# Efficient Analysis of Parallel Processor Scheduling Policies

Rajesh Kishin Mansharamani

Technical Report #1195

November 1993

# EFFICIENT ANALYSIS OF PARALLEL PROCESSOR SCHEDULING POLICIES

by
RAJESH KISHIN MANSHARAMANI[1]

A thesis submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
(Computer Sciences)

at the
UNIVERSITY OF WISCONSIN - MADISON
1993

[1] This is a slightly revised version of the dissertation. This research was partially supported by NSF grant CCR-9024144.

# Abstract

The widespread use of parallel systems has led to a number of proposals for high performance parallel processor scheduling policies. However, due to the specific nature of the workload assumptions and the performance evaluation techniques in previous work, the performance characteristics of processor scheduling policies are not well understood. This thesis unifies and generalizes previous policy analysis and comparisons using a general workload model that captures the essential features of parallel applications and a new performance evaluation technique. Our workload model includes general distributions of job parallelism and cumulative processing demand, controlled correlation between demand and parallelism, and a general nondecreasing deterministic execution rate function that captures the impact of synchronization and communication overheads.

The proposed new approach to performance modeling of parallel processor scheduling is that of interpolation approximations. The interpolation approximation approach yields closed form expressions for mean response times that provide ready insight into the functional dependence of policy performance on workload parameters, and can be easily evaluated for systems with hundreds of processors. We use interpolation approximations to evaluate and compare four policies shown in the literature to have high performance under various specific workloads. These include a dynamic spatial equipartitioning (EQS) policy, the Preemptive Smallest Available Parallelism First (PSAPF) policy, the dynamic First Come First Serve (FCFS) policy, and a run-to-completion policy called Adaptive Static Partitioning (ASP). The results show that, as in uniprocessor scheduling disciplines, the coefficient of variation of demand is a key parameter that distinguishes relative policy performance. Using the interpolation models we also derive other key parameters and delineate regions of the design space under which each policy performs best. We show that the EQS policy has highest performance over most of the expected practical regions of the workload space.

Finally, we thoroughly study the behavior of the EQS policy with respect to the workload parameters using both sample path analysis as well as approximate analysis. For example, we show that under our workload model the mean response time of EQS is smallest when all jobs are fully parallel and is highest when all jobs are fully sequential.

# Acknowledgements

I am grateful to my adviser Mary Vernon without whose advice this thesis would not have been possible. I thank her for teaching me analytic modeling techniques for parallel systems and fundamentals of performance modeling in general, which I could never have learned through coursework or text books. It has been my good fortune to learn from an expert in this area. Despite her busy schedule she gave valuable input in formulating the problem in a way that made it both analytically tractable and practically significant, provided me with a number of research ideas, gave me sound advice about which research directions were likely to be most fruitful, and ensured that the 'big picture' was always clear to me. Much of the presentation ideas in this work are also due to her and I thank her for helping me improve my writing style. I thank her for her constant support and encouragement, which also made this work possible. To sum up, the credit goes to her for building the foundation and structure though it doesn't show up on the title page of this thesis.

It has also been my good fortune to do joint research and course work with Rajeev Agrawal. I thank him for his advice and encouragement and for teaching me analytic techniques that seemed formidable to learn through textbooks and research papers. I also thank him for serving on my prelim and final defense committee as well as being a reader for this thesis.

I next thank Miron Livny for getting me interested in performance modeling of computer systems. Through coursework and joint research he taught me the fundamentals of modeling and simulation of computer systems. I am also grateful to him for his efforts to improve my writing style.

I thank Randy Nelson for being in my prelim committee and for providing useful feedback on this work. I thank Jeff Naughton for serving in my final defense committee and being a reader for this thesis, and Sigrun Andradottir and Rajan Suri for serving in my final defense committee. Suggestions from Vikram Adve, Peter Haas, Ravi Gujar, and other members of the UW performance seminar, namely, Fred Worley, Dan Ross, Alain Kägi, and Chandu Rao helped improve the presentation of part of this research. Scott Leutenegger was ever ready to answer questions about his thesis that helped me with my work. Joint work with Su-Hui Chiang provided a better understanding of one of the scheduling disciplines studied in this thesis.

The nonlinear programming faculty and students have been very helpful. Steve Dirkse implemented nonlinear programs in GAMS that were used to obtain results for this thesis. Michael Ferris helped in setting up the nonlinear programs and in explaining details on convexity of functions. A course on linear programming taught by Olvi Mangasarian proved useful in obtaining a key result of this thesis.

John Strikwerda was always ready to help with the numerical analysis related to this work, and joint research with him and Eric Bach gave me ideas on the applicability of this work. Eric Bach also provided assistance and references for random number generation.

A number of others have also helped out with my work in some way or the other. In particular, Vikram Adve helped me get through the depth screening exam. Mike Litzkow provided much needed assistance in running simulation jobs on the Condor distributed system. The systems lab was very prompt in fixing machine and printer problems. Lorene Webber was prompt in clearing up the paper work for the prelim and final defense. My office mates Jim Lewis and Babak Falsafi were very helpful and cooperative. V. Srinivasan provided good advice at the start of this thesis and also provided macros for figures in LaTeX.

Regarding non-technical matters a number of friends have been helpful. To be brief, I simply thank Vittaldas Prabhu to cover all of them.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

The widespread use of parallel processor systems has created the need for multiprogrammed job scheduling policies in order to reduce system response time and make efficient use of system processors. As a result, a number of proposals for high performance parallel processor scheduling policies have appeared in the literature.[1] More than 30 papers in the literature have evaluated and compared parallel processor scheduling policies under various specific workload assumptions, but the performance characteristics of these policies is still not well understood. This lack of understanding makes it difficult to select a particular policy for implementation in a given parallel system.

Parallel processor scheduling is not well understood primarily for several reasons. First and foremost, specific workload assumptions made in previous studies limit the applicability of processor scheduling results because the *relative* performance of scheduling disciplines can be sensitive to workload assumptions. Typical limiting assumptions in the literature include the exponential distribution for job processing requirement, specific distributions for available or maximum job parallelism, e.g., constant maximum parallelism, and uncontrolled or unspecified correlation between processing requirement and parallelism. Second, the performance evaluation techniques used thus far in the literature do not yield insight into key determinants of policy performance since the techniques are primarily based on numerical analysis (i.e., simulation or numerical solution of simultaneous equations). Third, even for a given set of specific workload assumptions, only a subset of the promising high-performance policies have been compared.

A fourth issue that has contributed to the lack of understanding of parallel scheduling is the

[1] We use mean response time as the performance metric throughout this thesis.

assumption of small system sizes in previous studies. While real systems are scaling to hundreds of processors, processor scheduling models in the literature have typically been solved only for systems with 10 to 20 processors primarily because of computational reasons. Note that for uniprocessor systems it is easy to model an increase in system processing power by maintaining a single parameter for processor capacity. But for parallel systems this is not necessarily the case since total processing power is usually increased by means of adding more processors to the system, which causes the underlying state space to grow exponentially and thus increases model complexity. Thus while current parallel systems have hundreds of processors [16], there does not exist any technique in the literature to model scheduling policies for these systems under general workloads.

Due the above restrictions in models for scheduling policies, previous studies of processor scheduling have arrived at different conclusions regarding which policy has highest performance. Some studies recommend policies that give high priority to jobs with low parallelism and low priority to jobs with high parallelism, for example, the Preemptive Smallest Available Parallelism[2] First (PSAPF) policy. The intuition for these policies is that parallel workloads are expected to have positive correlation between processing demand and parallelism, and thus policies that give high priority to jobs with low parallelism approximate the Shortest Job First policy which is optimal for uniprocessing systems. Another feature of these policies is that they tend to serve jobs with large parallelisms as late as possible and thus keep processor utilization high even when there are only few jobs in the system. Other studies recommend run-to-completion policies that adapt processor partitions to number of jobs in the system but do not change a job's partition during its execution, for example, the Adaptive Static Partitioning (ASP) policy. Still other studies show that spatial equipartitioning (EQS) policies that allocate an equal fraction of processing power to jobs in the system have high performance over various specific workloads. Finally, some studies show that a simple policy such as First-Come-First-Serve (FCFS) that allocates processors up to a maximum of a job's parallelism can have high performance under specific workloads and thus it may not be necessary to use more sophisticated policies.

These conclusions for high performance scheduling policies have been arrived at in the literature under various specific workload conditions, but the qualitative behavior of the policies under general workload conditions has not been examined. The workload parameters that are key determinants of policy performance need to be identified, and the policy comparison results need to be re-evaluated using more general workload assumptions in order to determine which policy has best overall performance.

[2] The available parallelism, $N$, of a job is the number of processors the system scheduler believes the job can productively use.

## 1.2 Goals of this Thesis

The purpose of this thesis is to understand the qualitative behavior of parallel processor policies as a function of workload parameters and to unify and generalize previous policy comparison results under a general workload model that captures the essential features of parallel applications. We choose to compare the ASP, FCFS, EQS, and PSAPF policies discussed above (and defined more precisely in Chapter 2) since each has been shown to have high performance under various specific workload assumptions. To accomplish our general goal we need to accomplish the following specific objectives:

1. Design and parameterize a general workload model that captures the essential features of parallel applications. The key properties to be represented include parallelism, process synchronization and communication, processing requirement, and correlation between processing requirement and parallelism.

   Design of the model requires judicious assumptions that permit broad applicability of policy comparison results and at the same time permit ease of analysis.

2. Develop mean response time models for the ASP, FCFS, EQS, and PSAPF policies, under the given workload model, that yield direct insight into key determinants of policy performance and are easy to evaluate for systems with hundreds of processors.

3. Obtain the workload parameters that are the principal determinants of relative performance of the given scheduling policies. For example, measures of job processing requirement, parallelism, and correlation between the two that influence the relative performance of the given policies.

4. (a) Compare ASP, FCFS, EQS, and PSAPF over the workload design space using the key parameters that affect relative policy performance.

   (b) Delineate regions of the design space over which each policy performs best, and if possible identify which policy has best performance for most of the design space or over important regions of the design space.

   (c) Unify and explain previous policy comparison results by showing how they relate to different regions of the design space.

## 1.3 Organization of this Thesis

Chapter 2 reviews existing literature on parallel processor scheduling and clarifies the limitations in previous work that were outlined to in Section 1.1. Chapter 3 focuses on our first goal and explains our choice of system and workload model to compare parallel processor policies. In this chapter we also derive constraints on workload parameters in order to delineate the parameter space for policy comparison. Chapter 4 presents a new approach to performance modeling of parallel processor policies, that of *interpolation approximations.* The key idea behind this approach is to first reduce processor scheduling policies to systems with known mean response time results for particular values of workload parameters, and then to interpolate between these points to obtain an approximation over a wider region of the parameter space. The interpolation approximation approach yields closed form expressions for mean response times that directly provide insight into policy performance as a function of workload parameters. Chapter 5 applies the interpolation approximation approach to model the EQS policy and Chapter 6 does the same for the ASP, FCFS, and PSAPF policies. Using the analytic models for these policies, Chapter 7 quantitatively compares their performance over the design space and delineates the regions of best policy performance. Key parameters obtained from the approximate mean response time expressions enable us to explore the design space in a more systematic way than done in the past. Chapter 7 also derives exact results for the sensitivity of policy performance to a key workload parameter. The EQS policy, which is shown to be the most promising policy, is further analyzed in Chapter 8 where we examine the sensitivity of its mean response time to various workload parameters. Chapter 9 provides a summary of this thesis and discusses directions for future research. Appendix A provides proofs and derivations that can be skipped without loss of continuity while reading this thesis.

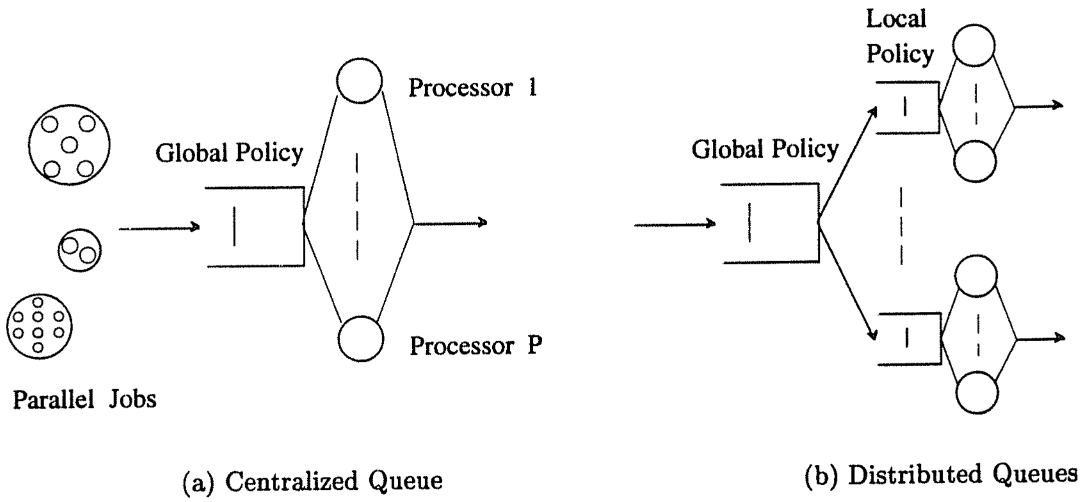
# Chapter 2

# Review of Previous Literature

In this chapter we review the literature on performance evaluation of parallel processor scheduling policies where the performance metric is mean response time. We first examine common models in the literature for parallel systems and parallel workloads in Sections 2.1 and 2.2. In order to categorize scheduling policy results that have appeared in the literature we classify parallel policies into four classes in Section 2.3, based on the presence or absence of two orthogonal system constraints. Section 2.4 reviews previous performance evaluation techniques, Section 2.5 reviews results for processor scheduling and finally Section 2.6 summarizes previous processor scheduling results and sets the stage for the work in this thesis.

The following overview on parallel processing should be useful in understanding the material in the rest of this chapter. A parallel application consists of units of work, called tasks, that can execute in parallel. Consider the simplest case of a parallel program where all tasks are independent, that is, they have no precedence constraints. The number of tasks in the application is either explicit, i.e., specified by the programmer, or implicit, i.e., generated by the compiler. If there are as many processors allocated to the application as the number of tasks then all tasks can simultaneously execute in parallel. However, in general the number of processors allocated to the application will be less than the number of tasks and thus the tasks will need to be allocated to processors according to some allocation rule. Two types of allocation are possible: (1) static allocation where each task is assigned to a specific processor at program initiation time, e.g., $n \cdot k$ tasks are allocated in units of $k$ to each of $n$ processors, and (2) dynamic allocation where tasks are placed in a common task queue and scheduled on to $n$ processors in some scheduling order, e.g., FCFS or processor sharing. In this thesis we focus on processor allocation to jobs and not on task allocation within a job. The nature of task allocation within jobs will be implicitly captured by our workload model and assumptions (see Chapter 3). For example, in many cases it will follow from the workload assumptions that work is redistributed across processors allocated, i.e, dynamic allocation.

## 2.1 Parallel System Models



(a) Centralized Queue (b) Distributed Queues

Figure 2.1: Queueing Models for Parallel Processor Systems

Job scheduling policies in computer systems allocate both processors and memory to jobs in the system. However, most of the parallel scheduling literature has focused only on processor allocation and processors are the only resource in most parallel system models. The simplest and most common parallel system model is the centralized queueing model shown in Figure 2.1a. Parallel jobs arrive at the system and join a central job queue. The system has $P$ identical processors which are allocated to the jobs according to some global processor allocation policy. Example systems for which the centralized queueing model is physically implemented include each processor partition on the CM-5, the "compute" partition on the Intel Paragon, and Uniform shared-Memory Access (UMA) machines such as the Sequent Symmetry and the Encore Multimax. While the central queue potentially achieves optimal load balancing it can also become a system bottleneck if implemented in a naive way. For systems with thousands of processors the central queue is only a conceptual model; efficient implementations of job schedulers will allow for distributed queue access.

This chapter mainly focuses on processor scheduling literature that assumes the centralized queueing model, viz. [1, 24, 25, 28, 27, 40, 41, 43, 44, 47, 48, 49, 50, 53, 54, 55, 63, 66, 69, 70, 71, 72, 80, 82, 84, 91]. A few studies have also examined a distributed queueing model shown in Figure 2.1b, which consists of a global policy for dispatching jobs to one or more processor partitions and a local policy for each partition. Such a distributed queueing model is appropriate for processor pool scheduling in Non Uniform shared-Memory Access (NUMA) machines (cf. [57, 68, 92]) and also for the Distributed Job Manager (DJM) on the CM-5.

A special case of the distributed queueing model is the fork-join queueing model which has been analyzed in [2, 13, 42, 45, 46, 52, 67, 83]. Each processor has a separate task queue associated with it which typically serves tasks in first-come-first-serve order. Jobs arrive at the system, split into one or more independent tasks and the tasks are dispatched to one or more queues based on a global policy, e.g., shortest queue routing. We believe that a fork-join queueing model may not be practical for many parallel systems since there is no attempt to coschedule the parallel tasks of a job. A discussion of the results for fork-join queues is beyond the scope of this thesis. A generalization of the distributed queueing model is the hierarchical task queue model (not shown), the details of which can be found in [12]. Policy comparison results obtained thus far for hierarchical task queues have been the same as those for centralized queues given identical workload conditions.

## 2.2 Parallel Workload Models

Essential features of parallel applications include parallelism, task synchronization and communication, processing requirement, and correlation between parallelism and processing requirement. At one extreme of models for parallel applications are models in which the program structure is explicitly represented by means of a task graph. At the other extreme are models in which a general job structure is captured by an Execution Rate Function (ERF) that measures the rate at which a program executes as a function of the number of processors allocated to the program. A task graph contains more parallelism and synchronization information that an ERF, but an ERF is simpler to parameterize. In between these two structures are application parallelism profile models that measure number of busy processors as a function of execution time of the application. Parallelism profile models provide more parallelism information than ERF models, e.g., parallelism measures such as fraction of sequential work and average parallelism, but less information than task graph models, e.g., less synchronization information. Parallel scheduling literature has mainly used workload models based on the task graph or ERF model for job structure, and hence we will review only these two application models in the rest of this section. For details of parallelism profile models for processor scheduling we recommend well written papers by Sevcik [71, 72].

Note that task graph and ERF models are only used to characterize parallel applications and the task graph or ERF itself need not be known to the scheduler. For example, for a workload based on the task graph model the scheduler model may only have information about instantaneous job parallelism. We next review workload models with task graph program structures and then review workload models with ERFs.

### 2.2.1 Workload models with task graph structures

Studies that explicitly model job parallelism and task synchronization using task graphs as program structures include [28, 27, 40, 41, 43, 47, 50, 53, 54, 66, 68, 80, 82, 84, 91, 92]. For task graph workloads we review models for job parallelism and processing requirement, and also models for correlation between parallelism and processing requirement.

A task graph explicitly shows the structure of a program in terms of task synchronization, and a program is said to complete when all its tasks have completed. The simplest non-trivial task graph structure is the fork-join structure which has been used in many analytic modeling studies. In such a job structure the instantaneous parallelism starts out with $N$ and decreases as tasks complete. Given task service times the task graph structure can be used to obtain several parallelism metrics such as fraction of sequential work, average parallelism, and variance of parallelism.

For workload models with task graph program structures a job's processing requirement has been typically specified in one of two ways[1]:

(1) For a job with $N$ tasks the task processing times, $T_1, \ldots, T_N$, are independent and identically distributed (i.i.d.), within and across all jobs, according to a specific distribution $\mathcal{F}_T$. Task service time distributions considered in processor scheduling studies have been: uniform [91, 92], (truncated) normal [68], exponential [40, 50, 53, 54, 55], and generalized exponential [80].

(2) Total demand $D$ is drawn from a distribution $\mathcal{F}_D$ and $T_i$ is a function of $D$, for $i = 1, \ldots, N$. Job demand distributions considered in processor scheduling studies have been: exponential and two-stage hyperexponential [41, 43]. Tasks service times in these two studies are obtained as follows: $T_i = D/N$ in [41], and $T_i = \frac{U_i}{\sum_{j=1}^{N} U_j} D$ in [43], where $\{U_i\}_{i=1}^{N}$ are i.i.d. uniform random variables in the interval (0,1].

Note that in approach (1) where $D$ is derived from $N$, jobs with few tasks have stochastically smaller demands than jobs with many tasks and demand and parallelism are positively correlated. Thus, correlation between demand and parallelism is implicit in this model. On the other hand, approach (2) for job processing requirement permits independent control of $D$ and

[1] We exclude measurement studies in which processing requirement is not modeled since it is fixed depending on the programs selected for the workload.

$N$. In this case $D$ and $N$ can either be independent, or $D$ can depend on $N$ as in the correlated workload of [43, 41], where the mean demand of a job is proportional to its number of tasks. More general correlation models are also possible by controlling the joint distribution of $D$ and $N$. In measurement studies the correlation between demand and parallelism depends on the workload mix and can be explicitly measured using a parameter such as correlation coefficient between demand and parallelism.

A job's completion time is computed by explicitly determining the time for the last task to complete. If the job was allowed to run stand alone on a number of processors greater than or equal to its maximum parallelism then its completion time would be the critical path time of its task graph. For example, for a fork-join program with $N$ tasks the completion time would be $\max(T_1, \ldots, T_N)$. If the job is assigned $k$ processors throughout its lifetime then its completion time is the critical path time for the schedule on $k$ processors. However, if the number of processors allocated to the job varies over time, including an allocation of zero processors while it is waiting for service, then computing the mean completion time needs more sophisticated techniques, which are reviewed in Section 2.4.

### 2.2.2 Workload models with ERFs

Studies that model job structures using ERFs include [1, 24, 25, 44, 48, 49, 63, 74, 69, 70, 71, 72]. For ERF based workloads we review models for job parallelism and processing requirement, and also models for correlation between parallelism and processing requirement.

An Execution Rate Function, $\gamma$, measures the rate at which a program executes as a function of processor allocation. Thus $\gamma(k)$ is the ratio of a program's execution time on 1 processor to its execution time on $k$ processors. The ERF $\gamma$, also called execution signature [18], implicitly captures task synchronization and communication as a function of processor allocation. The ERF is identical to the program's speedup curve when a program runs stand alone. In models of multiprogrammed systems the ERF is generally assumed to model the *instantaneous execution rate* of a program, even in cases where a job's processor allocation changes over time. Typical ERFs are concave and nondecreasing.

Unlike a task graph, an ERF does not give any information about a job's instantaneous parallelism. Typically, instantaneous parallelism is assumed to be constant throughout the lifetime of the program (i.e., available parallelism) in models where processor allocation to a job can dynamically change with time (cf. [69]). In systems where processor allocation to a job remains fixed throughout the job's execution, it is typically assumed that some parallelism information such as maximum parallelism is specified to the scheduler.

The ERF(s) may either be known to the scheduler as in [91, 72] or unknown to the scheduler as in [25, 63]. If the full ERF is unknown to the scheduler, the scheduler may have statistical information about the ERF such as average parallelism (avg), or processor working set[2] (pws). The job also has a fixed available or maximum parallelism, which is typically known to the scheduler.

In workloads that use ERFs, job processing requirement is modeled by specifying the distribution for total job demand $D$, e.g., an exponential distribution as in [63]. Correlation between demand and parallelism is explicit in ERF workload models and can be controlled by means of the joint distribution of $D$ and $N$. In some analytic models job demand has been assumed to be independent of the parallelism characteristic (cf. [71]), and in other analtyic models, such as the one in [69, 70], mean demand can be correlated with available parallelism, but it has been typical to use measures from specific workload mixes for which the correlation has not been explicitly specified. In simulation studies that use execution signatures of real programs the overall correlation is specific to the mix of programs.

Given an ERF $\gamma$, a job's completion time on $k$ processors is given by $D/\gamma(k)$, which is easy to compute as compared to analyzing the completion time of a task graph. By varying $\gamma$ one can examine the sensitivity of policy performance to ERF sublinearity independent of job demand. If the job's allocation varies over its lifetime due to contention from other jobs, then computing the mean completion time under ERF workloads requires analytic techniques which are reviewed in Section 2.4.

### 2.2.3 Summary of Workload Models

We have thus far classified workload models of parallel programs into two broad categories. A summary of these two types of workload models is given in Table 2.1. We note that ERFs capture fewer parallelism and synchronization details than task graphs, but still capture many essential features of parallel workloads.

The next section classifies processor scheduling policies for multiprogrammed parallel systems.

## 2.3 Classification of Processor Scheduling Policies

The function of a processor scheduling policy is to select jobs for processor allocation and determine how many processors to allocate to each of the selected jobs without violating underlying system constraints. Two types of system constraints have been addressed in the literature, fixed processor partitioning and run-to-completion processor allocation.

[2] The processor working set is defined in [25] as the minimum number of processors that maximizes $\gamma^2(k)/k$. It is also coincides with the number of processors corresponding to the knee of the execution-time efficiency profile [25].

Table 2.1: Parallel Workload Models

| | **Task Graph Models** | **ERF Models** |
|---|---|---|
| Program Structure | Specific task graph | General structure captured by an ERF $\gamma$ |
| Parallelism | Number of tasks per phase | Metrics such as maximum or average parallelism |
| Processing Requirement | (1) $T_i \sim \mathcal{F}_T$, $D = \sum_{i=1}^{N} T_i$<br>(2) $D \sim \mathcal{F}_D$, $(T_1, \ldots, T_N) = f(D)$ | $D \sim \mathcal{F}_D$ |
| Correlation | Implicit in (1), Explicit in (2) | Explicit |
| Service time on $k$ processors | Critical path time for job schedule on $k$ processors | $D/\gamma(k)$ |

Fixed partitioning means that processor partitions are fixed over all time as opposed to adaptive partitioning where processor partitions adapt to the system state as defined by the number and/or characteristics of jobs in the system. Fixed partitioning is motivated by simpler protection mechanisms, e.g., messages cannot be sent or memory cannot be accessed across fixed system boundaries, and thus is easier to implement than adaptive partitioning. Run-to-completion (RTC) processor allocation means that a job runs to completion on the set of processors initially allocated to it, as opposed to dynamic processor allocation where a job's allocation can change dynamically over its lifetime. RTC allocation is motivated by the elimination of context-switch and data movement overheads, and no requirement for applications to adapt to changes in processor allocation in the middle of execution. RTC allocation thus leads to lower scheduling overhead and conceptually simpler programming models than dynamic allocation.

Fixed partitioning and RTC allocation are orthogonal system constraints. Based on the presence or absence of each of these two constraints, four combinations are possible leading to four classes of processor scheduling policies.

**Fixed Partitioning/Run-To-Completion** (FP/RTC): Processors are partitioned at system initiation time and jobs are scheduled stand alone into the predefined partitions according to a specific scheduling rule such as first-come-first-serve. The partition sizes can equal as in [25, 44, 63, 70, 71], or unequal as in [48, 49] where partition size is based on priority classes.

**Fixed Partitioning/Dynamic Allocation**(FP/DA): Processors are partitioned as in FP/RTC but a job's allocation can change over its lifetime, e.g., time-sharing of jobs in a partition, as in the CM-5.

**Adaptive Partitioning/Run-To-Completion** (AP/RTC): Processor partitions can adapt to the system state as defined by number and/or characteristics of jobs in the system, with the constraint that a job's processor allocation remains unchanged throughout its execution, i.e., the job runs to completion once it receives its processor allocation.

**Adaptive Partitioning/Dynamic Allocation** (AP/DA): Processor partitions can adapt to the system state and a job's allocation can change over its lifetime. The processor allocation may be either preemptive or nonpreemptive.

FP/RTC and AP/RTC policies have been referred to as static policies in the literature whereas AP/DA policies have been referred to as dynamic policies. FP policies are simple to implement but they can lead to inefficient processor utilization as compared to AP policies among which the AP/DA policies potentially offer better processor utilization but at the cost of higher scheduling overhead. Several studies have analyzed FP/RTC policies [25, 44, 48, 49, 63, 69, 70, 68, 71, 74]. Surprisingly, FP/DA policies have not been examined at all in the literature. AP/RTC policies have been studied in [24, 25, 40, 48, 49, 63, 69, 70, 71, 72, 74, 91], and AP/DA policies have been studied in [1, 27, 28, 40, 41, 43, 44, 47, 48, 49, 50, 53, 54, 66, 68, 69, 70, 80, 82, 84, 91, 92].

We describe FP/RTC, AP/RTC and AP/DA policies that will be reviewed in Sections 2.4 and 2.5. In order to create more meaningful names for policy comparisons across studies and also to remove naming conflicts with the terms "AP" and "RTC", we have changed the names and acronyms of some of the scheduling policies in the literature. The policies below are listed in the order of FP/RTC, AP/RTC and AP/DA, and within each category in alphabetic order.

**FP($\frac{P}{n}$)** FP/RTC with $P/n$ partitions of size $n$ each, where $n$ evenly divides the number of processors $P$. Jobs are scheduled onto partitions in first-come-first-serve order.

**ASP** (Adaptive Static Partitioning) In this AP/RTC policy, proposed in [69, 70], free processors are assinged at job arrival and departure instants to jobs that have not received service, one at a time in round robin order, under the constraint that no job gets more processors than its maximum parallelism. [Note that the round robin order of allocation determines the number of processors allocated to each waiting job and it should not be confused with round robin service as in time sharing systems.]

**ASP($m$)** Same as ASP except that jobs receive a maximum allocation of $m$ instead of their maximum parallelisms, where $m$ is a fixed constant. This is similar to the EPM policy in [63].

**avg based policies** Four AP/RTC policies that allocate processors using average job parallelism and possibly additional parallelism characteristics have been proposed in the literature: two versions of **AVG**, and one each of **AVG+**, and **AVG+&mM**. (Sevcik [71] denotes these policies by A, A+, and A+&mM.) The AVG policy uses only average parallelism for processor allocation. In [71] the scheduler allocates free processors to waiting jobs in a first-fit manner with each selected job getting its average parallelism, whereas in [40] the scheduler allocates free processors are among waiting jobs *proportional* to their average parallelisms. We survey results for both versions of AVG in this chapter; the relevant version will be clear from context. The AVG+ and AVG+&mM policies proposed in [71] use more job information than average parallelism and adapt processor allocation to system load. Under AVG+ the allocation to a job is a function of its average and variance of parallelism and the offered system load, whereas under AVG+&mM the allocation is also based on minimum and maximum job parallelism. In each of these two policies the scheduler selects jobs in first-fit order.

**IP** (Insurance Policy) Same as ASP except that a fraction of the free processors are reserved for future arrivals. This fraction is a function of a prespecified parameter as well as the number of jobs in the system [63].

**METB** (Maximum Execution Time Benefit) This AP/RTC policy (called RTC in [91]) allocates free processors one at a time to waiting jobs in the order of maximum execution time benefit. More precisely, if $\tau_j(k)$ is the execution time of job $j$ on $k$ processors then the METB policy allocates the free processors one at a time to the waiting jobs, $Q^w$, up to a maximum of their maximum parallelisms in descending order of $\tau_j(p_j) - \tau_j(p_j + 1)$, where $p_j$ is the number of free processors already reserved for job $j$, $j \in Q^w$, and $\tau_j(0) = \infty$.

**pws based policies** Four AP/RTC policies based on processor working set (pws) are proposed in [25]. We name them as follows: **PWS**, **PWS-LA**, **PWS-FCFS**, **PWS-FCFS-LA**, the original names being FF, FF+LA, FF+FIFO, and FF+FIFO+LA, respectively [25]. Only the LA policies allow for more than pws allocation to a job. Under each of the four policies the scheduler scans jobs within a window of size $w$ and allocates free processors in a first-fit manner with each selected job getting its processor working set (pws) allocation. If there are idle processors after the first-fit allocation PWS-LA allocates them to the first job in the window whose pws is less than the number of free processors, PWS-FCFS allocates them to the first waiting job, and PWS-FCFS-LA is like PWS-FCFS except that if all jobs in the window get their pws allocations then all remaining free processors are given to the first job in the window whose pws is less than the number of remaining free processors.

**RA** The Robust Adaptive policy in [63, 74] computes a "target" partition size which equals the total number of processors divided by the number of waiting jobs, and allocates processors equal to the target size. (Jobs wait in the queue if the target number of processors are not available.) A number of minor variants of this policy are proposed in [63].

**EQ** The AP/DA EQuiallocation policy allocates an equal fraction of processing power to each job in the system unless a job has smaller parallelism than the equiallocation value, in which case each such job is allocated as many processors as its parallelism, and the equiallocation value is recursively recomputed for the remaining jobs. Reallocation of power can occur on job arrivals, job departures, and changes in a job's available parallelism.

**EQS** The Spatial EQuiallocation policy (EQS) is an EQ policy in which processing power is allocated spatially for integral allocation and temporally for fractional allocation. For example, if a job is to receive an allocation of 27.5 units of processing power, then it is allocated 27 processors and it receives an additional 0.5 units of processing power by time sharing an additional processor (i.e., the job alternately executes on 27 and 28 processors). Ignoring variations in implementation details, the EQS policy was first defined in [82].

**EQT** The Temporal EQuiallocation (EQT) policy is an EQ policy in which processing power is allocated temporally by means of idealized round robin scheduling. When a job with available parallelism $N$ is scheduled it receives $N$ quanta of size $Q/N$. Thus the overall quanta, $Q$, per job is independent of the available parallelism. When $Q \rightarrow 0$ this policy is an equiallocation policy [40]. Again ignoring implementation details, this policy was first called Round Robin Job and defined in [41].

**FCFS** This is an AP/DA policy in which processors are allocated to jobs on a First-Come-First-Serve basis. Each job is allocated processors as they become available up to a maximum of its parallelism. Processors released by a departing job are first allocated to the job in service (if any) whose allocation is less than its parallelism and then to jobs waiting for service.

**PSAPF** (Preemptive Smallest Available Parallelism First) The central job queue is a preemptive queue that is organized in ascending order of available job parallelism. Jobs with the same available parallelisms are served in first-come-first-serve order. 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.

**PSCDF** (Preemptive Smallest Cumulative residual Demand First) Like PSAPF except that job priority is cumulative residual processing demand instead of available parallelism and instantaneous parallelism is used for maximum allocation instead of available parallelism.

**PSNPF** (Preemptive Smallest Number of Processes First) Like PSAPF except that instantaneous parallelism is used instead of available parallelism.

**RRJ** Round Robin Job. See definition for EQT.

**RRP** Round Robin Process. There is a single queue of processes from all jobs in the system, and these processes are served in round robin order.

**RR-slot** This is a coscheduling policy that rotates processors among slots on a quantum basis. An arriving job is either assigned to a slot with the maximum number of unused processors or to a new slot, such that the average execution rate of the job is maximized, where the number of processors assigned to the job is less than or equal to its average parallelism. By average execution rate is meant the speedup of the job on its allocated processors divided by the total number of slots, $n$, in the system including the new slot for the job, if any (the job gets serviced once in $n$ slots). This policy was proposed in [91].

**SCDF** (Smallest Cumulative Demand First) Non-preemptive version of PSCDF.

**SNPF** (Smallest Number of Processes First) Non-preemptive version of PSNPF.

**SNQPF** (Smallest Number of Queued Processes First) Like SNPF except that job priority is the number of processes waiting in in the queue. This policy is called SQ in [53].

**UnequalDP** (Unequal Dynamic Partitioning) In this AP/DA policy, proposed in [91], processor allocation instants include job arrivals, job departures and changes in job parallelism. When a job requests one or more processors (including job arrival) idle processors are first used to satisfy the request. If there are no idle processors and the job is a new arrival then UnequalDP preempts a processor from a job with two or more processors and gives it to the incoming job. When processors are released either to due job departures or decrease in job parallelisms they are first assigned one at a time to waiting jobs and the rest of the processors (if any) are assigned on a FCFS basis. This policy is called Dynamic in [91], but has been referred to as UnequalDP or UnEqDP in more recent literature [39, 69].

The implementation of EQS in [82, 27] which we refer to by $\text{EQ}_{PC}$ is based on a process control approach where the runnable processes of an application are controlled in response to the number of processors it is supposed to use. $\text{EQ}_{PC}$ tries to achieve equipartitioning at job arrivals, departures, and periodic intervals of time that are system dependent (6 seconds in [82] and 300ms in [27]). The implementation of EQS in [47] reallocates processors among jobs in response to instantaneous parallelisms and we refer to it by $\text{EQ}_{IP}$ .

We have thus far surveyed parallel system and workload models and also classified processor allocation policies that have appeared in the literature. The next section surveys performance evaluation techniques for scheduling policies under various workload assumptions.

## 2.4 Performance Evaluation Techniques

The most important performance metric from a user's point of view is system response time and our goal is to find scheduling policies that minimize mean response time over a wide range of workload parameter values. With this objective in mind, Section 2.4.1 summarizes previous methods for estimating the mean response time of parallel processor policies under a variety of workload conditions, and Section 2.4.2 reviews optimal policy results and response time bounds under particular workload assumptions. We will use the notation $\overline{R}_\Psi$ to denote the mean response time under a parallel processor policy $\Psi$.

### 2.4.1 Mean Response Time Estimation

The mean response time of a policy can be quantitatively estimated using analytic modeling, simulation, or measurement. We review the use of each of these three techniques in the literature.

