}(N=1) = \overline{R}_{M/G/P\ PS}, \qquad \overline{R}_{EQ}(N=P) = \overline{R}_{M/G/1_P\ PS}$$

$$\overline{R}_{FCFS}(N=1) = \overline{R}_{M/G/P}, \qquad \overline{R}_{FCFS}(N=P) = \overline{R}_{M/G/1_P}$$

Proof. See appendix. ∎

The reduction for $EQ(N=k)$ in Proposition 4.1 holds only when $P \bmod k = 0$. The following reduction holds for all $k = 1, 2, \ldots, P$.

Proposition 4.2

$$\overline{R}_{EQ}(N=k) = \overline{R}_{Symmetric\ queue}[\phi(j) = \min(j \cdot k, P),\ \alpha(l,j) = 1/j], \quad k = 1, 2, \ldots, P.$$

Proof. See appendix. ∎

From Proposition 4.1 we estimate the mean response time of EQ when $N = k$ and $P \bmod k = 0$, by using expression (2) with $\overline{x} = \overline{D}/k$, and the mean response time of $FCFS$ by using expression (1) with $\overline{x} = \overline{D}/k$ and $C_x = C_D$. These expressions can be evaluated even when c is not an integer, yielding a simpler expression for EQ than the exact results from Proposition 4.2, as follows:

$$\overline{R}_{EQ}(N=k) \approx \frac{\overline{D}}{k} + \frac{\rho\sqrt{2(\frac{P}{k}+1)}}{\lambda(1-\rho)}, \quad k = 1, 2, \ldots, P, \tag{6}$$

$$\overline{R}_{FCFS}(N=k) \approx \frac{\overline{D}}{k} + \frac{\rho\sqrt{2(\frac{P}{k}+1)}(1+C_D^2)}{2\lambda(1-\rho)}, \quad k = 1, 2, \ldots, P. \tag{7}$$

Note that both these approximations are exact when $N = P$ since approximations (2) and (1) are exact for $c = 1$. We tested the validity of the EQ approximation (6) using the exact expression from the *symmetric queue* reduction [11], and found the relative errors to be typically less than 2%.

An important observation from approximations (6) *and* (7) is that $\overline{R}_{EQ}(N=k)$ depends only on mean demand ($\overline{D}$), whereas $\overline{R}_{FCFS}(N=k)$ depends on C_D^2 as well as $\overline{D}$

4.2.2 Light and Heavy Traffic ($\rho = 0, \rho = 1$)

At light traffic, that is, as $\rho \to 0$, the mean response time under either EQ or FCFS is simply the mean job service time on N processors, $\overline{S}$. Since $S = D/N$ and D and N are independent, we have

$$\lim_{\rho \to 0} \overline{R}_{EQ} = \lim_{\rho \to 0} \overline{R}_{FCFS} = \overline{S} = \overline{D}E[1/N]. \tag{8}$$

We present an informal derivation of the mean response time under heavy traffic for the EQ policy. As $\rho \to 1$, an arriving job finds greater than P jobs in the system with probability 1. Hence the processing power allocated to each job in the system under EQ is less than 1 when $\rho \to 1$. In this case the available parallelism of a job does not have an impact on system performance. In particular, when $\rho \to 1$ the mean system response time for any distribution of N reduces to the mean response time when $N = P$. By Proposition 4.1, $\overline{R}_{EQ}(N = P) = \overline{R}_{M/G/1_P\ PS} = (\overline{D}/P)/(1 - \rho)$ which follows from setting $c = 1$ and $\overline{x} = \overline{D}/P$ in (2). Thus, we obtain the following heavy traffic limit[3]:

$$\lim_{\rho \to 1} (1 - \rho)\overline{R}_{EQ} = \frac{\overline{D}}{P}. \tag{9}$$

We do not have a corresponding general heavy traffic limit for the FCFS policy. However, for the case of constant available parallelism we can obtain the following approximate heavy traffic limit from (7):

$$\lim_{\rho \to 1}(1 - \rho)\overline{R}_{FCFS}(N = k) \approx \frac{(1 + C_D^2)}{2}\frac{\overline{D}}{P}, \quad k = 1, 2, \ldots, P.$$

We note that this heavy traffic limit does not depend on k.

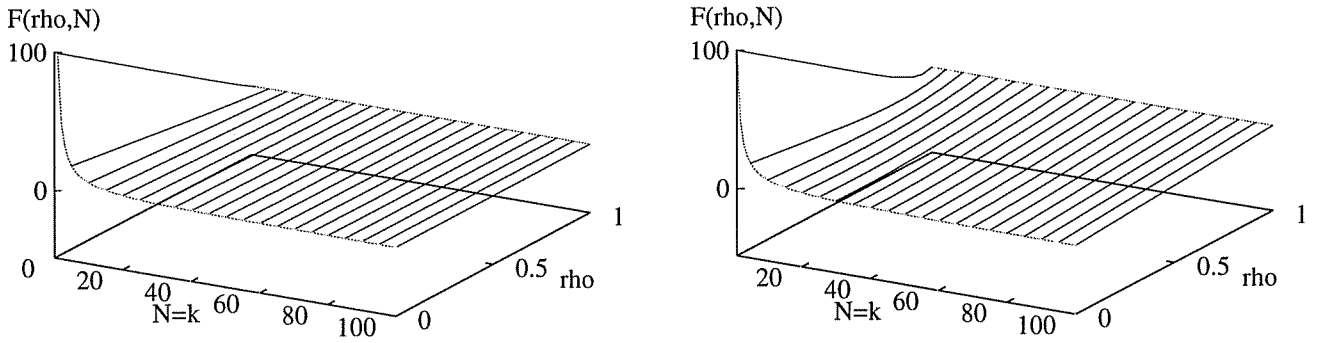
4.2.3 Summary of Results for EQ and FCFS

To summarize the results of the reductions derived thus far, figures 3(a) and (b) plot the *normalized* mean response time[4], $F(\rho, k) = (1 - \rho)\overline{R}_{\Psi}(\rho, N = k)$, $\Psi \in \{EQ, FCFS\}$, where $k = 1, 2, \ldots, P$ denotes the fixed value of parallelism assumed for all jobs. The curves are plotted for $P = 100$, and mean job demand $\overline{D} = P = 100$. Figure 3(a) contains the curves for the EQ policy, and for the FCFS policy when $C_D = 1$. (Note that the EQ curves hold for all values of C_D, and that the reductions for the FCFS policy yield the same values when $C_D = 1$.) Figure 3(b) contains the curves for the FCFS policy when $C_D = 5$.

[3]The independence assumption between D and N simplifies the derivation, but the result also holds for correlated workloads.

[4]The reason for normalizing the mean response time is that we can observe the behavior at low as well as very high utilizations on the same plot.

Several points are worth noting about the results in figure 3. First, for both policies and all values of C_D, $F(0,N) = \overline{D}E[1/N]$, which is equal to $\overline{D}/k$ when all jobs have parallelism k. Second, since $F(1,N)$ is equal to $\overline{D}/P$ for EQ and $F(1,N=k)$ is equal to $\overline{D}(1+C_D^2)/(2P)$ for FCFS, the curve for normalized mean response time at $\rho = 1$ is flat in both plots. Finally, for the EQ policy $F(\rho,P)$ is equal to $\overline{D}/P$, which yields a curve of constant value for $N = P$ in figure 3(a).



(a) EQ, $C_D \geq 0$
FCFS, $C_D = 1$

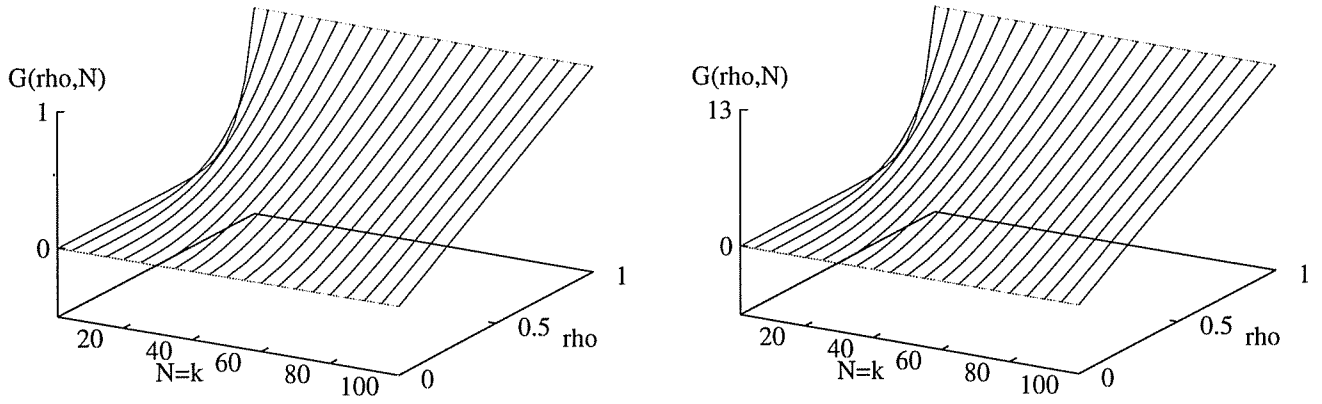
(b) FCFS, $C_D = 5$

Figure 3: Normalized Mean Response Time

$\overline{D} = P = 100$

In figure 4(a) and (b), we've plotted the normalized mean *extra* time, $G(\rho,k) = (1-\rho)\overline{X}_\Psi(\rho, N=k)$, for constant parallelism $k = 1, 2, \ldots, P$, and all other parameters as in figure 3(a). The extra time, $X = R - S$, is the time spent in the system other than the service time S. In other words, X is the penalty incurred due to resource contention. The mean extra time is thus given by $\overline{X} = \overline{R} - \overline{S}$, which equals $\overline{R} - \overline{D}/k$ when $N = k$. Note that the range on the Y-axis in figure 4(b) is 13 times that in figure 4(a) due to the influence of C_D^2 on system performance for the FCFS policy. We observe that $G(\rho,N)$ is constant at extreme values of ρ (0 at $\rho = 0$ and $\overline{D}/P$ at $\rho = 1$). For extreme values of N, it is linear for $N = P$, but highly convex for $N = 1$. That is, when $N = P$, $G(\rho,P) = \rho\overline{D}/P$ for EQ and $\rho(1+C_D^2)/(2P)$ for FCFS, and when $N = 1$, $G(\rho,1) = (1-\rho)\overline{W}_{M/M/P}$ for EQ and $(1-\rho)\overline{W}_{M/G/P}$ for FCFS as seen from Proposition 4.1.

In the next section we will interpolate between the response time and extra time values obtained for



(a) EQ, $C_D \geq 0$
FCFS, $C_D = 1$

(b) FCFS, $C_D = 5$

Figure 4: Normalized Mean Extra Time

$\overline{D} = P = 100$

particular points in the system parameter space. The plots in figures 3 and 4 will aid in determining how the interpolations should be constructed.

5 Interpolation Approximations for $\overline{R}_{EQ}$ and $\overline{R}_{FCFS}$

In this section we use the reductions of the previous section to derive interpolation approximations for $\overline{R}_{EQ}$ and $\overline{R}_{FCFS}$ that hold over the entire range of the system parameter space. We first consider interpolation on ρ to derive an approximation for $\overline{R}_{EQ}$. Second, we consider interpolations on $\overline{N}$ for both policies. Third, we derive interpolations on the distribution of N, $\underline{p} = (p_1, \ldots, p_P)$, for both policies. The interpolations are followed by validations using simulation and exact analysis. All three interpolations for EQ are exact when $\rho \to 1$, i.e., they yield the heavy traffic limit for EQ given by (9).

5.1 EQ: Interpolation on ρ

Let $F(\rho) = (1 - \rho)\overline{R}_{EQ}(\rho)$. The light and heavy traffic limits, $F(0)$ and $F(1)$, are given in equations (8) and (9) of section 4.2.2. Figures 3(a) and 4(a) suggest that a linear interpolation between $F(0)$ and $F(1)$

would be more accurate than a linear interpolation between $G(0)$ and $G(1)$, and that the former interpolation may be reasonably accurate (particularly for workloads with moderate to high parallelism). We thus proceed to define this interpolation.

A linear interpolation between $F(0)$ and $F(1)$ yields the following estimator for $F(\rho)$.

$$\begin{aligned}\hat{F}(\rho) &= (1-\rho)F(0) + \rho F(1) \\ &= (1-\rho)\overline{S} + \rho\overline{D}/P.\end{aligned}$$

Dividing $\hat{F}(\rho)$ by $(1-\rho)$ we obtain the desired estimator,

$$\begin{aligned}\overline{R}_{EQ} \approx \hat{R}^{\rho}_{EQ} = \frac{\hat{F}(\rho)}{1-\rho} &= \overline{S} + \frac{\rho}{1-\rho}\frac{\overline{D}}{P} \\ &= \overline{D}E[1/N] + \frac{\rho}{1-\rho}\frac{\overline{D}}{P}. \qquad (10)\end{aligned}$$

We note that this approximation is exact for the special case when $N = P$, which is easily seen by comparing equations (6) and (10) when $N = P$.

5.2 EQ and FCFS: Interpolation on $\overline{N}$

The next interpolation is applicable to both policies and uses the results derived in section 4.2 for extreme values of available parallelism ($N = 1$ and $N = P$), where $\overline{N} = 1$ and $\overline{N} = P$, respectively. Figures 3 and 4 suggest that a simple linear interpolation on $\overline{N}$ is likely to be more accurate if the approximation is for the mean extra time than for the mean response time, particularly for light to moderate traffic. We thus proceed to define this interpolation.

Let Ψ denote one of EQ or FCFS, and let $\overline{X}_{\Psi} = \overline{R}_{\Psi} - \overline{S}$. A linear interpolation on $\overline{N}$ yields the following estimator for $\overline{X}_{\Psi}$,

$$\hat{X}^{\overline{N}}_{\Psi} = \left(\frac{P-\overline{N}}{P-1}\right)\overline{X}_{\Psi}(\overline{N}=1) + \left(\frac{\overline{N}-1}{P-1}\right)\overline{X}_{\Psi}(\overline{N}=P), \qquad (11)$$

where $\overline{X}_{\Psi}(\overline{N}=1)$ and $\overline{X}_{\Psi}(\overline{N}=P)$ are derived from equations (6) and (7), by setting $k = 1$ and $k = P$, i.e.,

$$\overline{X}_{EQ}(\overline{N}=1) \approx \frac{\rho\sqrt{2(P+1)}}{\lambda(1-\rho)}, \qquad \overline{X}_{EQ}(\overline{N}=P) = \frac{\rho}{1-\rho}\frac{\overline{D}}{P}.$$

$$\overline{X}_{FCFS}(\overline{N}=1) \approx \frac{\rho\sqrt{2(P+1)}(1+C_D^2)}{2\lambda(1-\rho)}, \quad \overline{X}_{FCFS}(\overline{N}=P) = \frac{\rho(1+C_D^2)}{2(1-\rho)}\frac{\overline{D}}{P}.$$

Substituting the above values in equation (11), the full interpolation approximations are:

$$\overline{R}_{EQ} \approx \hat{R}^{\overline{N}}_{EQ} = \overline{D}E[1/N] + \left(\frac{P-\overline{N}}{P-1}\right)\frac{\rho\sqrt{2(P+1)}}{\lambda(1-\rho)} + \left(\frac{\overline{N}-1}{P-1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}. \quad (12)$$

$$\overline{R}_{FCFS} \approx \hat{R}^{\overline{N}}_{FCFS} = \overline{D}E[1/N] + \left\{\left(\frac{P-\overline{N}}{P-1}\right)\frac{\rho\sqrt{2(P+1)}}{\lambda(1-\rho)} + \left(\frac{\overline{N}-1}{P-1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}\right\}\left(\frac{1+C_D^2}{2}\right). \quad (13)$$

Note in the above approximations that $\hat{X}^{\overline{N}}_{FCFS} = \hat{X}^{\overline{N}}_{EQ}(1+C_D^2)/2$.

5.3 EQ and FCFS: Interpolation on $\underline{p}$

We now derive interpolation approximations for EQ and FCFS that use all of the reductions for constant available parallelism, $N = k$ for $k = 1, 2, \ldots, P$, derived in section 4.2. These approximations are more accurate than the previous interpolations on $\overline{N}$, as will be shown by validations.

The systems with constant parallelism have extreme values for the distribution of N, that is, $\underline{p} = \underline{e}_k$, $1 \leq k \leq P$, where $\underline{e}_k$ is a vector of length P having a 1 in the k^{th} component and $0's$ for all other components. An interpolation through the mean response times at these extreme points ($\overline{R}_\Psi(N=k)$) yields the following form of approximation for both policies.

$$\overline{R}_\Psi \approx \hat{R}^{\underline{p}}_\Psi = \sum_{k=1}^{P} p_k \overline{R}_\Psi(N=k).$$

From approximation (6) for $\overline{R}_{EQ}(N=k)$ (section 4.2) we get

$$\hat{R}^{\underline{p}}_{EQ} = \sum_{k=1}^{P} p_k \left\{\frac{\overline{D}}{k} + \frac{\rho\sqrt{2(\frac{P}{k}+1)}}{\lambda(1-\rho)}\right\} = \overline{D}E[1/N] + \frac{E\left[\rho\sqrt{2(\frac{P}{N}+1)}\right]}{\lambda(1-\rho)}. \quad (14)$$

Similarly, from approximation (7) for $\overline{R}_{FCFS}(N=k)$ (section 4.2) we get

$$\hat{R}^{\underline{p}}_{FCFS} = \sum_{k=1}^{P} p_k \left\{\frac{\overline{D}}{k} + \frac{\rho\sqrt{2(\frac{P}{k}+1)}}{\lambda(1-\rho)}\frac{(1+C_D^2)}{2}\right\} = \overline{D}E[1/N] + \frac{E\left[\rho\sqrt{2(\frac{P}{N}+1)}\right]}{\lambda(1-\rho)}\left(\frac{1+C_D^2}{2}\right). \quad (15)$$

We again note that $\hat{X}^{\underline{p}}_{FCFS} = \hat{X}^{\underline{p}}_{EQ}(1+C_D^2)/2$.

As in the interpolations on ρ and $\overline{N}$, the interpolation on $\underline{p}$ is an ad hoc approximation. There is, however, reason to believe that it can be more accurate. First, it uses P data points for interpolation as compared

to 2 each for the interpolations on ρ and $\overline{N}$. Second, from figure 3 we note that the mean response time of EQ and FCFS when $N = k$ changes very gradually with k in the range of moderate to high k. A linear combination of these mean response times could thus be expected to be an accurate estimator for workloads where all jobs have moderate to high parallelism. Third, when N takes on one of two extreme values, either 1 or P, the interpolation on $\underline{p}$ reduces to the interpolation on $\overline{N}$. Thus we might expect the interpolation on $\underline{p}$ to be accurate when C_N is low (e.g., constant N or N between two values of k that are moderate to high) and to perform as well as the interpolation on $\overline{N}$ when C_N is high (e.g., N takes on one of two extreme values). Validations will show that this intuition is largely correct and that the interpolation on $\underline{p}$ is in fact significantly more accurate than the interpolations on ρ and $\overline{N}$.

5.4 Model Validations

We validate the analytic interpolation approximations for the mean response time under EQ and FCFS against simulation results and against special cases of exact analysis. We first provide the parameter settings for the validation experiments, after which we present a summary of validations, and finally we present error plots for example parameter settings.

5.4.1 Validation parameter settings

For all validations, $\overline{D}$ is set to P. We varied the other model parameters as follows:

(i) P: 20,100,500,and 1000.

(ii) $\mathcal{F}_D$: Exponential, and 2-stage Hyperexponential (H_2) with $C_D = 5$.

As will be shown, the inaccuracy of the approximations for the FCFS policy increases as C_D increases. Thus, $C_D = 5$ serves as a stress test for those approximations. We also ran a few test experiments, and found no appreciable difference between the observed errors for cases with deterministic or two-stage Erlang demand distributions compared to cases for the exponential distribution, and no appreciable differences in observed errors for cases with Gamma ($C_D = 5$) distributions of job demand as compared with the cases with H_2.

(iii) ρ: 0.1 to 0.9. (Since $\overline{D} = P$, $\rho = \lambda$.)

(iv) $\mathcal{F}_N$: bounded-geometric, constant, and uniform.