#### 2.3.1.1 Analytic Modeling Techniques

Analytic modeling techniques for estimating mean response times of parallel processor scheduling policies have broadly fallen into five categories.

- Birth death models
- Matrix-geometric analysis
- Recurrence relations
- Reductions to known queueing systems

- Bulk arrival queues

Birth death models need very specific workload assumptions and thus have limited applicability. Matrix-geometric analysis and recurrence relations are based on state space enumeration and the numerical analysis involved provides no ready insight into policy performance. Moreover both matrix-geometric analysis and recurrence relations are limited to small system sizes. Some studies have reduced parallel scheduling policies to known queueing systems such as M/M/c or M/G/c queues, but to do so they have made very specific assumptions such as exponential demands and constant parallelism. Bulk arrival queueing theory has been used only in the context of workloads with i.i.d. task service times. The applicability of the results for i.i.d. task service times may be limited, since in general it is unlikely that task service times in real systems will be i.i.d. within and across all jobs and this assumption can influence relative policy performance.

Table 2.2 summarizes the analytic models for parallel processor scheduling on a per-policy basis. Both task graph and ERF models are summarized. The columns of Table 2.2 contain workload models, techniques, references, and applicability limitations for analytic models of specific parallel processor policies. The workload model column shows that task graph models for all policies analyzed have been only of the fork-join type, and that in nearly all studies for AP/RTC policies only ERF models have been used. The limitations column shows that common assumptions that may limit the applicability of analytic results include exponential demands or task service times, and no job arrivals.

We have seen that exact analysis such as matrix-geometric analysis or bulk arrival queueing theory is either limited to small systems sizes for computational reasons or requires specific assumptions about the workload such as i.i.d. task service times, which suggests that approximate analysis may be necessary for more general workload models. There has been one approximate analysis of specific types of AP/RTC policies by Gelenbe et al. [24]. To apply their analysis, however, 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 analytically for practical AP/RTC policies such as ASP.

#### 2.3.1.2 Simulation

Simulation studies that have compared processor scheduling policies include [25, 41, 43, 48, 49, 63, 66, 71, 74, 91, 92]. Assumptions that may limit the applicability of these studies include specific demand distributions such as exponential demands (per class), i.i.d. uniform task service times, implicit correlation between demand and parallelism, specific synchronization or division of job demand, and small system size. Note that the applicability limitations arise primarily on account of the technique since it is impossible to span across general distributions using simulation. Some simulation studies [43, 41] have broader applicability, on account of the use

Table 2.2: Analytic Models for Parallel Processor Scheduling

| Policy | Workload Type | Technique | Refs. | Limitations on Applicability |
|---|---|---|---|---|
| **FP/RTC** | | | | |
| FP($\frac{P}{n}$) | Fork-join jobs | Reduction | [68] | |
| | ERF Model | Reduction | [71, 63] | $D \sim$ exp |
| **AP/RTC** | | | | |
| ASP | ERF model | Matrix-geometric | [69, 70] | $P$ small, $D \sim$ exp per class |
| AVG | ERF model | Reduction | [71] | $D \sim$ exp, $N = k^+$, single ERF |
| pws Policies* | ERF model | Matrix-geometric | [69, 70] | $P$ small, $D \sim$ exp per class |
| RTC policies† | Fork-join jobs | Average of mean service times | [40] | $T_i$ iid exp, no arrivals |
| **AP/DA** | | | | |
| AVG | Fork-join jobs | Recurrence relatns | [40] | $T_i$ iid exp, no arrivals |
| RRJ | Fork-join jobs | Recurrence relatns | [40] | $T_i$ iid exp, no arrivals |
| EQ | ERF model | Matrix-geometric | [69] | $P$ small, $D \sim$ exp per class |
| FCFS | Fork-join jobs | Bulk arrival queue | [54, 55, 80] | $T_i$ iid exp or GE |
| | | Matrix-geometric | [50] | $T_i$ iid exp |
| | | Recurrence relatns | [40] | $T_i$ iid exp, no arrivals |
| Fixed priority | Fork-join jobs | Bulk arrival queue | [53] | $T_i$ iid exp |
| PSNPF | Fork-join jobs | Recurrence relatns | [40] | $T_i$ iid exp, no arrivals |
| RRP | Fork-join jobs | Recurrence relatns | [40, 80] | $T_i$ iid exp or GE, no arrivals in [40] |
| SNQTF | Fork-join jobs | Bulk arrival queue | [53] | $T_i$ iid exp |
| Threshold | ERF model | Birth death chain | [44] | $D \sim$ exp, $N = k$, single ERF |
| UnequalDP | ERF model | Matrix-geometric | [69] | $P$ small, $D \sim$ exp per class |

+ $N = k$ denotes an identical value of the parallelism metric for all jobs.

* PWS-FCFS, and PWS-FCFS-LA.

† The RTC policies are modeled are for a fixed number of jobs where each job is allocated processors once and for all at time $t = 0$.

of hyperexponential demands which have more variation than the exponential. (It is typical in computer system workloads to have high variation in processing demand.) It is unknown what constitutes a realistic distribution for parallelism and the specific distributions of parallelism in simulation studies cover only a narrow region of the design space. Moreover, simulation only provides numerical solutions and thus no ready insight into policy performance.

#### 2.3.1.3 Measurement

There have been very few measurement studies for parallel processor policies and these studies have evaluated only AP/DA policies on UMA systems with 4, 16, or 20 processors [27, 47, 82, 84]. A possible reason for the scarcity of measurement studies is that there has been no parallel workload characterization to date and thus it is unknown what constitutes a realistic program mix for policy evaluation. The workloads used in measurement studies consist of three to four parallel programs with different execution rate profiles. It is difficult to vary system parameters in measurement studies and thus limited insight is provided into policy performance. Moreover, the emphasis in these studies has been on implementation aspects rather than processor allocation issues (e.g., spatial versus temporal equipartitioning, affinity versus non-affinity scheduling, process control versus no process control).

We have thus far surveyed analytic, simulation, and measurement studies that estimate the mean response time of various parallel processor policies under different workloads. We next survey results for optimal policies and mean response time bounds.

### 2.4.2 Optimal Policies and Mean Response Time Bounds

Most of the optimal policy results and response time bounds derived in the literature use the technique of sample path analysis where one couples sample paths of equal probability between two systems and shows that over every pair of coupled sample paths the performance of one system is better than the other. The advantage of sample path analysis is that one does not have to solve for response times in order to compare systems and the comparisons do not involve detailed mathematics. Proofs using sample path analysis typically require inductive arguments. The applicability of sample path analysis is however very problem specific since (stochastic) dominance relationships may not always hold for the performance of two (or more) systems.

We first survey optimal policy results and then survey response time bounds for parallel processor scheduling policies under various workload assumptions.

#### 2.3.2.1 Optimal Policy Results

There have been few optimal policy results in the literature for parallel processor scheduling. These are summarized in Table 2.3. We note from the the table that a majority of the optimal policy results are for AP/DA policies. In each case, fairly restrictive assumptions are made such as i.i.d. exponential task service times or demands for the PSNTF and PSAPF results, and fully parallel jobs and linear ERFs for the shortest demand first policies. Thus the results need to be used with caution on account of their possible sensitivity to workload assumptions. For example, Agrawal et al. [1] show that PSAPF is not optimal when demand distribution has more variation than the exponential or when the ERF is sublinear. Leutenegger [39] gives a counterexample to show that PSCDF is not optimal under a specific workload setting.

#### 2.3.2.2 Response Time Bounds

There have been few results in the literature concerning response time bounds other than optimal policy bounds. However, the applicability of the results is broader than the applicability of the optimal policy results reviewed above on account of more general workload assumptions. Response time bounds for classes of parallel processor policies and for individual policies have appeared in [1, 80, 53]. Table 2.4 summarizes the results in these studies. For the last row of the table, the TP, JP, and NP policies are three fixed priority policies in [53], namely, task preemption (TP), job preemption of partially completed jobs (JP), and no preemption (NP), and $R_i$ is the response time of the $i^{th}$ priority class.

All optimal policy results and bounds reviewed in this section have been for centralized queueing system models. Optimal policy results and bounds for fork-join queueing models can be found in [2, 3, 13, 42, 45, 46, 67].

## 2.5 Results in the Literature

In Section 2.4 we reviewed models for many parallel processor policies under various workload assumptions. In this section we examine the results, emphasizing how processor scheduling policies compare against each other. Since previous studies have made specific workload assumptions our main objective is to summarize results for high performance policies over specific regions of the design space. On the way we also show examples of contradictory results reported in the literature because of different workload assumptions.

Section 2.5.1 discusses policy comparison results for FP/RTC policies, Section 2.5.2 for AP/RTC policies, and Section 2.5.3 for AP/DA policies. We compare results for the performance of AP/RTC policies versus the performance of AP/DA policies in Section 2.5.3. Throughout this section we use the notation $C_D$ to denote coefficient of variation in cumulative job demand $D$. As before we use the notation $N$ to denote number of tasks in fork-join program models and available or maximum parallelism in ERF models.

Table 2.3: Optimal Policy Results for Parallel Processor Scheduling

| **Optimal Policy** | **Workload Assumptions** | **Scheduler Information and Assumptions** | **Refs.** |
|---|---|---|---|
| **FP/RTC** | | | |
| FP(1) | Linear ERFs, $C_D = 0$, fully parallel jobs | FP($\frac{P}{n}$) scheduler that knows job arrival times | [68] |
| **AP/RTC** | | | |
| Allocate processors proportional to $\sqrt{D}$ | Fixed number of jobs. Execution time of job $j$ on $k$ processors: $\tau_j(k) = a_{1j}D_j/k + a_{2j} + a_3k$ | Scheduler has complete knowledge of $D_j$, $a_{1j}$ $a_{2j}$, and $a_3$ | [72] |
| **AP/DA** | | | |
| PSNTF | Fixed number, $K$, of fork-join jobs with iid exp. task service times | Scheduler knows instantaneous job parallelisms | [40] |
| PSAPF | General arrivals, available parallelisms, linear ERFs, and iid exp. job demands independent of everything else | Scheduler knows available parallelisms, arrival times, and any other information that does not reveal job demands | [1] |
| SCDF | Fixed set of fully parallel jobs with linear ERFs | Scheduler knows residual job demands | [72] |
| PSCDF | Poisson arrivals, fully parallel jobs with linear ERFs | Scheduler knows residual job demands | [72] |
| SEDF | Poisson arrivals, fully parallel jobs with linear ERFs | Non-preemptive scheduler that knows expected job demands | [72] |

Table 2.4: Response Time Bounds for Parallel Processor Scheduling

| **Result** | **Workload Assumptions** | **Refs.** |
|---|---|---|
| $\overline{R}_{\Psi} \geq \overline{R}_{\Psi_1^*}$, $\Psi$ is a parallel processor policy, and $\Psi_1^*$ is the corresponding optimal uniprocessor policy$^{+}$ | ERF model, general workload | [1] |
| $\overline{R}_{\Psi}(N = P) \leq \overline{R}_{\Psi} \leq \overline{R}_{\Psi}(N = 1)$, for all processor-conserving policies $\Psi$ | General arrivals, linear ERFs, $D \sim$ exp, $N$ general$^{\dagger}$ | [1] |
| $\overline{R}_{RRP} \geq \overline{R}_{DPS}$ | ERF model, general workload | [1] |
| Lower and upper bounding systems for RRP | Poisson arrivals, fork-join jobs, iid GE task service times, $N$ general | [80] |
| $R_1^{TP} \leq_{st} R_1^{JP} \leq_{st} R_1^{NP} \leq_{st} R_1^{FCFS}$,<br>$R_K^{TP} \geq_{st} R_K^{JP} \geq_{st} R_K^{NP} \geq_{st} R_K^{FCFS}$ | Poisson arrivals, fork-join jobs, iid exp. task service times, $N$ general | [53] |

$^{+}$ $\Psi_1^*$ uses same workload information as $\Psi$ and schedules workload on a uniprocessor of power $P$

$^{\dagger}$ $N$ denotes maximum parallelism in ERF models and number of tasks in fork-join models

### 2.5.1 FP/RTC Policies

Key issues in the design of high performance FP/RTC policies are:

- How should processors be partitioned, i.e., how many partitions and how many processors per partition.
- In what order should jobs be scheduled.

Studies in the literature that have compared FP/RTC policies have only focused on FP($\frac{P}{n}$) policies with $\frac{P}{n}$ equal sized partitions and FCFS scheduling. The only key question for FP($\frac{P}{n}$) policies is how many partitions are required for high performance. In Section 2.3.2.1 we noted that it has been shown in [68] that for a workload with $C_D = 0$, linear ERFs and fully parallel jobs, the FP(1) policy is optimal throughout the range of system utilization. However, experimental results for workloads with task synchronization and/or communication overheads show that the optimal number of partitions increases as a function of system load [71, 25, 44, 69, 70, 68, 63]. (In [25, 69, 70] the overall $C_D$ is greater than 1, in [71, 44, 63] $C_D$ is equal to 1, and in [68] $C_D$ is less than 1.) In all these studies it is shown that at very low loads ($\rho \to 0$) the optimal partition size is equal to the maximum parallelism in the workload and at very high loads ($\rho \to 1$) the optimal partition size is either 1 or 2 depending on the sublinearity of the ERF[3]. All studies explain this result by noting that at high loads a small partition size leads to more efficient processor utilization since job speedup curves tend to be close to linear at low values of processor allocation.

The experimental results mentioned above show that the optimal value of $\frac{P}{n}$ depends on the system load $\rho$, the overall $C_D$, workload parallelism, and ERF sublinearity, and it increases with $\rho$ in general. This has motivated the use of AP/RTC policies that give high allocation at low loads and low allocation at high loads.

### 2.5.2 AP/RTC Policies

Key issues in the design of high performance AP/RTC policies are:

- How many processors should be allocated to each job, and on what basis should the allocations be made.
- In what order should jobs be scheduled.

Studies have suggested that allocating processors equal to the knee of the execution-time efficiency profile, that is the number of processors that maximizes the ratio of efficiency to execution

[3] In case where the ERF has $\gamma(2) < 2$ the optimal partition size is 1.

time, achieves good performance for individual jobs [19, 44, 25]. The processor working set (pws) measure is the number of processors that coincides with the knee of the execution-time efficiency profile [25]. An allocation of average parallelism (avg) to a job has speedup and efficiency properties similar to an allocation of the knee [19]. This has motivated researchers to examine whether application characteristics such as avg and pws are useful for multiprogrammed scheduling in general.

Several AP/RTC policies have been compared under various workload assumptions in the literature [71, 91, 25, 40, 69, 70, 63, 74, 72]. All but the study of [63] compare policies that use application characteristics such as avg or pws against policies that don't. The study of [63] compares only policies that adapt their allocation to number of jobs in the system but do not use application characteristics for scheduling. We first review results for AP/RTC policies that use at most maximum job parallelism or job parallelism metrics such as avg or pws, and follow it by a review of results for AP/RTC policies that use additional application characteristics such as variance of parallelism and minimum parallelism. Finally, we discuss the results to date on the design of AP/RTC policies with good overall performance.

#### 2.4.2.1 Policies that use maximum parallelism, avg, or pws

Leutenegger and Nelson [40] give experimental data to show that for a fixed set of $K$ fork-join jobs with i.i.d. exponential task service times

$$\overline{R}_{AVG} < \overline{R}_{EQ-RTC} < \overline{R}_{MOD-FCFS}, \qquad \text{no arrivals, fork-join jobs with } T_i \text{ iid exp.}$$

In their model for RTC policies each of the $K$ jobs is allocated at least one processor. The AVG policy statically allocates processors proportional to average parallelisms, EQ-RTC statically allocates processors in an equipartitioned manner, and MOD-FCFS statically allocates processors in an FCFS manner with the additional constraint that each job receives at least one processor. Leutenegger and Nelson explain that AVG has higher performance under the given workload conditions since it leaves fewer processors idle compared to EQ-RTC and MOD-FCFS. As noted by them this is because jobs with few tasks have smaller average parallelisms and stochastically smaller demands and are thus allocated a smaller fraction of the processors by AVG, which results in fewer idle processors upon their departure. For most of their experiments Leutenegger and Nelson note that the allocation under AVG is close to the optimal allocation (the optimal allocation was determined using integer programming).

We now consider workloads in which jobs arrive continuously to the system as opposed to having only a fixed set of jobs to begin with. To examine whether avg is a good characteristic for processor allocation in a multiprogrammed mix of jobs Sevcik [71] compares AVG with first-fit scheduling against FP($\frac{P}{n}$) policies for a workload with exponential demands and identical avg for all jobs in the system. In his experiments if a job is allocated $k$ processors it executes with rate $\gamma(k)$, where $\gamma$ is derived from a given application structure. Sevcik first argues that for nearly linear ERFs AVG should perform better than FP($\frac{P}{n}$), and then provides experimental data to show that for sublinear ERFs

$$\overline{R}_{AVG} < \overline{R}_{FP(\frac{P}{n})}, \quad \text{for all } n, \quad D \sim \text{exp, and } 0.2 \leq \rho \leq 0.5, \text{ sublinear ERFs,}$$

and

$$\overline{R}_{FP(\frac{P}{n})} < \overline{R}_{AVG}, \quad \begin{array}{l} D \sim \text{exp and sublinear ERFs, where} \\ n > \text{avg for } \rho < 0.2 \text{ and } n < \text{avg for } \rho > 0.5, \end{array}$$

where $n$ is the size of processor partitions.

Majumdar et al. [44] compare PWS against FP($\frac{P}{n}$) for a similar workload as Sevcik [71] and obtain results similar to Sevcik's study, i.e., under exponential demands allocating at the knee is optimal for linear ERFs and is good for sublinear ERFs at low to moderate loads only.

Ghosal et al. [25] compare PWS-FCFS against FP($\frac{P}{n}$) and give experimental data for a workload with five job classes each with exponential demands, and different ERFs, pws's and mean demands per class, to show that in general [4],

$$\overline{R}_{PWS-FCFS} < \overline{R}_{FP(\frac{P}{n})}, \quad \begin{array}{l} \text{5 classes with different ERFs and} \\ \text{pws's, } D \sim \text{exp per class.} \end{array}$$

They explain that PWS-FCFS fragments processors with increasing load on account of the different pws values in the workload and thus the number of processor partitions increases with the number of jobs in the system.

Ghosal et al. also compare PWS-FCFS with PWS, PWS-LA, and PWS-FCFS-LA. They give experimental data to show that in general,

$$\overline{R}_{PWS-FCFS} < \overline{R}_{PWS-FCFS-LA} < \overline{R}_{PWS} < \overline{R}_{PWS-LA}, \quad \begin{array}{l} \text{5 classes, different ERFs and} \\ \text{pws's, } D \sim \text{exp per class.} \end{array}$$

Since LA policies can allocate more than the pws, Ghosal et al. conclude that allocating more processors than the pws is not beneficial for system performance. They also explain that the reason why PWS-FCFS and PWS-FCFS-LA perform better is that they are more adaptive to the number of waiting jobs, i.e., schedule more jobs into service, since they can allocate less than the pws, which is not so under PWS and PWS-LA.

Setia and Tripathi [69, 70] compare PWS-FCFS, PWS-FCFS-LA, and FP($\frac{P}{n}$) policies for a two class workload model with exponential demands, and different ERFs, pws's and mean demands per class. Their overall conclusions for these policies are the same as Ghosal et al.'s

[4] In their results FP(P) and FP(P/2) are better than better than PWS-FCFS only over very narrow ranges of the spectrum of offered load.

conclusions except that they give data to show that PWS-FCFS-LA has higher performance than PWS-FCFS at light load ($\rho < 1$).

Setia and Tripathi also compare PWS-FCFS against ASP to show that

$$\begin{array}{ll} \overline{R}_{ASP} \approx \overline{R}_{PWS-FCFS} \text{ for } \rho \leq 0.5, & \text{2 classes with different ERFs} \\ \overline{R}_{ASP} < \overline{R}_{PWS-FCFS} \text{ for } \rho > 0.5 & \text{and pws's, } D \sim \text{exp per class.} \end{array}$$

(PWS-FCFS performs marginally better than ASP at $0.1 \leq \rho \leq 0.5$.) They attribute the higher performance of ASP at moderate to high loads to the fact that PWS-FCFS saturates at lower loads than ASP since PWS-FCFS does not necessarily allocate one processor per job when $\rho \rightarrow 1$ as ASP does. They therefore conclude that at moderate to high loads it is more important to divide available processors among all waiting jobs than to satisfy the pws requirements of a subset of the waiting jobs.

ASP allocates processors up to the maximum parallelism of a job. A related policy is ASP($m$) which is like ASP except that it allocates up to a predefined maximum of $m$ processors per job. The ASP($m$) policy is studied by Rosti et al. [63] who compare AP/RTC policies that do not use parallelism characteristics but adapt processor allocation to number of free processors or number of jobs in the system. They give simulation data to show that for a single class workload with exponential job demands, fully parallel workloads, and a specific ERF, in general ASP($m$) has better performance than IP, which reserves a fraction of processors for future arrivals. However, they also show that the best choice of $m$ is sensitive to the workload. The RA policy is shown in [63] to have similar performance as the best ASP($m$) policy under exponential demands, but their data shows otherwise for nonexponential demands with high $C_D$.

#### 2.4.2.2 Policies that use additional parallelism characteristics

The policies compared above use at most maximum parallelism or parallelism characteristics such as avg or pws for processor allocation. We now examine policy comparison results for policies that use additional application characteristics such as variance in parallelism and minimum parallelism. Sevcik [71] examines policies that use variance of parallelism or variance, maximum and minimum parallelism in addition to average parallelism, and shows that in general for a workload with exponential job demands and specific application structures (with low and high variance in parallelism)

$$\begin{array}{ll} \overline{R}_{AVG+\&mM} \approx \overline{R}_{AVG+} \approx \overline{R}_{AVG} \text{ for } \rho \leq 0.4, & D \sim \text{exp, specific application} \\ \overline{R}_{AVG+\&mM} < \overline{R}_{AVG+} < \overline{R}_{AVG} \text{ for } \rho > 0.4 & \text{structures, avg} \sim \text{uniform.} \end{array}$$

AVG+&mM and AVG+ adapt to system load whereas AVG does not. $AVG$+ uses variance of parallelism in addition to average parallelism, and $AVG + \&mM$ uses variance of parallelism, minimum and maximum parallelism in addition to average parallelism. This gives some evidence to show that use of additional parallelism characteristics helps in reducing mean response time. These results are shown for specific policies under exponential demands and specific application structures and it remains to be seen whether additional parallelism characteristics are useful for other types of policies and also when $C_D$ has a higher value.

#### 2.4.2.3 Summary

To summarize, we have seen that high performance in AP/RTC policies can be achieved only if the processor allocation to jobs decreases with an increase in system load. For this reason ASP which does not use parallelism characteristics (except for maximum allocation) performs better than PWS-FCFS. At $\rho \rightarrow 0$ the optimal processor allocation is $P$ and at $\rho \rightarrow 1$ the optimal processor allocation is 1. ASP satisfies both conditions and has the potential to be a high performance AP/RTC policy. The performance of ASP, however, needs to be studied under a broader range of workload conditions than has been done in the past. These include higher values for $C_D$, more variability in $N$, and correlated as well as uncorrelated workloads.

In Sevcik's study we note that the AVG+ and AVG+&mM policies that outperform AVG are both adaptive to load unlike AVG. Thus we first need to understand AP/RTC policies with respect to their adaptation to number in the system, i.e., how well they divide processors among waiting jobs. The question of whether or not application characteristics in addition to avg are needed for scheduling can be fully answered only if policies with similar adaptive properties are studied. This is an interesting question in its own right and is beyond the scope of this thesis.

### 2.5.3 AP/DA Policies

AP/RTC policies have low partitioning overhead as compared to AP/DA policies, but their overall performance can suffer to due static allocation. For example, if the processor allocation to a job is low then the job cannot use additional processors even if they are idle, and if the processor allocation to a job is high then the job can delay waiting jobs if it has a large demand. This motivates the use of AP/DA policies in which the allocation to jobs can dynamically change over time. In this section we first review previous comparison results between AP/DA policies and AP/RTC policies. We then review comparison results for AP/DA policies.

AP/DA policies have higher repartitioning overheads as compared to AP/RTC policies but they have the potential to provide better service to jobs based on instantaneous job characteristics such as instantaneous parallelism. Zahorjan and McCann [91] show that for a workload with i.i.d. uniform task service times the UnequalDP policy which does not use information about job execution rates but preemptively attempts to allocate at least one processor per job outperforms the METB (Maximum Execution Time Benefit) AP/RTC policy that uses complete information about job execution times, even though the UnequalDP policy has high task switching overheads. Setia and Tripathi [69] show $\mathrm{EQ}_{PC}$ to perform better than ASP for exponential demands per class when the system load is moderate. In their experimental comparisons $\mathrm{EQ}_{PC}$ incurs higher scheduling overhead than ASP.

Although AP/RTC policies may perform worse than well designed AP/DA policies it is not true that dynamic allocation alone guarantees better performance. If the dynamic allocation is performed temporally rather than spatially it can degrade performance. For example, Zahorjan and McCann [91] show that the AP/RTC policy METB outperforms the AP/DA policy RR-slot which rotates jobs on a time-sharing basis. Under RR-slot jobs execute at less efficient points of their speedup curves than under METB and this causes it to perform worse. We next review the literature for comparisons among AP/DA policies.

AP/DA policies compared in the literature include Coscheduling policies, FCFS, RRP, EQ, dynamic AVG, SNPF, PSNPF, PSAPF, SNQTF, Fixed Priority policies, SCDF, and PSCDF. We review the literature in the order of results for RRP, Coscheduling and FCFS, followed by results for policies that attempt to approximate shortest job first but do not use job demand information, and we then review results for equiallocation policies and finally results for policies that use job demand information.

Ousterhout [58] proposed coscheduling policies for multiprogrammed parallel systems in which all tasks of a job are scheduled at the same time and jobs are scheduled in round-robin order. Seager and Stichnoth [66] give data to show that for a workload with correlation between demand and parallelism and high barrier synchronization overheads

$$\overline{R}_{Coscheduling} < \overline{R}_{RRP}, \qquad \begin{array}{l} T_{ij} = B_j + Normal(0,\sigma), B_j \sim \text{exp}, \\ \text{high barrier synchronization overheads.} \end{array}$$

On the other hand Leutenegger and Vernon [41] show that

$$\begin{array}{ll} \overline{R}_{Coscheduling} \leq \overline{R}_{RRP}, & \text{high lock contention, } D \sim H_2, T_i = D/N; \\ \overline{R}_{RRP} \leq \overline{R}_{Coscheduling}, & \text{independent processes, } D \sim H_2, T_i = D/N. \end{array}$$

Leutenegger [39] notes that the relative policy orderings remain unchanged for uneven division of demand among tasks.

Towsley et al. [80] compare FCFS, RRP, and PS for a workload with fork-join jobs having i.i.d. GE task service times, where the PS (processor sharing) policy schedules jobs as if they have only one task each. They show that, in general, when coefficient of variation, $C_v$, of task service times is less than 4,

$$\overline{R}_{FCFS} < \overline{R}_{RRP} < \overline{R}_{PS}, \qquad T_i \text{ i.i.d. GE}, C_v \leq 4.$$

They explain that RRP outperforms PS due to better processor utilization since processes of a job are not scheduled in parallel under PS. They explain that RRP performs worse than FCFS since it gives lower priority to jobs with fewer tasks, which also have stochastically smaller demands. This is opposite to the priority given by shortest demand first (SDF) scheduling.

Majumdar et al. [43] and Leutenegger and Vernon [41] give simulation data to show that under specific distributions of $N$ and $H_2$ job demands $\overline{R}_{RRP}$ is insensitive to $C_D$ at moderate loads and that $\overline{R}_{FCFS}$ increases with $C_D$. Their results show that

$$\overline{R}_{RRP} < \overline{R}_{FCFS}, \qquad \begin{array}{l}\text{correlated and uncorrelated workloads, } D \sim H_2 \text{ with} \\ C_D \text{ typically} \geq 2,\ T_i = D/N \text{ or } U_iD/\sum_{j=1}^{N} U_j,\ U_i \sim U(0,1].\end{array}$$

RRP performs badly for correlated workloads with low to moderate $C_D$ because it allocates priority in opposite order of shortest demand first. The same is true for dynamic AVG as shown in [40], where allocation is proportional to average parallelism. Under correlated workloads, policies that give higher priority to jobs with few processes are expected to perform well since they approximate shortest demand first. Thus Leutenegger and Nelson [40] show the PSNTF policy to be optimal under their workload model that assumes i.i.d. exponential task service times.

For i.i.d. exponential task service times Nelson and Towsley [53] show that

$$\overline{R}_{SNQTF} < \overline{R}_{TP} \approx \overline{R}_{JP} < \overline{R}_{NP} < \overline{R}_{FCFS}, \quad \text{3 classes, } T_i \text{ i.i.d. exp.,}$$

where TP, JP, and NP are fixed-priority policies with task preemption, job preemption, and no preemption.

Another study that proposes a "smallest parallelism first" policy to have high performance is the study of Majumdar et al. [43]. They show that for correlated as well as uncorrelated workloads

$$\overline{R}_{PSNPF} < \overline{R}_{RRP}, \overline{R}_{FCFS} \qquad \begin{array}{l}\text{correlated and uncorrelated workloads, } D \sim H_2, \\ T_i = U_iD/\sum_{j=1}^{N} U_j,\ U_i \sim U(0,1].\end{array}$$

Leutenegger and Vernon [41] give data for specific distributions of demand and parallelism to show that

$$\begin{array}{ll}\overline{R}_{PSNPF} < \overline{R}_{RRP}, & \text{uncorrelated workload with } C_D < 2, \text{ correlated workld with } C_D < 3\text{-}4; \\ \overline{R}_{PSNPF} > \overline{R}_{RRP}, & \text{uncorrelated workload with } C_D > 2, \text{ correlated workld with } C_D > 3\text{-}4.\end{array}$$

Agrawal et al. [1] show PSAPF to be optimal for a workload with the linear ERF and exponential demands independent of other workload variables. They also give counterexamples to show that PSAPF is not optimal under $C_D > 1$ or sublinear ERFs.

We now review results for equipartitioning policies. Tucker and Gupta [82] use measurements to show that

$$\overline{R}_{EQ_{PC}} < \overline{R}_{RRP}, \qquad \text{4 programs with different ERFs.}$$

They argue that EQ$_{PC}$ performs better because it controls the number of runnable processes per job by limiting it to number of processors allocated.

Leutenegger and Vernon [41] show that for specific distributions for demand and parallelism

$$\overline{R}_{RRJ} < \overline{R}_{PSNPF}, \overline{R}_{RRP}, \qquad \text{correlated and uncorrelated workloads, } D \sim H_2,\, T_i = D/N.$$

McCann et al. [47] give measurement data to show that the spatial EQ$_{IP}$ performs better than EQT since there are fewer preemptions and jobs execute at better operating points of their speedup curves.

To illustrate the high performance potential of EQ, Agrawal et al. [1] give experimental data to show that under linear job execution rates, and a specific $H_2$ distribution for demand with $C_D = 5$, and specific workloads for available parallelism $N$, $\overline{R}_{EQ}$ is within twice of the best achievable performance for the given workload conditions.

Leutenegger and Vernon [41] show RRJ to outperform PSNPF for specific demand and parallelism workloads with and without correlation. On the other hand the study of Leutenegger and Nelson [40] shows that RRJ performs worse than the optimal PSNPF policy. Thus there exists disparity in the literature regarding the performance of EQ[5] and PSNPF and one of the goals of this thesis is to clarify this disparity by using a more general workload model and delineating the regions of the design space where each of these policies has best performance.

When job demand characteristics are known PSCDF is good policy though it is not optimal like its uniprocessor counterpart. Majumdar et al. [43] and Leutenegger and Vernon [41] both show PSCDF to outperform all other policies they consider, which include PSNPF and RRJ.

To summarize, we note that the sensitivity of scheduling policy performance with respect to workload parameters has not been fully studied primarily because of the specific nature of the workload assumptions in previous studies and also because the solution techniques in these studies do not reveal the functional dependence of policy performance on workload parameters.

This completes our survey of policy comparison results. Table 2.5 summarizes the policy comparison results. The results column shows qualitative policy behavior whereas the comparisons column provides relative policy comparisons.

## 2.6 Motivation for this Thesis

We have reviewed performance evaluation studies of parallel processor policies by examining system models, workload models, four types of parallel processor policies, performance evaluation techniques, and policy comparison results. We now examine what needs to be done to obtain a better understanding of parallel processor scheduling.

[5] RRJ is a particular EQ policy.

Table 2.5: Policy Comparison Results for Parallel Processor Scheduling

| Policy | Results | Comparisons |
|---|---|---|
| **FP/RTC** | | |
| FP($\frac{P}{n}$) | FP(1) is optimal for $C_D = 0$, linear ERFs and fully parallel jobs. For sublinear ERFs, FP(1) is optimal at $\rho \rightarrow 0$ and FP(P) is optimal at $\rho \rightarrow 1$. | |
| **AP/RTC** | | |
| pws policies | | $\overline{R}_{PWS-FCFS} < \overline{R}_{PWS-FCFS-LA}$<br>$< \overline{R}_{PWS} < \overline{R}_{PWS-LA}$ |
| ASP | Optimal at extreme ends of $\rho$. | $\overline{R}_{ASP} < \overline{R}_{PWS-FCFS}$ |
| Avg based policies | For $D \sim$ exp, using application characteristics in addition to avg improves system performance. | $\overline{R}_{AVG+\&mM} < \overline{R}_{AVG+} < \overline{R}_{AVG}$ |
| AP/DA | | |
| UnequalDP | | $\overline{R}_{UnequalDP} < \overline{R}_{METB} < \overline{R}_{RR-slot}$. |
| RRP | $\overline{R}_{RRP}$ degrades with workload correlation, but is quite insenstive to $C_D$. | $\overline{R}_{RRP} < \overline{R}_{Coscheduling}$ except for high lock contention workloads.<br>$\overline{R}_{RRP} < \overline{R}_{FCFS}$ for uncorrelated workloads with low to high $C_D$.<br>$\overline{R}_{RRP} > \overline{R}_{FCFS}$ for correlated workloads with moderate $C_D$. |
| PSNPF | Optimal for $T_i$ iid exp and no arrivals. | $\overline{R}_{PSNPF} < \overline{R}_{FCFS}$<br>$\overline{R}_{PSNPF} < \overline{R}_{RRP}$ at low to moderate $C_D$. |
| PSAPF | Optimal for $D \sim$ exp and linear ERFs. | |
| RRJ | $\overline{R}_{RRJ}$ is quite insensitive to $C_D$, under $H_2$ demand distributions and specific parallelism workloads. | $\overline{R}_{RRJ} < \overline{R}_{RRP}$.<br>$\overline{R}_{RRJ} < \overline{R}_{PSNPF}$ in [41],<br>$\overline{R}_{RRJ} > \overline{R}_{PSNPF}$ in [40]. |
| PSCDF | Optimal for fully parallel workload with linear ERFs. Not optimal in general. | $\overline{R}_{PSCDF} < \overline{R}_{RRJ}$.<br>$\overline{R}_{PSCDF} < \overline{R}_{PSNPF}$. |

Most studies of parallel processor scheduling have assumed small system sizes with less than 20 processors, which is very small compared to real systems that can have hundreds of processors. Many studies have also made specific assumptions about the workload and have not clarified the implications of the assumptions for processor scheduling results. Some studies have assumed exponential demands or task service times, some studies have assumed no job arrivals, and some studies have assumed i.i.d. task service times which leads to an implicit correlation between demand and parallelism. Previous results under specific distributions of processing demand and parallelism show that policy performance can be sensitive to coefficient of variation in demand, $C_D$, as well as correlation between demand and parallelism and this calls for more general models in future studies that include nonexponential demands and a range of correlation between demand and parallelism.

In terms of performance evaluation techniques we believe that analytic modeling has the highest promise for a thorough understanding of policy performance. Measurement gives exact estimates that include system overheads but is limited to a specific mix of programs and specific system characteristics. Simulation allows a broader range of workloads but is not as general as analysis and does not give any direct insight into the dependence of policy performance on key parameters. Also, simulation is very time consuming for large systems. Furthermore, specific distributions of demand and parallelism are needed in simulation studies and it is unknown whether important portions of the parameter space are being ignored by doing so. In contrast, analytic modeling allows for general distributions of demand and parallelism and general ERFs and if closed form expressions are available for mean response time then the dependence of policy performance on workload parameters is readily observable.

Exact analysis for parallel processor systems has so far not been as successful as exact analysis for uniprocessor systems since space enumeration techniques, such as matrix-geometric analysis, have typically been required. Moreover, previous exact estimates of mean response time in parallel scheduling have rarely take the form of closed form expressions except over narrow ranges of system parameters. This indicates that approximate analysis is likely to be a future trend in performance evaluation of parallel systems both from the viewpoint of obtaining closed form expressions and from the viewpoint of scalability to systems with hundreds or thousands of processors.