The bounded-geometric distribution [17, 15], is specified by

$$N = \begin{cases} P, & \text{with probability } P_{max}, \\ \min(G, P), & \text{with probability } 1 - P_{max}, \end{cases} \qquad \text{where } G = \text{Geometric(p)}.$$

In the validations we ensured coverage of extreme values of C_N and $\overline{N}$ which served as stress tests.

Table 2 and 3 list the parameter settings for all distributions of N considered in the validations. In table 2 the parameter settings for the bounded geometric distributions are arranged in three groups of three, and within each group in decreasing $\overline{N}$. It can be shown that for a fixed value of $\overline{N}$, the bounded-geometric distribution with lowest C_N has $P_{max} = 0.0$ and the bounded-geometric distribution with highest C_N has $p = 1$ [20]. Thus, the first group of three are low C_N workloads, the last group are high C_N workloads, and the middle group are workloads with intermediate C_N. There are fewer workloads in table 3 than in table 2 mainly because the simulations were very time-consuming for $P = 500, 1000$. However, workloads for which significant errors were observed in the approximations at $P = 20, 100$ are also included in the $P = 500, 1000$ experiments.

Table 2: Validation Workloads for N: P=20,100

Distribution	**Parameter Settings**									
Bounded-	P_{max}	0.0	0.0	0.0	0.1	0.1	0.1	0.9	0.5	0.1
Geometric	p	0.005	1/(0.5P)	1/(0.1P)	0.01	1/(0.4P)	0.9	1	1	1
Constant	N=1, N=P/4, N=P/2, N=3P/4, N=P									
Uniform	(1,P), (1,P/2), (P/2,P)									

Table 3: Validation Workloads for N: P=500,1000

Distribution	**Parameter Settings**			
Bounded-	P_{max}	0.9	0.1	0.1
Geometric	p	1	1/(0.4P)	0.9
Constant	N=P/10, N=P/4, N=P/2, N=3P/4, N=P			
Uniform	(1,P), (1,P/2)			

All approximations were validated against exact analysis when $N = k$, and against simulation otherwise. Exact estimates for $\overline{R}_{EQ}(N = k)$ were obtained by reducing the system to a *symmetric queue* (see Proposition 4.2). Exact estimates for $\overline{R}_{FCFS}(N = k)$ were obtained using matrix-geometric analysis [29, 26, 38]. For the estimates obtained by simulation almost all had 95% confidence intervals with less than 5% half-widths [14]. To obtain the confidence intervals, we used the regenerative method for many of the data points and the method of batch means whenever the regenerative method was too time consuming.

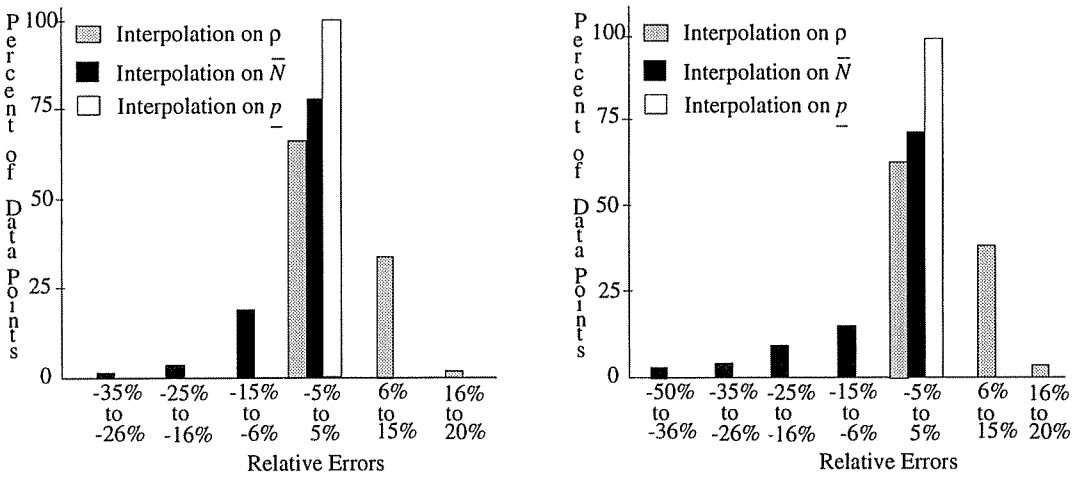
5.4.2 Summary of Validation Results

Figures 5 and 6 present histograms that summarize all of the validation experiments for the EQ and FCFS approximations. The total number of data points for the EQ validations was 306 for P=20,100, and 172 for P=500,1000. The same is true for FCFS at each value of $C_D = 1$ and $C_D = 5$, thus leading to a total of 956 validations for FCFS.[5]

First, consider the EQ histograms in figure 5. Since simulation estimates for $\overline{R}_{EQ}$ were statistically the same for different values of C_D we do not specify any value of C_D in the histograms for EQ. We observe that all three approximations for $\overline{R}_{EQ}$ are fairly accurate for small and large numbers of processors, and that the interpolation on $\underline{p}$ has extremely low error for all cases examined. In fact, the maximum relative error that was observed for $\hat{R}^{\underline{p}}_{EQ}$ was only -2.6%. The interpolation on $\overline{N}$ tends to underestimate $\overline{R}_{EQ}$ and the interpolation on ρ tends to overestimate $\overline{R}_{EQ}$. These trends can be predicted from the plots in figure 4(a). The worst case errors for the interpolation on $\overline{N}$ were for $(N = P/4,\ \rho = 0.9)$. This is consistent with the data in figure 4(a), noting that the error at higher ρ will be magnified when the normalized mean extra time is divided by $1 - \rho$. The worst case errors for the interpolation on ρ were for $(N = P/4,\ \rho = 0.7)$, which is also consistent with the data in figure 4(a), noting that as $\overline{N}$ decreases the mean response time is dominated by the mean job service time (e.g., at $\overline{N} = 1$). Note that for these cases of constant N the interpolation on $\underline{p}$ is extremely accurate.

Now consider the FCFS histograms in figure 6. We first note that for $C_D = 1$ the FCFS histograms are almost the same as the EQ histograms. The worst case errors at $C_D = 1$ for $\hat{R}^{\overline{N}}_{FCFS}$ were for the same workloads as the worst case errors for $\hat{R}^{\overline{N}}_{EQ}$. Comparing the results for $C_D = 5$ we note that the performance of both the FCFS approximations degrades with C_D. However, most of the data points are still within an

[5] Many simulation experiments were run on the Condor distributed system [1].



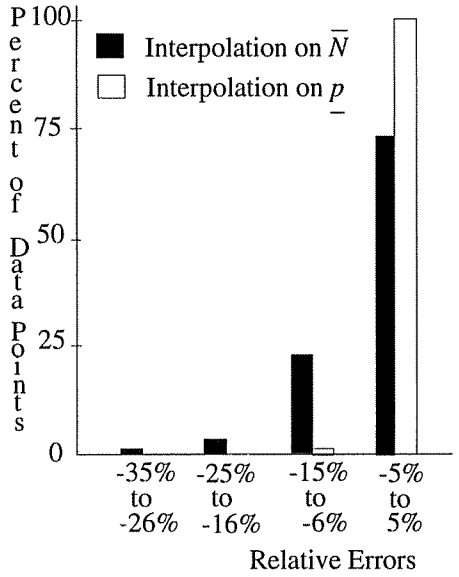
(a) P=20,100 (b) P=500,1000

Figure 5: Summary of Validations: EQ

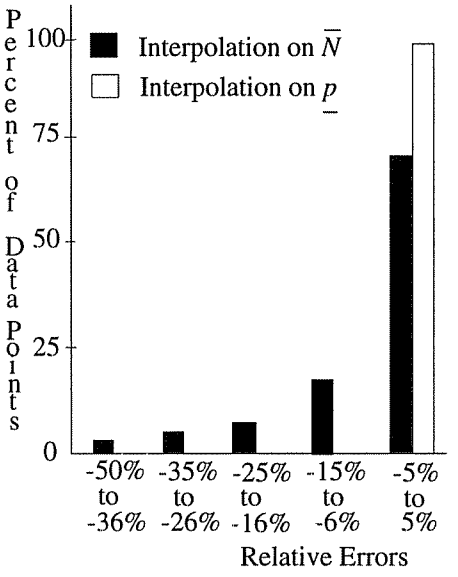
acceptable range of error, i.e., within -5% to 35% error for the interpolation on $\underline{p}$ and within $\pm 35\%$ for the interpolation on $\overline{N}$. We also observe that in general the interpolation on $\underline{p}$ is more accurate than the interpolation on $\overline{N}$ and that the interpolation on $\underline{p}$ overestimates mean response time (i.e., is conservative) in the majority of cases examined. At $C_D = 5$, the worst case errors for the interpolation on $\overline{N}$ were located at ($P = 100,\ N = 75,\ \rho = 0.2$) and ($P = 1000,\ N = 100,\ \rho = 0.9$). Interestingly, the worst case errors for the interpolation on $\underline{p}$ were also located at constant N, that is, ($N = 3P/4,\ \rho = 0.2$). This is non-intuitive since $\hat{R}^{\underline{p}}_{FCFS}$ interpolates among $\overline{R}_{FCFS}(N = k)$ and we had an *off-the-shelf* solution available for $\overline{R}_{FCFS}(N = k)$. The explanation is that approximation (7) for $\overline{R}_{FCFS}(N = k)$ turns out to be somewhat inaccurate at high C_D, low to moderate utilization, and k between $P/4$ and $3P/4$. The trade-offs between accuracy and simple approximations that readily yield insight still favor the use of this available solution for the M/G/c queue, but the validation results suggest that the approximation for FCFS scheduling in a parallel system could be improved if a more accurate closed-form approximation can be found for the M/G/c queue.

5.4.3 Example Validation Experiments

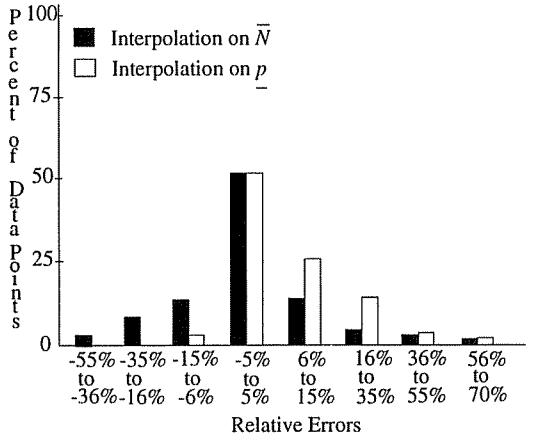
To illustrate how the interpolation approximation accuracy varies with various model parameters, we present example plots of relative error versus utilization for specific distributions of N, specific values of P, and in the case of the FCFS policy, specific values of C_D. The distributions of N considered are bounded-geometric



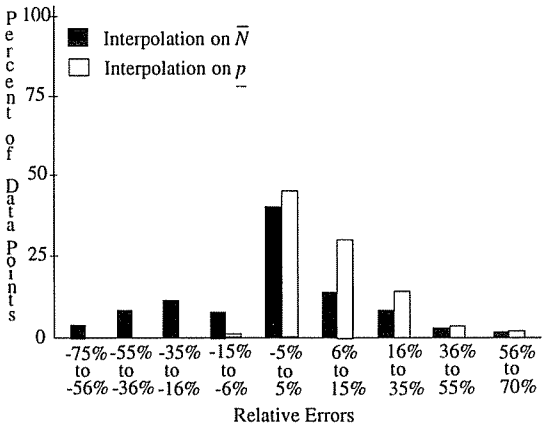
(a) Cd=1: P=20,100



(b) Cd=1: P=500,1000



(c) Cd=5: P=20,100



(d) Cd=5: P=500,1000

Figure 6: Summary of Validations: FCFS

with parameter settings given in table 4. Note that these workloads have high (H), moderate (M) and low (L) average parallelism, respectively. We found the errors for these three workloads to be fairly representative for bounded-geometric distributions. We observed that the accuracy of the interpolation on $\overline{N}$ decreases with decrease in C_N, this is also true for the uniform distribution. For the constant N distribution C_N is lowest and the errors were also higher for the interpolation on $\overline{N}$. For the interpolation on $\underline{p}$ the constant N distribution reflects errors in the reductions rather than in the interpolation itself.

Table 4: Three example workloads for N

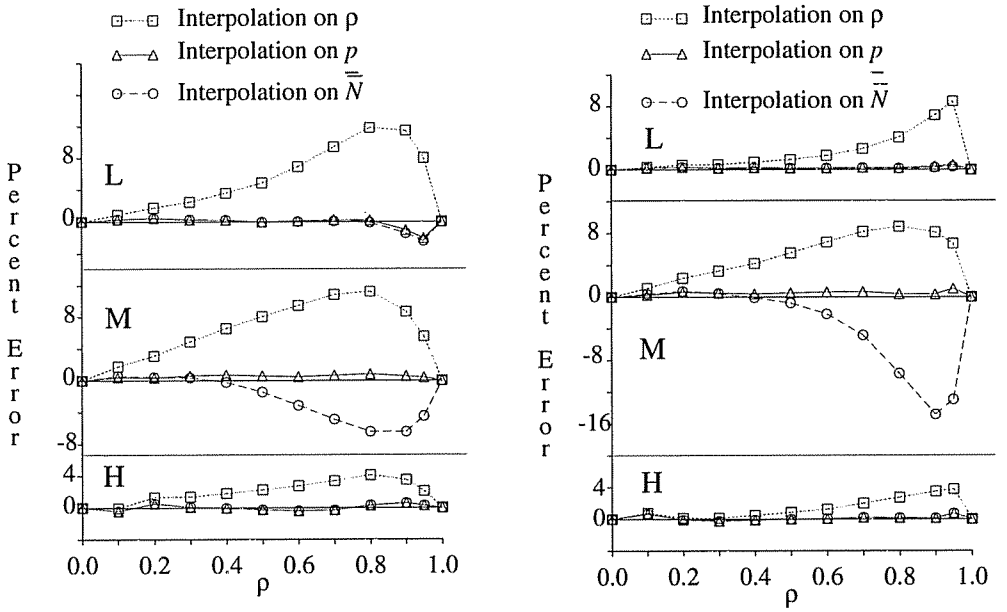
Symbol	Parallelism	P_{max}	p	$\underline{P}$=20		$\underline{P}$=100	
				$\overline{N}$	C_N	$\overline{N}$	C_N
H	High	0.9	1.0	18.10	0.31	90.10	0.33
M	Moderate	0.1	1/(0.4P)	8.70	0.77	43.14	0.80
L	Low	0.1	0.9	3.00	1.89	11.00	2.70

In figures 7(a) and (b) we plot the relative percent error for each of the three interpolation approximations for $\overline{R}_{EQ}$ as compared to simulation estimates, for the H, M, and L workloads. These figures show that, as expected, the interpolation on ρ accurately predicts $\overline{R}_{EQ}$ for the H workload, but overestimates $\overline{R}_{EQ}$ for the M and L workloads. The interpolation on $\overline{N}$ is accurate for the H and L workloads as expected, but it underestimates $\overline{R}_{EQ}$ for the M workload. The interpolation on $\underline{p}$ is the most accurate approximation and its estimation is very close to the simulated values.

Figure 8 presents example percent errors for the FCFS interpolation approximations for $C_D = 1, 5$ and $P = 100$. We observe that the interpolation on $\underline{p}$ performs fairly well for all three example workloads, with errors within 10% of the simulation estimates for both low and high C_D. The interpolation on $\overline{N}$ performs as well for the H and L workloads, but its accuracy is significantly lower for the M workload when $\rho > 0.5$.

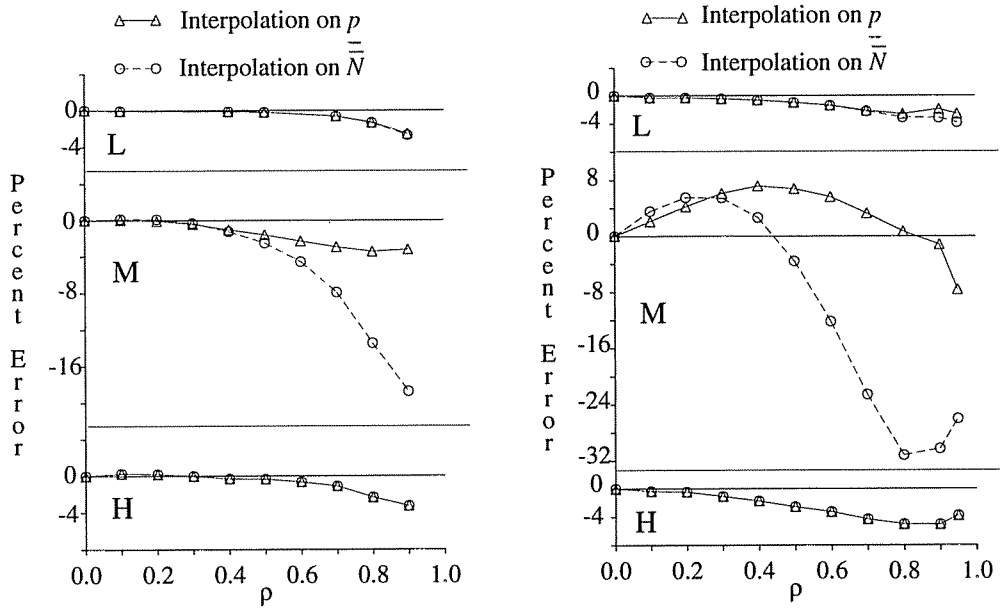
6 Analysis for PSAPF

In this section we consider interpolation approximations for the PSAPF policy. The analysis using interpolation approximations thus further illustrates the utility of this approach for analyzing and understanding the relative performance of parallel scheduling policies. We first present reductions for PSAPF and then use the reductions to derive interpolation approximations for $\overline{R}_{PSAPF}$. As before we validate the approximations



(a) P=20 (b) P=100

Figure 7: Example Validations for EQ



(a) $C_D = 1$ (b) $C_D = 5$

Figure 8: Example Validations for FCFS, $P = 100$

using simulation and exact analysis.

6.1 Reductions

We derive reductions for PSAPF under the case of constant available parallelism. When all jobs have the same available parallelism the PSAPF policy reduces to simple FCFS scheduling. Hence the reductions for PSAPF when $N = k$ are the same as the reductions for FCFS that were presented in section 4.2. Thus,

$$\overline{R}_{PSAPF}(N = k) \;=\; \overline{R}_{M/G/c}, \quad c = \frac{P}{k}, \;\; P \bmod k = 0.$$

In particular,

$$\overline{R}_{PSAPF}(N = 1) = \overline{R}_{M/G/P}, \quad \text{and} \quad \overline{R}_{PSAPF}(N = P) = \overline{R}_{M/G/1_P}.$$

Using the M/G/c approximation in (1), the reduction for $\overline{R}_{PSAPF}(N = k)$ is thus as in (7), i.e.,

$$\overline{R}_{PSAPF}(N = k) \approx \frac{\overline{D}}{k} + \frac{\rho\sqrt{2(\frac{P}{k}+1)}(1 + C_D^2)}{2\lambda(1-\rho)}, \quad k = 1, 2, \ldots, P. \tag{16}$$

Note that the fact that PSAPF reduces to FCFS when all jobs have constant parallelism enables the use of interpolation approximations to analyze a policy that might otherwise be very difficult to analyze. Also note that the reductions for the PSAPF policy are summarized in figures 3 and 4.