From the literature reviewed in this chapter we have seen that under specific workload assumptions ASP has been shown to have high performance among AP/RTC policies, EQS has been shown to have high performance among AP/DA policies, and PSNPF has been shown to have high performance for highly correlated workloads. One study [80] also points out that a simple policy such as FCFS may have high performance under specific workloads. Policy comparisons in previous studies have been limited to specific workload assumptions and it is not clear how ASP, EQS, PSNPF and FCFS perform over a broad range of workload parameters that includes arbitrary $C_D$, arbitrary correlation, as well as general job execution rates.

In the remainder of this thesis we shall develop approximate analytic models for ASP, EQS, PSAPF, and FCFS under a general workload model for systems with hundreds of processors. The approximate analysis and the workload model are chosen such that the functional dependence of policy performance on workload parameters is readily apparent and thus we can obtain key parameters that influence qualitative policy behavior. The key parameters help us explore the design space in a more systematic way than done in the past.

# Chapter 3

# System and Workload Model

In this chapter we address our first goal, that is, to design a workload model that captures the essential features of parallel applications and is easy to parameterize. The goal is to have a simple system and workload model that is broadly applicable, characterizes the essential features of parallel workloads with respect to scheduling disciplines, contains a small number of parameters, and is easy to analyze. With this end in mind we shall use the centralized queueing system model and the ERF workload model that were reviewed in Chapter 2. The ERF model implicitly captures job synchronization and communication, allows independent control of demand, parallelism, and parallel program overhead, and is easier to analyze than the task graph model.

To achieve broad applicability for policy comparison results, we make little or no restrictions on the distribution of important system parameters, such as job parallelism and total service demand. Since variation in demand has been shown to be a critical factor in the performance of uniprocessor scheduling policies [33] and there is also experimental data that shows this to be the case for parallel processor policies [43, 41] we place no restrictions on job demand distribution. Since parallel programs are still in the early stages of design it is unknown what is a typical distribution for available job parallelism. We therefore assume a general distribution for available parallelism. There is experimental evidence in the literature to show that correlation between demand and parallelism can affect the relative performance of some scheduling policies (cf. [43, 41]). We therefore include a control parameter to vary the amount of correlation in our workload model. Finally, practical programs have synchronization and communication overheads and these overheads can be different for different workloads. We therefore assume a general nondecreasing execution rate function (ERF) for the workload and to permit ease of analysis we assume that all jobs use the same ERF.

We give details and further justification of our system and basic workload model in Sections 3.1 and 3.2. More details of our correlation model are provided in Section 3.3. The notation used in this thesis and example workload settings are given in Section 3.4 and finally in Section 3.5 we provide constraints on workload parameters that delineate the design space. These constraints will be used to study the qualitative behavior of scheduling policies and to obtain stress tests for validating mean response time approximations.

## 3.1 System Model

We consider an open system model with $P$ identical processors and a central job queue. The centralized queueing model is only conceptual; practical implementations of the policies considered will in general allow for distributed queue access. Jobs arrive at the system according to a Poisson process with rate $\lambda$ as shown in Figure 3.1. We assume zero job 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 define our workload model.

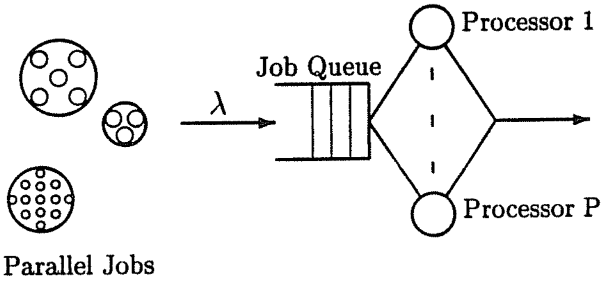


Figure 3.1: Open System Model

## 3.2 Basic Workload Model

We assume that all jobs are statistically identical. Every 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 $E : [0, P] \rightarrow [0, P]$, 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, $t$, the $P$ processors are allocated to jobs present in the queue according to the processor allocation policy, $\Psi$. If $a(t)$ processors (possibly fractional) 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 assume the following about $N$ and $E$.

- $N$ has a general (bounded) distribution, $\mathcal{F}_N$, with mean[1] $\overline{N}$, coefficient of variation[2] $C_N$, and probability mass function $\underline{p} = (p_1, \ldots, p_P)$, where $p_k = \Pr[N = k]$, $k = 1, \ldots, P$.

- $E$ is derived from a *deterministic* function $\gamma$, 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 $\gamma$ as the execution rate function (ERF) of the workload. The ERF $\gamma$ is said to be *linear* if $\gamma(x) = x$, for all $0 \leq x \leq P$.

  For all jobs with available parallelism $N$, $E(N) = \gamma(N)$. When fewer than $N$ processors are allocated to the job, the execution rate $E$ depends on more detailed characteristics of the applications. In this thesis it is assumed 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 had available parallelism equal to the processor allocation, i.e., $E(j) = \gamma(j)$, for $1 \leq j < N$. This could be appropriate for applications based on the work queue model, or in some cases where the processes of a job are timeshared on the allocated processors. 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 $E(x)$.

Note that other assumptions about job execution rate on fewer than $N$ processors are possible. For example, one might assume that the parallelism overhead is about the same on fewer processors as on $N$ processors, i.e., $E(j) = \frac{j}{N}\gamma(N)$ for $1 \leq j < N$, which could represent a system with

[1] We interchangeably use mean and expectation throughout this thesis.

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

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. As another example, if communication overheads are fixed for a given available parallelism but the load is only balanced when $j$ evenly divides $N$, then $E(j) = \frac{1}{\lceil N/j \rceil}\gamma(N)$, for $1 \leq j < N$.

The service time of a job on $N$ processors is given by the random variable $S = D/\gamma(N)$, and we denote its mean by $\overline{S}$.

The workload model defined above contains three simplifications 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 execution rate function $\gamma$ for all jobs.

The first assumption is realistic for RTC processor allocation policies. 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.

## 3.3 Correlation Model

It is unknown whether or how job demand is correlated with parallelism in real workloads. The most general way to model correlation is to specify an arbitrary joint distribution of $D$ and $N$, $F(D,N)$, but this approach can complicate both the analysis and exploration of the design space. A simpler model that still permits a wide range of correlation, can be obtained by assuming that for a job with available parallelism $N$, its *mean demand* is either independent of $N$ with probability $q$ or is linearly correlated with $N$ with probability $1-q$. Varying $q$ from 0 to 1 thus allows us to control the workload correlation in the model. Below we define the parameters of the correlation model more precisely.

The mean demand of the job with available parallelism $N$ is given by

$$\Delta_N = \begin{cases} A, & \text{with probability } q, \\ cN, & \text{with probability } 1-q. \end{cases}$$

In the first case the demand is drawn from a general distribution, $\mathcal{F}^u_D$, with mean $A$ and coefficient of variation $C_v$, where $A$ and $C_v$ are fixed constants independent of $N$. In the second case, the demand is stochastically equal to a demand that is drawn from the same distribution and then scaled by the factor $\frac{cN}{A}$. In the latter case the mean demand is $cN$ as required, and the coefficient of variation of is equal to $C_v$, which does not depend on $N$.

Let $r$ denote the *correlation coefficient* of $\Delta_N$ and $N$. That is,

$$r \equiv \frac{E[\Delta_N N] \; - \; E[\Delta_N]\,E[N]}{\sigma_{\Delta_N}\sigma_N}, \quad \sigma_{\Delta_N}, \sigma_N \neq 0 \tag{3.1}$$

Define $r$ to be 0 when $\sigma_{\Delta_N} = 0$ or $\sigma_N = 0$. The following lemma shows how $A$ and $c$ are related to $\overline{D}$ (i.e., the mean demand of the workload across all jobs) and $\overline{N}$, and how $q$ is related to $r$. This lemma shows that the workload correlation is specified by the single parameter $r$.

**Lemma 3.3.1** *For the correlation model given by (3.1),*

$$A = \overline{D}, \quad c = \overline{D}/\overline{N}, \quad \textit{and} \quad q = 1 - r^2.$$

**Proof.** By definition of $\Delta_N$,

$$\Delta_N = \begin{cases} A, & \text{with probability } q, \\ cN, & \text{with probability } 1-q. \end{cases}$$

Thus, $E[\Delta_N] = \overline{D} = qA + (1-q)c\overline{N}$, for all $0 \leq q \leq 1$. Setting $q = 1$ yields $A = \overline{D}$, and setting $q = 0$ yields $c = \overline{D}/\overline{N}$.

To prove that $q = 1 - r^2$, we first note that either $\sigma_{\Delta_N} = 0$ or $\sigma_N = 0$ implies that $\Delta_N = \overline{D}$ with probability 1. Thus $q = 1 - r^2$ for these cases. For $\sigma_{\Delta_N} > 0$ and $\sigma_N > 0$, we evaluate the

RHS of equation (3.1). First note that

$$E[\Delta_N\, N] = q\, A\, \overline{N} \;+\; (1-q)\, c\, E[N^2].$$

Using this and $E[\Delta_N] = \overline{D}$ and further simplifying we find,

$$E[\Delta_N\, N] \;-\; E[\Delta_N]\, \overline{N} \;=\; (1-q)\, \overline{D}\, \overline{N}\, C_N^2. \tag{3.2}$$

Also,

$$\begin{aligned} E[\Delta_N^2] &= q\, A^2 \;+\; (1-q)\, c^2\, E[N^2], \\ \sigma_{\Delta_N}^2 &= E[\Delta_N^2] \;-\; E[\Delta_N]^2 = (1-q)\, \overline{D}^2\, C_N^2. \end{aligned}$$

Substituting $\sigma_{\Delta_N} = \sqrt{1-q}\,\overline{D}\, C_N$ and the RHS of (3.2) in (3.1), we have

$$r = \frac{(1-q)\, \overline{D}\, \overline{N}\, C_N^2}{\sqrt{1-q}\, \overline{D}\, C_N\, \sigma_N} = \sqrt{1-q},$$

which results in $q = 1 - r^2$ as required. ■

A consequence of this lemma is that $r = 0$ implies that $q = 1$ and thus that $D$ and $N$ are independent.

## 3.4 Notation

Table 3.1 summarizes the notation for system parameters and variables. Under the implicit assumption of Poisson arrivals with rate $\lambda$, and dynamic redistribution of work whenever a job is allocated fewer processors than its available parallelism, i.e., $E(j) = \gamma(j)$, we use the following notation to characterize the system and workload model.

$$(\Psi,\ \mathcal{F}_N,\ \mathcal{F}_D^u,\ r,\ \gamma),$$

$\Psi$ = processor allocation policy

$\mathcal{F}_N$ = distribution of $N$, e.g., $N = P$, Uniform(1,P)

$\mathcal{F}_D^u$ = distribution of demand for jobs with mean demand independent of parallelism e.g., $\exp(\mu)$

$r$ = correlation coefficient as defined by (3.1)

$\gamma$ = execution rate function. By default we assume that $\gamma$ is a general nondecreasing ERF. We use the notation $\gamma \in \mathcal{E}^c$, to specify that $\gamma$ belongs to the class of concave and nondecreasing ERFs, $\mathcal{E}^c$. To specify the linear ERF we use the notation $\gamma^l$.

Table 3.1: System Notation

| | |
|---|---|
| $P$ | Number of processors in the system |
| $\lambda$ | Arrival rate of jobs |
| $D$ | Total job demand |
| $\mathcal{F}_D^u$ | Distribution of demand for "uncorrelated" jobs |
| $\overline{D}$ | Marginal mean job demand |
| $C_D$ | Coefficient of variation of marginal demand |
| $\rho$ | Offered load $\lambda\overline{D}/P$ |
| $N$ | Available job parallelism |
| $\mathcal{F}_N$ | Distribution of available parallelism |
| $p_k$ | Probability$[N = k]$, $k = 1, \ldots, P$ |
| $\underline{p}$ | $(p_1, p_2, \ldots, p_P)$ |
| $\overline{N}$ | Average available parallelism |
| $C_N$ | Coefficient of variation of available parallelism |
| $r$ | Correlation coefficient (as defined by (3.1)) |
| $\gamma$ | General execution rate function of the workload |
| $\mathcal{E}^c$ | Class of concave and nondecreasing ERFs |
| $\gamma^l$ | Linear execution rate function |
| $\overline{S}$ | Mean job service time |
| $S_n$ | Normalized mean service time $\overline{S}/\overline{D}$ |
| $\overline{R}_\Psi$ | Mean response time of policy $\Psi$ |
| $\hat{R}_\Psi^x$ | Estimator for mean response time under $\Psi$ (obtained using an interpolation approximation on parameter $x$) |
| $M/G/1_P$ | An $M/G/1$ system with a processor of power $P$ |

To indicate a general distribution of demand or available parallelism, general workload ERF, or arbitrary value of $r$ between 0 and 1, we simply leave the notation as $\mathcal{F}_D^u$, $\mathcal{F}_N$, $\gamma$, or $r$, respectively.

For experimental results in this chapter, we will make use the following bounded-geometric distribution for available job parallelism (similar to the distribution in [41, 39]):

**Definition 3.4.1 (Bounded-geometric($P_{\max}$, $p$))** *A random variable $N$ has a bounded-geometric distribution with parameters $P_{\max}$ and $p$ if*

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

In some experiments, three specific bounded-geometric distributions for N will be examined. These distributions are given in Table 3.2. More details of these workloads are given in Chapter 4. Another distribution for $N$ that will be used is the following two-point pmf:

**Definition 3.4.2 ($K_2(a, b, \alpha)$)** *$N$ has a $K_2(a, b, \alpha)$ distribution if*

$$N = \begin{cases} a, & \textit{with probability } \alpha, \\ b, & \textit{with probability } 1 - \alpha. \end{cases} \qquad 0 \leq \alpha \leq 1$$

We use the following two types of ERFs in our experiments:

- $\gamma(k) = k^c$, $k = 1, 2, \ldots$, $0 \leq c \leq 1$,
- $\gamma(k) = (1 + \beta)k/(k + \beta)$, $k = 1, 2, \ldots$, $0 \leq \beta \leq \infty$.

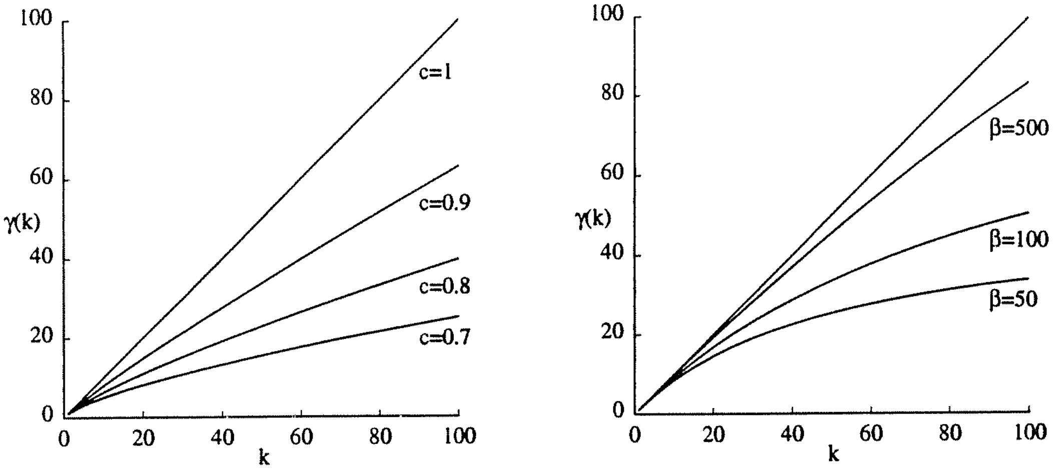
Both ERFs are concave and nondecreasing as shown in Figure 3.2. The second ERF is derived from a type of execution signature in [18].

Table 3.2: Three Bounded-geometric Distributions for $N$

| Symbol | Parallelism | $P_{\max}$ | $p$ | P=20 | | 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 |

## 3.5 Constraints on Workload Parameters

The workload model defined above is not only general but is also easy to parameterize. Important generalizations in the workload model include the general distribution of available parallelism, general distribution of job demand for jobs with no correlation, general nondecreasing



(a) $\gamma(k) = k^c$

(b) $\gamma(k) = (1+\beta)k/(k+\beta)$

Figure 3.2: Two types of ERFs

deterministic execution rate function, and controlled correlation between demand and parallelism. Varying workload parameters, such as $C_D$ and $r$, allows us to explore the design space more thoroughly than in the past. Nearly all previous performance studies of parallel processor scheduling policies have assumed specific distributions for demand and/or parallelism. Furthermore, we are not aware of any study that has considered a range of correlation between demand and parallelism. (Some previous studies have considered specific extremes of our correlation model such as $r = 0$ and $r = 1$, cf. [43, 41, 92]. In i.i.d. task service time models there is implicitly a high correlation between demand and parallelism and there is no opportunity to vary demand and parallelism parameters independently.)

Workload parameters of immediate interest to us are mean and coefficient of variation in demand, i.e., $\overline{D}$ and $C_D$, mean and coefficient of variation of available parallelism, i.e., $\overline{N}$ and $C_N$, correlation coefficient $r$, execution rate function $\gamma$, and mean service time $\overline{S}$. These parameters must satisfy certain relationships which constrain the system design space. Without loss of generality, we let $\overline{D}$, $\overline{N}$, $r$, and $\gamma$ be the free parameters in the model, where $0 \le \overline{D} < \infty$, $1 \le \overline{N} \le P$, $0 \le r \le 1$, and $0 \le \gamma(x) \le x$, $\gamma(x^-) \le \gamma(x)$ and $\gamma(x) = x$ for $0 \le x \le 1$, and then consider how these parameters constrain the other parameters of interest, i.e., $C_N$ (Section 3.5.1), $C_D$ (Section 3.5.1), and $\overline{S}$ (Section 3.5.3). The constraints delineate the model parameter space and will be useful in evaluating the qualitative behavior of scheduling policies as well as for identifying stress tests for validating mean response time approximations.

Several bounds in this section are derived using simple properties of convex functions. A convex function is defined as follows (cf. [61]).

**Definition 3.5.1 (Convex Function)** *A function $f : (a, b) \to I\!R$ is called* convex *if*

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha)f(y), \quad \textit{for all } x, y \in (a,b) \textit{ and } \alpha \in (0,1).$$

We shall also make use of **Jensen's inequality** (see [26]) which states that

> If $h : I\!R \to I\!R$ is convex and $X$ is a random variable with a finite mean, then
>
> $$E[h(X)] \ge h(E[X]).$$

### 3.5.1 Constraints on $C_N$

Since $N$ is bounded above by $P$, it follows that $C_N$ cannot be unbounded. For a fixed $\overline{N}$ we first derive achievable bounds on $C_N$ for general distributions of $N$ in Theorem 3.5.1. We then focus on bounded-geometric distributions of $N$ and in Theorem 3.5.2 we derive distributions in this class that have minimum and maximum $C_N$.

**Theorem 3.5.1 (Bounds on $C_N$)** *Given $\overline{N}$, for a general distribution of $N$ we have*

$$0 \;\le\; C_N \;\le\; \sqrt{\frac{\overline{N}(P+1)-P}{\overline{N}^2} - 1}.$$

*The lower bound is attained when $N$ is constant and integer-valued for all jobs. The upper bound is attained for the $K_2(1, P, \cdot)$ distribution.*

**Proof.** The lower bound is trivial. The derivation of the upper bound is as follows. Since $C_N = \sigma_N/\overline{N}$ we need to derive an upper bound for $\sigma_N$. By definition,

$$\sigma_N^2 = E[N^2] - \overline{N}^2$$

$$E[N^2] \;=\; \sum_{i=1}^{P} p_k k^2 \;=\; \sum_{i=1}^{P} p_k f(k), \tag{3.3}$$

where $f(x) = x^2$. We derive an upper bound on $E[N^2]$ by observing that $f$ is a convex function, that is,

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha)f(y), \qquad 0 \le \alpha \le 1.$$

Choosing $\alpha$ such that $\alpha \cdot 1 + (1-\alpha)P = k$, that is, $\alpha = (P-k)/(k-1)$, we get the following bound for $f(k)$,

$$f(k) \;=\; f(\alpha \cdot 1 + (1-\alpha)P) \;\le\; \alpha f(1) + (1-\alpha)f(P) = \frac{P-k}{P-1} \cdot 1 + \frac{k-1}{P-1}P^2.$$

Using this bound in (3.3) it follows that,

$$\begin{aligned}E[N^2] &\leq \sum_{k=1}^{P} p_k \left(\frac{P-k}{P-1}\cdot 1 + \frac{k-1}{P-1}P^2\right)\\ &= \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1}P^2\\ &= \overline{N}(P+1) - P.\end{aligned}$$

Hence,

$$C_N^2 = \frac{E[N^2] - \overline{N}^2}{\overline{N}^2} \leq \frac{\overline{N}(P+1) - P}{\overline{N}^2} - 1,$$

which yields the upper bound of proposition 3.5.1. This upper bound is attained when $N$ has a two point pmf with mass only at 1 and $P$. ∎

We now derive bounded-geometric distributions with minimum and maximum $C_N$.

**Theorem 3.5.2 (Bounded-Geometric distributions with minimum and maximum $C_N$)** *Let $N$ have a bounded-geometric distribution with parameters $P_{\max}$ and $p$. For a given $\overline{N}$, $C_N$ is maximum when $p = 1$ and $C_N$ is minimum when $P_{\max} = 0$.*

**Proof.** When $p = 1$, N can take one of two values, either 1 or P. As shown in the proof of Theorem 3.5.1, for a given value of $\overline{N}$, $C_N$ is highest (over all distributions of N with mean $\overline{N}$) if

$$N = \begin{cases} 1, & \text{with probability } p_1,\\ P, & \text{with probability } 1 - p_1,\end{cases}$$

which is the bounded-geometric distribution with $p = 1$ and $P_{\max} = 1 - p_1$. Hence it trivially follows than over all bounded-geometric distributions that have the same $\overline{N}$, $C_N$ is maximum when $p = 1$.

The proof for the second result that over all bounded-geometric distributions with the same $\overline{N}$, $C_N$ is minimum when $P_{\max} = 0$ is rather long and has detailed algebraic manipulation. We therefore provide it in the Appendix A. The intuitive reason for this result is that the pmf of N is more evenly "spread" out when $P_{\max} = 0$. ∎

### 3.5.2 Constraints on $C_D$

In Section 3.3 we saw that a job with available parallelism $k$ could belong to one of two types in terms of its demand, either "uncorrelated" with mean $A = \overline{D}$ and coefficient of variation $C_v$, or "correlated" with mean $c\,k$ and coefficient of variation $C_v$, where $c = \overline{D}/\overline{N}$. The probability that the job is uncorrelated is $q = 1 - r^2$ and the probability that it is correlated is $1 - q = r^2$. Thus,

$$\begin{aligned} E[D^2|N=k] &= q \times A^2(1+C_v^2) \; + \; (1-q) \times c^2 k^2 (1+C_v^2) \\ &= (1-r^2) \times \overline{D}^2 (1+C_v^2) \; + \; r^2 \times \frac{\overline{D}^2}{\overline{N}^2} k^2 (1+C_v^2) \\ &= (1+C_v^2)\overline{D}^2 \left(1 - r^2 + r^2 \frac{k^2}{\overline{N}^2}\right). \end{aligned}$$

Unconditioning on $N = k$, we get

$$\begin{aligned} E[D^2] &= (1+C_v^2)\overline{D}^2 \sum_{k=1}^{P} \left(1 - r^2 + r^2 \frac{k^2}{\overline{N}^2}\right) p_k \\ &= (1+C_v^2)\overline{D}^2 \left(1 - r^2 + r^2 \frac{\overline{N^2}}{\overline{N}^2}\right) \\ &= (1+C_v^2)\overline{D}^2 \{1 - r^2 + r^2(1+C_N^2)\} \\ &= (1+C_v^2)\overline{D}^2 (1 + r^2 C_N^2). \end{aligned}$$

As a result,

$$C_D^2 = \frac{E[D^2]}{\overline{D}^2} - 1 \; = \; (1+C_v^2)(1+r^2C_N^2) - 1, \tag{3.4}$$

which shows how $C_D$ is related to $C_v$, $r$, and $C_N$. We see that $C_D$ increases with each of $C_v$, $r$, and $C_N$. When there is no correlation ($r = 0$), $C_D$ is simply $C_v$ in which case $C_N$ has no impact on $C_D$. At $r = 0$ any value of $C_D$ is possible between 0 and $\infty$ since we have no constraints on $C_v$. However, when $r$ increases so does the influence of $C_N$ on $C_D$. This causes $C_D$ to be lower bounded by $r^2 C_N^2$ and thus for correlated workloads in our model not all marginals distributions are allowed for $D$. For example, if $r > 0$ and $C_N > 0$ it is not possible for $D$ to be deterministic, since $C_D > 0$. However, for uncorrelated workloads the distribution of $D$ can be general including deterministic.

### 3.5.3 Constraints on $\overline{S}$

We first derive bounds for $\overline{S}$ when $r = 0$ and then derive bounds for $\overline{S}$ when $r = 1$. Using the bounds we examine the qualitative behavior of $\overline{S}$ with workload correlation. The following expressions for mean job service time, $\overline{S}$, will be useful in deriving bounds for $\overline{S}$ under $r = 0$ and $r = 1$ and in developing interpolation approximations for correlated workloads in Chapters 5 and 6.

$$\overline{S} = E\left[\frac{D}{\gamma(N)}\right] \;\; = \;\; \sum_{k=1}^{P} p_k \left\{ q \frac{A}{\gamma(k)} \; + \; (1-q) \frac{c\,k}{\gamma(k)} \right\}$$

$$= \quad q\overline{D}E\left[\frac{1}{\gamma(N)}\right] \;+\; (1-q)\frac{\overline{D}}{\overline{N}}E\left[\frac{N}{\gamma(N)}\right] \tag{3.5}$$

$$= \quad (1-r^2)\,\overline{S}(r=0) \;+\; r^2\overline{S}(r=1), \tag{3.6}$$

where $\overline{S}(r=0)$ denotes the mean job service time (on N processors) when $r = 0$ and $\overline{S}(r=1)$ denotes the mean job service time when $r = 1$.

#### 3.5.3.1 Results for uncorrelated workloads

When $r = 0$, we have $\overline{S} = \overline{D}E[1/\gamma(N)]$. Assuming $\gamma$ to be concave (which is typical for ERFs) we derive bounds on $E[1/\gamma(N)]$ and use them to understand the behavior of $\overline{S}$ as a function of $N$. We use the following definition for concave functions (cf. [61]).

**Definition 3.5.2 (Concave Function)** *A function $f : (a,b) \to I\!R$ is called* concave *if*

$$f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha)f(y), \quad \textit{for all } x, y \in (a,b) \textit{ and } \alpha \in (0,1).$$

**Lemma 3.5.1** *Given a concave ERF $\gamma$, we have for a given value of $\overline{N}$ that*

$$\frac{1}{\overline{N}} \leq \frac{1}{\gamma(\overline{N})} \;\leq\; E\left[\frac{1}{\gamma(N)}\right] \;\leq\; \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1}\cdot\frac{1}{\gamma(P)}.$$

**Proof.** Since $\gamma(x) \leq x$, for $0 < x \leq P$, we have $1/x \leq 1/\gamma(x)$ and thus $1/\overline{N} \leq 1/\gamma(\overline{N})$. Since $\gamma$ is concave, $1/\gamma$ is convex [61]. Therefore by Jensen's inequality

$$\frac{1}{\gamma(\overline{N})} \leq E\left[\frac{1}{\gamma(N)}\right].$$

For integer $\overline{N}$ this bound is attained when $N$ is constant for all jobs, that is, $N \equiv \overline{N}$.
To derive the upper bound on $E[1/\gamma(N)]$ we proceed as follows.

$$E[1/\gamma(N)] = \sum_{k=1}^{P} p_k \frac{1}{\gamma(k)} = \sum_{k=1}^{P} p_k g(k), \tag{3.7}$$

where $g(x) = 1/\gamma(x)$. Since $g$ is convex we choose $\alpha = (P-k)/(k-1)$ so that

$$g(k) = g(\alpha\cdot 1 + (1-\alpha)P) \leq \alpha g(1) + (1-\alpha)g(P) = \frac{P-k}{P-1}\cdot 1 + \frac{k-1}{P-1}\cdot\frac{1}{\gamma(P)}.$$

Using this upper bound for $g(k)$ in (3.7) we get,

$$E[1/\gamma(N)] \leq \sum_{k=1}^{P} p_k\left(\frac{P-k}{P-1} + \frac{k-1}{P-1}\cdot\frac{1}{\gamma(P)}\right) = \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1}\cdot\frac{1}{\gamma(P)}.$$

This upper bound is attained when $N$ can take only one of two values, either 1 or $P$. ■

The bounds on $E[1/\gamma(N)]$ from Lemma 3.5.1 yield the following bounds on $\overline{S}$.

**Theorem 3.5.3 (Bounds on $\overline{S}$ when $r = 0$)** *For a concave ERF $\gamma$, we have for a given value of $\overline{N}$ that when $r = 0$*

$$\frac{\overline{D}}{\overline{N}} \le \frac{\overline{D}}{\gamma(\overline{N})} \le \overline{S} \le \overline{D}\left(\frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1} \cdot \frac{1}{\gamma(P)}\right).$$

For the linear ERF the bounds on $\overline{S}$ reduce to the following form:

**Corollary 3.5.1** *For linear ERFs, we have for a given value of $\overline{N}$ that when $r = 0$*

$$\frac{\overline{D}}{\overline{N}} \le \overline{S} \le \overline{D}\left(1 - \frac{\overline{N}-1}{P}\right).$$

The implications of Theorem 3.5.3 are that at $r = 0$, $\overline{S}$ is minimum when $C_N = 0$ and $\overline{S}$ is maximum when $C_N$ is maximum. (Note from Theorem 3.5.1 that for a given $\overline{N}$, $C_N$ is maximum when $N$ is either 1 or $P$ which is also the case when $\overline{S}$ is maximum.) This result seems to contradict a result by Nelson in [50] which shows that for a workload with i.i.d. exponential task service times, $\overline{S}$ is minimum when $C_N$ is maximum. However, the i.i.d. task service time model is a fully correlated workload, whereas Theorem 3.5.3 was derived for uncorrelated workloads. We will see below that for fully correlated workloads we can get the same result as Nelson.

#### 3.5.3.2 Results for fully correlated workloads

Now consider the case where $r = 1$, that is, we have a fully correlated workload. We note from (3.5) that at $r = 1$, we have

$$\overline{S} = \frac{\overline{D}}{\overline{N}} E\left[\frac{N}{\gamma(N)}\right].$$

Thus when $r = 1$, to derive bounds on $\overline{S}$ we need to derive bounds on $N/\gamma(N)$. $N/\gamma(N)$ is often concave for many ERFs, e.g., for the ERF $\gamma(k) = k^c$ where $0 \le c \le 1$, and for the ERF $\gamma(k) = (1+\beta)k/(k+\beta)$ where $0 \le \beta \le \infty$. For concave $N/\gamma(N)$ we have the following bounds on $E[N/\gamma(N)]$.

**Lemma 3.5.2** *If $N/\gamma(N)$ is concave then for a given $\overline{N}$*

$$1 \le \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1} \cdot \frac{P}{\gamma(P)} \le E\left[\frac{N}{\gamma(N)}\right] \le \frac{\overline{N}}{\gamma(\overline{N})}.$$

**Proof.** We first prove the upper bound on $E[N/\gamma(N)]$. Since $N/\gamma(N)$ is concave $-N/\gamma(N)$ is convex and it follows by Jensen's inequality that

$$-\frac{\overline{N}}{\gamma(\overline{N})} \le -E\left[\frac{N}{\gamma(N)}\right],$$

which leads to the upper bound of the lemma. For integer $\overline{N}$ the upper bound is attained when $N$ is constant for all jobs, that is, $N \equiv \overline{N}$.

To derive the lower bounds on $E[N/\gamma(N)]$ we proceed as follows.

$$E[N/\gamma(N)] = \sum_{k=1}^{P} p_k \frac{k}{\gamma(k)} = \sum_{k=1}^{P} p_k g(k), \tag{3.8}$$

where $g(x) = x/\gamma(x)$. Since $g$ is concave we choose $\alpha = (P-k)/(k-1)$ so that

$$g(k) = g(\alpha \cdot 1 + (1-\alpha)P) \geq \alpha g(1) + (1-\alpha) g(P) = \frac{P-k}{P-1} \cdot 1 + \frac{k-1}{P-1} \cdot \frac{1}{\gamma(P)}.$$

Using this lower bound for $g(k)$ in (3.8) we get,

$$E[N/\gamma(N)] \geq \sum_{k=1}^{P} p_k \left( \frac{P-k}{P-1} + \frac{k-1}{P-1} \cdot \frac{1}{\gamma(P)} \right) = \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1} \cdot \frac{P}{\gamma(P)}.$$

This lower bound is attained when $N$ can take one of two values, either 1 or $P$. Finally, since $\gamma(P) \leq P$, we have

$$\frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1} \cdot \frac{P}{\gamma(P)} \geq 1.$$

■

Lemma 3.5.2 readily yields the following bounds on $\overline{S}$ when $r = 1$.

**Theorem 3.5.4 (Bounds on $\overline{S}$ when $r = 1$)** *If $N/\gamma(N)$ is concave then we have for a given $\overline{N}$ that when $r = 1$*

$$\frac{\overline{D}}{\overline{N}} \leq \frac{\overline{D}}{\overline{N}} \left( \frac{P-\overline{N}}{P-1} + \frac{\overline{N}-1}{P-1} \cdot \frac{P}{\gamma(P)} \right) \leq \overline{S} \leq \frac{\overline{D}}{\gamma(\overline{N})}.$$

From Theorem 3.5.4 we have the following corollary, where we use $\gamma^l$ to denote a linear ERF.

**Corollary 3.5.2** *If $\gamma$ is a concave ERF such that $N/\gamma(N)$ is concave, then for a given $\overline{N}$*

$$\overline{S}(r=1,\gamma^l) = \frac{\overline{D}}{\overline{N}} \leq \frac{\overline{D}}{\overline{N}} E\left[\frac{N}{\gamma(N)}\right] = \overline{S}(r=1,\gamma) \leq \overline{D} E\left[\frac{1}{\gamma(N)}\right] = \overline{S}(r=0,\gamma),$$

*where $\gamma^l$ is the linear ERF.*

**Proof.** The lower bounds for $\overline{S}(r=1,\gamma)$ follow directly from Theorem 3.5.4. For the upper bound on $\overline{S}(r=1,\gamma)$ note that from Theorem 3.5.4 $\overline{S}(r=1,\gamma) \leq \overline{D}/\gamma(\overline{N})$ which from Theorem 3.5.3 is less than or equal to $\overline{S}(r=0,\gamma)$. ■

The implications of Theorem 3.5.4 are that if $N/\gamma(N)$ is concave, then for fully correlated workloads $\overline{S}$ is maximum when $C_N = 0$ and $\overline{S}$ is minimum when $C_N$ is maximum. ($C_N$ is maximum when $N$ can take one of two values either 1 or $P$.) This concurs with Nelson's result for i.i.d. exponential task service times. We also note from Corollary 3.5.2 that for a concave ERF $\gamma$ such that $N/\gamma(N)$ is concave, $\overline{S}(r = 1) \leq \overline{S}(r = 0)$. That is, if $\overline{D}$ and $\overline{N}$ remain unchanged mean service time decreases with workload correlation. For example, for the linear ERF we have $N/\gamma^l(N) = 1$ which is trivially concave and thus $\overline{S}(r = 1) = \overline{D}/\overline{N} \leq \overline{D}E[1/N] = \overline{S}(r = 0)$. For this example when $r = 1$ a job with parallelism $k$ has mean demand $\overline{D}/\overline{N} \times k$ and thus mean service time $\overline{D}/\overline{N}$ which is the same for all $k$. On the other hand when $r = 0$ a job with low parallelism has higher mean service time than a job with high parallelism and this increase in variance causes the overall mean service time to increase as compared to the case for $r = 1$.

#### 3.5.3.3 Summary

To summarize our results we note that for a concave ERF $\gamma$, for uncorrelated workloads $\overline{S}$ is minimum when $C_N$ is minimum and $\overline{S}$ is maximum when $C_N$ is maximum. For a concave ERF $\gamma$ such that $N/\gamma(N)$ is concave, we have that for fully correlated workloads $\overline{S}$ is maximum when $C_N$ is minimum and $\overline{S}$ is minimum when $C_N$ is maximum. For the latter conditions on $\gamma$, $\overline{S}$ decreases with workload correlation $r$. (We only proved $\overline{S}(r = 1) \leq \overline{S}(r = 0)$, but using (3.6) in addition to this bound shows that that $\overline{S}$ decreases with $r$.)

# Chapter 4

# The Interpolation Approximation Approach

In this chapter we pursue our second goal, which is, to develop alternative models of parallel processor scheduling disciplines. That is, how to develop models for scheduling policies that are broadly applicable, yield insight into policy performance, and are easy to evaluate for large systems. Analytic models have the potential to satisfy these desirable features. However, previous analytic models of parallel processor scheduling disciplines typically involve numerical solutions of sets of simultaneous equations that grow superlinearly in the number of processors and/or yields no direct insight into the functional relationship between performance measures and particular workload parameters. Furthermore, all but one previous models either assume exponential distributions of job service time (per class if needed) or assume i.i.d. task service times implying a specific degree of correlation between total job demand and the number of tasks in a job. These assumptions potentially limit the applicability of the results, and also prohibit the analysis of how scheduling policy performance varies with workload parameters that characterize service time distributions and/or correlation.