6.2 Interpolation Approximations

The estimates for $\overline{R}_{PSAPF}(N = k)$ can now be interconnected to yield interpolation approximations for $\overline{R}_{PSAPF}$ over the entire parameter space. As before, the reductions at constant parallelism provide the basis for two types of interpolations: (1) interpolation on $\overline{N}$ between the endpoints $\overline{X}_{PSAPF}(N = 1)$ and $\overline{X}_{PSAPF}(N = P)$, and (2) interpolation on $\underline{p}$ among all of the reductions $\overline{R}_{PSAPF}(N = k)$. Furthermore, since the workloads analyzed in this paper have no correlation between demand and parallelism, we will again derive a simple linear interpolation on $\overline{N}$ and a simple weighted sum interpolation on $\underline{p}$, yielding:

$$\hat{R}^{\overline{N}}_{PSAPF} \;\approx\; \overline{S} + \left(\frac{P - \overline{N}}{P - 1}\right)\overline{X}_{PSAPF}(\overline{N} = 1) + \left(\frac{\overline{N} - 1}{P - 1}\right)\overline{X}_{PSAPF}(\overline{N} = P) \tag{17}$$

$$= \;\overline{D}E[1/N] + \left\{\left(\frac{P - \overline{N}}{P - 1}\right)\frac{\rho\sqrt{2(P+1)}}{\lambda(1-\rho)} + \left(\frac{\overline{N} - 1}{P - 1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}\right\}\left(\frac{1 + C_D^2}{2}\right), \tag{18}$$

and

$$\hat{R}^{p}_{PSAPF} \approx \sum_{k=1}^{P} p_k \overline{R}_{PSAPF}(N=k) \tag{19}$$

$$\approx \overline{D}E[1/N] + \frac{E\left[\rho\sqrt{2(\frac{P}{N}+1)}\right]}{\lambda(1-\rho)}\left(\frac{1+C_D^2}{2}\right). \tag{20}$$

Note that these approximations are identical to the corresponding interpolation approximations for mean response time under the FCFS policy. One might expect lower accuracy in the simple interpolations for the PSAPF policy, since the interpolations do not reflect the priority given to jobs with lower available parallelism. However, there are specific cases where FCFS and PSAPF can be expected to have similar performance (e.g., exponential job demands and high system utilization), and a previous simulation study [17] has shown that for specific distributions of D and N, PSAPF is not significantly better than FCFS when D and N are independent and when $C_D \leq 5$. We thus believe that it is worthwhile to start with the simple interpolations, and to improve upon these interpolations if validations show that improvement is needed. Note that *if the simple interpolations validate well*, then the interpolation approximations yield the substantial insight that the FCFS and PSAPF policies generally have similar performance when demand and parallelism are uncorrelated.

It is worth noting that correlated workloads are also of interest, but are beyond the scope of this paper. Uncorrelated workloads are of interest because the actual degree of correlation in real workloads is unknown and may be quite weak. (Certainly any given parallel system is likely to have both fully parallel jobs that execute quickly and jobs with lower parallelism that require large amounts of CPU time.) Furthermore, a complete understanding of policy behavior includes the uncorrelated case.

6.3 Validation Experiments

Figure 9 presents histograms that summarize the validations of the PSAPF approximations. The validation parameter settings are the same as those of section 5.4. The total number of data points was 306 for each value of C_D when P=20,100, and 172 for each value of C_D when P=500,1000, thus leading to a total of 956 validations. The approximations were validated against exact analysis for constant N and against simulation otherwise.

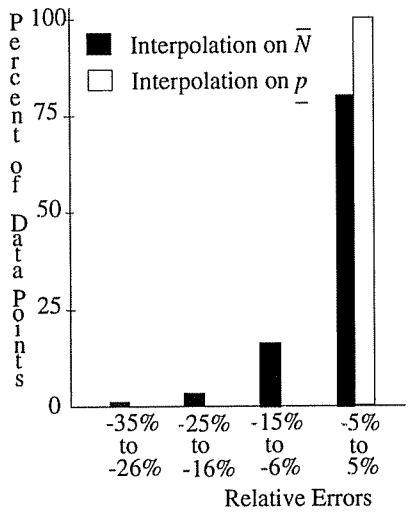
From figure 9 we observe that the relative errors in the PSAPF approximations are *very similar to* those

for FCFS in figure 6. In particular, the interpolation on $\underline{p}$ is highly accurate at $C_D = 1$, the overall accuracy of both PSAPF approximations degrades with C_D, and at $C_D = 5$ the interpolation on $\underline{p}$ tends to overestimate mean response time whereas the interpolation on $\overline{N}$ shows no strong tendency towards underestimation or overestimation of mean response time. We note that at $C_D = 5$ the errors for PSAPF are somewhat higher on average than those for FCFS. The worst case errors for $\hat{R}^{\overline{N}}_{PSAPF}$ and $\hat{R}^{\underline{p}}_{PSAPF}$ in the P=20,100 histogram for $C_D = 5$ were located at ($P = 100$, $N = Uniform(50, 100)$, $\rho = 0.3, 0.4$). The worst case errors in the P=500,1000 histogram for $C_D = 5$ were located at ($P = 1000$, $N = 100$, $\rho = 0.9$) for the interpolation on $\overline{N}$, and ($N = 3P/4$, $\rho = 0.2$) for the interpolation on $\underline{p}$. Thus, the approximation tends to be most inaccurate for workloads with constant parallelism or with very low values of C_N.

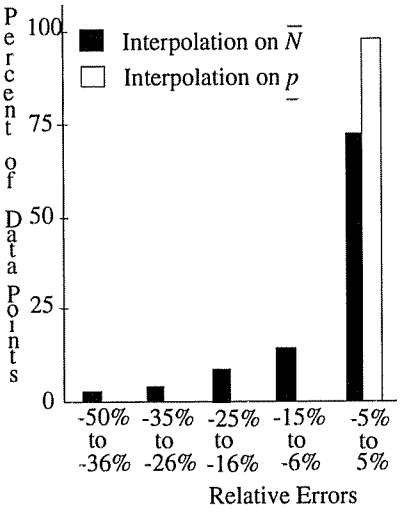
Figure 10 presents example percent errors for the PSAPF approximations for $C_D = 1, 5$ and $P = 100$. The example workloads are the same as those in figures 7 and 8. We observe that both interpolations on $\overline{N}$ and $\underline{p}$ are very accurate for the H and M workloads, the accuracy of the interpolations for the M workload degrades with C_D, with the interpolation on $\underline{p}$ having more positive errors.

One source of the error at high values of C_D is that we used approximate estimates at the end points $N = k$ as given by (16) instead of exact solutions. To estimate the amount of error due to this factor we computed exact solutions for $\overline{R}_{PSAPF}(N = k)$ by means of matrix-geometric analysis and then used the same interpolation methods as (17) and (19). Using this approach for $P = 20, 100$ we found that worst case errors (for Uniform N) at $C_D = 5$ went down to about 60% and in the great majority of cases examined the approximation is within 15% of the simulation estimates. Although the use of exact solutions at the end points improves the accuracy of the PSAPF approximations, we note again that the exact estimates at $N = k$ are obtained using numerical analysis and thus they yield no direct insight into policy behavior as a function of the system and workload parameters. Since for most cases the simple approximations have relative error within -35% to 35% range, these approximations are sufficiently accurate for the policy insights and comparisons discussed in the next section.

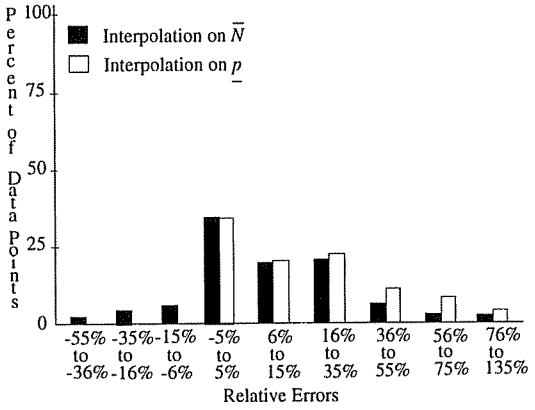
In concluding this section we note that another possible way to improve on the interpolation approximations for PSAPF is to consider modifying the interpolation on $\underline{p}$ to account for the priority given to jobs with smaller parallelism [22]. Due to space constraints, and because it is not needed for the comparisons in the next section, we do not pursue this approach further in this paper.



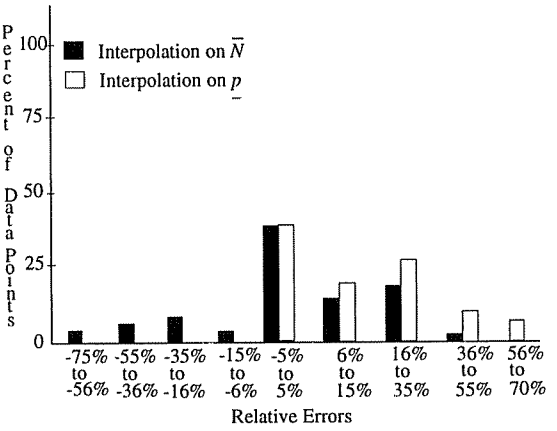
(a) Cd=1: P=20,100



(b) Cd=1: P=500,1000

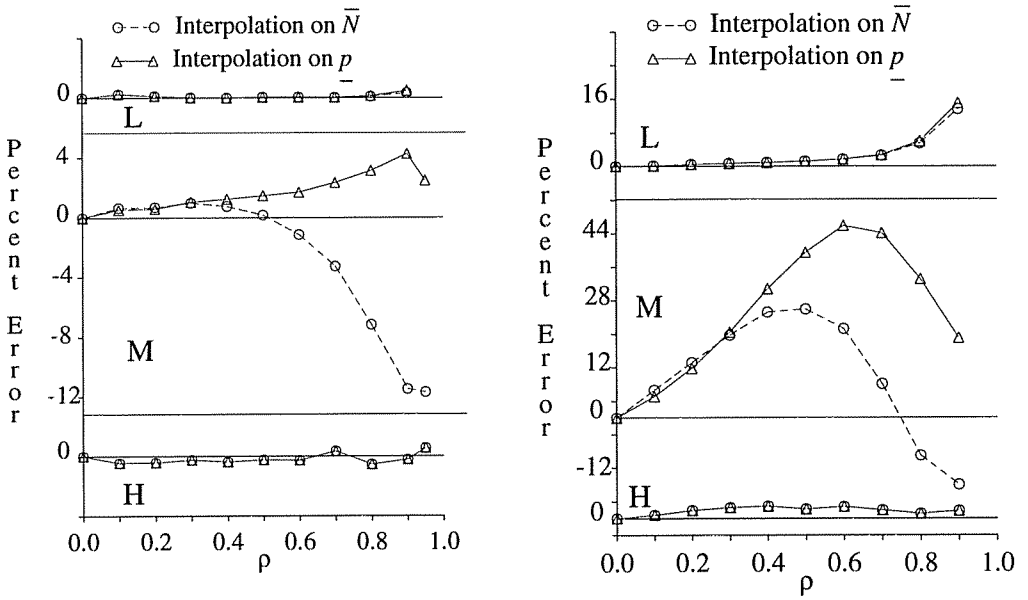


(c) Cd=5: P=20,100



(d) Cd=5: P=500,1000

Figure 9: Summary of Validations: PSAPF



(a) $C_D = 1$ (b) $C_D = 5$

Figure 10: Example Validations for PSAPF, $P = 100$

7 Preliminary Insights and Results

In this section we illustrate the use of interpolation approximations for deriving ready insight into (1) policy behavior as a function of model parameters, and (2) relative policy performance. Both the insights and the policy performance comparisons will be interpreted relative to the accuracy of the models as determined by the validation experiments in sections 5 and 6. In cases where we focus on particular regions of the design space to sharpen the insights, we will select regions where the interpolation approximations have high accuracy.

We first discuss the key workload parameters for the mean response times of the EQ, FCFS, and PSAPF policies that are directly obtained from the interpolation approximations. We then compare the performance of these three policies on the basis of one of the key workload parameters. The results in this section are for the workload assumptions that were used in deriving the interpolation approximations, i.e., general distributions of D and N, no correlation between D and N, and linear job execution rates. We also point to how some of the results generalize for correlated workloads and/or nonlinear execution rates.

7.1 Insights into Key Workload Parameters

The interpolation approximations from sections 5 and 6 readily indicate that that C_D is a key determinant of policy performance, under the workload assumptions stated above. The impact of C_D on scheduling policy performance has been observed in some previous studies. The interpolation approximations clarify and generalize those results. The approximations also indicate that $E[1/N]$ is perhaps a parallelism measure that is a key determinant of policy performance. The significance of the parallelism measure $E[1/N]$ has not to our knowledge been previously pointed out or tested. Below we discuss these points in greater detail.

7.1.1 Functional Dependence on Job Demand

All three interpolation approximations for $\overline{R}_{EQ}$ (equations (10), (12), and (14)) depend only on $\overline{D}$ and not on higher moments of job demand. In particular, the approximations for $\overline{R}_{EQ}$ are independent of C_D. This result generalizes the observation in a previous simulation study [17], which showed that for specific distributions of D and N, $\overline{R}_{EQ}$ is independent of C_D.

The mean response time estimates of FCFS and PSAPF (equations (13), (15), (18) and (20)) depend not only on $\overline{D}$ but also on C_D. In particular, these response time estimates increase linearly in C_D^2. Two previous simulation studies [18, 17] have shown that for specific distributions of demand and parallelism $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase with C_D; however, they did not show the (approximate) linear dependence on C_D^2.

For workloads with sublinear execution rates and/or correlation between D and N, one might expect the same functional dependencies on parameters of the demand distribution for each policy. This intuition is born out by extensions to the interpolation approximations for these workloads [22].

7.1.2 Functional Dependence on Parallelism

Measures of workload parallelism include $\overline{N}$ and $E[1/N]$. While $\overline{N}$ captures the average available parallelism of jobs, $E[1/N]$ captures the mean execution time of jobs (since $E[1/N] = \overline{S}/\overline{D}$). Below we discuss the insights into the impact of these parallelism measures on policy performance that can be obtained from the interpolation approximations, and corroborate that insight with experimental results.

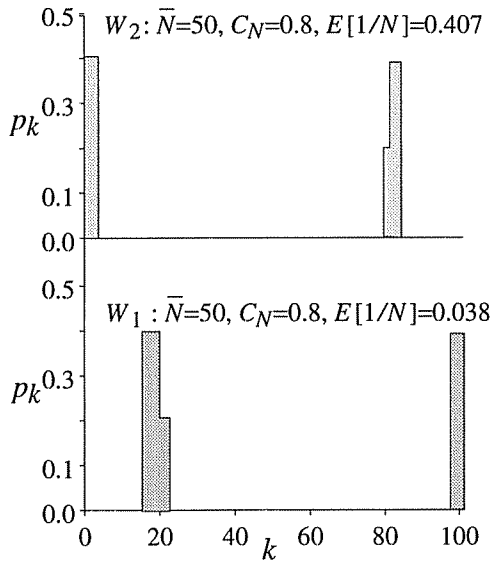
The interpolation on ρ for the EQ policy (approximation (10)) readily shows that $\hat{R}^{\rho}_{EQ}$ increases linearly with $E[1/N]$. Approximations (12) and (14) show a similar dependence on $E[1/N]$ at low utilization, but it is

not clear how $E[1/N]$ impacts the mean extra time component of these two approximations. It is nevertheless clear from all three approximations that at low utilizations $E[1/N]$ is the key workload parallelism parameter.

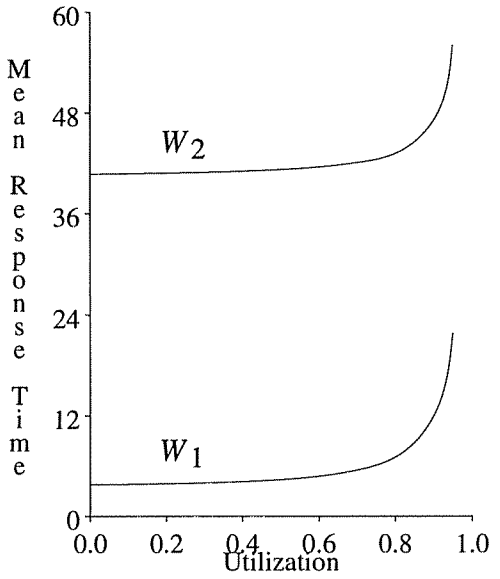
In figures 11 and 12 we give experimental results that are motivated by the interpolation approximation results and that suggest that $E[1/N]$ is a strong determinant of EQ policy performance throughout the range of utilization, including very high utilizations such as $\rho = 0.95$. Each of these figures gives the full distribution of available parallelism for each of two workloads as well as the approximate mean response time for the workloads as a function of ρ, estimated by the interpolation on $\underline{p}$. Both figures are for systems with $P = 100$ processors. In figure 11 workload W_1 has the same mean and coefficient of variation of available parallelism as workload W_2, but much lower $E[1/N]$. The figure shows that W_1 has significantly lower mean response time than W_2 throughout the range of system utilization. In figure 12 workloads W_3 and W_4 have very different values for mean available parallelism but the same value for $E[1/N]$. These two workloads have very similar mean response time, for all values of ρ. Thus $\overline{N}$ does not have (much) impact on the performance of EQ when $E[1/N]$ is constant, while $E[1/N]$ can have significant impact on performance when $E[N]$ is constant. The figures provide evidence that the key parallelism measure for $\overline{R}_{EQ}$ is $E[1/N]$ rather than $\overline{N}$. Note that $E[1/N]$ contains higher moments of N than simply $\overline{N}$ and thus it conveys more information about the available parallelism distribution. For workloads with sublinear execution rates and/or correlation between D and N the natural generalization of $E[1/N]$ is the normalized mean service time $\overline{S}/\overline{D}$, which will again convey information about the distribution of N. (Further results are needed to corroborate the hypothesis that this is the parallelism metric that is the key determinant of performance for those workloads.)

The interpolations on $\overline{N}$ and $\underline{p}$ for the FCFS and PSAPF policies, like the corresponding interpolations for EQ, do not directly show which parameters of the parallelism distribution are the key determinants of policy performance, except at light load. However, the reductions for the FCFS and PSAPF policies under constant parallelism, summarized in figure 3, show that when $C_D = 5$ $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ can *decrease* with decrease in $E[1/N]$.[6] To see this more clearly, Figure 13 plots the points in figures 3(a) and (b) for $\rho = 0.9$, yielding the approximate mean response time under constant parallelism (from approximations (6),(7), and (16)) versus parallelism k, for P=100, $C_D = 5$, $\overline{D} = P$, and $\rho = 0.9$. We see that $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase with available parallelism (i.e., a decrease in $E[1/N]$) after $k = 5$. In contrast, $\overline{R}_{EQ}$ decreases as available parallelism increases (i.e., as $E[1/N]$ decreases) for this workload. Similar trends have been observed in a

[6] At $C_D = 1$, the behavior of FCFS and PSAPF is similar to the behavior of EQ.



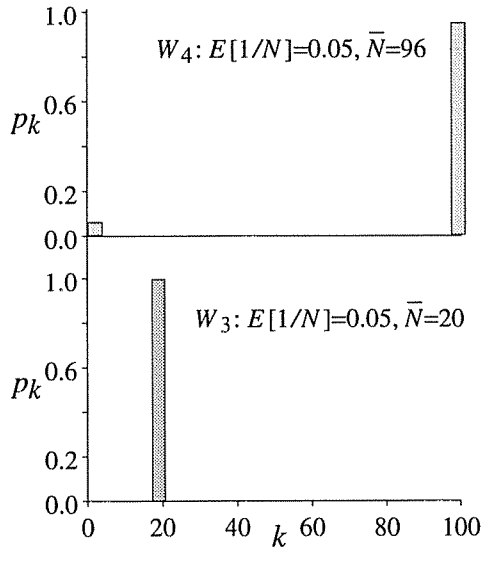
(a) Workloads W_1 and W_2



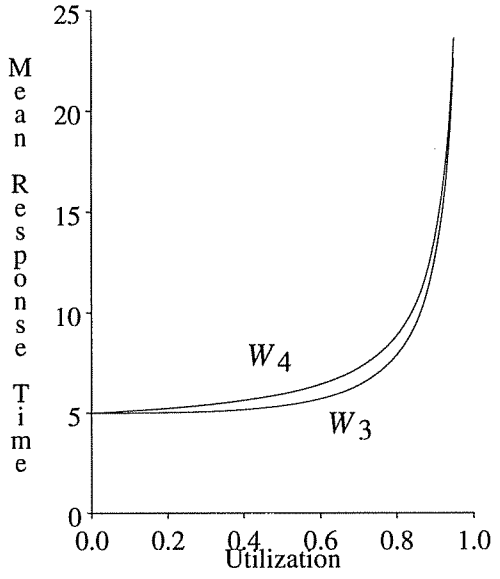
(b) $\hat{R}^p_{EQ}$ versus ρ

Figure 11: Sensitivity of $\hat{R}^p_{EQ}$ to $E[1/N]$ for fixed $\overline{N}$ and C_N

$\overline{D} = P = 100$



(a) Workloads W_3 and W_4



(b) $\hat{R}^p_{EQ}$ versus ρ

Figure 12: Sensitivity of $\hat{R}^p_{EQ}$ to $\overline{N}$ for fixed $E[1/N]$

$\overline{D} = P = 100$

previous simulation study [15] for a different distribution of N.