In this chapter, a new approach is proposed for the modeling of parallel scheduling policies, that of *interpolation approximations*. We reduce parallel processor systems under specific values of system parameters to single server or multiple server queues with known solutions. Using these *reductions* mean job response time formulas are derived that readily yield insight into policy behavior, and are easy to evaluate for large systems (say hundreds of processors or more). To illustrate the interpolation approximation approach we use the linear ERF ($\gamma^l$) and assume no correlation between demand and parallelism. We, however, assume a general distribution of demand, $D$, and a general distribution of available parallelism, $N$. no correlation between $D$

and $N$. As per the notation introduced in Chapter 3 our assumptions are $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$. The assumptions of linear ERFs and independence between $D$ and $N$ make it easier to present our approach and we relax these assumptions for specific policies in the succeeding chapters.

In this chapter example interpolation approximations are developed for the mean response times of the EQ, FCFS, and PSAPF policies. (Since execution rates are assumed to be linear, spatial EQS and temporal EQT have the same performance and we generically refer to them as EQ.) These approximations result in simple closed form expressions that show how policy performance depends on coefficient of variation in demand, $C_D$, and on job parallelism parameters. The approximations are shown to be accurate for most of the parameter space by means of validations against discrete event simulation and special cases of exact analysis. Section 4.1 of this chapter discusses the interpolation approximation approach and surveys literature that has used interpolation approximations for single server, multiserver, and fork-join queues. Section 4.2 reduces EQ and FCFS under special cases of parameter values to single server or multiserver queues with known solutions. Section 4.3 presents interpolation approximations for the mean response times under each of EQ and FCFS based on the results derived in Section 4.2, and Section 4.4 does the same for PSAPF. Validations for the interpolation approximations are provided in Section 4.5 and finally the approximation approach is summarized in Section 4.6.

## 4.1 Background

In this section we outline the interpolation approximation approach and summarize such approximations that have appeared in previous literature.

The underlying principle behind interpolation approximations is simple: *use the known to predict the unknown.* The first step is to derive exact results under extreme values of system parameters, for example, light and heavy traffic limits of mean response time, or mean response times under fully sequential and fully parallel workloads. The next step is to form a function that interpolates among 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, $\rho$, 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 [15] 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 $C_x \leq 1$. The parameter of interpolation is $C_x^2$.

(The approximation can be used as an extrapolation for $C_x > 1$.) Burman and Smith [9] 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. In [10] 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 [22] interpolates between light and heavy traffic limits of the moments of the waiting time distribution in an M/G/1 Round Robin queue. Simon and Willie [73] estimate response time characteristics in priority queueing networks using interpolation approximations based on simulation and heavy traffic limits. Reiman and Simon [59], and Reiman et al. [60] 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 [23] derive interpolation approximations for response time distributions in several single server queues, based on a similar approach. Whitt [87], Fendick and Whitt [21] 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 [83] propose interpolation approximations (on system utilization) for the mean response times of a symmetric fork-join queue (FCFS scheduling in each 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.

## 4.2 Reductions to Known Queueing Systems: Examples for EQ and FCFS

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.

### 4.2.1 Queueing Systems with Known Solutions

Consider an open multiserver queue with sequential work as shown in Figure 4.1. We consider the special case of an M/G/c queue in which jobs arrive according to a Poisson process with rate

$\lambda$, and have i.i.d. service times with mean $\overline{x}$ and coefficient of variation $C_x$. Server utilization is given by $\rho = \lambda\overline{x}/c$, where $c$ is the number of servers. We assume that the scheduling discipline is FCFS unless otherwise stated.

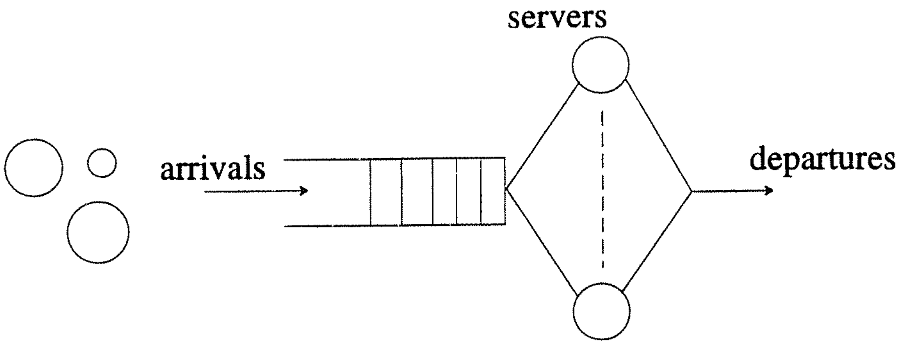


Figure 4.1: Multiserver queue with sequential work

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 [64, 77, 79, 88, 89]. Of particular interest to us is the simple approximation proposed in [64] for the mean number in a GI/G/c queue, which leads to the following approximate formula for $\overline{R}_{M/G/c}$:

$$\overline{R}_{M/G/c} \approx \overline{x} + \frac{\rho^{\sqrt{2(c+1)}}(1 + C_x^2)}{2\lambda(1-\rho)}. \tag{4.1}$$

Note that this approximation is exact for $c = 1$ and $c = \infty$. Using this approximation and the fact that $\overline{R}_{M/G/c\ PS} = \overline{R}_{M/M/c}$ [65], one can derive the following approximation:

$$\overline{R}_{M/G/c\ PS} \approx \overline{x} + \frac{\rho^{\sqrt{2(c+1)}}}{\lambda(1-\rho)}. \tag{4.2}$$

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 [64] for the M/M/c queue. We use the approximate expressions given by (4.1) and (4.2) for the reductions in section 4.2.2.

### 4.2.2 Reductions for EQ and FCFS

We consider two cases of reductions. First, when available parallelism, $N$, is constant across all jobs, for both FCFS and EQ. Second, light and heavy traffic limits, where the heavy traffic limit is derived for EQ and for a restricted case for FCFS.

#### 4.2.2.1 Constant Available Parallelism ($N = k$)

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

**Proposition 4.2.1** *For the workload assumptions* $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$*:*

$$\overline{R}_{EQ}(N = k) = \overline{R}_{M/G/c\ PS}, \qquad c = \frac{P}{k}, \quad P \bmod k = 0. \tag{4.3}$$

$$\overline{R}_{FCFS}(N = k) = \overline{R}_{M/G/c}, \qquad c = \frac{P}{k}, \quad P \bmod k = 0. \tag{4.4}$$

*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.** First consider the proof for the EQ reduction. Let $\Gamma = (EQ, N = k, \mathcal{F}_D^u, r = 0, \gamma^l)$, where $P \bmod k = 0$. When there are $Q \leq c$ jobs in $\Gamma$, each job receives $k$ amount of processing power. When there are $Q > c$ jobs in $\Gamma$, each job receives $P/Q$ amount of processing power. This is how a processor sharing (PS) discipline allocates processing power to jobs, when there are $c$ servers, each with a processing power of $k$. (Note this reduction is valid only for the linear ERF.)

Now consider the proof for the FCFS reduction. Let $\Gamma = (FCFS, N = k, \mathcal{F}_D^u, r = 0, \gamma^l)$, where $P \bmod k = 0$. System $\Gamma$ operates as follows. A job that arrives when system $\Gamma$ 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. Since processors are allocated and released in units of $k$, the system $\Gamma$ behaves like an $M/G/c$ system with $c = P/k$ processors, in which each job has one task with service requirement $x = D/k$. (Note this reduction also holds for nonlinear ERFs.) ∎

From Proposition 4.2.1 we estimate the mean response time of $EQ$ when $N = k$ and $P$ mod $k = 0$, by using expression (4.2) with $\overline{x} = \overline{D}/k$, and the mean response time of $FCFS$ by using expression (4.1) with $\overline{x} = \overline{D}/k$ and $C_x = C_D$. Thus, we have,

$$\overline{R}_{EQ}(N = k) \approx \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(c+1)}}}{\lambda(1-\rho)}, \quad c = \frac{P}{k}, \quad P \bmod k = 0,$$

$$\overline{R}_{FCFS}(N = k) \approx \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(c+1)}}(1 + C_D^2)}{2\lambda(1-\rho)}, \quad c = \frac{P}{k}, \quad P \bmod k = 0,$$

where $\rho = \lambda\overline{D}/P$. These expressions can be evaluated even when $c$ is not an integer. Therefore, we use the same approximations with $c = P/k$ even when $P \bmod k \neq 0$, to get

$$\overline{R}_{EQ}(N=k) \approx \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(\frac{P}{k}+1)}}}{\lambda(1-\rho)}, \quad k = 1, 2, \ldots, P. \tag{4.5}$$

$$\overline{R}_{FCFS}(N=k) \approx \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(\frac{P}{k}+1)}}(1+C_D^2)}{2\lambda(1-\rho)}, \quad k = 1, 2, \ldots, P. \tag{4.6}$$

Note that both these approximations are exact when $N = P$ since approximations (4.2) and (4.1) are exact for $c = 1$.

*An important observation from approximations* (4.5) *and* (4.6) 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 $\overline{D}$.

**4.2.2.2 Light and Heavy Traffic Limits ($\rho = 0, \rho = 1$)**

At light traffic, that is, $\rho \to 0$, the mean response time under each of EQ and FCFS is simply the mean job service time on $N$ processors, $\overline{S}$. Due to the assumption of linear execution rates, the service time of a job with available parallelism of $N$ is $D/N$. Since $D$ and $N$ are independent, the mean job service time is given by

$$\lim_{\rho\to 0} \overline{R}_{EQ} = \lim_{\rho\to 0} \overline{R}_{FCFS} = \overline{S} = \overline{D}E[1/N]. \tag{4.7}$$

We present an informal derivation of the mean response time under heavy traffic for the EQ policy. A more rigorous derivation is given in Chapter 5. At heavy traffic, an arriving job finds more than $P$ jobs in the system with probability 1 as $\rho \to 1$. The EQ policy allocates an equal fraction of processing power to all jobs if there are more than $P$ jobs in the system. Hence the processing power allocated to each job in the system is less than 1 when $\rho \to 1$, in which case only the total job demand matters for mean response time and not the available parallelism. In particular, when $\rho \to 1$ the mean response time in the system for any distribution of $N$ reduces to the mean response time when $N = P$. By Proposition 4.2.1, $\overline{R}_{EQ}(N = P) = \overline{R}_{M/G/1_P\ PS} = (\overline{D}/P)/(1-\rho)$, which follows from setting $c = 1$ and $\overline{x} = \overline{D}/P$ in (4.2). Thus, we obtain the following heavy traffic limit[1]:

$$\lim_{\rho\to 1} (1-\rho)\overline{R}_{EQ} = \frac{\overline{D}}{P}. \tag{4.8}$$

We do not have a corresponding 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 (4.6):

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

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

which we note does not depend on $k$.

#### 4.2.2.3 Summary of Results for EQ and FCFS

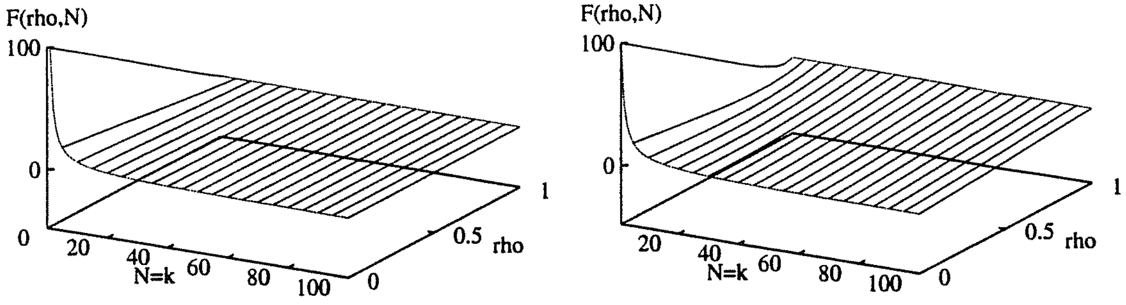
To summarize the results of the reductions derived thus far, Figures 4.2a and 4.2b plot the *normalized* mean response time[2], $F(\rho, k) = (1-\rho)\overline{R}_{\Psi}(\rho, N = k)$, $\Psi \in \{EQ, FCFS\}$, where $k = 1, 2, \ldots, P$ denotes the fixed value of parallelism assumed for all jobs. The curves are plotted for $P = 100$, and mean job demand $\overline{D} = P = 100$. Figure 4.2a 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 4.2b contains the curves for the FCFS policy when $C_D = 5$.

Several points are worth noting about the results in Figure 4.2. 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 4.2a.

In Figure 4.3a and 4.3b, we have 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 4.2a. 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 4.3b is 13 times that in Figure 4.3a 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 $\rho$ (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 4.2.1.

In the next section we will interpolate between the response time and extra time values obtained for particular points in the system parameter space. The plots in figures 4.2 and 4.3 will aid in determining how the interpolations should be constructed.

[2] 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.

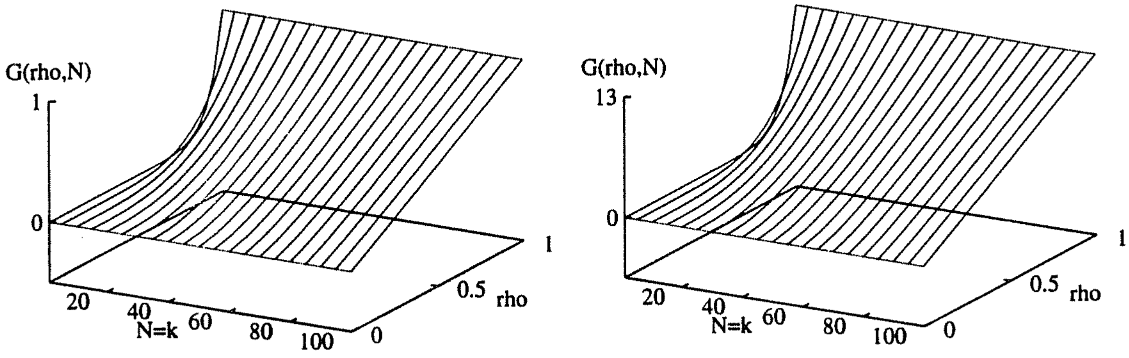


(a) EQ, $C_D \geq 0$
FCFS, $C_D = 1$

(b) FCFS, $C_D = 5$

Figure 4.2: Normalized Mean Response Time

$\overline{D} = P = 100$



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

(b) FCFS, $C_D = 5$

Figure 4.3: Normalized Mean Extra Time

$\overline{D} = P = 100$

## 4.3 Example Interpolation Approximations for EQ and 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 $\rho$ 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 (4.8).

### 4.3.1 Interpolation on $\rho$: EQ

Let $F(\rho) = (1-\rho)\overline{R}_{EQ}(\rho)$. The light and heavy traffic limits, $F(0)$ and $F(1)$, are given in equations (4.7) and (4.8) of Section 4.2.2. Figures 4.2(a) and 4.3(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,

$$\begin{aligned}\overline{R}_{EQ} \approx \hat{R}^{\rho}_{EQ} &= \frac{\hat{F}(\rho)}{1-\rho} \\ &= \overline{S} + \frac{\rho}{1-\rho}\frac{\overline{D}}{P} \\ &= \overline{D}E\left[\frac{1}{N}\right] + \frac{\rho}{1-\rho}\frac{\overline{D}}{P}. \qquad (4.9)\end{aligned}$$

This approximation is exact for the special case when $N = P$, which is easily seen by comparing equations (4.5) and (4.9) when $N = P$.

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

The next interpolation is applicable to both policies and uses the results derived in Section 4.2.2 for extreme values of available parallelism ($N = 1$ and $N = P$), where $\overline{N} = 1$ and $\overline{N} = P$, respectively. Figures 4.2 and 4.3 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 $\Psi$ 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 $\overline{X}_\Psi$,

$$\hat{X}^{\overline{N}}_{\Psi} = \left(\frac{P-\overline{N}}{P-1}\right)\overline{X}_\Psi(\overline{N}=1) + \left(\frac{\overline{N}-1}{P-1}\right)\overline{X}_\Psi(\overline{N}=P), \tag{4.10}$$

where $\overline{X}_\Psi(\overline{N}=1)$ and $\overline{X}_\Psi(\overline{N}=P)$ are derived from equations (4.5) and (4.6), 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 (4.10), the full interpolation approximations are:

$$\overline{R}_{EQ} \approx \hat{R}^{\overline{N}}_{EQ} = \overline{D}E[1/N] + \left(\frac{P-\overline{N}}{P-1}\right)\frac{\rho^{\sqrt{2(P+1)}}}{\lambda(1-\rho)} + \left(\frac{\overline{N}-1}{P-1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}. \tag{4.11}$$

$$\overline{R}_{FCFS} \approx \hat{R}^{\overline{N}}_{FCFS} = \overline{D}E[1/N] + \left\{\left(\frac{P-\overline{N}}{P-1}\right)\frac{\rho^{\sqrt{2(P+1)}}}{\lambda(1-\rho)} + \left(\frac{\overline{N}-1}{P-1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}\right\}\left(\frac{1+C_D^2}{2}\right). \tag{4.12}$$

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

### 4.3.3 Interpolation on the pmf cf $N$: EQ and FCFS

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 4.2.2. 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, $\underline{p} = \underline{e}_k$, $1 \le k \le P$, 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 (4.5) for $\overline{R}_{EQ}(N = k)$ (Section 4.2.2) we get

$$\hat{R}^{\underline{p}}_{EQ} = \sum_{k=1}^{P} p_k \left\{ \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(\frac{P}{k}+1)}}}{\lambda(1-\rho)} \right\} = \overline{D}E[1/N] + \frac{E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}}\right]}{\lambda(1-\rho)}. \tag{4.13}$$

Similarly, from approximation (4.6) for $\overline{R}_{FCFS}(N = k)$ (Section 4.2.2) we get

$$\hat{R}^{\underline{p}}_{FCFS} = \sum_{k=1}^{P} p_k \left\{ \frac{\overline{D}}{k} + \frac{\rho^{\sqrt{2(\frac{P}{k}+1)}}}{\lambda(1-\rho)} \frac{(1+C_D^2)}{2} \right\} = \overline{D}E[1/N] + \frac{E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}}\right]}{\lambda(1-\rho)} \left(\frac{1+C_D^2}{2}\right). \tag{4.14}$$

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

As in the interpolations on $\rho$ 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 $\rho$ and $\overline{N}$. Second, from Figure 4.2 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 $\rho$ and $\overline{N}$.

## 4.4 Example Interpolation Approximations 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 $\overline{R}_{PSAPF}$.

### 4.4.1 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 4.2.2. Thus,

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

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 (4.1), the reduction for $\overline{R}_{PSAPF}(N=k)$ is thus as in (4.6), i.e.,

$$\overline{R}_{PSAPF}(N=k) \approx \frac{\overline{D}}{k}+\frac{\rho^{\sqrt{2(\frac{P}{k}+1)}}(1+C_D^2)}{2\lambda(1-\rho)}, \quad k=1,2,\ldots,P. \tag{4.15}$$

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 4.2 and 4.3.

### 4.4.2 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:

$$\begin{aligned}
\hat{R}^{\overline{N}}_{PSAPF} &\approx \overline{S}+\left(\frac{P-\overline{N}}{P-1}\right)\overline{X}_{PSAPF}(\overline{N}=1)+\left(\frac{\overline{N}-1}{P-1}\right)\overline{X}_{PSAPF}(\overline{N}=P) && (4.16)\\
&= \overline{D}E[1/N]+\left\{\left(\frac{P-\overline{N}}{P-1}\right)\frac{\rho^{\sqrt{2(P+1)}}}{\lambda(1-\rho)}+\left(\frac{\overline{N}-1}{P-1}\right)\frac{\rho}{1-\rho}\frac{\overline{D}}{P}\right\}\left(\frac{1+C_D^2}{2}\right) && (4.17)
\end{aligned}$$

and

$$\begin{aligned}
\hat{R}^{\underline{p}}_{PSAPF} &\approx \sum_{k=1}^{P} p_k\overline{R}_{PSAPF}(N=k) && (4.18)\\
&\approx \overline{D}E[1/N]+\frac{E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}}\right]}{\lambda(1-\rho)}\left(\frac{1+C_D^2}{2}\right). && (4.19)
\end{aligned}$$

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 [41] 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.

## 4.5 Model Validations

We validate the analytic interpolation approximations for the mean response time under EQ, FCFS, and PSAPF 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.

### 4.5.1 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) $\mathcal{F}_D^u$: 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) $\rho$: 0.1 to 0.9. (Since $\overline{D} = P$, $\rho = \lambda$.)

(iv) $\mathcal{F}_N$: bounded-geometric, constant, and uniform. In the validations we ensured coverage of extreme values of $C_N$ and $\overline{N}$ which served as stress tests.

Table 4.1 and 4.2 list the parameter settings for all distributions of $N$ considered in the validations. In Table 4.1 the parameter settings for the bounded geometric distributions are arranged in three groups of three, and within each group in decreasing $\overline{N}$. As shown by Theorem 3.5.2, 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 $p = 1$. 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 4.2 than in Table 4.1 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 4.1: Validation Workloads for $N$: P=20,100

| **Distribution** | **Parameter Settings** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Bounded- | $P_{\max}$ | 0 | 0 | 0 | .1 | .1 | .1 | .9 | .5 | .1 |
| Geometric | $p$ | .005 | 1/(.5P) | 1/(.1P) | .01 | 1/(.4P) | .9 | 1 | 1 | 1 |
| Constant | N=1, N=P/4, N=P/2, N=3P/4, N=P | | | | | | | | | |
| Uniform | (P/2,P), (1,P), (1,P/2) | | | | | | | | | |

Table 4.2: Validation Workloads for $N$: P=500,1000

| **Distribution** | **Parameter Settings** | | | |
|---|---|---|---|---|
| Bounded- | $P_{max}$ | .9 | .1 | .1 |
| Geometric | $p$ | 1 | 1/(.4P) | .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* [30] (see Theorem 5.1.1 in Chapter 5). Exact estimates for $\overline{R}_{FCFS}(N = k)$ or $\overline{R}_{PSAPF}(N = k)$ for $H_2$ demand distributions were obtained using matrix-geometric analysis [56, 51, 75]. For the estimates obtained by simulation almost all had 95% confidence intervals with less than 5% half-widths [37]. 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.

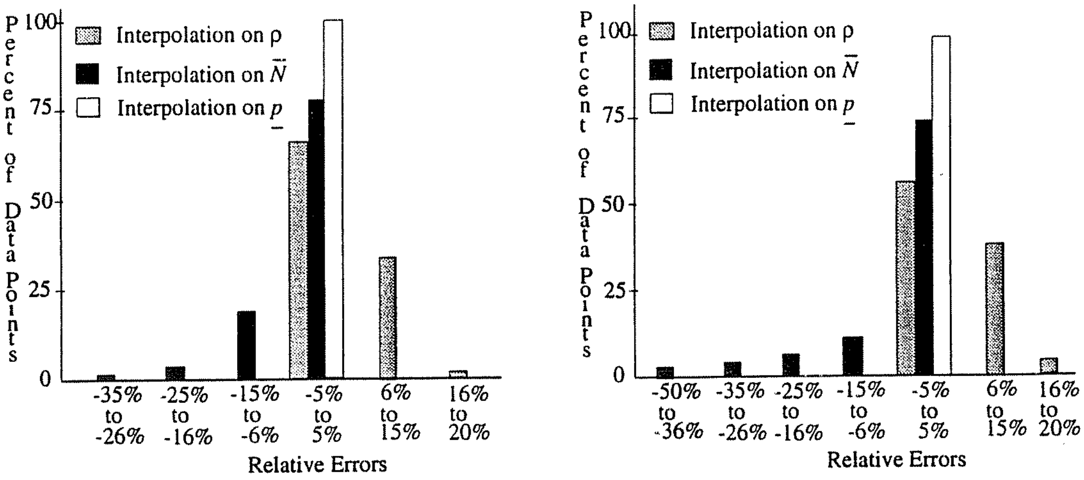
### 4.5.2 Summary of Validations

Figures 4.4, 4.5, and 4.6 present histograms that summarize all of the validation experiments for the EQ, FCFS, and PSAPF approximations discussed in this chapter. 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 and for PSAPF at each value of $C_D = 1$ and $C_D = 5$, thus leading to a total of 956 validations for each of FCFS and PSAPF.[3]

First, consider the EQ histograms in Figure 4.4. 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 $\rho$ tends to overestimate $\overline{R}_{EQ}$. These trends can be predicted from the plots in Figure 4.3a. 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 4.3a, noting that the error at higher $\rho$ will be magnified when the normalized mean extra time is divided by $1 - \rho$. The worst case errors for the interpolation on $\rho$ were for $(N = P/4,\ \rho = 0.7)$, which is also consistent with the data in Figure 4.3a, 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 4.5. 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 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 (4.6) 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

[3] Many simulation experiments were run on the Condor distributed system [6].



(a) P=20,100

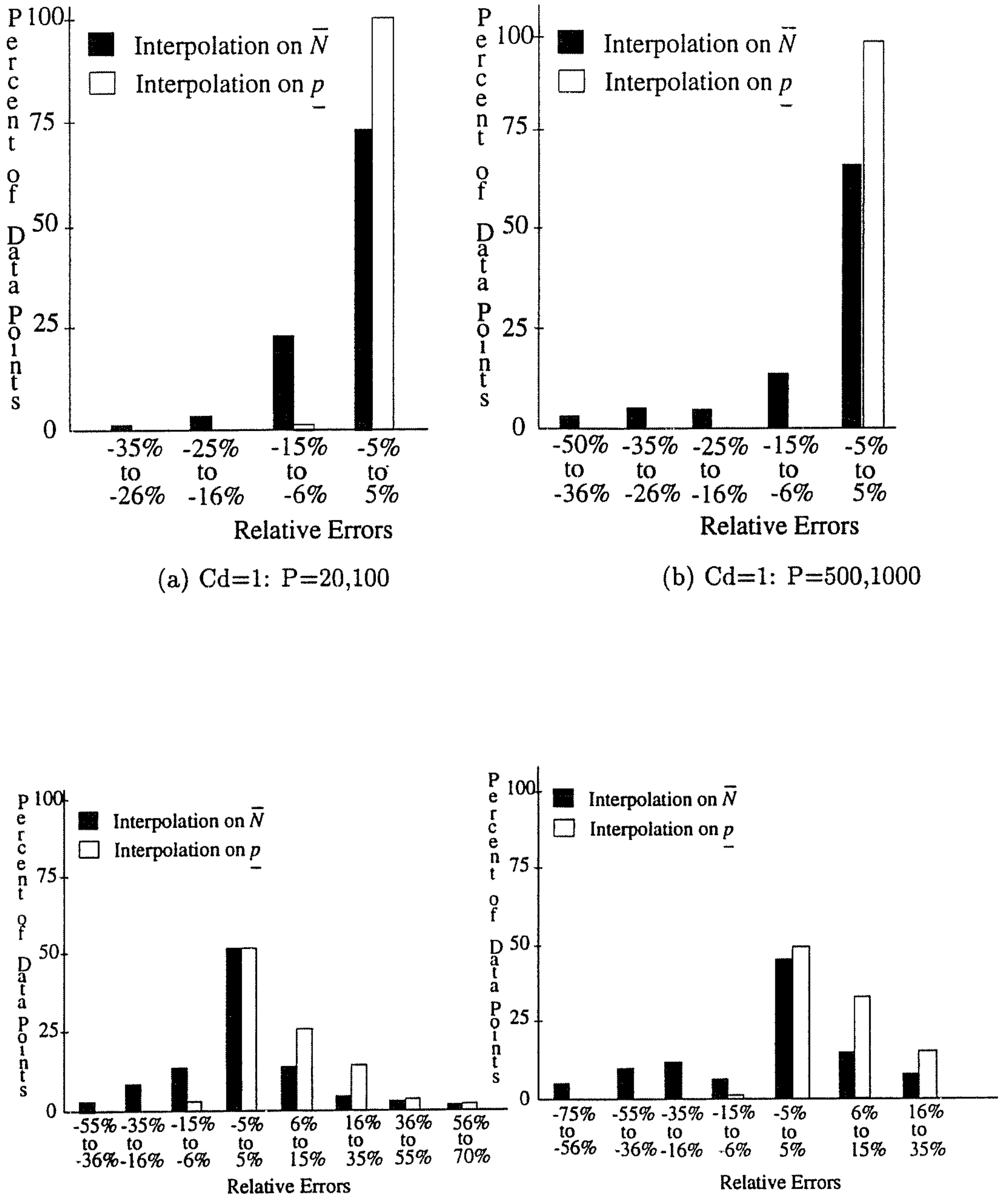
(b) P=500,1000

Figure 4.4: Summary of Validations: EQ

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.

Figure 4.6 presents histograms that summarize the validations of the PSAPF approximations. From Figure 4.6 we observe that the relative errors in the PSAPF approximations are *very similar to* those for FCFS in Figure 4.5. 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 $C_N$.

One source of the error for $\hat{R}_{PSAPF}$ at high values of $C_D$ is that we used approximate estimates at the end points $N = k$ as given by (4.15) 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



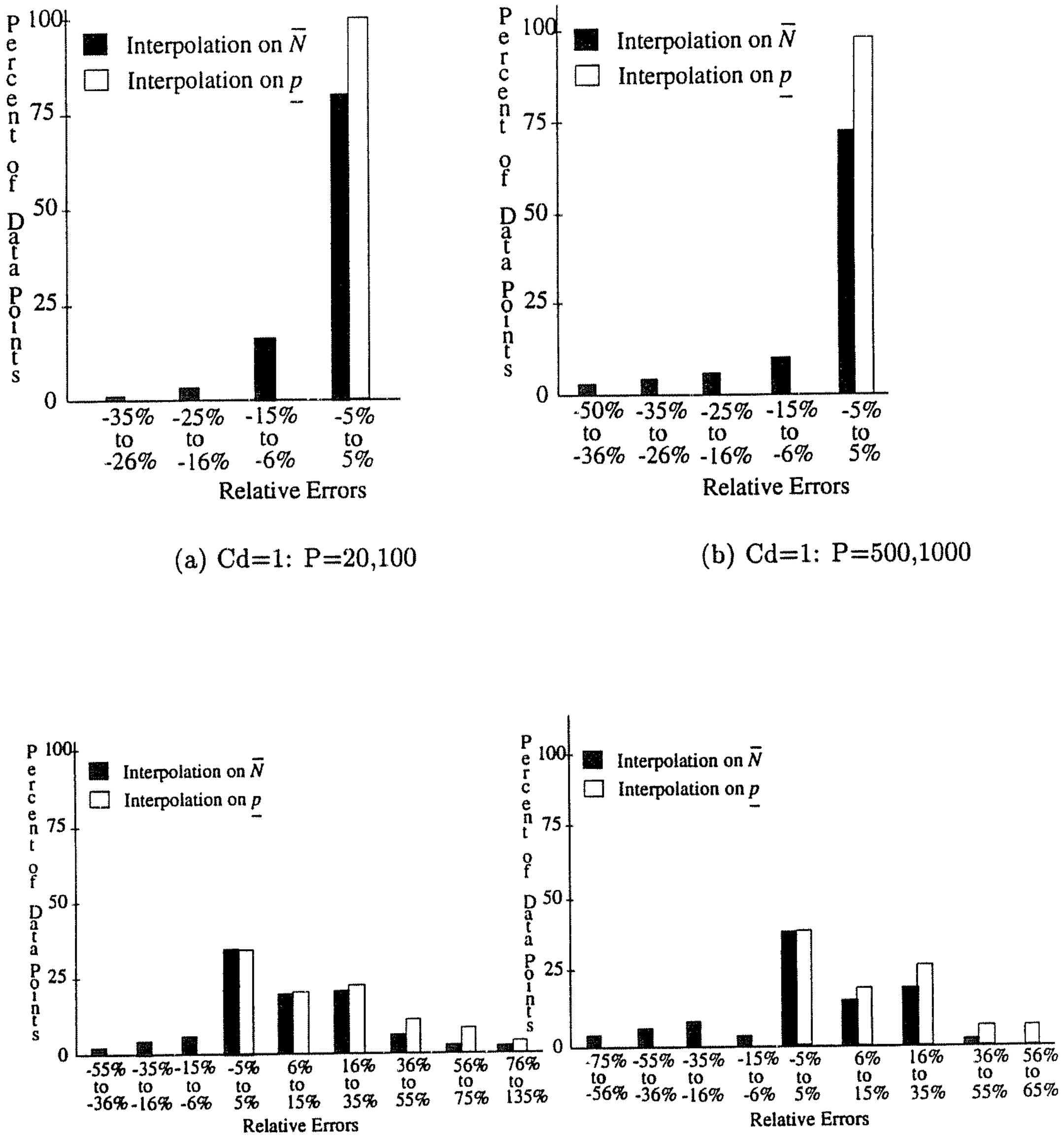
(a) Cd=1: P=20,100

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

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

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

Figure 4.5: Summary of Validations: FCFS



(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 4.6: Summary of Validations: PSAPF

of matrix-geometric analysis and then used the same interpolation methods as (4.16) and (4.18). 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 policy insights and comparisons. In Chapter 6 we improve on the interpolation approximations for PSAPF by modifying the interpolation on $\underline{p}$ to account for the priority given to jobs with smaller parallelism.

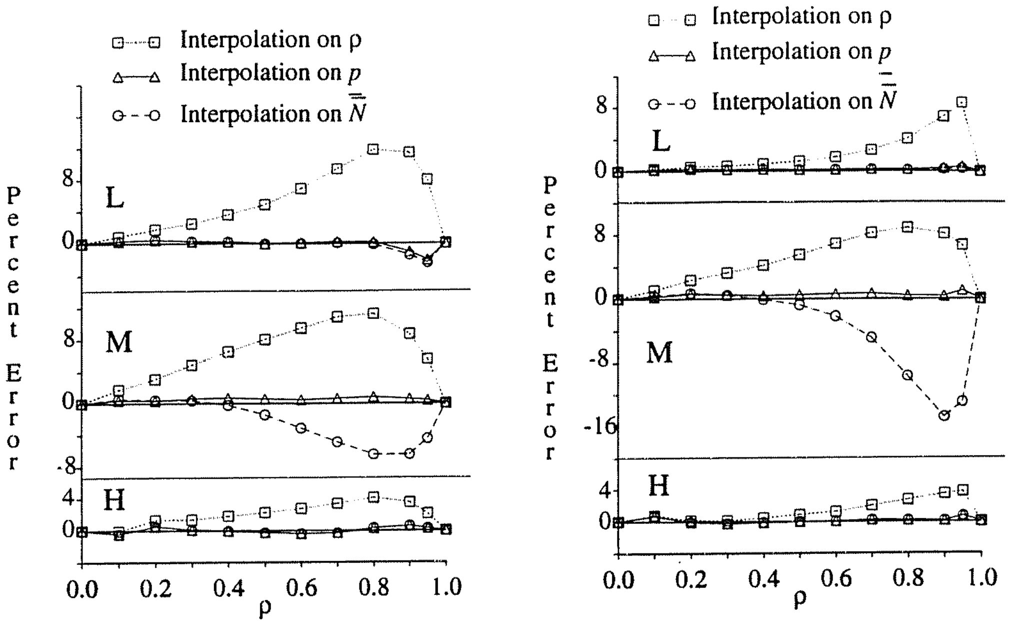
### 4.5.3 Example Validation Experiments

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 FCFS and PSAPF, specific values of $C_D$. The distributions of $N$ considered are bounded-geometric with parameter settings given in Table 3.2. 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.

In Figures 4.7a and 4.7b 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 $\rho$ 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 4.8 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 $\rho > 0.5$.

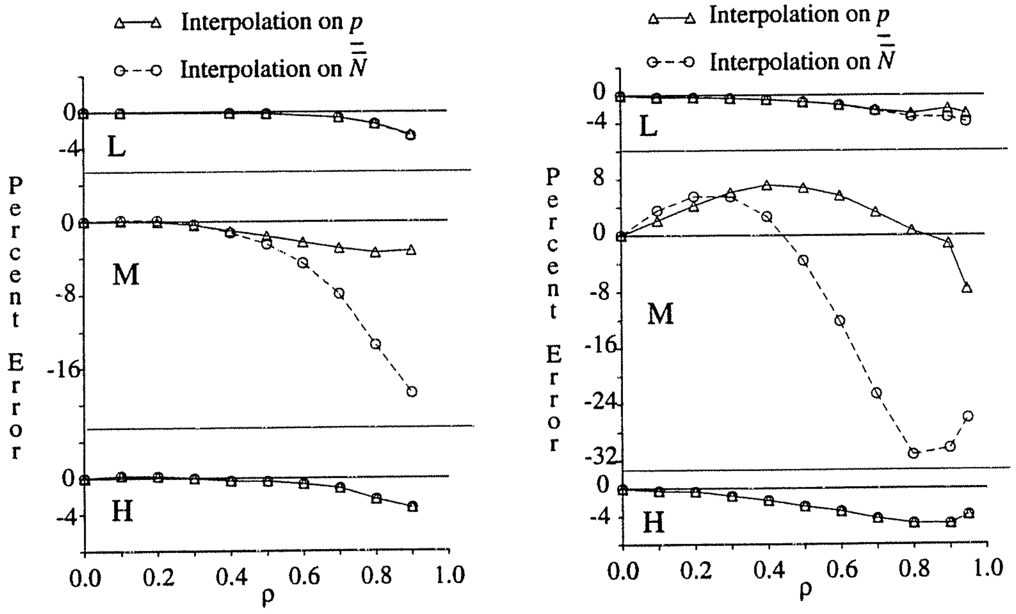
Figure 4.9 presents example percent errors for the PSAPF approximations for $C_D = 1, 5$ and



(a) P=20

(b) P=100

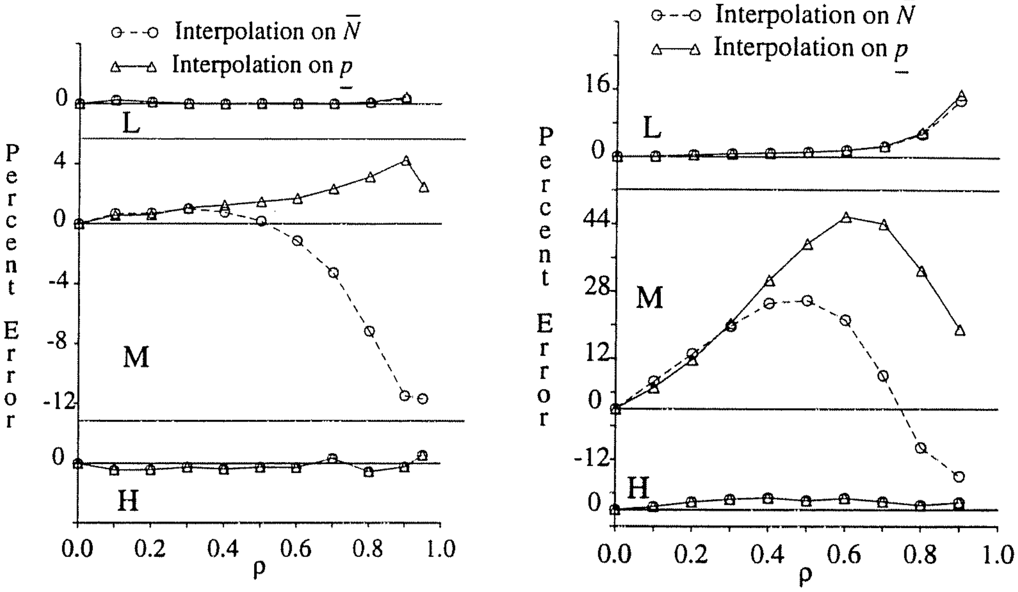
Figure 4.7: Example Validations for EQ



(a) $C_D = 1$

(b) $C_D = 5$

Figure 4.8: Example Validations for FCFS, $P = 100$



(a) $C_D = 1$

(b) $C_D = 5$

Figure 4.9: Example Validations for PSAPF, $P = 100$

$P = 100$. 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.

## 4.6 Conclusion

In the survey of analytic performance evaluation techniques in Chapter 2 we saw that exact techniques that exist in literature are either limited to small system sizes for computational reasons or require specific workload assumptions such as i.i.d. task service times within and across jobs. This motivates the use of approximate analysis for large systems and general workloads.

In this chapter we have explored the new approach of interpolation approximations to estimate mean response times for parallel processor policies. We first found points in the parameter space for which the parallel system reduces to a queueing system that has a known solution. We then showed how three types of interpolations can be used to obtain mean response time estimates over the entire parameter space of the assumed workload. 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.

The interpolation approximation approach was applied to three scheduling policies: EQ, FCFS, and PSAPF for a workload with general demands, general available parallelism, no correlation between demand and parallelism, and linear job execution rates. We will relax the last two assumptions for the EQ policy in the next chapter, and obtain approximations for $\overline{R}_{PSAPF}$ under correlated workloads in Chapter 6. We will also apply the interpolation approximation approach to the ASP policy in Chapter 6.

We have shown how interpolation approximations yield closed form expressions for $\overline{R}_{EQ}$, $\overline{R}_{FCFS}$, and $\overline{R}_{PSAPF}$. Validations showed these approximate estimates of mean response time to be fairly accurate over the parameter space. More accurate estimates could be obtained by using tighter approximations or exact analysis at the extreme values of interpolation parameters, but we preferred to choose the analysis at extreme values of system parameters such that it yielded simple expressions that readily show the dependence of policy performance on workload parameters. Using the closed form expressions for mean response time we saw that the approximations for $\overline{R}_{EQ}$ are independent of coefficient of variation in job demand, $C_D$, whereas the approximations for $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_D^2$. Thus $C_D$ is a key determinant of relative policy performance, which will be used to compare the policies in Chapter 7.

# Chapter 5

# Analysis of the EQS Policy

This chapter derives mean response time solutions for the spatial EQuipartitioning (EQS) policy so that we can study its qualitative behavior as a function of key parameters and compare its performance with the other policies, namely, ASP, FCFS, and PSAPF. The previous chapter illustrated that the interpolation approximation approach for the EQ policy yields accurate mean response time estimates under general distributions of marginal demand and available parallelism, and under the assumptions of linear ERFs and independence between demand and parallelism. In this chapter we use the interpolation approximation approach to obtain mean response time solutions for EQS assuming a general ERF, under both uncorrelated and correlated workloads. To do this, in Section 5.1 we first reduce the EQS policy under constant available parallelism to a *symmetric queue* and also generalize the light and heavy traffic limits for $\overline{R}_{EQ}$ derived in Chapter 4. We then use the reductions for $\overline{R}_{EQ}$ in Section 5.2 to derive four interpolation approximations under uncorrelated workloads and two under correlated workloads assuming general demand and available parallelism distributions and a general ERF. In Section 5.3 we propose a different approximate analysis approach under general workload assumptions and show that it is a generalization of an accurate interpolation approximation of Section 5.2. Finally, Section 5.5 summarizes the analysis of this chapter.

## 5.1 Reductions for EQS under a General ERF

In this section we analyze the system $(EQS, \mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma)$ under special cases of system parameters. We first consider a constant value of available parallelism for all jobs, i.e., $N = k$, $1 \leq k \leq P$, and then consider light and heavy traffic limits. We finally summarize both cases (constant parallelism and traffic limits) by means of three dimensional plots of normalized mean response time versus offered load and parallelism.

### 5.1.1 Analysis under Constant $N$

The EQS system under constant available parallelism will be shown to reduce to a *symmetric queue*, which is defined as follows [30].

**Definition 5.1.1 (Symmetric Queue)** *A queue is a* symmetric *queue if it operates in the following manner:*

*(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 $l \in \{1,2,\ldots,j\}$; 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 $l \in \{1,\ldots,j+1\}$ 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 $\phi(j) > 0$ if $j > 0$, and $\sum_{l=1}^{j} \alpha(l,j) = 1$.*

The exact solution for mean response time of the system (EQS, $N = k$, $\mathcal{F}_D^u$, $r = 0$, $\gamma$), for $k = 1, 2, \ldots, P$, is given by the following theorem.

**Theorem 5.1.1** *For the system $(EQS, N = k, \mathcal{F}_D^u, r = 0, \gamma)$, $1 \le k \le P$,*

$$\overline{R}_{EQS}(N = k, r = 0) \;=\; \frac{b}{\lambda}\left\{\sum_{i=1}^{P} \frac{(P\rho)^i}{(i-1)!\,\gamma(k)^{\min(i,m)} \prod_{j=m+1}^{i} \gamma(P/j)} + \frac{(P\rho)^P}{P!\,\gamma(k)^m \prod_{j=m+1}^{P} \gamma(P/j)} \frac{\rho}{1-\rho}\left(\frac{1}{1-\rho} + P\right)\right\} \quad (5.1)$$

*where $m = \lfloor P/k \rfloor$, $\rho = \lambda\overline{D}/P$, and*

$$b \;=\; \left[1 + \sum_{i=1}^{P} \frac{(P\rho)^i}{i!\,\gamma(k)^{\min(i,m)} \prod_{j=m+1}^{i} \gamma(P/j)} + \frac{(P\rho)^P}{P!\,\gamma(k)^m \prod_{j=m+1}^{P} \gamma(P/j)} \frac{\rho}{1-\rho}\right]^{-1}.$$

**Proof.** Let $\Gamma_k$ denote the system $(EQS, N = k, \mathcal{F}_D^u, r = 0, \gamma)$, $1 \leq k \leq P$. We first show that $\Gamma_k$ is a special case of a *symmetric queue* [30], and then derive the mean response time of $\Gamma_k$ under a general distribution of job demand.

System $\Gamma_k$ satisfies conditions (i) through (iv) for a symmetric queue in Definition 5.1.1. The total service effort supplied when there are $j$ jobs in $\Gamma_k$ is $\phi(j) = j \cdot \min(E(k), E(P/j))$ since $E$ is non-decreasing.. (Note that for $j \geq P$, $\phi(j) = P$, because $E(x) = \gamma(x) = x$ for $0 \leq x \leq 1$.) From the definition of the EQ policy, $\alpha(l, j) = 1/j$, $l = 1, \ldots, j$, since each job in $\Gamma_k$ gets an equal fraction of processing power. Note that this does not hold if available parallelism is not constant across all jobs.

The mean response time of a job in system $\Gamma_k$ can be derived from Theorems 3.8 and 3.10 of [30], which give the following steady state probability of $i$ jobs in the queue for the stationary symmetric queue with arbitrary distribution of job service time:

$$\pi_i = \frac{ba^i}{\prod_{l=1}^{i} \phi(l)}, \quad i = 0, 1, 2, \ldots \tag{5.2}$$

where

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

Substituting $\phi(l) = l \min(E(k), E(P/l))$ into equation (5.2) and using Little's Result

$$\overline{R}_{EQS}(N = k) = \frac{\sum_{i=1}^{\infty} i\pi_i}{\lambda},$$

we obtain the mean response time for $\Gamma_k$ as given in (5.1), where $m = \lfloor P/k \rfloor$. That is, $m$ is the maximum number of jobs that can execute simultaneously without contention for processors. To derive (5.1) we used the fact that if there are $P$ or more jobs in the system then the total service effort of the symmetric queue is $P$. ∎

**Remark:** Although the above theorem is derived for the case $E = \gamma$, the symmetric queue reduction and equation (5.2) actually hold for any nondecreasing ERF $E$, (e.g., for $E(j) = (j/N)\gamma(N)$) and the mean response time formula (5.1) holds for any nondecreasing $E$ such that $E(x) = x$ for $0 \leq x \leq 1$.

An important observation from equation (5.1) is that $\overline{R}_{EQS}(eq - N = k)$ *depends only on the mean job demand and not on higher moments of job demand.* This property is a generalization of the corresponding property for Processor Sharing (PS) systems. Note that when $P = 1$ or when $N = 1$ the EQS policy is identical to the PS policy, and thus $\overline{R}_{EQS}(N = 1)$ equals $\overline{R}_{M/G/P\ PS} = \overline{R}_{M/M/P}$. For the case of linear ERFs it was shown in Proposition 4.2.1 that when $P \bmod k = 0$, $\overline{R}_{EQS}(N = k) = \overline{R}_{M/G/c\ PS} = \overline{R}_{M/M/c}$, where $c = P/k$. The same does not hold for nonlinear ERFs, however, if $k > 1$.

### 5.1.2 Light and Heavy Traffic Analysis

We present exact results for the limiting cases of job arrival rates, $\lambda \rightarrow 0$ and $\lambda \rightarrow P/\overline{D}$. For this analysis we will assume that $D$ and $N$ are independent.

**Theorem 5.1.2** *The following light and heavy traffic limits hold for the system (EQS, $\mathcal{F}_N$, $\mathcal{F}_D^u$, $r = 0$, $\gamma$).*

$$\lim_{\rho \to 0} \overline{R}_{EQS} = \overline{S} = \overline{D}E\left[\frac{1}{\gamma(N)}\right] = \overline{D}\sum_{k=1}^{P} p_k \frac{1}{\gamma(k)} \tag{5.3}$$

$$\lim_{\rho \to 1}(1-\rho)\overline{R}_{EQS} = \frac{\overline{D}}{P}. \tag{5.4}$$

*where $\rho = \lambda\overline{D}/P$.*

**Proof.** The light traffic limit is straightforward. The proof for the heavy traffic limit is as follows.

1. Pr[An arriving job finds less than P jobs in the system] $\rightarrow 0$ as $\rho \rightarrow 1$. Hence the steady state probability of there being less than P jobs in the system goes to 0 as $\rho \rightarrow 1$ (because a Poisson arrival takes a random look at the system).

2. From 1, it follows that the fraction of time that the system is in states with less than P jobs goes to 0 as $\rho \rightarrow 1$. Therefore, the fraction of time that jobs hold more than one processor goes to 0 as $\rho \rightarrow 1$. In other words, the fraction of time that jobs hold less than or equal to one processor goes to 1 as $\rho \rightarrow 1$.

3. When a job has less than or equal to one processor, its parallelism does not determine its execution rate. That is, the job completes at rate $a$ where $a \leq 1$ is the amount of processing power allocated to the job and is independent of $N$. Since $D$ and $N$ are independent a job's parallelism does not determine its completion time if it is allocated less than or equal to one processor throughout its lifetime.

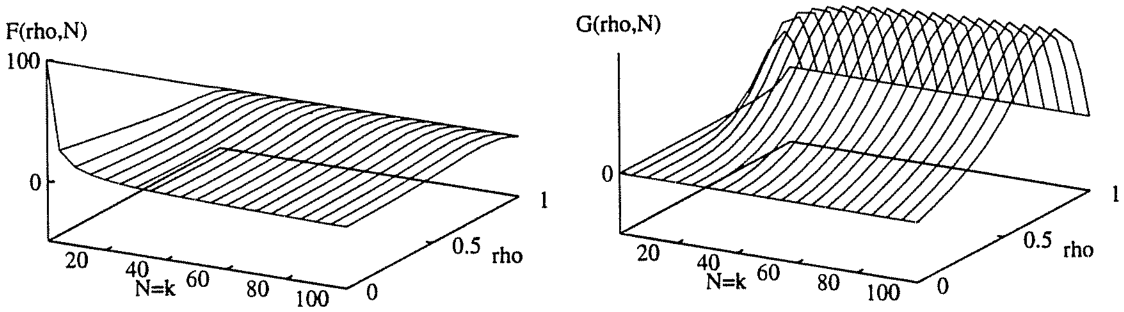
4. From 2 and 3 it follows that $\overline{R}_{EQS}$ is independent of the distribution of parallelism as $\rho \rightarrow 1$. As a result we use the mean response time for $N = 1$ (equation (5.1)) to obtain the heavy traffic limit given in (5.4). (Other choices of $N$ such as $N = k$, $2 \leq k \leq P$, yield the same heavy traffic limit.)

■

In the above proof we assumed that the system is stable when $\rho \rightarrow 1$ even when the ERF $\gamma$ is sublinear. The reason is that once there are P or more jobs in the system all jobs are processed with a linear execution rate. Thus $\lambda < P/\overline{D}$ (i.e., $\rho < 1$) ensures a stable system.

### 5.1.3 Summary of Reductions

To summarize the results of the reductions derived in Sections 5.1.1 and 5.1.2, Figure 5.1a plots the *normalized* mean response time under EQS, $F(\rho, k) = (1-\rho)\overline{R}_{EQS}(\rho, N = k)$, and Figure 5.1b plots the normalized mean extra time $G(\rho, k) = (1-\rho)\overline{X}_{EQS}(\rho, k)$, where $k = 1, 2, \ldots, P$. The curves are plotted for $P = 100$, mean job demand $\overline{D} = P = 100$, and the ERF $\gamma(k) = k^{0.8}$. The curve trends at $\rho = 0$ and $\rho = 1$ are similar to those in Figures 4.2a and 4.3a, which were for the linear ERF. We however, note that for moderate to high $\rho$ the trends are different when the ERF is sublinear. In Figure 5.1b we observe that for $k \geq 5$, $G(\rho, k)$ initially rises with $\rho$ and after reaching a maximum sharply decreases as $\rho \to 1$. The reason for the sharp decrease when $\rho \to 1$ is that as $\rho \to 1$ all jobs in the system execute with a linear execution rate.



(a) Normalized Mean Response Time

(b) Normalized Mean Extra Time

Figure 5.1: Summary of Exact Results for EQS

$$\text{ERF } \gamma(i) = i^{0.8},\ i = 1, 2, \ldots$$
$$\overline{D} = P = 100$$

## 5.2 Interpolation Approximations

In this section we first present four interpolation approximations for $\overline{R}_{EQS}$ for the workload $(EQS, \mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma)$, that is, for an uncorrelated workload with general distributions for demand and available parallelism, and a general ERF $\gamma$. These approximations are derived from the exact results of Section 5.1. We then extend our analysis to correlated workloads by deriving two new interpolations. The first three interpolations under uncorrelated workloads are similar to the interpolation approximations of Chapter 4. The fourth interpolation approximation under $r = 0$ is new and generalizes to one of the interpolations under correlated workloads. Validations of our approximations are provided in Section 5.4. All the interpolation approximations in this section are ad hoc; in Section 5.3 and Chapter 8 we provide partial explanations for why they work out to be accurate.

### 5.2.1 Interpolation on $\rho$: $r = 0$

Let $F(\rho) = (1-\rho)\overline{R}_{EQS}(\rho)$. A linear interpolation between $F(0)$ and $F(1)$ yields the following estimator for $F(\rho)$.

$$\hat{F}(\rho) = (1-\rho)F(0) + \rho F(1) = (1-\rho)\overline{S} + \rho\overline{D}/P,$$

where the values of $F(0)$ and $F(1)$ are obtained from Theorem 5.1.2 in Section 5.1.2. Dividing $\hat{F}(\rho)$ by $(1-\rho)$ we obtain the desired estimator for $\overline{R}_{EQS}$ as

$$\begin{aligned} \overline{R}_{EQS}(r=0) \approx \hat{R}^{\rho}_{EQS} &= \overline{S} + \frac{\rho}{1-\rho}\frac{\overline{D}}{P} \\ &= \overline{D}E[1/\gamma(N)] + \frac{\rho}{1-\rho}\frac{\overline{D}}{P}, \text{ under } (\mathcal{F}_N, \mathcal{F}^u_D, \cdot, \gamma). \end{aligned} \qquad (5.5)$$

Validations in Section 5.4 will show that approximation (5.5) is accurate when ERFs are close to linear, but can have relative errors as high as −86% for extremely sublinear ERFs. Typical errors in all our validations (1302 data points) were in the range of −35% to 15%. We note from (5.5) that $\hat{R}^{\rho}_{EQS}$ is independent of $C_D$ which corroborates our observation in Section 5.1.1 that $\overline{R}_{EQS}(N = k)$ is independent of $C_D$.

### 5.2.2 Interpolation on $\overline{N}$: $r = 0$

Let $\overline{X}_{EQS} \equiv \overline{R}_{EQS} - \overline{S}$. A linear interpolation on $\overline{N}$ between $\overline{X}_{EQS}(N = 1)$ and $\overline{X}_{EQS}(N = P)$ yields the following estimator for $\overline{R}_{EQS}$

$$\hat{R}^{\overline{N}}_{EQS} = \overline{S} + \left(\frac{P-\overline{N}}{P-1}\right)\overline{X}_{EQS}(\overline{N}=1) + \left(\frac{\overline{N}-1}{P-1}\right)\overline{X}_{EQS}(\overline{N}=P).$$

Therefore, under the assumptions $(\mathcal{F}_N, \mathcal{F}^u_D, r = 0, \gamma)$

$$\begin{aligned} \overline{R}_{EQS}(\mathcal{F}_N, r=0) \approx \hat{R}^{\overline{N}}_{EQS} &= \overline{D}E[1/\gamma(N)] + \left(\frac{P-\overline{N}}{P-1}\right)\overline{W}_{M/M/P} + \\ &\quad \left(\frac{\overline{N}-1}{P-1}\right)(\overline{R}_{EQS}(N=P) - \overline{D}/\gamma(P)), \end{aligned} \qquad (5.6)$$

where $\overline{R}_{EQS}(N = P)$ can be computed exactly using equation (5.1) (Section 5.1.1). Validations in Section 5.4 will show this interpolation on $\overline{N}$ to be more accurate in general than the interpolation on $\rho$. However, the interpolation on $\overline{N}$ is quite sensitive to the distribution of $N$. For workloads with high and low mean parallelism, the interpolation on $\overline{N}$ is vey accurate, but for workloads with moderate parallelism and low $C_N$ its accuracy is low for sublinear ERFs. Typical relative errors for this approximation ranged between $-35\%$ to $5\%$ in all our validations. We note that $\hat{R}^{\overline{N}}_{EQS}$ is also independent on $C_D$ just like the interpolation on $\rho$. This follows because each of $\overline{R}_{EQS}(N = 1)$ and $\overline{R}_{EQS}(N = P)$ is independent of $C_D$.

### 5.2.3 Interpolation on the pmf of $N$: $r = 0$

In Section 5.1.1 we derived solutions for $\overline{R}_{EQS}(N = k)$, $k = 1, 2, \ldots, P$ (see equation (5.1)). These points of constant parallelism are extreme values of the pmf of $N$, that is, $\underline{p} = \underline{e}_k$, $k = 1, \ldots, P$, where $\underline{e}_k$ is a vector with a 1 in the $k^{th}$ component and $0's$ elsewhere. An interpolation through the mean response times at these points ($\overline{R}_{EQS}(N = k)$ ) yields the following approximation

$$\overline{R}_{EQS}(\mathcal{F}_N, r = 0) \approx \hat{R}^{\underline{p}}_{EQS} = \sum_{k=1}^{P} p_k \overline{R}_{EQS}(N = k, r = 0), \qquad \text{under } (\cdot, \mathcal{F}^u_D, \cdot, \gamma). \qquad (5.7)$$

Approximation (5.7) is the most accurate approximation among the interpolations on $\rho$, $\overline{N}$ and $\underline{p}$. Section 5.4 will show that for nearly all data points in the validations the relative errors for the interpolation on $\underline{p}$ were between $-5\%$ to $5\%$. The maximum relative error among all validations was about 14%. Note that the interpolation on $\underline{p}$ was also the most accurate interpolation approximation for $\overline{R}_{EQS}$ in Chapter 4, where the workload assumptions were restricted to linear ERFs. From approximation (5.7) we note that $\hat{R}^{\underline{p}}_{EQ}$ is independent of $C_D$ since $\overline{R}_{EQS}(N = k)$ is independent of $C_D$ for all $k = 1, \ldots, P$.

### 5.2.4 Interpolation on $E[1/\gamma(N)]$: $r = 0$

At light loads $\overline{R}_{EQS} = \overline{S}$ which at $r = 0$ equals $\overline{D}E[1/\gamma(N)]$. Thus at $\rho = 0$, $\overline{R}_{EQS}$ is linear in $E[1/\gamma(N)]$ with slope $\overline{D}$. When $\rho \to 1$ we note that $(1 - \rho)\overline{R}_{EQS} = \overline{D}/P$ by the heavy traffic limit and thus $\overline{R}_{EQS}$ is independent of $E[1/\gamma(N)]$ when $\rho \to 1$. More precisely, it is linear in $E[1/\gamma(N)]$ with slope zero. Thus at the extreme values of $\rho$, $\overline{R}_{EQS}$ is linear in $E[1/\gamma(N)]$. We assume a similar behavior at in between values of $\rho$ as well to get an interpolation on $E[1/\gamma(N)]$. That is, we consider extreme values of $E[1/\gamma(N)]$, i.e., 1 at $N = 1$ and $1/\gamma(P)$ at $N = P$, and then interpolate between the mean response time values at the two endpoints to obtain for the workload ($\mathcal{F}_N$, $\mathcal{F}_D^u$, $r = 0$, $\gamma$):

$$\overline{R}_{EQS}(\mathcal{F}_N, r=0) \approx \hat{R}_{EQS}^{E[1/\gamma(N)]} = \frac{E\left[\frac{1}{\gamma(N)}\right] - \frac{1}{\gamma(P)}}{1-\frac{1}{\gamma(P)}}\overline{R}_{EQS}(N=1) + \frac{1-E\left[\frac{1}{\gamma(N)}\right]}{1-\frac{1}{\gamma(P)}}\overline{R}_{EQS}(N=P), \tag{5.8}$$

where $\overline{R}_{EQS}(N=1)$ and $\overline{R}_{EQS}(N=P)$ are obtained from equation (5.1) in Section 5.1.1. Not only will validations show this to be an accurate approximation (more than 95% of validation data points had relative errors between −5% to 15%) but we will also corroborate in Chapter 8 that $E[1/\gamma(N)]$ (almost) uniquely determines the mean response time of the EQS policy when $r = 0$.

### 5.2.5 Interpolation on $S_n$: $0 \leq r \leq 1$

The interpolation approximations for $\overline{R}_{EQS}$ derived above are accurate when there is no correlation between demand and parallelism. We wish to obtain an approximation for $\overline{R}_{EQS}$ for correlated workloads for all values of the correlation coefficient, $r$, between mean demand and available parallelism. To develop approximations for $\overline{R}_{EQS}$ under arbitrary values of $r$ ($0 \leq r \leq 1$) consider the following two approaches:

(i) Interpolation on $r$:
Obtain $\overline{R}_{EQS}$ at $r = 0$ and $r = 1$ and then interpolate between these two endpoints. This approximation has the form

$$\overline{R}_{EQS} \approx (1 - f(r))\,\overline{R}_{EQS}(r=0) \;+\; f(r)\,\overline{R}_{EQS}(r=1),$$

where $f(r)$ is a suitable function of $r$ such that $0 \leq f(r) \leq 1$, $f(0) = 0$, and $f(1) = 1$. Whatever be the choice of $f(r)$ the interpolation cannot be done unless we know the values of $\overline{R}_{EQS}(r=0)$ and $\overline{R}_{EQS}(r=1)$. $\overline{R}_{EQS}(r=0)$ can easily be approximated using one of the interpolation approximations for $r = 0$. However, for the present we do not have a simple way of estimating $\overline{R}_{EQS}(r=1)$.

(ii) Generalize one of the previous approximations for $r = 0$.

We followed the second approach in developing an approximation for $\overline{R}_{EQS}$ when $0 \leq r \leq 1$. We will show that this approximation has the useful property of being rewritten as an interpolation on $r$, and thus can easily be used to obtain the qualitative behavior of EQS as a function or $r$. The approximation for $\overline{R}_{EQS}$ when $0 \leq r \leq 1$ is derived from the interpolation on $E[1/\gamma(N)]$ for $r = 0$. Since $\overline{S} = \overline{D}E[1/\gamma(N)]$ when $r = 0$, $E[1/\gamma(N)] = \overline{S}/\overline{D}$. Let $S_n = \overline{S}/\overline{D}$, $0 \leq r \leq 1$, denote the normalized mean service time of the workload. We generalize the interpolation on $E[1/\gamma(N)]$ given by approximation (5.8), by replacing $E[1/\gamma(N)]$ by $S_n$. Hence for $0 \leq r \leq 1$, we obtain for the workload $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$:

$$\overline{R}_{EQS}(\mathcal{F}_N, r) \quad \approx \quad \hat{R}_{EQS}^{Sn} = \frac{S_n - \frac{1}{\gamma(P)}}{1 - \frac{1}{\gamma(P)}} \overline{R}_{EQS}(N = 1) + \frac{1 - S_n}{1 - \frac{1}{\gamma(P)}} \overline{R}_{EQS}(N = P). \qquad (5.9)$$

Note that at $N = 1$ and $N = P$ the question of correlation does not arise since $N$ is constant. Hence in this approximation $S_n$ alone captures complete information about correlation. Validations for approximation (5.9) are given in Section 5.4. Typical relative errors from simulation estimates in all our validations (2866 data points) were in the range $-5\%$ to $15\%$, which shows that this approximation is very accurate. We will corroborate in Chapter 8 that $S_n$ is (almost) uniquely determines $\overline{R}_{EQS}$ for correlated workloads. Note that since $\overline{R}_{EQS}(N = 1)$ and $\overline{R}_{EQS}(N = P)$ are independent of $C_D$ so is $\hat{R}_{EQS}^{Sn}$. Thus all interpolation approximations for $\overline{R}_{EQS}$ derived in this section are independent of $C_D$.

### 5.2.6 Interpolation on $r$: $0 \leq r \leq 1$

Approximation (5.9) can be rewritten in a form that explicitly shows the dependence of $\overline{R}_{EQS}$ on $r$. From equation (3.6) we note that $S_n = (1 - r^2)S_n(r = 0) + r^2 S_n(r = 1)$. Substituting this in (5.9) and simplifying we obtain for the workload $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ that

$$\overline{R}_{EQS}(r) \approx (1 - r^2)\overline{R}_{EQS}(r = 0) + r^2 \overline{R}_{EQS}(r = 1). \qquad (5.10)$$

## 5.3 Generalized Approximate Analysis: New derivation of interpolation approximations

In Section 5.2 we developed an interpolation approximation on $\underline{p}$ for $\overline{R}_{EQS}(r = 0)$ and this approximation will be shown to be extremely accurate in the validations section of this chapter. Likewise, the interpolation on $r$ that holds for correlated workloads is also very accurate. However, all interpolations derived thus far have been very ad hoc. That is, they are accurate but there does not seem to be satisfactory explanation for why they are accurate. In this section we use an alternate approach, under general workload assumptions, the results of which provide a justification for the interpolation on $\underline{p}$ when $r = 0$ and for the interpolation on $r$. Furthermore, the new approximation also shows how the interpolation on $\underline{p}$ generalizes for correlated workloads.

The approximate mean response time for the EQS policy for the workload $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ is derived by (1) classifying jobs according to their available parallelism, (2) computing the mean response time for each class of jobs by approximating the *average interference* from other classes of jobs, and (3) computing the overall mean response time as a weighted sum of the approximate mean response times per class. The particular approximate representation of average interference by other job classes yields a system for each class that reduces to a symmetric queue, from which the class mean response time is computed.

Let a job with available parallelism $k$ belong to class $C_k$, for $k = 1, \ldots, P$. Let $\overline{R}_{EQS,C_k}$ denote the mean response time of class $C_k$ under the workload $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$. Clearly,

$$\overline{R}_{EQS} = \sum_{k=1}^{P} p_k \overline{R}_{EQS,C_k}. \tag{5.11}$$

The approximate processor contention from classes other than $C_k$ is modeled by assuming each such class has available parallelism $k$, but retains its total service demands as before. More precisely, we approximate $\overline{R}_{EQS,C_k}$ to be the mean response time of class $C_k$ in a system $\Gamma_k$ which is like the original system except that a class $C_j$ job in $\Gamma_k$ has demand $D_j$ and available parallelism $k$, where $\overline{D}_j = q\overline{D} + (1-q)cj$, $q = 1 - r^2$ and $c = \overline{D}/\overline{N}$, as per the correlation model in Section 3.3. The instantaneous load of class $C_j$ jobs is not accurately modeled by assuming that class $C_j$ jobs have parallelism $k$. However, the average load by class $C_j$ jobs is accurately modeled since the arrival rate and distribution of processing requirement of the class are as in the actual system. Thus, the overall interference of $C_j$ with $C_k$ may be reasonably well represented.

An approximation for $\overline{R}_{EQS,C_k}$ can be derived by solving for the mean response time of class $k$ in system $\Gamma_k$. Note that in system $\Gamma_k$ there are $P$ job classes, $C_1, \ldots, C_P$, where $C_j$ has available parallelism $k$ and demand $D_j$. Since all jobs have the same available parallelism and since the definition of a symmetric queue permits multiple job classes with different service demand distributions (see Definition 5.1.1), the system again reduces to a symmetric queue. In this case, the total service effort with $j$ jobs in the queue is $\phi(j) = j \cdot \min(\gamma(k), \gamma(P/j))$, $j \geq 0$, and the fraction of effort for job $i$ is $\alpha(i,j) = 1/j$, for $i = 1, \ldots, j$. Furthermore, equation (5.2) holds also for the case of multiple classes with different distribution of demand (see Theorem 3.8 and 3.10 of [30]). Hence, $\overline{R}_{\Gamma_k} = \overline{R}_{EQS}(N = k, r = 0)$, and the overall mean response time for $\Gamma_k$ depends on the overall mean demand but not the demand distributions per class.

The mean response time of class $k$ in $\Gamma_k$ is obtained from part (ii) of Theorem 3.10 of Kelly [30]. Using the notation in this thesis, this theorem can be stated as follows.

> Given there are $Q$ customers in the symmetric queue, the classes of the customers are independent and the probability the customer in a given position is of class $C_k$ is $\dfrac{\lambda_k \overline{D}_k}{\lambda \overline{D}}$, where $\lambda_k$ is the arrival rate of class $C_k$ and $\overline{D}_k$ is the mean demand of class $C_k$.

Thus given $Q$ jobs in system $\Gamma_k$, the of number, $Q_k$, of jobs of class $C_k$ is binomially distributed with parameters $Q$ and $u_k$ where $u_k := \lambda_k \overline{D}_k/(\lambda \overline{D}) = p_k \overline{D}_k/\overline{D}$. Therefore $\overline{Q}_k = \overline{Q}u_k$ and using Little's law we obtain the mean response time of class $C_k$ in $\Gamma_k$ as

$$\overline{R}_{\Gamma_k,C_k} = \frac{\overline{Q}_k}{\lambda_k} = \frac{\overline{Q}u_k}{\lambda p_k} = \frac{\overline{D}_k}{\overline{D}}\overline{R}_{\Gamma_k}.$$

Since $\overline{R}_{\Gamma_k} = \overline{R}_{EQS}(N = k, r = 0)$ and $\overline{R}_{\Gamma_k,C_k}$ is the proposed approximation for $\overline{R}_{EQS,C_k}$ we obtain

$$\overline{R}_{EQS,C_k} \approx \frac{\overline{D}_k}{\overline{D}}\overline{R}_{EQS}(N = k, r = 0).$$

Substituting this in (5.11), we obtain under the workload assumptions $(\cdot, \mathcal{F}_D^u, \cdot, \gamma)$ that

$$\overline{R}_{EQS}(\mathcal{F}_N, r) \approx \sum_{k=1}^{P} p'_k \overline{R}_{EQS}(N = k, r = 0), \quad p'_k = p_k \frac{\overline{D}_k}{\overline{D}} = p_k \left(1 - r^2 + r^2 \frac{k}{\overline{N}}\right), \tag{5.12}$$

where the expression for $p'_k$ was derived as per the correlation model described in Section 3.3.

Further insight can be obtained from equation (5.12) by making the following observations. When $r = 0$, $p'_k = p_k$, for $k = 1, 2, \ldots, P$, and approximation (5.12) reduces to the interpolation approximation in (5.7). On the other hand when $r = 1$ it follows from (5.12) that under the assumptions $(\cdot, \mathcal{F}_D^u, \cdot, \gamma)$,

$$\overline{R}_{EQS}(\mathcal{F}_N, r = 1) \approx \sum_{k=1}^{P} p_k \frac{k}{\overline{N}} \overline{R}_{EQS}(N = k, r = 0).$$

Finally, for $r$ between 0 and 1 and $(\cdot, \mathcal{F}_D^u, \cdot, \gamma)$,

$$\begin{aligned}
\overline{R}_{EQS}(\mathcal{F}_N, r) &\approx \sum_{k=1}^{P} p_k \left\{1 - r^2 + r^2 \frac{k}{\overline{N}}\right\} \overline{R}_{EQS}(N = k, r = 0) \\
&= (1 - r^2) \sum_{k=1}^{P} p_k \overline{R}_{EQS}(N = k, r = 0) + r^2 \sum_{k=1}^{P} p_k \frac{k}{\overline{N}} \overline{R}_{EQS}(N = k, r = 0) \\
&\approx (1 - r^2)\overline{R}_{EQS}(\mathcal{F}_N, r = 0) + r^2 \overline{R}_{EQS}(\mathcal{F}_N, r = 1),
\end{aligned}$$

which is the interpolation on $r$ (5.10).

The next section will show that this general approximation is extremely accurate for uncorrelated workloads as well as for correlated workloads.

## 5.4 Validations

In Section 5.2 we presented interpolations on $\rho$, $\overline{N}$, $\underline{p}$, $E[1/\gamma(N)]$, and $S_n$, and in Section 5.3 we presented a generalized approximation in order to estimate $\overline{R}_{EQS}$ for uncorrelated as well as correlated workloads. Note that the interpolation on $S_n$ is a generalization of the interpolation on $E[1/\gamma(N)]$ and thus we do not treat the latter as a separate interpolation. We validate the approximations derived in Sections 5.2 and 5.3 against simulation and also against special cases of exact analysis. We first discuss stress tests for the approximations, then provide the settings of validation parameters, after which we present a summary of validations, and then error plots for example validations.