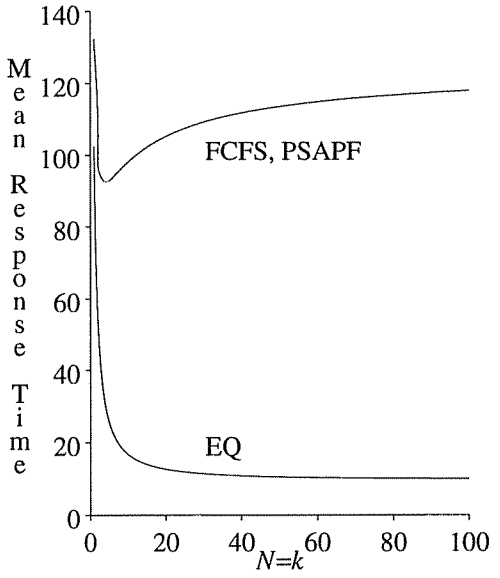


Figure 13: Sensitivity of $\overline{R}_{\Psi}(N = k)$ to available parallelism, $\Psi \in \{EQ, FCFS, PSAPF\}$

$\overline{D} = P = 100$, $C_D = 5$, $\rho = 0.9$

7.2 Policy Comparison

We now focus on a quantitative comparison of EQ against FCFS and PSAPF under the assumption of no correlation between D and N, and linear execution rates. As discussed above, C_D is a key parameter that influences the relative performance of EQ, FCFS, and PSAPF. From the interpolations on $\overline{N}$ and $\underline{p}$ in sections 5.2 and 5.3 we saw that $\hat{X}_{FCFS} = \hat{X}_{EQ}(1 + C_D^2)/2$, which implies that

$$\overline{R}_{EQ} \begin{cases} > \overline{R}_{FCFS}, & C_D < 1, \\ = \overline{R}_{FCFS}, & C_D = 1, \\ < \overline{R}_{FCFS}, & C_D > 1. \end{cases}$$

Since our estimators for $\overline{R}_{PSAPF}$ are the same as the estimators for $\overline{R}_{FCFS}$ the same relation holds between $\overline{R}_{EQ}$ and $\overline{R}_{PSAPF}$. We illustrate these comparisons in figure 14 using the M and L workloads of parallelism (see table 4 in section 5.4.3). Note that for this figure we estimated $\overline{R}_{EQ}$ using the interpolation on $\underline{p}$ which is most accurate for this policy. For $\overline{R}_{PSAPF}$ and $\overline{R}_{FCFS}$ we give both the interpolation on $\overline{N}$ and the

interpolation on $\underline{p}$, in order to illustrate the range of estimated performance for these policies under these workloads. Note that the uncertainty in the estimated policy performance is small compared to the performance differential between EQ and the other two policies and thus the interpolation approximations derived in this paper have enabled a broad comparison of the policies for the workloads that satisfy the model assumptions. A previous simulation study [17] showed similar policy comparison results for a hyperexponential distribution for D and specific distributions of N.

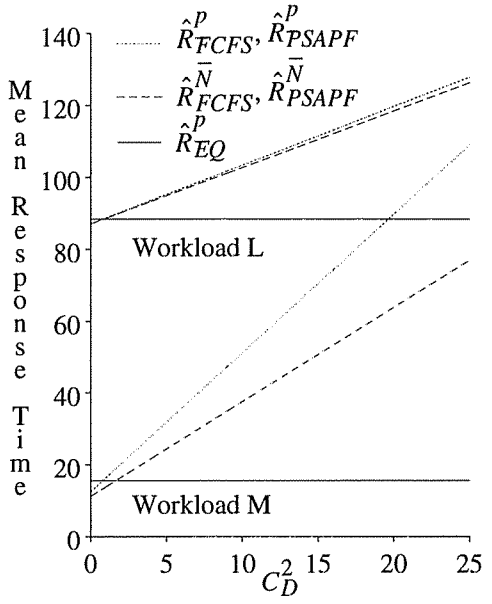


Figure 14: $\hat{R}_\Psi$ versus C_D^2, $\Psi \in \{EQ, FCFS, PSAPF\}$

$\overline{D} = P = 100$, $\rho = 0.9$

From figure 14 we note that for $C_D >> 1$ the performance of FCFS and PSAPF is considerably worse than that of EQ. Since general purpose computer system workloads typically have high C_D [33, 43] this result is relevant to practical systems.[7] Furthermore, the results show that *relative* policy performance can be strongly influenced by C_D, indicating that it is important to interpret the results of scheduling policy performance comparisons in the context of the assumed distribution(s) of job demand. Had we restricted the workload model to only exponential job demands we would have concluded that all three of EQ, FCFS, and PSAPF perform about the same. We also would have concluded that $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ decrease

[7]Note that we have also measured the squared coefficient of variation in service times on our local CM-5 to be ranging from 7.84 to about 25, with the higher end being more typical.

with available parallelism, which is untrue when C_D is high.

8 Conclusions

In this paper we have shown how the technique of interpolation approximations can be applied to the analysis of parallel system scheduling policies to yield very simple and highly efficient performance models that readily yield insight. First we defined a workload model that enables the analysis yet includes *general distribution of total job processing requirement*, general distribution of available job parallelism, and sublinear as well as linear job execution rates (although, for the sake of space and clarity in the exposition, the sublinear execution rates were not included in the interpolation approximations). Second, for each of three parallel scheduling policies, EQ, FCFS, and PSAPF, we found points in the parameter space for which the parallel system reduces to a queueing system that has a known solution. Note that the general distribution of job demand is tractable since there are known solutions of single server and multiserver queues that allow for general job demands. Third, we showed how three types of interpolations can be used to obtain mean response time estimates over the entire parameter space. All three interpolations have a very simple structure and, coupled with the closed-form estimates at the end-points, they readily yield insight into the parameters that are key determinants of system performance. Thus, in much the same way that current parallel systems are built by interconnecting off-the-shelf microprocessors, we have interconnected off-the-shelf solutions at extreme values of the model parameters to obtain a parallel system performance model. Furthermore, just as different parallel processor interconnection networks provide different levels of performance, validation experiments reveal that different interpolation techniques provide different degrees of accuracy.

Extensive validations show that the interpolation models are sufficiently accurate for the purposes of comparing policy performance and gaining insight into policy behavior. Also, the loss in accuracy due to the use of closed form approximations rather than exact analysis at the end-points is small compared to the gain in insight provided by the model.

The interpolation approximations yielded the insights that (1) FCFS and PSAPF have similar performance (for all distributions of total job service requirement) when available parallelism and total service requirement are independent, (2) the coefficient in job demand (C_D) is a key determinant of FCFS and PSAPF performance, whereas mean response time under EQ is insensitive to the second and higher moments of job demand, and (3) $E[1/N]$ where N is the available job parallelism is a key determinant of the

EQ policy performance for workloads with linear execution rates and independence between job demand and available parallelism. The functional dependence of policy performance on $E[1/N]$ is a new observation. The results for the C_D parameter clarify and generalize results from previous work [17], and show that *relative* policy performance is sensitive to job demand distribution. For example, the EQ policy outperforms FCFS and PSAPF when C_D is greater than 1, whereas FCFS and PSAPF outperform EQ when $C_D < 1$. Since the typical job demand distributions in parallel workloads are as yet unknown, conclusions that are reached in studies of relative policy performance generally need to be interpreted in the context of the assumed demand distribution(s).

In further research we are extending the applicability of the models to include static scheduling policies, sublinear job execution rates, and workloads with (a controlled degree of) correlation between demand and available parallelism. We are also developing interpolations on other system parameters that have even higher accuracy than those considered in this paper. These models can then be used to further explore the parameter space for scheduling policy performance.

Acknowledgements

We are grateful to Peter Haas, Vikram Adve, Rajeev Agrawal, Randy Nelson, and the UW-Madison Performance Seminar group for providing useful comments on the material in this paper.

Appendix

In this appendix we prove Propositions 4.1 and 4.2.

Proposition 4.1

$$\overline{R}_{EQ}(N=k) = \overline{R}_{M/G/c\ PS}, \qquad c = \frac{P}{k},\ P \bmod k = 0.$$
$$\overline{R}_{FCFS}(N=k) = \overline{R}_{M/G/c}, \qquad c = \frac{P}{k},\ P \bmod k = 0.$$

In particular,

$$\overline{R}_{EQ}(N=1) = \overline{R}_{M/G/P\ PS}, \qquad \overline{R}_{EQ}(N=P) = \overline{R}_{M/G/1_P\ PS}$$
$$\overline{R}_{FCFS}(N=1) = \overline{R}_{M/G/P}, \qquad \overline{R}_{FCFS}(N=P) = \overline{R}_{M/G/1_P}$$

Proof. We first give the proof for the EQ reduction. Let $\Gamma = (EQ, P, \lambda, \mathcal{F}_D, N=k)$, $P \bmod k = 0$. When there are $Q \leq c$ jobs in Γ, each job receives k amount of processing power. When there are $Q > c$ jobs in Γ, each job receives P/Q amount of processing power. This is precisely how an $M/G/c$ processor sharing (PS) queue allocates processing power to jobs, where each of the c servers has a processing power of k.

Now consider the proof for the FCFS reduction. Let $\Gamma = (FCFS, P, \lambda, \mathcal{F}_D, N=k)$, $P \bmod k = 0$. System Γ operates as follows. A job that arrives when system Γ is empty gets k processors. Subsequent jobs that arrive also get k processors unless all processors are occupied. When a job departs it releases all k of its processors as a single unit. The first job waiting in the queue (if any) thus gets all k processors released by the departing job, and so on. Thus an arriving job waits for service in FCFS order and upon service gets k units of processing power throughout its lifetime. At any point in time there are at most $c = P/k$ jobs in the servers. This means that system Γ behaves like an $M/G/c$ system with $c = P/k$ processors, in which each job has one task with service requirement $x = D/k$. ∎

Proposition 4.2

$$\overline{R}_{EQ}(N=k) = \overline{R}_{Symmetric\ queue}[\phi(j) = \min(j \cdot k, P),\ \alpha(l,j) = 1/j], \quad k = 1, 2, \ldots, P.$$

(Note that k does not need to evenly divide P as in Proposition 4.1.)