### 5.4.1 Stress Tests for Validations

We first note that all three interpolations are exact when $\rho = 0$ and when $\rho = 1$ (by exact at $\rho = 1$ we mean that $\lim_{\rho \to 1}(1-\rho)\hat{R}^{x}_{EQS} = \lim_{\rho \to 1}(1-\rho)\overline{R}_{EQS}$ where $x \in \{\rho, \overline{N}, \underline{p}\}$). The interpolation on $\rho$ overpredicts mean response time when $N = 1$, and can underpredict mean response time for higher values of $N$ if the ERF is sublinear. To see this consider Figure 5.1b. The curve for $N = 1$ lies below the straight line that connects the points $G(0,1)$ and $G(1,1)$. On the other hand the curves for higher values of $N = k$ lie above the straight line that connects $G(0,k)$ and $G(1,k)$, specially at moderate to high utilizations. The interpolation on $\rho$ is however quite accurate for linear ERFs and moderate to high parallelism as seen from Section 4.5 in Chapter 4. To stress test the interpolation on $\rho$ we should therefore use workloads with sublinear ERFs, and also workloads with low parallelism and linear ERFs.

The interpolation on $\overline{N}$ is exact at $N = 1$ and $N = P$. We therefore expect it to be accurate at the low and high ends of parallelism. For linear ERFs $\hat{R}^{\overline{N}}_{EQS}$ usually underestimates $\overline{R}_{EQS}$ for moderate parallelism as seen in Section 4.5 in Chapter 4. This can also been seen for sublinear ERFs from Figure 5.1b for constant available parallelism. To stress test this interpolation we consider workloads with moderate parallelism. The interpolation on $\underline{p}$ is exact whenever $N$ is constant. We therefore expect it to be accurate when there is little variation in the distribution of $N$ (low $C_N$). Thus, to stress test this approximation we use workloads with high $C_N$. The interpolation on $S_n$ is exact at the extreme values of $S_n$, i.e., $S_n = 1/\gamma(P)$ and $S_n = 1$. To stress test this approximation we need to vary $S_n$ using various distributions for $N$, several settings for correlation coefficient $r$, and several ERFs. Finally, the generalized approximation (5.12) has properties similar to the interpolation on $\underline{p}$ and we use the same stress tests for it as for the interpolation on $\underline{p}$.

### 5.4.2 Validation Parameters Settings

The parameters of the workload model that we need to set are: number of processors $P$, distributions for job demand $D$, offered load $\rho$, distributions for available parallelism $N$, types of ERFs, and values for correlation coefficient $r$. We considered systems with $P = 20$ and $P = 100$.

The settings for $D$, $\rho$, and $N$ were identical to the settings of the validations for the interpolation approximations for linear ERFs discussed in Section 4.5.1 in Chapter 4. We used the following settings $\gamma$ and $r$.

(i) $\gamma$: $\gamma(k) = k$, $\gamma(k) = k^c$ $0 < c < 1$, $\gamma(k) = (1+\beta)k/(k+\beta)$ $0 < \beta < \infty$, $k = 1, 2, \ldots, P$
In the absence of extensive data for real workloads we validate our models against three types of ERFs. The first is simply the linear ERF. The second is a simple algebraic choice of a concave sublinear ERF, whereas the third is derived from a type of *execution signature* given in [18]. For the ERF $\gamma(k) = k^c$, we used c = 0.7, 0.8, and 0.9, which are plotted for $P = 100$ in Figure 3.2a. At $c = 0.7$, $\gamma(20) = 8.14$ and $\gamma(100) = 25.12$ which are quite low compared to their linear counterparts of 20 and 100, respectively. The value of $c = 0.7$ therefore stress tests the accuracy of the models for highly sublinear ERFs. For the ERF $\gamma(k) = (1+\beta)k/(k+\beta)$ the following values of $\beta$ are used in the validations: $\beta = 20, 50, 100$ for $P = 20$, and $\beta = 50, 100, 500$ for $P = 100$. The smaller values of $\beta$ are used as stress tests whereas the larger values are used to evaluate the accuracy of the models when the ERF is close to linear, but not exactly linear. Figure 3.2b plots these ERFs for $P = 100$.

(ii) $r$: 0, 0.5, 1.

The interpolation approximations were validated against exact analysis for workload settings with $N = k$, and against simulation otherwise. Exact values of $\overline{R}_{EQS}(N = k)$ were obtained using equation (5.1). Simulation estimates of $\overline{R}_{EQS}$ had 95% confidence intervals with less than 5% half-widths. They were obtained using the regenerative method for many data points and the method of batch means whenever the regenerative method was too time consuming (specially for workloads with low parallelism).

### 5.4.3 Summary of Validations

Figure 5.2 presents a histogram of percentage of data points versus relative error that summarizes all our validations for the interpolation approximations on $\rho$, $\overline{N}$, and $\underline{p}$, which assume that $r = 0$. The total number of data points was 1302. The histogram shows that the interpolation on $\underline{p}$ is the most accurate approximation (the maximum error for this interpolation was 13.92%). The interpolations on $\overline{N}$ and $\rho$ can considerably underestimate $\overline{R}_{EQS}$, but they are reasonably accurate for most of the data points. The worst case errors were located at ($P = 100$, $N = 100$, $\rho = 0.7,$, $\gamma(k) = k^{0.7}$) for the interpolation on $\rho$, ($P = 100$, $N = 25$, $\rho = 0.9$, $\gamma(k) = k^{0.8}$) for the interpolation on $\overline{N}$, and ($P = 100$, $N$ = Bounded-geometric($P_{max} = 0.5, p = 1$), $\rho =$

0.7, $\gamma(k) = k^{0.7}$) for the interpolation on $\underline{p}$. In general the interpolation on $\rho$ is quite inaccurate when the ERF has moderate or high sublinearity, but it is accurate for ERFs that are close to linear. The interpolation on $\overline{N}$ performs badly when the ERF has high sublinearity and $\overline{N}$ is low to moderate and $C_N$ is low. However, it is accurate for workloads with high or very low $\overline{N}$, and also for workloads with high $C_N$.

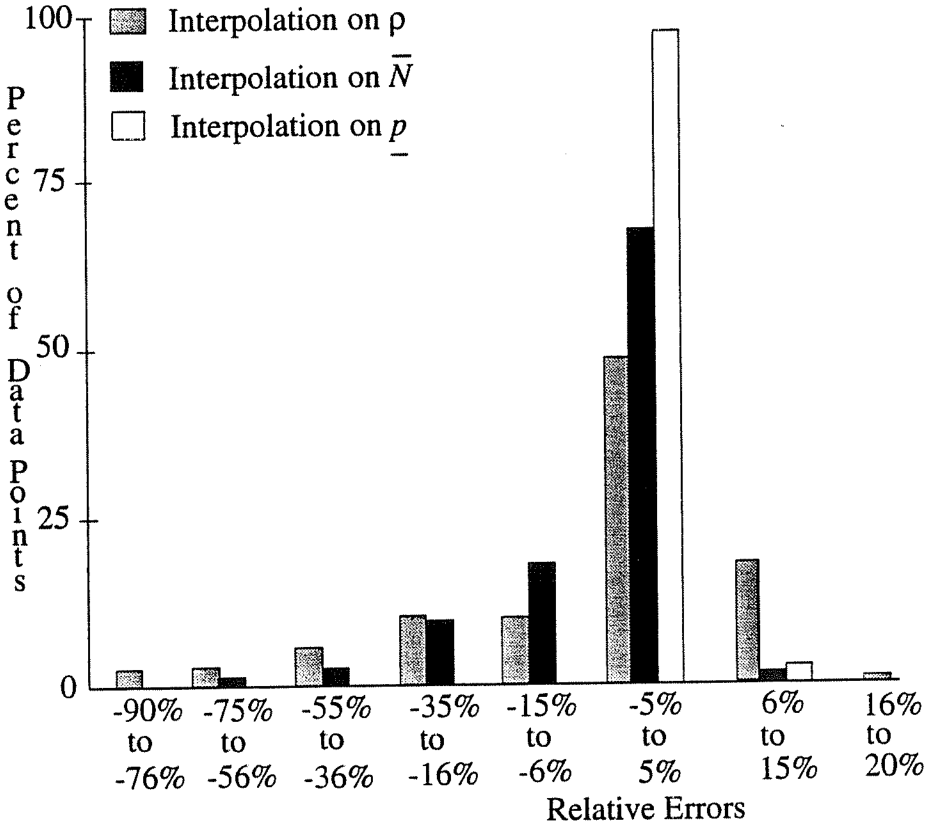


Figure 5.2: Summary of Validations for Interpolations on $\rho$, $\overline{N}$, and pmf

$r = 0$,

P=20,100

(1302 data points)

Figure 5.3 summarizes the validations for the interpolation on $S_n$ and the generalized approximation (5.12) for a total of 2866 data points. We used three values of $r$ in our validations, $r = 0$, $r = 0.5$, and $r = 1$. From Figure 5.3 we note that the generalized approximation is more accurate and that for both approximations at least 95% of the validations in each case have a relative error between -5% and 15%. The maximum relative error for the generalized approximation was the same as for the interpolation on $\underline{p}$ (see summary for $r = 0$) and the maximum relative error for the interpolation on $S_n$ of 34.18% was for the was for the data point ($P = 100$, $N =$ Bounded-geometric($P_{max} = 0, p = 0.1$), $\rho = 0.5$, $r = 1$, $\gamma(k) = k^{0.7}$). Approximation (5.12) is extremely accurate since all data points in Figure 5.3 are with 15% of simulation estimates. The highest errors (> 10%) for this approximation were observed for correlated workloads with low to moderate $\overline{N}$ (0.1P to 0.5P), high $C_N$, and moderate to high execution rate sublinearity.

### 5.4.4 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 $P$, specific $\gamma$, and $r$. The distributions of $N$ considered are bounded-geometric with parameter settings given in Table 3.2. For these example validations we varied $\rho$ from 0.0 to 1.0.

We first provide example validations for the interpolations on $\rho$, $\overline{N}$, and $\underline{p}$ which assume $r = 0$. In Figures 5.4a and b we plot the percent error of these approximations of $\overline{R}_{EQS}$ against simulation estimates of $\overline{R}_{EQS}$ as a function of $\rho$ for the H, M, and L workloads, for the linear ERF. This is one stress test for the performance of the approximations since the linear ERF is the limiting case of sublinear ERFs. In Figure 5.4b we present another stress test by using the ERF with maximum sublinearity among the ERF considered in the validations. The ERF used for this figure is $\gamma(k) = k^{0.7}$, $k = 1, 2, \ldots, P$. From Figures 5.4a and b we note that the interpolation on $\underline{p}$ is clearly the most accurate approximation. The interpolation on $\overline{N}$ is accurate for the H and L workloads, but it does not perform that well for the M workload, specially for the highly sublinear ERF. The interpolation on $\rho$ is reasonably accurate for linear ERFs, but it considerably underpredicts mean response time for high and moderate parallelism at extremely sublinear ERFs, specially at moderate to high utilizations.

We now present example validations for the interpolation approximation on $S_n$ and the generalized approximation (5.12). Figure 5.5 presents example validations for the interpolation on $S_n$ and the generalized approximation (5.12) for $r = 0.5$. We note that both approximations typically overestimate mean response time, specially when the ERF is sublinear. The generalized approximation is more accurate than the interpolation on $S_n$. For the interpolation on $S_n$ the error is less for the H and L workloads as compared to the $M$ workload. The reason is that the interpolation is exact at extreme ends of parallelism ($N = 1$ and $N = P$).

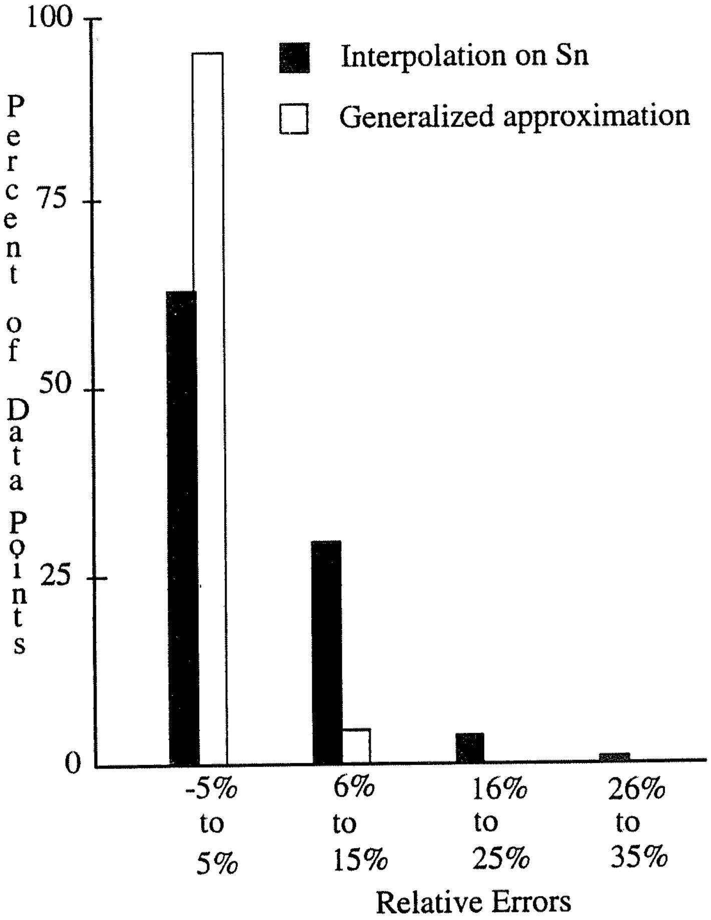
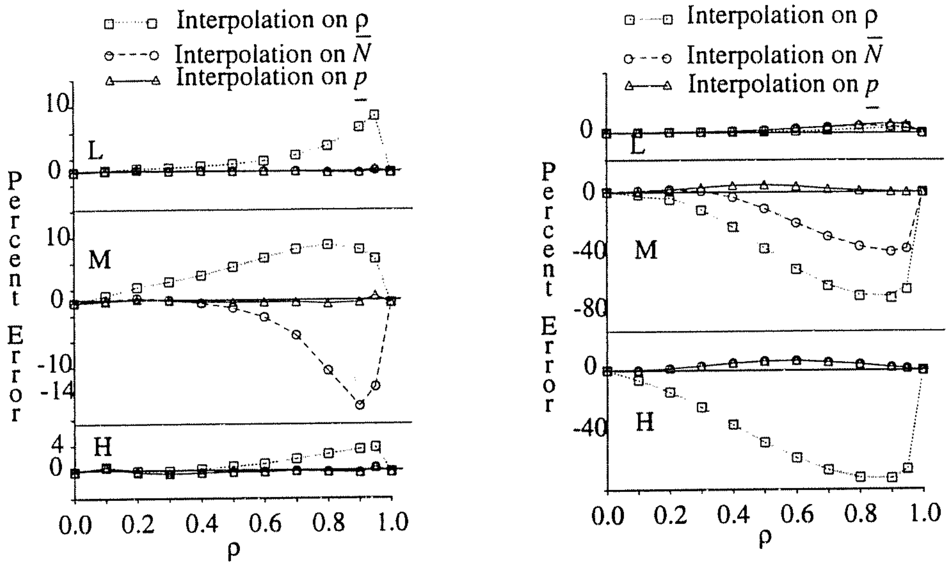


Figure 5.3: Summary of Validations for Interpolation on $S_n$ and generalized approximation

r=0, 0.5, 1,
P=20,100
(2866 data points)

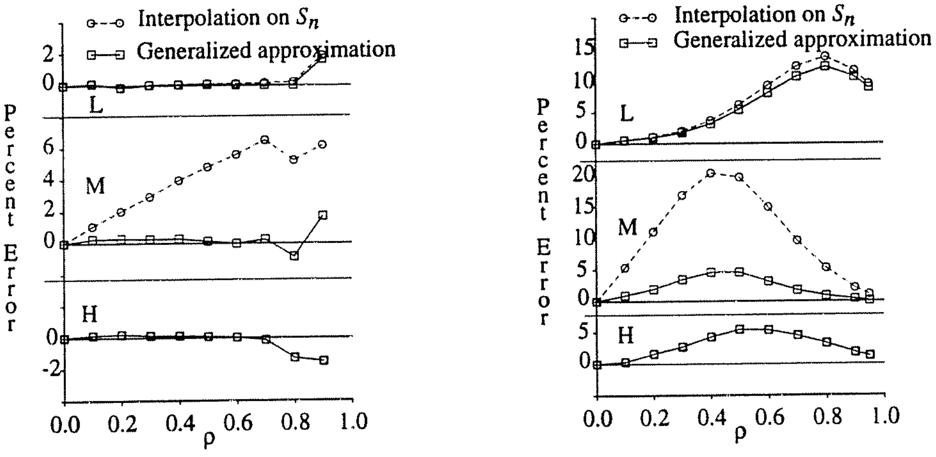


(a) Linear ERF

(b) $\gamma(k) = k^{0.7}$

Figure 5.4: Example Validations for Interpolation Approximations: r=0

$\overline{D} = P = 100$



(a) Linear ERF

(b) $\gamma(k) = k^{0.7}$

Figure 5.5: Example Validations for the Interpolation on $S_n$ and generalized approximation

$r = 0.5$,
$\overline{D} = P = 100$

## 5.5 Summary of Analysis and Relation to Previous Work

In this chapter we have developed interpolation approximations for $\overline{R}_{EQS}$ under the assumptions of general available parallelism, general demand, a general nondecreasing ERF, and correlated as well as uncorrelated workloads. This shows the potential of the interpolation approximation approach to model parallel systems with nonlinear ERFs and correlation between demand and parallelism. Our validations showed the interpolation approximations for EQS to be fairly accurate throughout the parameter space.

The only previous analyses in the literature that are related to this chapter are the analysis of EQS in [40] using recurrence relations for a workload model consisting of a fixed number of fork-join jobs with i.i.d. exponential task service times, and the analysis of $\text{EQ}_{PC}$ in [69] using matrix-geometric techniques for a workload model consisting of $C$ job classes with exponential demands, a fixed available parallelism, and a fixed ERF per class. The analysis in [69] is for a more practical workload than [40] and it is also extensible to phase-type distributions for demand, but it suffers from the drawback that it is computationally prohibitive to solve for systems beyond 10-20 processors. Furthermore, the numerical nature of the solution does not yield any insight into the performance of $\text{EQ}_{PC}$ . As shown in this chapter both these drawbacks are overcome by the interpolation approximation approach. The qualitative behavior of EQ has been studied using simulation in [41, 39] and we review their results in Chapter 8 where we further analyze the EQS policy.

Before ending this chapter we compare the interpolation approximations for $\overline{R}_{EQS}$ under linear ERFs as derived in Chapter 4 versus the corresponding interpolation approximations derived under general ERFs in this chapter. The interpolation on $\rho$ has the same structure for both linear as well as general ERFs but it is quite inaccurate for ERFs with moderate to high sublinearity. The interpolation approximations on $\overline{N}$ and $\underline{p}$ in this chapter have more complex expressions than the corresponding interpolations for $\overline{R}_{EQS}$ in Chapter 4 that were derived for the linear ERF. This is because we used the symmetric queue to model the EQS policy under constant available parallelism and we do not know of simple mean response time approximations for the symmetric queue. However, the increase in complexity of mean response time expressions does not prohibit the obtainment of key parameters since we could derive that the interpolations approximations are independent of $C_D$ and in Chapter 8 we will obtain that $S_n$ is a key workload parameter. Neither does the increase in complexity of mean response time approximations affect the time to compute the expressions. We note, however, that for very large systems (say with $P > 1000$) the mean response time expression for $\overline{R}_{EQS}(N = k)$ as given in (5.1) may have to be evaluated using special rearrangements since it involves factorials. We have not encountered any problem in evaluating (5.1) for $P = 500$, but for $P = 1000$ a straightforward evaluation of the expression caused numerical overflow. Special purpose rearrangements to avoid problems with factorials is outside the scope of this thesis and we will not dwell further on this issue.

# Chapter 6

# Analysis of ASP, FCFS, and PSAPF

The goal is to solve for the performance of the FCFS, PSAPF, and ASP policies under general distributions of demand $\mathcal{F}_D^u$ and available parallelism $\mathcal{F}_N$, arbitrary correlation coefficient $r$, and any general ERF $\gamma$, as was done for the EQS policy in Chapter 5. However, the FCFS, ASP, and PSAPF policies are difficult to analyze under such completely general workload assumptions and therefore suitable restrictions are made below for the sake of analytic tractability. Fortunately, the restrictions do not limit the applicability of the policy comparison results, because as will be shown in Chapter 7, one can extrapolate from the comparisons under the restricted assumptions to the general case.

In this chapter we review the interpolation approximation from Chapter 4 for $\overline{R}_{FCFS}$ under $(\mathcal{F}_N,\ \mathcal{F}_D,\ r = 0,\ \gamma^l)$ and derive new interpolation approximations or reductions for the following policies and workloads assuming an arrival rate of $\lambda$ and $E(j) = \gamma(j)$:

| | |
|---|---|
| FCFS: | $(N = k,\ \mathcal{F}_D,\ r = 0,\ \gamma)$, |
| ASP: | $(\mathcal{F}_N,\ \exp(1/\overline{D}),\ r = 0,\ \gamma^l)$, |
| ASP: | $(N = P,\ \exp(1/\overline{D}),\ r = 0,\ \gamma^l)$, |
| PSAPF: | $(N = k,\ \mathcal{F}_D,\ r = 0,\ \gamma)$, |
| PSAPF: | $(\mathcal{F}_N,\ \mathcal{F}_D,\ r = 0,\ \gamma^l)$, and |
| PSAPF: | $(\mathcal{F}_N,\ \mathcal{F}_D^u,\ r > 0,\ \gamma^l)$. |

As will be shown in Chapter 7, the restrictive assumptions $(r = 0, \gamma^l)$ for FCFS under general $\mathcal{F}_N$ provides the best case performance of FCFS relative to PSAPF. Likewise, the assumptions $(\exp(1/\overline{D}), r = 0, \gamma^l)$ are favorable for ASP relative to EQS, and the assumption $(\gamma^l)$ will be shown favorable for PSAPF relative to EQS. Chapter 7 will show that conclusions from these "favorable" comparisons can be extrapolated to other regions of the general workload parameter space. Regarding absolute policy performance, mean response time estimates for FCFS and PSAPF under $(N = k, r = 0, \gamma)$ provide insight about how the performance of each of these policies behaves with respect to ERF sublinearity.

In Section 6.1 we provide the approximations for $\overline{R}_{FCFS}$. We then develop the (interpolation) approximations for $\overline{R}_{ASP}$ (Section 6.2), and for $\overline{R}_{PSAPF}$ under no correlation between mean demand and available parallelism (Section 6.3) and under full correlation (Section 6.3.4). The new approximations for $\overline{R}_{ASP}$ and $\overline{R}_{PSAPF}$ are validated in Section 6.4.

## 6.1 FCFS

We first review the interpolation approximation on $\underline{p}$ for $\overline{R}_{FCFS}$ from Chapter 4 that holds for $(\mathcal{F}_N, \mathcal{F}_D, r = 0, \gamma^l)$. We then develop a new approximation for $\overline{R}_{FCFS}$ under constant available parallelism and general $\gamma$.

### 6.1.1 Analysis under General N: $r = 0$ and $\gamma^l$

For the system $(FCFS, \mathcal{F}_N, \mathcal{F}_D, r = 0, \gamma^l))$ the following interpolation approximation on the pmf of $N$, $\underline{p}$, is an accurate estimator for $\overline{R}_{FCFS}$ (see Chapter 4)

$$\begin{aligned} \overline{R}_{FCFS}(\mathcal{F}_N, r = 0) &\approx \sum_{k=1}^{P} p_k \overline{R}_{FCFS}(N = k, r = 0), \quad \text{under } (\cdot, \mathcal{F}_D^u, \cdot, \gamma^l) \\ &= \overline{S} + \frac{E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}}\right]}{1-\rho}\left(\frac{1+C_D^2}{2\lambda}\right), \end{aligned} \tag{6.1}$$

where the solution for $\overline{R}_{FCFS}(N = k, r = 0)$ is derived by reducing the system to the M/G/c queue, under similar reasoning to the reduction under the more general assumptions of $(N = k, r = 0, \gamma)$, given next.

### 6.1.2 Analysis under Constant N

Let $\Gamma_{FCFS,k} = (FCFS, N = k, \mathcal{F}_D, r = 0, \gamma)$. In Chapter 4 mean response time estimates were provided for this system under the assumption that $\gamma = \gamma^l$. The extension to general nondecreasing $\gamma$ is straightforward and uses the following reduction.

First consider the case where $k$ evenly divides P. A job arriving at an empty system is allocated $k$ processors. Subsequent jobs that arrive are also allocated $k$ 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 obtains all $k$ processors released by the departing job, and so on. Since processors are allocated and released in units of size $k$, the system $\Gamma_{FCFS,k}$ behaves like a system with $c = P/k$ processors in which each job has one task with service requirement $S = D/\gamma(k)$. That is, under $(N = k, \mathcal{F}_D, r = 0, \gamma)$

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

To compute $\overline{R}_{M/G/c}$ we use the following approximation which is derived using Sakasegawa's approximation [64] for the mean number in a GI/G/c queue:

$$\overline{R}_{M/G/c} \approx \overline{S} + \frac{\nu^{\sqrt{2(c+1)}}}{1-\nu}\left(\frac{1+C_S^2}{2\lambda}\right), \quad \text{where} \quad \nu = \frac{\lambda \overline{S}}{c},$$

$C_S$ being the coefficient of variation in job service time, $S$. Using $S = D/\gamma(k)$, we obtain $C_S^2 = C_D^2$ and thus

$$\overline{R}_{FCFS}(N = k, r = 0) \approx \frac{\overline{D}}{\gamma(k)} + \frac{\nu^{\sqrt{2(P/k+1)}}}{1-\nu}\left(\frac{1+C_D^2}{2\lambda}\right), \quad \text{under } (\cdot, \mathcal{F}_D^u, \cdot, \gamma) \qquad (6.2)$$

where $\nu = \dfrac{\lambda \overline{D}}{P} \cdot \dfrac{k}{\gamma(k)}$. Since (6.2) can also be computed when $k$ does not evenly divide P, it can be used as an approximation for $\overline{R}_{FCFS}(N = k)$ for all $k = 1, 2, \ldots, P$.

It is tempting to believe that the interpolation on $\underline{p}$ using estimates from (6.2) as the interpolation end-points can be used to approximate $\overline{R}_{FCFS}$ for a sublinear ERF $\gamma$. Validations so far have shown that this approximation is accurate for low values of $\rho$, but that the accuracy degrades with $\rho$ and at high load the accuracy can be quite bad even when $C_D = 1$ (a case under which the approximation is very accurate for the linear ERF). With some "fine tuning" it may be possible to obtain an accurate estimator by this approach, but this is not pursued further in this thesis.

Note that both approximations (6.1) and (6.2) show that $\overline{R}_{FCFS}$ increases linearly with $C_D^2$.

## 6.2 ASP

We develop an approximation for $\overline{R}_{ASP}$ under $(\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l)$. Setia and Tripathi [69] derive an exact solution for $\overline{R}_{ASP}$ under exponential per class job demands and general job execution rates, which is based on matrix-geometric analysis [56, 51]. Two drawbacks of this exact analysis are that the underlying state space grows exponentially in the number of processors (making the analysis computationally prohibitive even for systems with 20 processors) and that the analysis does not yield direct insight into the dependence of $\overline{R}_{ASP}$ on workload parameters.

In contrast, we derive a closed form approximation for $\overline{R}_{ASP}$ under the restrictive assumptions of linear execution rates and exponential demands. The assumption of linear execution rates yields estimates of the best possible performance of ASP (i.e., under no synchronization and communication overheads). The exponential demand assumption should also result in lower estimates for $\overline{R}_{ASP}$ than for workloads with high $C_D$, since ASP is a static allocation policy. This is discussed further in Chapter 7.

Section 6.2.1 presents an interpolation approximation under general $\mathcal{F}_N$ and Section 6.2.2 derives reductions and interpolation approximations for the extreme cases of constant $N$, i.e., $N = 1$ and $N = P$.

### 6.2.1 Analysis under General N: $\exp(1/\overline{D})$, $r = 0$, $\gamma^l$

To derive an approximation for $\overline{R}_{ASP}$ we note from the definition of ASP in Chapter 2 that at each allocation point ASP divides processors equally among waiting jobs (with no fewer than one processor per job). This resemblance to the EQS policy suggests using the same form of interpolation for $\overline{R}_{ASP}$ as we used for $\overline{R}_{EQS}$ in (5.9), that is, an interpolation on $S_n$. Thus we have the following interpolation approximation on $S_n$ for the mean response time of $(ASP, \mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l)$.

$$\begin{aligned}\overline{R}_{ASP}(\mathcal{F}_N) \quad \approx \quad & \left(\frac{S_n - 1/P}{1 - 1/P}\right)\overline{R}_{ASP}(N = 1) + \\ & \left(\frac{1 - S_n}{1 - 1/P}\right)\overline{R}_{ASP}(N = P), \qquad \text{under } (\cdot, \exp(1/\overline{D}), r = 0, \gamma^l). \qquad (6.3)\end{aligned}$$

Solutions for $\overline{R}_{ASP}(N = 1)$ and $\overline{R}_{ASP}(N = P)$ are given next.

### 6.2.2 Analysis for $N = 1$ and $N = P$: $\exp(1/\overline{D})$, $\gamma^l$

When $N = 1$, ASP is the same as FCFS. Therefore for exponential job demands, $\overline{R}_{ASP}(N = 1)$ is simply the mean response time in an $M/M/P$ queue, i.e.,

$$\overline{R}_{ASP}(N = 1) = \overline{R}_{M/M/P}, \quad (\cdot, \exp(1/\overline{D}), r = 0, \gamma^l).$$

Under nonexponential demands the extension is that

$$\overline{R}_{ASP}(N = 1) = \overline{R}_{M/G/P}, \quad (\cdot, \mathcal{F}_D, r = 0, \gamma). \qquad (6.4)$$

We do not have an exact solution for $\overline{R}_{ASP}(N = P)$ and develop an approximation for $\overline{R}_{ASP}(N = P)$ by observing the behavior of ASP at extreme ends of system utilization. When $\rho = 0$, $\overline{R}_{ASP}(N = P)$ is simply $\overline{S} = \overline{D}/P$ (since execution rates are linear). On the other hand when $\rho \rightarrow 1$ the queue length increases and a waiting job is allocated just one processor upon service (assuming that there are at least as many jobs as free processors). Therefore, for exponential job demands, as $\rho \rightarrow 1$ the system under ASP tends to behave like an M/M/P queue, i.e., $\overline{R}_{ASP} \rightarrow \overline{R}_{M/M/P}$ as $\rho \rightarrow 1$. More generally,

$$\lim_{\rho \rightarrow 1} \overline{R}_{ASP} \approx \overline{R}_{M/G/P}, \quad (\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma). \tag{6.5}$$

Combining these two estimates at extreme ends of $\rho$, we get the following approximation for $\overline{R}_{ASP}(N = P)$ when job demand is exponential.

$$\overline{R}_{ASP}(N = P) \approx (1 - \alpha(\rho))\frac{\overline{D}}{P} + \alpha(\rho)\overline{R}_{M/M/P}, \quad \text{under } (\cdot, \exp(1/\overline{D}), r = 0, \gamma^l) \tag{6.6}$$

where $\alpha(0) = 0$, $\alpha(1) = 1$, and $0 < \alpha(\rho) < 1$, for $0 < \rho < 1$.[1]

We empirically derived $\alpha(\rho)$ by comparing the right hand side of (6.6), for various choices of $\alpha(\rho)$, against simulation estimates of $\overline{R}_{ASP}(N = P)$ at P=10, 20, 50, and 100. We tried to bias the choice of $\alpha(\rho)$ so that it would be more accurate for P=50 and 100 as compared to P=10 and 20. Our empiric estimation proceeded as follows. We first note that it is likely for $\overline{R}_{ASP}$ to contain high powers of $\rho$ as seen from the approximation (6.4) for $\overline{R}_{ASP}(N = 1)$. We therefore started out with the simple form $\alpha(\rho) = \rho^c$, and we ranged $c$ from 1 to P. For all values of $c$ this choice of $\alpha(\rho)$ resulted in inaccurate estimates of $\overline{R}_{ASP}(N = P)$ when validated against simulation. As a result we next tried $\alpha(\rho) = a_1\rho^{c_1} + a_2\rho^{c_2}$, for $0 < a_1 < 1$, $a_1 + a_2 = 1$, and $1 \leq c_1, c_2 \leq P$. This approach did not yield accurate estimates at $P = 100$ motivating us to try out $\alpha(\rho) = a_1\rho^{c_1} + a_2\rho^{c_2} + a_3\rho^{c_3}$, for $0 < a_1, a_2, a_3 < 1$, $a_1 + a_2 + a_3 = 1$, and $1 \leq c_1, c_2, c_3 \leq P$. This last form of $\alpha(\rho)$ produced satisfactory estimates for $\overline{R}_{ASP}(N = P = 100)$ for certain choices of coefficients and powers of $\rho$ that were "fine tuned" to yield the following estimator which is accurate at lower values of $P$ as well.

$$\alpha(\rho) = \frac{2}{P}\rho + (0.5 - 2/P)\rho^s + 0.5\rho^{\lceil P/3 \rceil},$$

where

$$s = \begin{cases} 3.5 & P \leq 20, \\ 4.5 & P = 50, \\ 6.0 & P = 100. \end{cases}$$

Note that since approximation (6.3) will be shown to validate well in Section 6.4, $S_n$ is the key parameter for job parallelism under the given workload. That is, $\overline{R}_{ASP}$ is approximately the same for all distributions of $N$ that yield the same value for $S_n$.

The interpolation approximation approach in (6.3) and (6.6) also resulted in an accurate approximation when job demand is deterministic ($C_D = 0$) and $N$ has a general distribution,

[1] Note that the interpolation approximation on $\rho$ might also be applied for general $D$, $N$, and/or $\gamma$; however the function $\alpha(\rho)$ is difficult to derive in these cases.

$\mathcal{F}_N$. In this case the functional form of $\alpha(\rho)$ was less carefully constructed and the approximation also has so far been less extensively validated than the approximation for exponential demands ($C_D = 1$). (For the 50 data points validated for this approximation, 42 are with 15% of the simulation estimates and the maximum error is about 28%.) The following summarizes the approximation for $C_D = 0$:

$$\begin{aligned}\overline{R}_{ASP}(\mathcal{F}_N) \quad &\approx \quad \left(\frac{S_n - 1/P}{1 - 1/P}\right) \overline{R}_{ASP}(N = 1) + \\ &\qquad \left(\frac{1 - S_n}{1 - 1/P}\right) \overline{R}_{ASP}(N = P), \qquad \text{under } (\cdot, D = \overline{D}, r = 0, \gamma^l). \qquad (6.7)\end{aligned}$$

where

$$\overline{R}_{ASP}(N = 1) = \overline{R}_{M/D/P} \approx \overline{D} + \frac{\rho^{\sqrt{2(P+1)}}}{2(1 - \rho)\lambda},$$

and

$$\overline{R}_{ASP}(N = P) \approx (1 - \alpha(\rho))\frac{\overline{D}}{P} + \alpha(\rho)\overline{R}_{M/D/P},$$

$$\alpha(\rho) = \frac{1}{P}\rho + (0.5 - 1/P)\,\rho^{\sqrt{P/2}} + 0.5\,\rho^{P/2}.$$

## 6.3 PSAPF: $r = 0$

We first review the interpolation approximation on $\underline{p}$ from Chapter 4 for $\overline{R}_{PSAPF}$ under $(\mathcal{F}_N, \mathcal{F}_D, r = 0, \gamma^l)$, then derive a more accurate estimator under the same workload assumptions, and finally provide solutions for constant $N$ and general $\gamma$. Section 6.3.4 derives estimates for $\overline{R}_{PSAPF}$ for $r > 0$.

### 6.3.1 Review of Analysis for General N: $r = 0$, $\gamma^l$

The following interpolation approximation on the pmf of $N$ was shown in Chapter 4 to provide reasonably accurate estimates of $\overline{R}_{PSAPF}$ under $(\mathcal{F}_N, \mathcal{F}_D, r = 0, \gamma^l)$ and is noted to be the same as the interpolation approximation (6.1) for $\overline{R}_{FCFS}$:

$$\begin{aligned}\overline{R}_{PSAPF}(\mathcal{F}_N, r = 0) \quad &\approx \quad \sum_{k=1}^{P} p_k \overline{R}_{PSAPF}(N = k, r = 0), \qquad \text{under } (\cdot, \mathcal{F}_D^u, \cdot, \gamma^l) \\ &= \quad \overline{S} + \frac{E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}}\right]}{1 - \rho}\left(\frac{1 + C_D^2}{2\lambda}\right). \qquad (6.8)\end{aligned}$$

In many validations this approximation results in less than 35% errors from simulation estimates. However, it does not validate well in some cases with high $C_D$ and low $C_N$ (e.g., more than 100% relative errors have been observed), which motivates a more accurate approximation for

$\overline{R}_{PSAPF}$. We note that approximation (6.8) gives the "coarse" result that $\overline{R}_{PSAPF}(r=0) \approx \overline{R}_{FCFS}(r=0)$ under the given workload assumptions. The new approximation derived next will yield a more refined comparison.

### 6.3.2 More Accurate Estimator for General N: $r = 0$, $\gamma^l$

We derive a more accurate approximation for $\overline{R}_{PSAPF}$ by observing that PSAPF is essentially a Preemptive Resume (PR) priority scheduling policy. A known heuristic for obtaining performance estimates of PR for a multiserver system with sequential jobs is to compare PR with FCFS in a uniprocessor system and then map the comparison to the multiserver system (cf. [11, 5, 78]). For example, in [5], Buzen and Bondi approximated the mean extra time (i.e., mean response time minus mean service time) of an M/G/c PR queue by

$$\overline{X}_{M/G/c\ PR} \approx \frac{\overline{X}_{M/G/1_c\ PR}}{\overline{X}_{M/G/1_c\ FCFS}} \overline{X}_{M/G/c\ FCFS}, \tag{6.9}$$

where the $M/G/1_c\ PR$ queue is obtained by replacing all $c$ servers of the $M/G/c\ PR$ queue by a single server of power $c$. (Likewise for FCFS.) We use a similar heuristic to estimate the mean extra time of a parallel system under PSAPF, $\overline{X}_{PSAPF} \equiv \overline{R}_{PSAPF} - \overline{S}$, as

$$\overline{X}_{PSAPF} \approx \frac{\overline{X}_{M/G/1_P\ PR}}{\overline{X}_{M/G/1_P\ FCFS}} \overline{X}_{FCFS}, \tag{6.10}$$

where job priorities in the $M/G/1_P\ PR$ queue are the same as those in the PSAPF system (i.e., inversely proportional to available parallelism).

A closed form expression for $\overline{R}_{PSAPF}$ is derived by obtaining closed form expressions for each of $\overline{X}_{M/G/1_P\ FCFS}$, $\overline{X}_{FCFS}$, and $\overline{X}_{M/G/1_P\ PR}$ in (6.10). $\overline{X}_{M/G/1_P\ FCFS}$ is simply $\rho^2(1+C_D^2)/(2\lambda(1-\rho))$ [32], and approximation (6.1) yields a closed form expression for $\overline{X}_{FCFS} \equiv \overline{R}_{FCFS} - \overline{S}$. The analysis in [33] for an M/G/1 PR queue (under the given workload assumptions) yields,

$$\overline{X}_{M/G/1_P\ PR} = \sum_{k=1}^{P} p_k \left[\frac{\sigma_{k-1}}{1-\sigma_{k-1}} + \frac{\sigma_k}{(1-\sigma_{k-1})(1-\sigma_k)} \left(\frac{1+C_D^2}{2}\right)\right] \frac{\overline{D}}{P}, \quad \text{where} \quad \sigma_k = \rho \sum_{i=1}^{k} p_i. \tag{6.11}$$

Thus, under the assumptions $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$, we have the following closed form expression for $\overline{R}_{PSAPF} = \overline{S} + \overline{X}_{PSAPF}$:

$$\overline{R}_{PSAPF}(r=0) \approx \overline{S} + \left\{ \sum_{k=1}^{P} p_k \left[\frac{\sigma_{k-1}}{1-\sigma_{k-1}} + \frac{\sigma_k}{(1-\sigma_{k-1})(1-\sigma_k)} \left(\frac{1+C_D^2}{2}\right)\right] \frac{\overline{D}}{P} \right\} E\left[\rho^{\sqrt{2(\frac{P}{N}+1)}-2}\right]. \tag{6.12}$$

Note that the accuracy of approximation (6.10) can be improved if we use a more accurate approximation for $\overline{X}_{FCFS}$; however, the use of numerical analysis entails significant loss of insight.

### 6.3.3 Analysis under Constant N

When available parallelism is constant, i.e., $N = k$, PSAPF is identical to FCFS and approximation (6.2) is valid for the system $(PSAPF, \lambda, N = k, \mathcal{F}_D, r = 0, \gamma)$ as well. We therefore have,

$$\overline{R}_{PSAPF}(N = k, r = 0) \approx \frac{\overline{D}}{\gamma(k)} + \frac{\nu\sqrt{2(P/k+1)}}{1-\nu}\left(\frac{1+C_D^2}{2\lambda}\right), \quad \text{under } (\cdot, \mathcal{F}_D, \cdot, \gamma) \tag{6.13}$$

where $\nu = \dfrac{\lambda\overline{D}}{P} \cdot \dfrac{k}{\gamma(k)}$.

Note that as in the case of FCFS, approximations (6.12) and (6.13) for $\overline{R}_{PSAPF}$ increase linearly in $C_D^2$.

### 6.3.4 PSAPF: $r > 0$

The estimate $\overline{R}_{PSAPF}$ is first derived for fully correlated workloads ($r = 1$) and then combined with approximation (6.12) for uncorrelated workloads ($r = 0$) to yield an estimate for arbitrary partial correlation ($0 < r < 1$).

### 6.3.5 Analysis for $r = 1$: $\gamma^l$

The approximation for $\overline{R}_{PSAPF}$ under ($\mathcal{F}_N$, $\mathcal{F}_D^u$, r=1, $\gamma^l$) is derived by: (1) classifying jobs according to their available parallelism, (2) computing the mean response time for each class of jobs by approximating the *average interference* from other classes of jobs, and (3) computing the overall mean response time as a weighted sum of the approximate mean response times per class. This general approach yields very accurate estimates of $\overline{R}_{EQS}$ under the given workload conditions (see Chapter 5). In the case of PSAPF the particular approximate representation of average interference by other job classes yields a system for each class that reduces to a preemptive resume queue, from which the class mean response time is computed.[2]

Let a job with available parallelism $k$ belong to class $C_k$, for $k = 1, \ldots, P$. Let $\overline{R}_{PSAPF,C_k}$ denote the mean response time of class $C_k$ in the system $(PSAPF, \mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$. Clearly,

$$\overline{R}_{PSAPF} = \sum_{k=1}^{P} p_k \overline{R}_{PSAPF,C_k}. \tag{6.14}$$

[2] This general approach validates well not only for $r = 1$ but also for $0 \leq r < 1$. However, the separate approximations for $\overline{R}_{PSAPF}(r = 0)$ and $\overline{R}_{PSAPF}(r)$, $0 < r < 1$, yield more insight.

The approximate processor contention from classes other than $C_k$ is modeled by assuming each such class has available parallelism $k$, but retains its total service requirements and job priority as before. More precisely, we approximate $\overline{R}_{PSAPF,C_k}$ to be the mean response time of class $C_k$ in a system $\Gamma_k$ which is like the original system except that a class $C_j$ job in $\Gamma_k$ has demand $D_j$, priority $j$, and available parallelism $k$, where $\overline{D}_j = \frac{\overline{D}}{\overline{N}} \cdot j$, as per the correlation model in Chapter 3. The instantaneous load of class $C_j$ jobs is not accurately modeled by assuming that class $C_j$ jobs have parallelism $k$. However, the priority and average load of class $C_j$ jobs are accurately modeled. Thus, the overall interference of $C_j$ with $C_k$ may be reasonably well represented.

An approximation for $\overline{R}_{PSAPF,C_k}$ is derived by solving for the mean response time of class $k$ in system $\Gamma_k$. Since jobs from classes $k+1$ to P have lower priority than $k$ it is only necessary to consider arrivals from classes 1 through $k$ to obtain $\overline{R}_{\Gamma_k,C_k}$. Recall that in $\Gamma_k$ all jobs have an available parallelism of $k$. First assume that $k$ evenly divides $P$. Thus processors are allocated or preempted in units of $k$ at a time. If processors are grouped $k$ at a time and each such cluster is thought of as a superprocessor, then we realize that $\Gamma_k$ essentially functions as an $M/G/c$ PR queue with $c = P/k$ servers each of power $k$, and with $k$ priority classes. Therefore, $\overline{R}_{\Gamma_k,C_k}$ is equal to the mean response time of the $k^{th}$ priority class in this $M/G/c\ PR$ queue. Tabetaeoul and Kouvatsos [78] derive an approximation for per class mean response times of a $GI/G/c\ PR$ queue using a heuristic similar to (6.9). Using their heuristic we obtain the following expression for $\overline{R}_{\Gamma_k,C_k}$, which is derived in Appendix A.2.

$$\overline{R}_{\Gamma_k,C_k} \approx c\,\overline{x}_k + \frac{1}{p_k}\left(\sum_{i=1}^{k-1} g_i\right)\left(\sigma_k^{\sqrt{2(c+1)}-2} - \sigma_{k-1}^{\sqrt{2(c+1)}-2}\right) + \frac{1}{p_k}\, g_k \sigma_k^{\sqrt{2(c+1)}-2}, \quad p_k > 0, \tag{6.15}$$

where

$$g_i = p_i \frac{\sigma_{i-1}}{1-\sigma_{i-1}}\overline{x}_i + \frac{\lambda p_i \sum_{j=1}^{i}\{p_j \overline{x}_j^2\}}{(1-\sigma_{i-1})(1-\sigma_i)}\left(\frac{1+C_v^2}{2}\right),$$

$$\overline{x}_i = \frac{\overline{D}i}{\overline{N}P}, \quad \text{and} \quad \sigma_i = \lambda \sum_{j=1}^{i} p_i \overline{x}_i, \quad i = 1, \ldots, k.$$

We use the approximation (6.15) even when $k$ does not evenly divide P and thus obtain

$$\overline{R}_{PSAPF} \approx \sum_{k=1}^{P} p_k\, h(k, \lambda, (p_1, \ldots, p_k), (\overline{D}, C_v)), \qquad \text{under } (\mathcal{F}_N, \mathcal{F}_D^u, r = 1, \gamma^l), \tag{6.16}$$

where $h(k, \lambda, (p_1, \ldots, p_k), (\overline{D}, C_v))$ is given by the RHS of (6.15). We note from (6.15) that $\overline{R}_{PSAPF}$ grows linearly in the squared coefficient of variation of demand, $C_v$, of each job class when $r = 1$ and $\gamma = \gamma^l$.

### 6.3.6 Analysis for $0 < r < 1$: $\gamma^l$

We have thus far obtained estimates for $\overline{R}_{PSAPF}(r = 0)$ and $\overline{R}_{PSAPF}(r = 1)$. To estimate $\overline{R}_{PSAPF}$ for a general $r$ between 0 and 1, consider an interpolation approximation on $r$. That is,

$$\overline{R}_{PSAPF}(r) \approx (1 - f(r))\,\overline{R}_{PSAPF}(r = 0) + f(r)\,\overline{R}_{PSAPF}(r = 1),$$

where $f(r)$ is a suitable function of $r$. We note from (3.6) that at $\rho \to 0$, $\overline{R}_{PSAPF} = \overline{S} = (1 - r^2)\overline{S}(r = 0) + r^2\overline{S}(r = 1)$ and we therefore obtain at $\rho \to 0$ that $f(r) = r^2$. We found that for $\rho > 0$ this choice of $f(r)$ continues to yield accurate estimates for $\overline{R}_{PSAPF}$. Therefore, we have,

$$\overline{R}_{PSAPF}(r) \approx (1 - r^2)\overline{R}_{PSAPF}(r = 0) + r^2\overline{R}_{PSAPF}(r = 1), \quad \text{under } (\mathcal{F}_N, \mathcal{F}_D^u, \cdot, \gamma^l), \quad (6.17)$$

where we estimate $\overline{R}_{PSAPF}(r = 0)$ and $\overline{R}_{PSAPF}(r = 1)$ using approximations (6.12) and (6.16), respectively.

Note that the form of approximation (6.17) is identical to (5.10), which was proposed for the EQS policy. This will prove useful in comparing the performance of the EQS and PSAPF policies in the range $0 < r < 1$.

## 6.4 Validations of Approximations for $\overline{R}_{ASP}$ and $\overline{R}_{PSAPF}$

In this section we validate the approximations for $\overline{R}_{ASP}$ and $\overline{R}_{PSAPF}$ derived in Sections 6.2, 6.3, and 6.3.4. The parameter settings for the validations are as follows:

- For most of validations P=20 or P=100 processors.[3]

- We used three different distributions for available parallelism $N$. First, the bounded-geometric distribution. Second, a uniform distribution with several values for the lower and upper limits. Third, constant $N$, i.e., $N = k$.

- For validating the ASP approximations (6.6) and (6.3) we used an exponential distribution for demand, and for validating the PSAPF approximations we used exponential ($C_v = 1$) as well as two-stage hyperexponential ($H_2$) demands with $C_v = 5$. In a few cases we also validated the PSAPF approximations for deterministic and Gamma distributions of demand. The accuracy of the approximations for deterministic demands was nearly the same as the accuracy for exponential demands and for the Gamma distribution the

[3] In some cases we considered systems with 10 or 50 processors and in some other cases systems with 500 or 1000 processors. The accuracy of the approximations was approximately the same in these cases as the accuracy for 20 or 100 processors.

accuracy was the same as for $H_2$ demands with the same $C_v$. We also ran a few test cases for $C_v < 5$ and noted that the accuracy of the PSAPF approximations was higher at lower $C_v$.

- For all validations $\overline{D}$ was set to $P$ so that $\rho \equiv \lambda\overline{D}/P = \lambda$. In the validations $\rho$ was varied from 0.1 to 0.9.

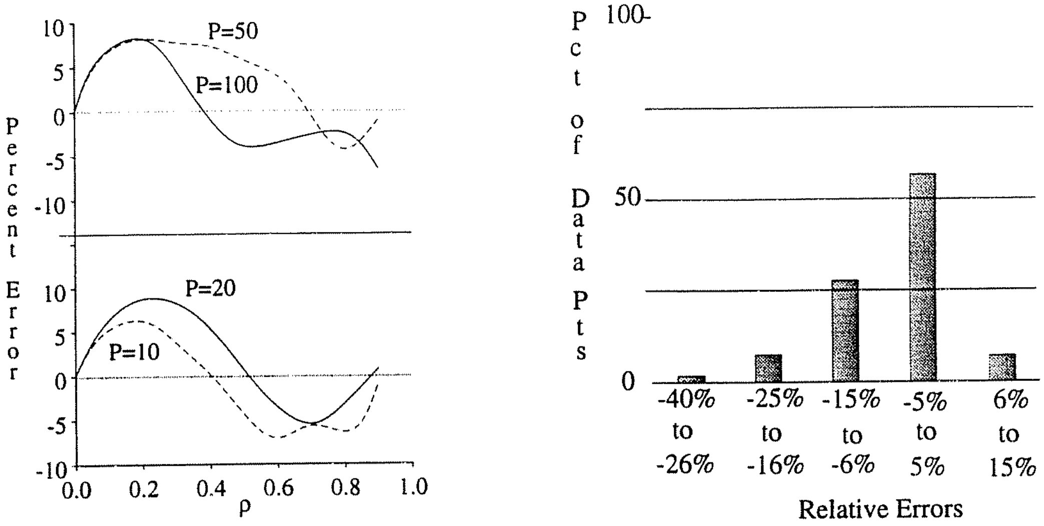
The approximations for $\overline{R}_{PSAPF}$ for constant available parallelism were validated using exact matrix-geometric analysis [56, 51]. In all other cases, our approximations were validated using discrete event simulation. All simulation estimates of mean response time had 95% confidence intervals with less than 10% half-widths, and in nearly all cases the half-widths were less than 5%. The batch means method was used if obtaining the regenerative cycles was too time consuming.

### 6.4.1 ASP Validations

Figure 6.1a depicts the relative errors for approximation (6.6) for systems with 10, 20, 50, and 100 processors. Observe that for all four system sizes approximation (6.6) overestimates $\overline{R}_{ASP}(N = P)$ at low utilizations, but underestimates $\overline{R}_{ASP}(N = P)$ at moderate to high utilizations. In all cases the relative errors are less than 10% in magnitude. We expect approximation (6.3) to have higher relative errors since it uses approximation (6.6). To validate approximation (6.3) we ran simulation experiments for many bounded-geometric distributions for $N$ with different values of $P_{max}$ and $p$, and for uniform and constant distributions for $N$. The total number of data points in the validations (excluding the points for $N = P$) was about 140 for systems with 20 and 100 processors. Figure 6.1b summarizes these validation results by plotting histograms of relative error. The figure shows that approximation (6.3) is very accurate. For more than 90% of the data points the approximation is within 15% of the simulation estimates. The highest error (-36.4%) occurred for a U[50,100] distribution for N at $\rho = 0.3$. In general, we noticed that the errors were larger and more negative for workloads with high average available parallelism (say $\overline{N} > 3P/4$) when load was low to moderate (around $\rho = 0.5$).

### 6.4.2 PSAPF Validations

We present validations for approximations (6.12), (6.16), and (6.17). For approximation (6.12), and $r = 0$, the number of data points for each of $C_v = 1$ and $C_v = 5$ was 306 leading to a total of 612 validations. Figure 6.2a presents the histograms of relative errors for approximation (6.12). We see that approximation (6.12) is extremely accurate when $C_v = 1$, and reasonably accurate at $C_v = 5$ (about 90% of the data points have less than 35% error at $C_v = 5$). This approximation is more accurate than the PSAPF approximations in Chapter 4, which had more than 100%



(a) Approximation for N=P

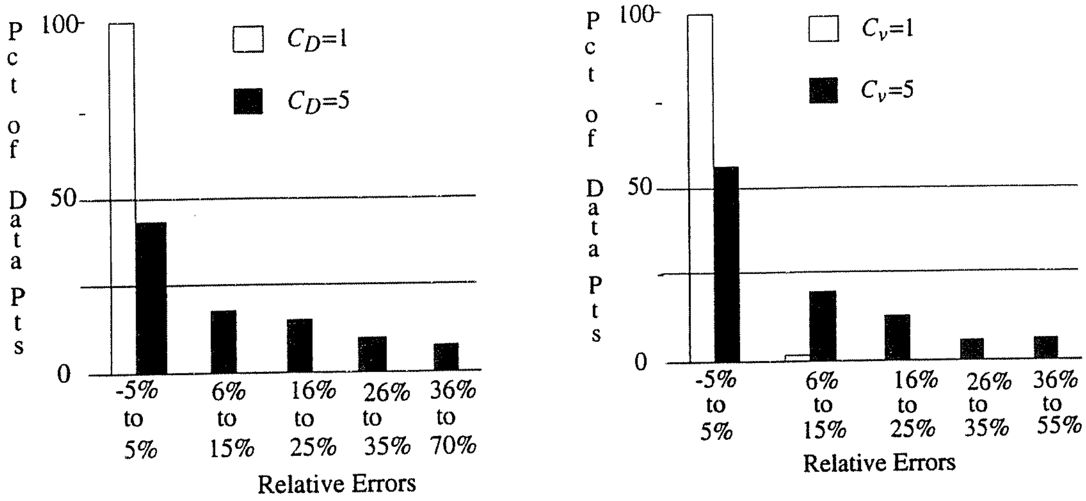
(b) Approximation for general N

Figure 6.1: Validations of ASP Approximations

error in some cases. The maximum error at $C_v = 5$ occurred at $N = 3/4P$ and $\rho = 0.2$ for both P=20 and P=100. In general, the largest errors at $C_v = 5$ were observed for distributions of $N$ with moderate to high $\overline{N}$ and low $C_N$. The overall accuracy might improve further if a more accurate approximation for $\overline{X}_{FCFS}$ is used in approximation (6.12).

Approximation (6.16) was validated against 178 data points for each of $C_v = 1$ and $C_v = 5$, leading to a total of 356 validations. (Since the constant N distribution cannot be used when $r = 1$ the total number of validations is fewer than when for $r = 0$.) Figure 6.2b summarizes the validations for this approximation. As seen from the figure approximation (6.16) is very accurate at low $C_v$ and quite accurate at high $C_v$ (about 95% of the data points have less than 35% error when $C_v = 5$). The maximum error at $C_v = 5$ occurred for the data point N=U[1,100], $\rho = 0.5$. The approximation errors when $r = 1$ were highest for low $C_N$ workloads at low to moderate load.

For approximation (6.17) the validations consist of 226 data points for each of $C_v = 1$ and $C_v = 5$, leading to a total of 452 validations (excluding the cases for $r = 0$ and $r = 1$). We considered three values of $r$, viz., $r = 0.25$, $r = 0.5$, and $r = 0.75$. Figure 6.3 displays histograms of the relative error at $C_v = 1$ and $C_v = 5$. The accuracy of approximation (6.17) is very high at $C_v = 1$ and fairly good at $C_v = 5$. Thus all three approximations for PSAPF (i.e., for r=0, r=1, and for general r) are reasonably accurate in general, as long as $C_v \leq 5$. The maximum error for approximation (6.17) was encountered for a specific bounded-geometric distribution for



(a) Approximation for r=0

(b) Approximation for r=1

Figure 6.2: Relative Error Histograms for PSAPF Approximations

N with low $C_N$, $C_v = 5$, $r = 0.5$, and $\rho = 0.8$. As for $r = 0$ and $r = 1$, approximation (6.17) is more accurate for distributions of $N$ with high $C_N$.

## 6.5 Summary of Analysis and Relation to Previous Work

In this chapter we have developed analytic models for the ASP, FCFS, and PSAPF policies. We derived reductions for each policy under a general ERF and showed that their mean response times are sensitive to $C_D$, unlike the mean response time of the EQS policy. We noted from the reductions that assuming the linear ERF provides the same sensitivity results to $C_D$ as assuming a general ERF. All interpolation approximations in this chapter are valid for a general distribution of available parallelism and the linear ERF. We derived an interpolation on $S_n$ for $\overline{R}_{ASP}$ for an uncorrelated workload with exponential demand. We derived a more accurate approximation for $\overline{R}_{PSAPF}$ under uncorrelated workloads than the interpolation approximations for $\overline{R}_{PSAPF}$ from Chapter 4. We also developed an accurate approximation for $\overline{R}_{PSAPF}$ under fully correlated workloads with general demands, and used the the approximations at the extreme ends of correlation to obtain an interpolation on $r$ for $\overline{R}_{PSAPF}$ that holds for arbitrary workload correlation, i.e., $0 \leq r \leq 1$.

Previous analytic models for ASP have appeared in [69, 70] and for FCFS in [54, 55, 80, 50, 40]. There have been no analytic models for PSAPF but there is one analytic model in

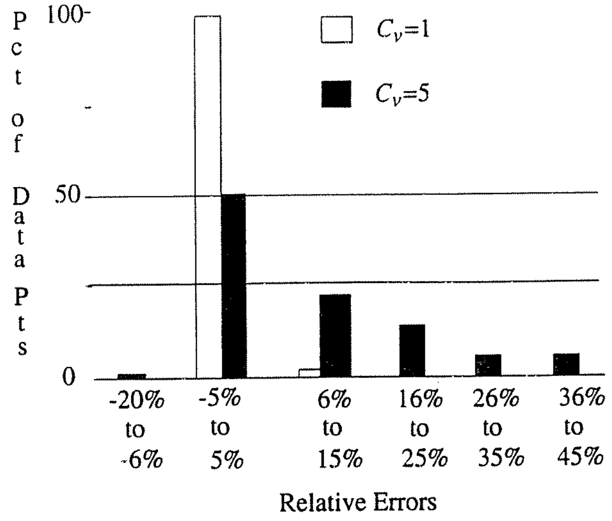


Figure 6.3: Relative Error Histograms for PSAPF Interpolation on $r$

$r = 0.25, \quad 0.5, \quad 0.75$

the literature for PSNPF [40]. In [50] an approximation is derived for $\overline{R}_{FCFS}$ under the i.i.d. exponential task service time model using a generalized version of Amdahl's law. Other than this approximation all the analytic models for FCFS, ASP, and PSNPF in the literature are based on numeric solution techniques such as matrix-geometric analysis or recurrence relations. None of these models shows how policy performance varies as a function of coefficient of variation in demand. Moreover, the models based on matrix-geometric relations or recurrence relations have computational limitations since the state space grows exponentially in the number of processors. Three simulation studies [43, 41, 39] give experimental data for the behavior of FCFS and PSNPF as a function of demand and parallelism parameters. All three studies show that $\overline{R}_{FCFS}$ and $\overline{R}_{PSNPF}$ increase with $C_D$ for specific two stage hyperexponential distributions but do not examine whether they increase linearly with $C_D^2$ as done in this chapter. In [39] it is shown for a specific distribution of demand and parallelism that $\overline{R}_{FCFS}$ and $\overline{R}_{PSNPF}$ can increase with $\overline{N}$ if $C_D$ is high. We corroborate this result in the next chapter.

This chapter and the previous one have focused on developing analytic models for ASP, FCFS, EQS, and PSAPF. Table 6.1 summarizes the model solutions derived in this chapter and in Chapters 4 and 5. We have observed from all the analytic models that a key workload parameter that influences the mean response time of these policies is coefficient of variation, $C_D$, in demand. We will therefore use this parameter to explore the design space while comparing policies in the next chapter. For the EQS policy under general workload assumptions and for the

ASP policy under $D \sim \exp$, $r = 0$ and $\gamma^l$ we have noted that the normalized mean service time, $S_n \equiv \overline{S}/\overline{D}$, is a key determinant of policy performance, which will prove useful in our policy comparison study in the next chapter. This shows the utility of the interpolation approximation approach in obtaining key parameters for policy comparison, besides being an effective approach for easily evaluating policy performance for large systems.

Table 6.1: Summary of Model Solutions

| Policy | Reductions | | Approximations | |
|---|---|---|---|---|
| ASP | $(N = 1, \mathcal{F}_D, r = 0, \gamma)$ | (6.4) | Interpolation on $\rho$: $(N = P, \exp(1/\overline{D}), r = 0, \gamma^l)$ | (6.6) |
| | $(\rho \rightarrow 1, \mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ | (6.5) | Interpolation on $S_n$: $(\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l)$ | (6.3) |
| EQS | $(N = k, \mathcal{F}_D, r = 0, \gamma)$ | (5.1) | Interpolation on $\underline{\rho}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma)$ | (5.5) |
| | $(\rho \rightarrow 1, \mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ | (5.4) | Interpolation on $\overline{N}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma)$ | (5.6) |
| | | | Interpolation on $\underline{p}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma)$ | (5.7) |
| | | | Interpolation on $\overline{S}_n$: $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ | (5.9) |
| | | | Interpolation on $r$: $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ | (5.9) |
| | | | Generalized approximation: $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$ | (5.12) |
| FCFS | $(N = k, \mathcal{F}_D, r = 0, \gamma)$ | (6.2) | Interpolation on $\overline{N}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$ | (4.12) |
| | | | Interpolation on $\underline{p}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$ | (4.14) |
| PSAPF | $(N = k, \mathcal{F}_D, r = 0, \gamma)$ | (6.13) | Interpolation on $\overline{N}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$ | (4.17) |
| | | | Interpolation on $\underline{p}$: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$ | (4.19) |
| | | | PR heuristic: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l)$ | (6.12) |
| | | | Generalized approximation: $(\mathcal{F}_N, \mathcal{F}_D^u, r = 1, \gamma^l)$ | (6.16) |
| | | | Interpolation on $r$: $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma^l)$ | (6.17) |

# Chapter 7

# Policy Comparison Results

The main goal of this thesis is to study policy performance with respect to workload parameters and to determine which scheduling policy out of ASP, EQS, FCFS, and PSAPF has the highest performance over most of the design space. Our analytic models in Chapters 5 and 6 reveal the dependence of policy performance on workload parameters and clearly show that a key parameter that affects the relative performance of these four policies is the coefficient of variation, $C_D$, in job demand. It remains to quantitatively determine the relative performance of these policies with respect to $C_D$ and also other workload parameters in cases where $C_D$ does not uniquely determine relative policy performance.

In the first part of this chapter we compare ASP, EQS, FCFS, and PSAPF using our approximate analytic models (as well as simulation in some cases) and we delineate regions of the design space over which each policy performs best. In the second part of this chapter we qualitatively corroborate the results from the first part using exact analysis under job dependent ERFs and more general correlation between demand and parallelism. In the second part we assume a generalized exponential (GE) distribution for demand, which is completely parameterized by the mean and coefficient of variation of demand. Section 7.3 shows how the performance comparison results in this chapter generalize and unify previous work. Finally, in Section 7.4 we summarize our policy comparison results and relate them to previous work.

## 7.1 Policy Comparison Using Interpolation Approximations

The goal of this section is to compare the performance of ASP, EQS, FCFS, and PSAPF under the general workload assumptions $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$. The mean response times of ASP, FCFS, and

PSAPF were each derived under one or more restrictive assumptions, i.e., linear execution rates for all three policies, no correlation for ASP and FCFS, and exponential demands for ASP. However, if it turns out that the restrictive assumptions are more favorable to one policy, $\Psi_1$, over another, $\Psi_2$, and yet $\Psi_1$ performs worse, then the same relative ordering between $\Psi_1$ and $\Psi_2$ will hold under more general conditions that are less favorable to $\Psi_1$. In this way we will be able to generalize the results from comparisons of the four policies under the restrictive assumptions.

The following theorem will prove useful in understanding the impact of execution rate assumptions on policy comparisons. This theorem shows that for any fixed set of jobs with a common workload ERF, $\gamma$, the total execution rate of all jobs (or equivalently the processor efficiency) is maximum for the EQS policy.

**Theorem 7.1.1** *Consider a set of $K$ jobs with available parallelisms $(n_1, \ldots, n_K)$. Let $\Psi$ be a processor allocation policy that allocates $a_i^\Psi$ processors to job $i$, for $i = 1, \ldots, K$. Then for a workload ERF $\gamma$ that is concave and nondecreasing, and for $E(j) = \gamma(j)$, i.e., jobs dynamically and efficiently redistribute their work,*

$$\sum_{i=1}^{K} E(a_i^{EQS}) \geq \sum_{i=1}^{K} E(a_i^{\Psi}), \quad \textit{for any processor allocation policy } \Psi. \tag{7.1}$$

**Proof.** See Appendix A.3.1. ∎

**Remark:** An extension to Theorem 7.1.1 is that the available parallelisms can be random variables $(N_1, \ldots, N_K)$, in which case one must take the expected value of the sums in (7.1).

The intuition behind the result is that when $\gamma$ is concave the total execution rate decreases with variability in allocation. EQS tends to allocate an equal fraction of processors to jobs and this leads to high overall efficiency. As per the theorem, the assumption of the linear ERF is more favorable to the ASP, FCFS, and PSAPF policies as compared to EQS. We discuss the favorability of other workload parameter settings as they arise in the comparisons below.

This section first compares ASP and EQS and shows that EQS performs as well or better than ASP for essentially the entire parameter space. We then compare FCFS and PSAPF and show that PSAPF outperforms FCFS for most of the parameter space. Section 7.1.3 compares EQS and PSAPF and delineates the regions under which each policy performs best. Note that all experiments in this section have $\overline{D}$ set to $P$ so that $\rho \equiv \lambda\overline{D}/P = \lambda$.

### 7.1.1 ASP versus EQS

Section 7.1.1.1 compares the performance of ASP and EQS for uncorrelated workloads first using approximations (6.3) and (5.9) under the assumptions of exponential job demands and the linear ERF $\gamma^l$. Then the performance of the two policies is compared for $C_D = 0$ using approximation (6.7) for $\overline{R}_{ASP}$, and for $C_D > 1$ using simulation for $\overline{R}_{ASP}$. (Note that approximation (5.9) for $\overline{R}_{EQS}$ is independent of $C_D$.) In all cases, the linear ERF is most favorable to the ASP policy, which allows extrapolation of the policy comparisons to sublinear ERFs. Section 7.1.1.2 uses simulation for $\overline{R}_{ASP}$ to compare the performance of ASP and EQS for correlated workloads.

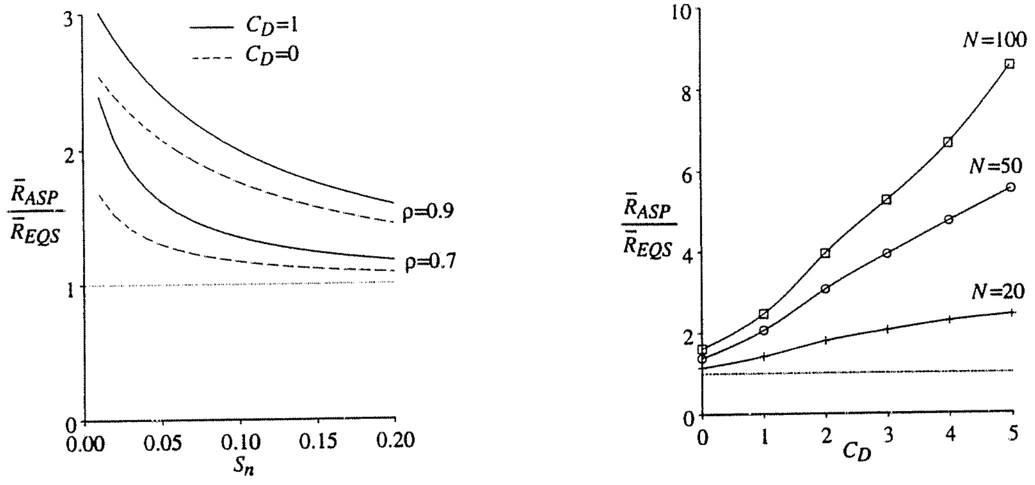
#### 7.1.1.1 ASP versus EQS: r=0

Key to the comparison of $\overline{R}_{ASP}$ and $\overline{R}_{EQS}$ under the assumptions $(\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l)$ are the following observations from (5.9) and (6.3). First, $S_n$, $\overline{D}$, and $\rho$ are the key determinants of $\overline{R}_{ASP}$ and $\overline{R}_{EQS}$ under the given assumptions. Second, for fixed $S_n$ and $\rho$ the ratio $\overline{R}_{ASP}/\overline{R}_{EQS}$ is insensitive to $\overline{D}$ because each of the formulas in (5.9) and (6.3) is directly proportional to $\overline{D}$. Therefore, if we keep $\overline{D}$ fixed and plot the ratio $\overline{R}_{ASP}/\overline{R}_{EQS}$ against $S_n$ for different values of $\rho$, the results will hold for all $\overline{D}$ and all distributions of $N$ under $(\exp(1/\overline{D}), r = 0, \gamma^l)$

Consider the maximum value of $S_n$, i.e, $S_n = 1$, in which case all jobs are fully sequential. When $N = 1$, the EQS system is identical to an M/M/P processor sharing (PS) system and thus $\overline{R}_{EQS}(N = 1) = \overline{R}_{M/M/P\ PS}$. From [65] we obtain $\overline{R}_{M/M/P\ PS} = \overline{R}_{M/M/P\ FCFS}$. On the other hand ASP is identical to FCFS when $N = 1$ and thus for exponential demands $\overline{R}_{ASP}(N = 1) = \overline{R}_{M/M/P\ FCFS}$. Therefore, for $C_D = 1$, $r = 0$, and $S_n = 1$, $\overline{R}_{ASP} = \overline{R}_{EQS}$.

Next consider how these policies compare as job parallelism increases, that is, as $S_n$ decreases. Figure 7.1a plots $\overline{R}_{ASP}/\overline{R}_{EQS}$ versus $S_n$ for the workload $(\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l)$. (Consider only the solid lines in the figure for now.) The range of $S_n$ in Figure 7.1a covers the likely practical values of $\overline{N}$ (i.e., $\overline{N}$ = 0.05P to P). Recall that when $S_n = 1$ (not shown) $\overline{R}_{EQS} = \overline{R}_{ASP}$ and thus the ratios for $C_D = 1$ will converge to 1 when $S_n = 1$. Also recall that the assumption of the linear ERF results in the lowest possible ratio of response times, as per Theorem 7.1.1, at each value of $\rho$. Thus the curves for workloads with sublinear ERFs will lie above the curves shown for the linear ERF. Figure 7.1a reveals that over the entire range of $S_n < 1$, the EQS policy outperforms the ASP policy. The ASP policy becomes more competitive with the EQS policy as $S_n$ increases, but is significantly less competitive for workloads that are (nearly) fully parallel. The reason for the poor performance of ASP is its lack of flexibility in processor allocation. Unlike the dynamic allocation under EQS, the (adaptive) static allocation under ASP can leave processors idle when a parallel job could otherwise use them.

To compare the policies for distributions of demand other than the exponential, we first note from (5.9) that for fixed $\overline{D}$, $\overline{R}_{EQS}$ is insensitive to $\mathcal{F}_D$. At $S_n = 1$, i.e., N=1, and $\gamma = \gamma^l$, $\overline{R}_{ASP} = \overline{R}_{M/G/P}$ and thus ASP policy performance is sensitive to $C_D$. When $P$ is large (say $P \geq 100$), then for $S_n = 1$, $\overline{R}_{ASP}$ is only slightly smaller than $\overline{R}_{EQS}$ for $C_D < 1$, equal to $\overline{R}_{EQS}$ at $C_D = 1$, and then increases with respect to $\overline{R}_{EQS}$ with further increase in $C_D$. The intuition for the increase of $\overline{R}_{ASP}$ with respect to $C_D$ at $S_n = 1$ is that a scheduled job runs to completion without interruption and each large demand job in execution reduces the number of system processors available for serving small jobs. This intuition should also apply for $S_n < 1$.