Proof. Let $\Gamma = (EQ, P, \lambda, \mathcal{F}_D, N=k)$, $1 \leq k \leq P$. If there are not enough jobs to utilize all processors then the total service effort of Γ is $\phi(j) = j \cdot k$, since each job is allocated exactly k processors. If there are enough jobs to utilize all processors then the service effort is $\phi(j) = P$. Thus $\phi(j) = \min(j \cdot k, P)$.[8] Since the EQ policy allocates an equal fraction of processing power to all jobs it follows that $\alpha(l,j) = 1/j$, $l = 1, \ldots, j$. This would not hold if maximum parallelism was not constant across jobs because jobs with small available parallelisms could get fewer processors than the equiallocation value. ∎

[8] Note that for a sublinear ERF γ and spatial partitioning the service effort is $\phi(j) = j \cdot \min(k, \gamma(P/j))$.

References

[1] A. Bricker, M. Litzkow, and M. Livny. Condor Technical Summary. TR 1069, Computer Sciences Dept., Univ. of Wisconsin-Madison, January 1992.

[2] D. Burman, and D. Smith. Approximate Analysis of a Queueing Model with Bursty Traffic. *Bell System Tech. Jnl. 62* (1983), 1433-1453.

[3] D. Burman, and D. Smith. An Asymptotic Analysis of a Queueing System with Markov-Modulated Arrivals. *Operations Research 34*, 1 (1986), 105-119.

[4] G. Cosmetatos. Some Approximate Equilibrium Results for the Multi-Server Queue (M/G/r). *Operational Research Quarterly 27*, 3 (1976), 615-620.

[5] K. Fendick, and W. Whitt. Measurements and Approximations to Describe the Offered Traffic and Predict the Average Workload in a Single-Server Queue. *Proc. of the IEEE 77*, 1 (Jan. 1989), 171-194.

[6] P. Fleming. An Approximate Analysis of Sojourn Times in the M/G/1 Queue with Round-Robin Service Discipline. *AT&T Bell Labs. Tech. Jnl. 63*, 8 (Oct. 1984), 1521-1535.

[7] P. Fleming, and B. Simon. Interpolation Approximations of Sojourn Time Distributions. *Operations Research 39*, 2 (1991), 251-260.

[8] E. Gelenbe, D. Ghoshal, and S. Tripathi. Analysis of Processor Allocation in Large Multiprocessor Systems. *Proc. of the Interntl. Conf. on the Performance of Distributed Systems and Integrated Comm. Networks*, Kyoto, Japan, Sept. 1991.

[9] D. Ghosal, G. Serazzi, and S. Tripathi. The Processor Working Set and Its Use in Scheduling Multiprocessor Systems. *IEEE Trans. on Software Engg. 17*, 5 (May 1991), 443-453.

[10] A. Gupta, A. Tucker, and L. Stevens. Making Effective Use of Shared Memory Multiprocessors: The Process Control Approach. Tech. Report, Computer Sciences Dept., Stanford University, July 1991.

[11] F. Kelly. *Reversibility and Stochastic Networks.* John Wiley & Sons, 1979.

[12] L. Kleinrock. *Queueing Systems, Vol I: Theory.* John Wiley & Sons, New York 1975.

[13] L. Kleinrock. *Queueing Systems, Vol II: Computer Applications.* John Wiley & Sons, New York 1976.

[14] S. Lavenberg (Ed). *Computer Performance Modeling Handbook.* Academic Press, New York 1983.

[15] S. Leutenegger. Issues in Multiprogrammed Multiprocessor Sharing. Ph.D. Thesis, Tech. Report #954, Computer Sciences Dept., Univ. of Wisconsin-Madison, Aug. 1990

[16] S. Leutenegger, and R. Nelson. Analysis of Spatial and Temporal Scheduling Policies for Semi-Static and Dynamic Multiprocessor Environments. Research Report - IBM T.J. Watson Research Center, Yorktown Heights, Aug. 1991.

[17] S. Leutenegger, and M. Vernon. The Performance of Multiprogrammed Multiprocessor Scheduling Policies. *Proc. of the ACM SIGMETRICS Conf. on Measurement & Modeling of Computer Systems 18*, 1 (May 1990), 226-236.

[18] S. Majumdar, D. Eager, and R. Bunt. Scheduling in Multiprogrammed Parallel Systems. *Proc. of the ACM SIGMETRICS Conf. on Measurement & Modeling of Computer Systems 16*, 1 (May 1988), 104-113.

[19] S. Majumdar, D. Eager, and R. Bunt. Characterisation of programs for scheduling in multiprogrammed parallel systems. *Performance Evaluation 13* (1991), 109-130.

[20] R. Mansharamani. Efficient Analysis of Parallel Processor Scheduling Policies. Ph.D. Thesis, Computer Sciences Department, University of Wisconsin, Madison, WI, November 1993.

[21] R. Mansharamani, and M. Vernon. Performance Analysis of the EQuipartitioning Parallel Processor Allocation Policy. *In preparation.*

[22] R. Mansharamani, and M. Vernon. Comparison of Processor Allocation Policies for Parallel Systems. *In preparation.*

[23] C. McCann, R. Vaswani, and J. Zahorjan. A Dynamic Processor Allocation Policy for Multiprogrammed, Shared Memory Multiprocessors. *ACM Transactions on Computer Systems 11*, 2 (May 1993), 146-178.

[24] V. Naik, S. Setia and M. Squillante. Scheduling of Large Scientific Applications on Distributed Memory Multiprocessor Systems. Research Report RC 18621, IBM T. J. Watson Research Center, Yorktown Heights, Jan. 1993. *Proc. of the 6th SIAM Conf. on Parallel Processing for Scientific Computation.*

[25] R. Nelson. A Performance Evaluation of a General Parallel Processing Model. *Proc. of the ACM SIGMETRICS Conf. on Measurement & Modeling of Computer Systems 18*, 1 (May 1990), 13-26 .

[26] R. Nelson. Matrix Geometric Solutions in Markov Models - A Mathematical Tutorial. Research Report - IBM T.J. Watson Research Center, Yorktown Heights, Apr 1991.

[27] R. Nelson, and D. Towsley. A Performance Evaluation of Several Priority Policies for Parallel Processing Systems. COINS Tech. Report 91-32, Computer and Info. Sciences, Univ. of Mass.-Amherst, May 1991. (To appear in JACM.)

[28] R. Nelson, D. Towsley, and A. Tantawi. Performance Analysis of Parallel Processing Systems. *IEEE Trans. on Software Engg.*, April 1988, 532-540.

[29] M. Neuts. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. The John Hopkins University Press, 1981.

[30] M. Reiman, and B. Simon. An Interpolation Approximation for Queueing Systems with Poisson Input. *Operations Research 36*, 3 (1988), 454-469.

[31] M. Reiman, B. Simon, and S. Willie. Simterpolation: A Simulation Based Interpolation Approximation for Queueing Systems. *Operations Research 40*, 4 (1992), 706-723.

[32] H. Sakasegawa. An Approximation Formula $L_q \doteq \alpha\rho^{\beta}/(1-\rho)$. *Annals of the Institute of Statistical Mathematics 29*, 1 (1977), 67-75.

[33] C. Sauer, and K. M. Chandy. Computer System Performance Modeling. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[34] S. Setia, M. Squillante, and S. Tripathi. Processor Scheduling on Multiprogrammed, Distributed Memory Parallel Systems. *Proc. of the ACM SIGMETRICS Conf. on Measurement & Modeling of Computer Systems 21*, 1 (May 1993), 158-170.

[35] S. Setia, and S. Tripathi. An Analysis of Several Processor Partitioning Policies for Parallel Computers. Tech. Report CS-TR-2684, Univ. of Maryland, May 1991.

[36] S. Setia, and S. Tripathi. A Comparative Analysis of Static Processor Partitioning Policies for Parallel Computers. *Proc. of the Interntl. Workshop on Modeling, Analysis and Simulation of Computer and Telecomm. Systems (MASCOTS'93)*, January 1993.

[37] B. Simon, and S. Willie. Estimation of Response Time Characteristics in Priority Queueing Networks via an Interpolation Methodology based on Simulation and Heavy Traffic Limits. *Computer Science and Statistics: Proc. of the 18th Symposium on the Interface*, American Statistical Association, 1986, 251-256.

[38] M. Squillante. MAGIC: A Computer Performance Modeling Tool Based on Matrix-Geometric Techniques. *Proc. of the 5^{th} Interntl. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, Feb. 1991.

[39] D. Stoyan. Comparison Methods for Queues and Other Stochastic Models. Wiley 1983.

[40] Y. Takahashi. An Approximation Formula for the Mean Waiting Time of a M/G/c Queue. *Jnl. of the Operations Research Society of Japan 20*, 3 (1977), 150-163.

[41] Thinking Machines Corporation. The Connection Machine CM-5 Technical Summary. Cambridge, Massachusetts, October 1991.

[42] D. Towsley, C. Rommel, and J. Stankovic. Analysis of Fork-Join Program Response Times on Multiprocessors. *IEEE Trans. on Parallel and Distributed Systems*, July 1990, 286-303.

[43] K. Trivedi. Probability and Statistics, with Reliability, Queueing and Computer Science Applications. Prentice-Hall, 1982, pp. 130.

[44] A. Tucker and A. Gupta. Process Control and Scheduling Issues for Multiprogrammed Shared-Memory Multiprocessors. *Proc. of the 12th ACM Symp. on Operating System Principles*, Dec. 1989, 159-166.

[45] S. Varma, and A. Makowski. Interpolation Approximations for Symmetric Fork-Join Queues. To appear in *Proceedings of Performance'93.*

[46] R. Vaswani and J. Zahorjan. The Implications of Cache Affinity on Processor Scheduling for Multiprogrammed, Shared Memory Multiprocessors. *Proc. of the 13th ACM Symposium on Operating System Principles*, October 1991, 26-40.

[47] W. Whitt. An Interpolation Approximation for the Mean Workload in a GI/G/1 Queue. *Operations Research 37*, 6 (1989), 936-952.

[48] R. Wolff. Stochastic Modeling and the Theory of Queues. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[49] D. Yao. Refining the Diffusion Approximation for the M/G/m Queue. *Operations Research 33* (1985), 1266-1277.

[50] J. Zahorjan, and C. McCann. Processor Scheduling in Shared Memory Multiprocessors. *Proc. of the ACM SIGMETRICS Conf. on Measurement & Modeling of Computer Systems 18*, 1 (May 1990), 214-225.