(a) $\theta = S_n$, $C_D = 0,\ 1$

(b) $\theta = C_D$

Figure 7.1: $\overline{R}_{ASP}/\overline{R}_{EQ}$ versus workload parameter $\theta$: r=0, $\gamma^l$

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

Figure 7.1a also plots $\overline{R}_{ASP}/\overline{R}_{EQS}$ versus $S_n$ when $C_D = 0$ (using approximations (6.7) and (5.9)), and as suggested by intuition the ratios are lower than when $C_D = 1$. However, the ratio is greater than one throughout the range of $S_n$ shown in the figure although at $S_n = 1$ and $C_D = 0$, as noted above, $\overline{R}_{ASP}$ is marginally smaller than $\overline{R}_{EQS}$.

For $C_D > 1$ we do not have analytic estimates of $\overline{R}_{ASP}$ and thus we resort to simulation to show the trends in relative policy performance. For $\overline{R}_{EQS}$, however, approximation (5.9) is valid for all $C_D$. Figure 7.1b plots $\overline{R}_{ASP}/\overline{R}_{EQS}$ versus $C_D$ for constant available parallelism and two-stage hyperexponential ($H_2$) demand distributions. The figure shows that $\overline{R}_{ASP}$ increases significantly with $C_D$ and the rise is sharper for larger available parallelism. The intuition for the latter observation is that jobs with higher parallelism and larger processing demand can occupy a larger number of servers, thus more significantly reducing the processors available to serve waiting jobs. Using the same intuition it appears likely that $\overline{R}_{ASP}$ should increase with $C_D$ for general demand and parallelism distributions. This was partially verified for specific nondeterministic distributions of $N$ (not shown).

Before concluding this section, it may be of interest to compare how EQS performs for workloads with sublinear $\gamma$ versus how ASP performs when $\gamma$ is linear. Assuming that job demand is exponential, $S_n(\gamma)$ and $S_n(\gamma^l)$ are the respective key parallelism parameters for EQS and ASP. To compare EQS($\gamma$) against ASP($\gamma^l$) we need relationships between $S_n(\gamma)$ and $S_n(\gamma^l)$. We consider the ERF

$$\gamma(k) = \frac{(1+\beta)k}{k+\beta}, \quad k = 1, \ldots, P. \tag{7.2}$$

For $r = 0$ and the ERF (7.2) we obtain

$$S_n(\gamma) = E\left[\frac{1}{\gamma(N)}\right] = E\left[\frac{N+\beta}{(1+\beta)N}\right] = \frac{1}{1+\beta}\left(1 + \beta E\left[\frac{1}{N}\right]\right) = \frac{1}{1+\beta}[1 + \beta S_n(\gamma^l)],$$

or equivalently,

$$S_n(\gamma^l) = \frac{1}{\beta}\left[(1+\beta)S_n(\gamma) - 1\right]. \tag{7.3}$$

For the ERF in (7.2) $S_n(\gamma^l)$ is related to $S_n(\gamma)$ by (7.3), and thus $S_n(\gamma)$ uniquely determines the performance of both policies. For $\rho = 0.7$ and $\rho = 0.9$ Figure 7.2 plots the ratio $\overline{R}_{ASP}(\gamma^l)/\overline{R}_{EQS}(\gamma)$ versus $S_n(\gamma)$ for the ERF (7.2) with $\beta = 100$, which is considerably sublinear as shown in Figure 3.2. The ratios converge to 1 when $S_n = 1$ and are thus greater than 1 throughout the range of $S_n < 1$. For (low) values of $\rho$ where mean service time dominates mean response time, the ratio will be less than 1 (except at $S_n = 1$). However, Figure 7.2 shows that (at moderate to high loads) a poor choice of scheduling policy, perhaps dictated by existing system software or hardware can be more detrimental to overall mean system response time than parallel program overheads.

#### 7.1.1.2 ASP versus EQS: r=1

In Section 7.1.1.1 we noted that lack of flexibility of processor allocation under ASP causes it perform worse than EQS when $r = 0$. For example, if a highly parallel job is allocated fewer processors than its available parallelism it cannot make use of additional processors when they become idle. When $r = 1$, the more parallel jobs also have larger demands and therefore the static allocation under ASP should hurt their performance even more than when $r = 0$. Thus intuition suggests that the differential between $\overline{R}_{ASP}$ and $\overline{R}_{EQS}$ will increase with $r$.

We compare $\overline{R}_{ASP}$ with $\overline{R}_{EQS}$ under the assumptions $(\cdot, \mathcal{F}_D^u = \exp(1/\overline{D}), r = 1, \gamma^l)$. The assumption of the linear ERF is more favorable to ASP and so is the assumption of exponential demands compared with $C_v > 1$. Due to lack of analytic estimates it is unknown whether $S_n$ is a key parameter for $\overline{R}_{ASP}$. As a result we use simulation and compare the performance of EQS and ASP under specific distributions of $N$. More specifically, we use the bounded-geometric distribution with parameters $P_{\max}$ and $p$. We note from Theorem 3.5.2 that for bounded-geometric distributions with a given $\overline{N}$, over all values of $P_{\max}$ and $p$, $C_N$ is maximum when $p = 1$ and $C_N$ is minimum when $P_{\max} = 0$. We refer to workloads with these two extremes of $C_N$ as **high** $C_N$ and **low** $C_N$ workloads, respectively.

Figure 7.3a plots $\overline{R}_{ASP}/\overline{R}_{EQS}$ versus $S_n$ for the high $C_N$ and low $C_N$ workloads for the linear ERF at two values of $\rho$. The curves for workloads with sublinear ERFs will lie above

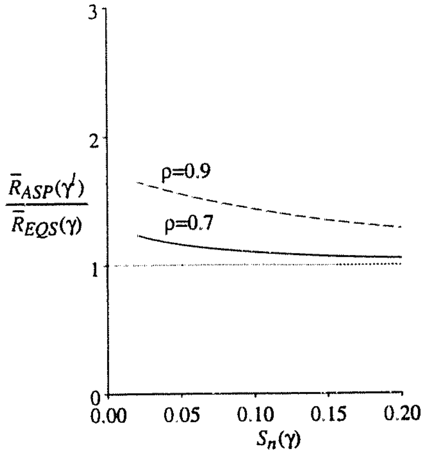


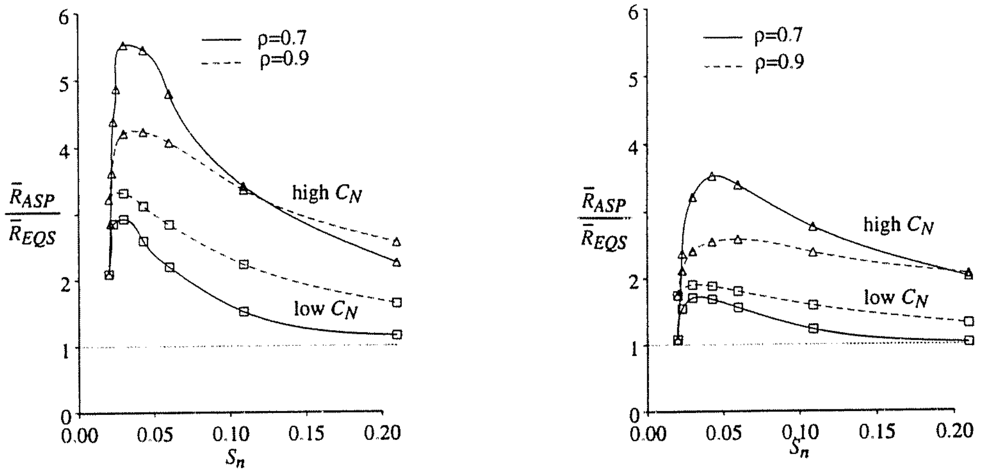
Figure 7.2: $\overline{R}_{ASP}(\gamma^l)/\overline{R}_{EQ}(\gamma)$ versus $S_n(\gamma)$: r=0

P=100, $C_D = 1$, $\beta = 100$

those for the linear ERF. We note from Figure 7.3a that $\overline{R}_{ASP}/\overline{R}_{EQS}$ is higher for the high $C_N$ workload than the low $C_N$ workload. Moreover, the ratios for the high $C_N$ workload are markedly higher than the ratios in Figure 7.1. What causes the performance of ASP to degrade at $r = 1$ when $C_N$ is high? The intuition for this behavior is as follows. For all data points in Figure 7.3 we observed that the mean waiting time under ASP is negligible compared to the overall mean response time of ASP[1]. Thus for the given range of utilizations and for the given workload assumptions, for the ASP policy the mean response time of an arriving job is primarily determined by the number of processors it is allocated when it begins service. The high $C_N$ workload has a much higher percentage of fully parallel jobs as compared to the low $C_N$ workload and fully parallel jobs under ASP have the highest mean service time among all parallelism classes. Furthermore, these jobs are allocated a smaller fraction of the number of processors that they can productively use on average. This phenomenon is exaggerated for ASP than for EQS since under ASP a job's partition cannot expand beyond its initial allocation.

From Figure 7.3a we also observe that all curves initially increase sharply with $S_n$, reach a peak at moderate parallelism, and then decrease with further increase in $S_n$. We clarify this behavior by explaining two opposing trends that occur when $S_n$ increases. The first trend is

[1] This observation does not concur with the observations of Setia and Tripathi [69] because we examine a system with $P = 100$ whereas they examined systems with $P \leq 10$ in which it is less likely for a job to find an idle processor upon arrival.



(a) linear ERF

(b) $\beta = 100$

Figure 7.3: $\overline{R}_{ASP}(\gamma^l)/\overline{R}_{EQ}(\gamma)$ versus $S_n(\gamma)$: r=1

P=100, $\overline{D} = P$, $C_v = 1$

that when $S_n$ increases the mean demand of highly parallel jobs increases[2] which causes their mean response time under ASP to increase relative to EQS because a highly parallel job can make use of idle processors under EQS but not under ASP. The second trend is that when $S_n$ increases the percentage of fully parallel jobs decreases which decreases their contribution to overall mean response time. When $S_n$ is low the first trend dominates causing the curves in Figure 7.3 to increase and as $S_n$ increases further the second trend dominates causing the curves to decrease.

As in the case of $r = 0$ we also plot $\overline{R}_{ASP}(\gamma^l)/\overline{R}_{EQS}(\gamma)$ versus $S_n(\gamma)$ for the highly sublinear ERF with $\beta = 100$. Figure 7.3b shows that the ratios are greater than 1 throughout the range of $S_n$ (they converge to 1 at $S_n = 1$). At lower utilizations where $\overline{S}$ dominates mean response time, the ratios will be less than 1. Thus at moderate to high utilizations EQS performs better even when the EQS workload has a sublinear ERF and the ASP workload has the linear ERF.

#### 7.1.1.3 Summary of ASP versus EQS Comparison

To summarize the policy comparison results for ASP and EQS, we conclude that EQS has

[2] As $S_n$ increases more and more of the probability mass shifts to lower values of parallelism where jobs have smaller mean demands (since $r = 1$). Therefore, to keep the overall mean demand as $\overline{D}$ the mean demand of highly parallel jobs increases.

significantly better performance because (1) it utilizes processors better, that is, jobs make use of idle processors whenever possible, and (2) its mean response time is not sensitive to variation in job demand. ASP becomes more competitive with EQS as $S_n$ decreases, $C_v$ decreases, and ERF linearity increases. While the last two observations follow from intuition and from Theorem 7.1.1 the first observation follows from the results shown in Figure 7.1 (and 7.3) and would be difficult to obtain in the absence of simple approximations such as (6.3) and (6.7) for $\overline{R}_{ASP}$.

### 7.1.2 PSAPF versus FCFS

The purpose of this section is to quantify the difference in performance between PSAPF and FCFS over the model parameter space $(\mathcal{F}_N, \mathcal{F}_D^u, r, \gamma)$. When N is deterministic, PSAPF is identical to FCFS. For nondeterministic N, however, we expect PSAPF to perform differently than FCFS. In general, one can expect PSAPF to perform better than FCFS for three reasons. First, by delaying service of more parallel jobs, PSAPF tends to keep processor utilization high for a larger portion of each busy period [1]. Second, for correlated workloads PSAPF gives higher priority to jobs with smaller mean demands. Third, at high instantaneous load the overall efficiency is higher under PSAPF for sublinear $\gamma$ because jobs that receive higher priority also execute more efficiently. Due to the last two reasons the most favorable parameter values for FCFS relative to PSAPF are no correlation ($r = 0$) and the linear ERF ($\gamma = \gamma^l$). Using the analytic models of Chapter 6 we show how the policies compare under these favorable conditions and extrapolate the results to the case of sublinear ERFs. We then provide simulation data to show how PSAPF and FCFS compare under correlated workloads.

#### 7.1.2.1 PSAPF versus FCFS: r=0

Consider the workload settings $r = 0$ and $\gamma = \gamma^l$. Using approximation (6.10) we have that

$$\overline{X}_{PSAPF} \approx \overline{X}_{FCFS} \times \frac{\overline{X}_{M/G/1_P\ PR}}{\overline{X}_{M/G/1_P}}.$$

To compare $\overline{R}_{PSAPF}$ with $\overline{R}_{FCFS}$ we need to compare $\overline{X}_{M/G/1_P\ PR}$ with $\overline{X}_{M/G/1_P\ FCFS}$. From Section 6.3.2, under the given workload

$$\overline{X}_{M/G/1_P\ PR} = \sum_{k=1}^{P} p_k \left[\frac{\sigma_{k-1}}{1-\sigma_{k-1}} + \frac{\sigma_k}{(1-\sigma_{k-1})(1-\sigma_k)}\left(\frac{1+C_D^2}{2}\right)\right]\frac{\overline{D}}{P}, \quad \text{where} \quad \sigma_k = \rho\sum_{i=1}^{k} p_i,$$

and

$$\overline{X}_{M/G/1_P\ FCFS} = \frac{\rho}{1-\rho}\left(\frac{1+C_D^2}{2}\right)\frac{\overline{D}}{P}.$$

Note that relative policy performance is only sensitive to the first two moments of $D$. When $D = \exp(1/\overline{D})$ then $\overline{X}_{M/G/1_P\ PR} = \overline{X}_{M/M/1_P}$ and thus for all $\mathcal{F}_D$ with $C_D = 1$ and fixed $\overline{D}$, we have $\overline{X}_{M/G/1_P} = \overline{X}_{M/M/1_P}$ which means that $\overline{X}_{PSAPF} \approx \overline{X}_{FCFS}$ when $C_D = 1$. Setting $C_D = 1$ in the formulas for $\overline{X}_{M/G/1_P\ PR}$ and $\overline{X}_{M/G/1_P\ FCFS}$ we find that

$$\sum_{k=1}^{P} p_k \left[\frac{\sigma_{k-1}}{1-\sigma_{k-1}} + \frac{\sigma_k}{(1-\sigma_{k-1})(1-\sigma_k)}\right] \frac{\overline{D}}{P} = \frac{\rho}{1-\rho}\frac{\overline{D}}{P}.$$

Now consider $C_D > 1$. Since $(1 + C_D^2)/2 > 1$, we obtain

$$\overline{X}_{M/G/1_P\ PR} < \left(\frac{1+C_D^2}{2}\right) \sum_{k=1}^{P} p_k \left[\frac{\sigma_{k-1}}{1-\sigma_{k-1}} + \frac{\sigma_k}{(1-\sigma_{k-1})(1-\sigma_k)}\right] \frac{\overline{D}}{P} = \left(\frac{1+C_D^2}{2}\right) \frac{\rho}{1-\rho}\frac{\overline{D}}{P} = \overline{X}_{M/G/1_P}.$$

Thus $\overline{X}_{PSAPF} < \overline{X}_{FCFS}$ when $C_D > 1$. Likewise, when $C_D < 1$, $\overline{X}_{PSAPF} > \overline{X}_{FCFS}$. Thus, at $r = 0$, and $\gamma = \gamma^l$, the relative performance of PSAPF and FCFS as determined by $C_D$ is as follows

$$\overline{R}_{PSAPF} \begin{cases} > \overline{R}_{FCFS}, & C_D < 1, \\ = \overline{R}_{FCFS}, & C_D = 1, \\ < \overline{R}_{FCFS}, & C_D > 1. \end{cases} \tag{7.4}$$

These results are illustrated for the three bounded-geometric distributions for $N$ given in Table 7.1.

Table 7.1: Three Bounded-Geometric Distributions for $N$

P=100

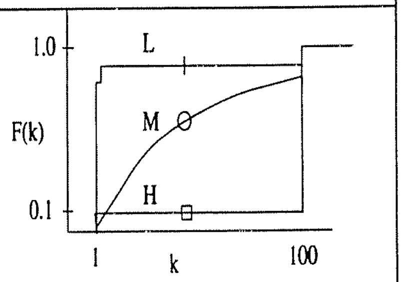
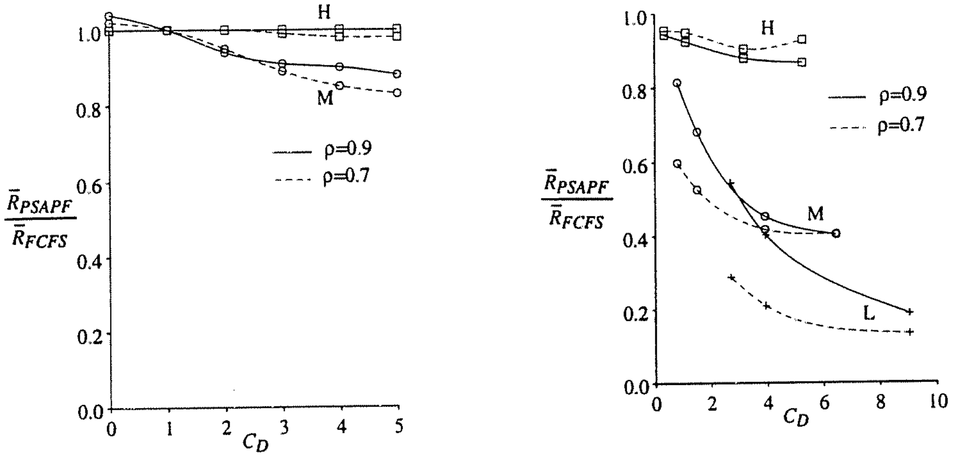
| Symbol | Parallelism | $P_{max}$ | $p$ | $\overline{N}$ | $C_N$ | CDF of $N$ |
|---|---|---|---|---|---|---|
| H | High | 0.9 | 1.0 | 90.10 | 0.33 |  |
| M | Moderate | 0.1 | 1/(0.4P) | 43.14 | 0.80 | |
| L | Low | 0.1 | 0.9 | 11.00 | 2.70 | |

Figure 7.4a plots $\overline{R}_{FCFS}/\overline{R}_{PSAPF}$ versus $C_D$ for the H and M workloads. The response time ratios for the L workload are not shown because they lie very close to those for the H workload. We note from Figure 7.4a that $C_D$ has a much stronger effect for the M workload as compared to the H and L workloads. This is because PSAPF is more highly differentiated from FCFS for the M workload in which there is a wider range of values for available parallelism as opposed to the H and L workloads, as shown by the cdfs in Table 7.1.



(a) $r = 0$

(b) $r = 1$

Figure 7.4: $\overline{R}_{FCFS}/\overline{R}_{PSAPF}$ versus $C_D$: $\gamma^l$

P=100

#### 7.1.2.2 PSAPF versus FCFS: r=1

Let us now compare PSAPF and FCFS at $r = 1$. We expect the performance of PSAPF to improve as correlation increases and thus $\overline{R}_{PSAPF}/\overline{R}_{FCFS}$ should be lower at $r = 1$ than at $r = 0$. We ran simulation experiments to obtain $\overline{R}_{FCFS}$ at $r = 1$ and $C_v$ between 0 and 5 for the H, M, and L parallelism workloads. $\overline{R}_{PSAPF}$ is approximated using (6.16). Figure 7.4b plots the ratios $\overline{R}_{FCFS}/\overline{R}_{PSAPF}$ as a function of $C_D$ for the three parallelism workloads. (For the L workload the results are shown up to $C_v = 3$ which corresponds to $C_D = 9.04$ using (3.4).) We observe that the mean response time ratios at r=1 are significantly lower than the ratios at r=0. The ratios also decrease faster with increasing $C_D$, and also decrease more substantially for the M workload compared with the H workload, and for the L workload compared with the M workload. The reason for the marked improvement in the L workload is that when $r = 1$ 90% of the jobs have lower mean demands than the other 10%. This differentiation in mean demands when $r = 1$ increases the performance differential between PSAPF and FCFS.

#### 7.1.2.3 Summary of PSAPF versus FCFS Comparison

To summarize the comparison between PSAPF and FCFS, the results have shown that PSAPF performs better than FCFS for most of the parameter space. FCFS performs marginally better when $r = 0$ and $C_D < 1$. The quantitative results above were for the linear ERF. PSAPF should perform relatively even better if the ERF is sublinear, as explained above.

### 7.1.3 PSAPF versus EQS

Sections 7.1.1 and 7.1.2 showed that in general, EQS performs better than ASP and PSAPF performs better than FCFS, respectively. The EQS policy has high performance since it efficiently utilizes processors and its response time is insensitive to $C_D$. The PSAPF policy has high performance for workloads with high correlation since it favors jobs with small mean demand in these workloads. In this section we first compare PSAPF and EQS under no correlation and the linear ERF ($r = 0$ and $\gamma = \gamma^l$) and then under full correlation and the linear ERF ($r = 1$ and $\gamma = \gamma^l$). We use the comparisons at these two extreme ends of correlation to obtain results for partial workload correlation. Note that on account of Theorem 7.1.1 the parameter setting $\gamma = \gamma^l$ is more favorable to PSAPF relative to EQS.

**7.1.3.1 PSAPF versus EQS: r=0**

We compare PSAPF and EQS at $r = 0$ and $\gamma = \gamma^l$ using accurate approximations for the mean response times of both policies. Using approximation (6.10) we have that

$$\overline{R}_{PSAPF} \approx \overline{S} + \overline{X}_{FCFS} \cdot \frac{\overline{X}_{M/G/1_P\ PR}}{\overline{X}_{M/G/1_P}}. \tag{7.5}$$

Using this approximation for $\overline{R}_{PSAPF}$ we first compare PSAPF against EQS when $C_D = 1$ by comparing $\overline{X}_{FCFS}$ against $\overline{X}_{EQS}$ when $C_D = 1$. We then extend the comparison over the entire range of $C_D$.

At $r = 0$ and $\gamma = \gamma^l$ the interpolation on $\underline{p}$ was shown to very accurate for $\overline{X}_{FCFS}$ as well as for $\overline{X}_{EQS}$. Using this interpolation approximation (see (4.14) and (4.13)) under the given assumptions we get

$$\overline{X}_{FCFS} \approx \frac{E\left[\rho^{\sqrt{2(\frac{\underline{P}}{N}+1)}}\right]}{\lambda(1-\rho)} \left(\frac{1+C_D^2}{2}\right),$$

and

$$\overline{X}_{EQS} \approx \frac{E\left[\rho^{\sqrt{2(\frac{\underline{P}}{N}+1)}}\right]}{\lambda(1-\rho)}.$$

Using these formulas for $\overline{X}_{FCFS}$ and $\overline{X}_{EQS}$ we have that

$$\overline{X}_{FCFS} \approx \overline{X}_{EQS}, \qquad \text{under } (\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l).$$

It therefore follows from (7.5) that

$$\overline{R}_{PSAPF} \approx \overline{S} + \overline{X}_{EQS} \cdot \frac{\overline{X}_{M/G/1_P\ PR}}{\overline{X}_{M/G/1_P\ FCFS}}, \qquad \text{under } (\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l).$$

From Section 7.1.2.1 we observe that

$$\overline{X}_{M/G/1_P\ PR} = \overline{X}_{M/G/1_P\ FCFS}, \qquad \text{when } C_D = 1.$$

As a result

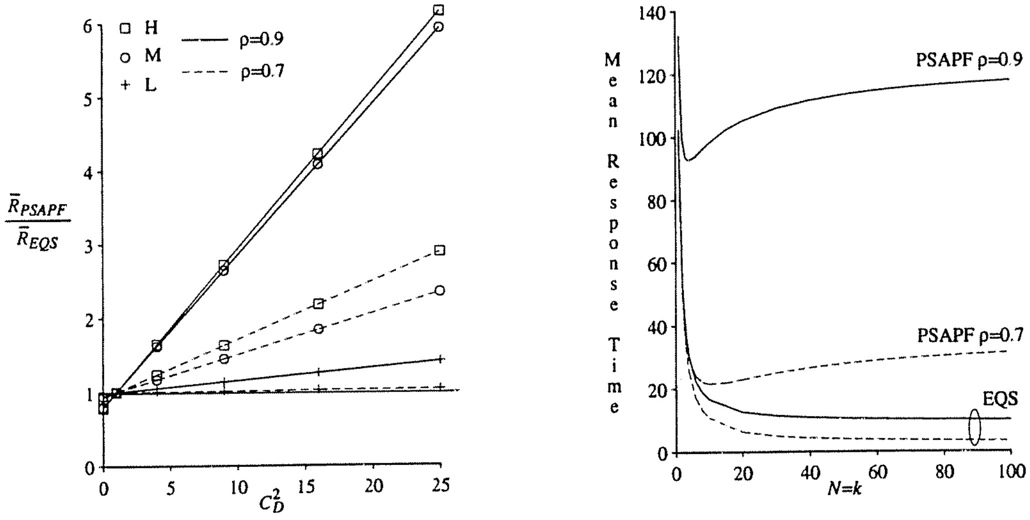
$$\overline{R}_{PSAPF} \approx \overline{S} + \overline{X}_{EQS} = \overline{R}_{EQS}, \qquad \text{under } (\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma^l). \tag{7.6}$$

This comparison may seem limited since we have just examined one value of $C_D$ and that too an impractical one. We, however, claim that this comparison reveals full information about the relative performance of PSAPF and EQS at $r = 0$ and $\gamma = \gamma^l$. The reason is that from approximation (6.12) we have that $\overline{R}_{PSAPF}$ increases linearly in $C_D^2$ and we have also repeatedly seen from all approximations for $\overline{R}_{EQS}$ that $\overline{R}_{EQS}$ is independent of $C_D$. If we plot $\overline{R}_{PSAPF}$ and $\overline{R}_{EQS}$ versus $C_D^2$ we will observe the curve for PSAPF to be a straight line with a positive slope and the curve for EQS to be a horizontal line. From (7.6) we note that these two lines will intersect at $C_D = 1$, regardless of the distribution of $N$. This means that for all distributions of $N$, when $r = 0$ and $\gamma = \gamma^l$

$$\overline{R}_{PSAPF} \begin{cases} < \overline{R}_{EQS}, & C_D < 1, \\ = \overline{R}_{EQS}, & C_D = 1, \\ > \overline{R}_{EQS}, & C_D > 1. \end{cases} \qquad \text{under } (\mathcal{F}_N, \mathcal{F}_D^u, r = 0, \gamma^l). \tag{7.7}$$

Thus, when $r = 0$ and $\gamma = \gamma^l$, we can uniquely determine the relative performance of PSAPF and EQS using $C_D$ alone. Leutenegger and Vernon [41] have identical comparison results for PSNPF and RRJ (temporal EQS) (using simulation) as (7.7) in the range $C_D \geq 1$, assuming $H_2$ job demands and specific distributions of $N$. The result we have shown holds for all distributions of $D$ and $N$. To graphically illustrate this result and get a quantitative estimate of the difference between $\overline{R}_{PSAPF}$ and $\overline{R}_{EQS}$ at $r = 0$ we use the three bounded-geometric distributions for $N$ given in Table 7.1. Figure 7.5a plots $\overline{R}_{PSAPF}/\overline{R}_{EQS}$ versus $C_D^2$ for the H, M, and L workloads for $N$. As explained above the curves are linear in $C_D^2$ because $\overline{R}_{PSAPF}$ increases linearly in $C_D^2$ whereas $\overline{R}_{EQS}$ is insensitive to $C_D$. We observe that for $C_D < 1$, $\overline{R}_{EQS}$ is fairly close to $\overline{R}_{PSAPF}$ and that at high values of $C_D$ such as $C_D = 5$ EQS significantly outperforms PSAPF for the H and M workloads which have high and moderate parallelism respectively. This is particularly true at $\rho = 0.9$. For the L workload the difference between PSAPF and EQS is not as significant because the mean response time of this workload is dominated by the mean service time $\overline{S}$, which is the same for both PSAPF and EQS. To observe how these two policies compare over a range of parallelism values for workloads with practical values of $C_D$ such as $C_D = 5$ we plot $\overline{R}_{EQS}(N = k)$ and $\overline{R}_{PSAPF}(N = k)$ as a function of $k$ in Figure 7.5b. We see that PSAPF can perform as much as 12 times worse than EQS at $\rho = 0.9$ and close to 10 times worse at

$\rho = 0.7$. We observe that $\overline{R}_{PSAPF}$ can increase with an increase in parallelism, particularly, at $\rho = 0.9$, whereas $\overline{R}_{EQS}$ decreases with an increase in parallelism. The same observation was made by Leutenegger [39] for the PSNPF and RRJ policies for specific workloads of demand and parallelism. We will study the behavior of $\overline{R}_{EQS}$ as a function of available parallelism in more detail in Chapter 8.



(a) $\overline{R}_{PSAPF}/\overline{R}_{EQS}$ vs $C_D^2$

(b) $\overline{R}_\Psi$(N=k) vs k, $C_D$=5, $\Psi \in \{PSAPF, EQS\}$

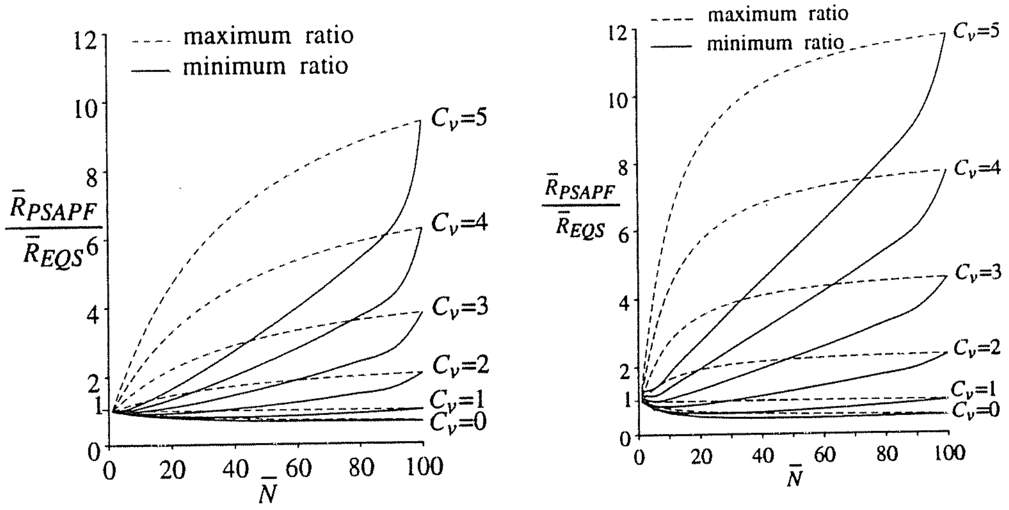
Figure 7.5: Comparison of PSAPF and EQS at $r = 0$: $\gamma^l$

$\overline{D} = P = 100$

#### 7.1.3.2 PSAPF versus EQS: r=1

When $r = 1$ and $\gamma = \gamma^l$, we have $\overline{S} = \overline{D}/\overline{N}$ (see the equation above (3.6)) and thus $S_n = 1/\overline{N}$. Since $S_n$ is the key parallelism parameter for EQS, and $\overline{N}$ is uniquely determined from $S_n$ at $r = 1$ and $\gamma = \gamma^l$, it follows that $\overline{N}$ is equivalently the key parallelism parameter for EQS under these workload conditions. That is, at $r = 1$ and $\gamma = \gamma^l$ $\overline{R}_{EQS}$ is approximately the same across all distributions of $N$ with the same $\overline{N}$. This is not necessarily true for $\overline{R}_{PSAPF}$. We therefore use approximation (6.16) and nonlinear programming to obtain the minimum and maximum values of $\overline{R}_{PSAPF}$ across all distributions of $N$ that have the same $\overline{N}$, and then use these values to determine the relative performance of PSAPF with respect to EQS. The details of the nonlinear programming solution method are given in Appendix A.3.2.

Figure 7.6a plots the minimum and maximum of $\overline{R}_{PSAPF}/\overline{R}_{EQS}$ as a function of $\overline{N}$ for



(a) $\rho = 0.7$

(b) $\rho = 0.9$

Figure 7.6: $\overline{R}_{PSAPF}/\overline{R}_{EQS}$ versus $\overline{N}$: $r = 1$, $\gamma^l$

P=100

various $C_v$ at $\rho = 0.7$ for a 100-processor system. We observe that the ratios increase with $C_v$ since PSAPF is sensitive to $C_v$, whereas EQS is not. When $C_v \leq 1$ PSAPF performs better than EQS. However, the reverse is mostly true for $C_v = 2$ (in the range $\overline{N} \geq 30$) and is true for all cases with $C_v \geq 3$. We also observe that the minimum and maximum ratios increase with $\overline{N}$ for $C_v \geq 2$, due to the improvement in EQS performance with increase in average available parallelism when $r = 1$. Figure 7.6b plots similar ratios for $\rho = 0.9$ and we note that the difference between PSAPF and EQS increases with an increase in $\rho$.

Under sublinear execution rates the ratios of Figure 7.6 should be higher by virtue of Theorem 7.1.1. We can therefore conclude that for general workload conditions with $r = 1$, EQS outperforms PSAPF as long as $C_v > 2$.[3]

To summarize the policy comparison results between EQS and PSAPF at $r = 1$ we have

$$\overline{R}_{EQS} \begin{cases} > \overline{R}_{PSAPF}, & 0 \leq C_v < 1, \\ ?\ \overline{R}_{PSAPF}, & 1 \leq C_v < 2, \\ < \overline{R}_{PSAPF}, & C_v = 2,\ \overline{N} \geq 0.3P, \\ < \overline{R}_{PSAPF}, & C_v > 2, \end{cases} \qquad \text{under } (\mathcal{F}_N, \mathcal{F}_D^u, r = 1, \gamma^l),$$

[3] Note that given $C_v$, $C_N$, and $r$, we can compute $C_D$ using (3.4). For example, when $\overline{N} = 50$, $C_N \leq 0.99$ from (3.5.1) and thus if $C_v = 2$, we get $C_D \leq 3$.

where the question mark for $1 \leq C_v < 2$ reflects that under the given assumptions the exact relationship between $\overline{R}_{EQS}(r=1)$ and $\overline{R}_{PSAPF}(r=1)$ is sensitive to the distribution of $N$.

### 7.1.3.3 PSAPF versus EQS: $0 < r < 1$

Coupling the results from Sections 7.1.3.1 and 7.1.3.2 for $r = 0$ and $r = 1$ we have the following relationships between $\overline{R}_{EQS}$ and $\overline{R}_{PSAPF}$ at the extreme ends of correlation.

$$\overline{R}_{EQS}(r=0) \begin{cases} > \overline{R}_{PSAPF}(r=0), & 0 \leq C_v < 1, \\ = \overline{R}_{PSAPF}(r=0), & C_v = 1, \\ < \overline{R}_{PSAPF}(r=0), & C_v > 1, \end{cases} \tag{7.8}$$

and

$$\overline{R}_{EQS}(r=1) \begin{cases} > \overline{R}_{PSAPF}(r=1), & 0 \leq C_v < 1, \\ ?\ \overline{R}_{PSAPF}(r=1), & 1 \leq C_v < 2, \\ < \overline{R}_{PSAPF}(r=1), & C_v = 2,\ \overline{N} \geq 0.3P, \\ < \overline{R}_{PSAPF}(r=1), & C_v > 2, \end{cases} \tag{7.9}$$

Note that in (7.8) we used the fact that at $r = 0$, we have $C_v = C_D$ as can be seen from (3.4). Now consider the case where $0 < r < 1$. The approximations for $\overline{R}_{EQS}$ and $\overline{R}_{PSAPF}$ (see (5.10) and (6.17)) for $0 < r < 1$ have the following forms

$$\begin{aligned} \overline{R}_{EQS} &\approx (1-r^2)\overline{R}_{EQS}(r=0) + r^2\overline{R}_{EQS}(r=1) \\ \overline{R}_{PSAPF} &\approx (1-r^2)\overline{R}_{PSAPF}(r=0) + r^2\overline{R}_{PSAPF}(r=1). \end{aligned}$$

Using these approximations and the relationships in (7.8) and (7.9) it follows that for general $r$, $\overline{R}_{EQS} > \overline{R}_{PSAPF}$ for $C_v < 1$ and $\overline{R}_{EQS} < \overline{R}_{PSAPF}$ for $C_v > 2$. For $C_v$ between 1 and 2, the relative performance of EQS and PSAPF depends on the value of $r$ and on the distribution of $N$. In general, workloads for computer systems have high variation in demand [65, pg16],[81][4] and in these systems we expect EQS to perform significantly better than PSAPF.

## 7.2 Policy Comparison using Exact Analysis

In the previous section we saw that the EQS policy has highest performance for most of the design space, particularly, when coefficient of variation in job demand is moderate to high. In this section we explain why it has high performance with respect to variation in demand. We analyze the sensitivity of EQ (all EQ policies) for a workload model with job dependent ERFs, more general correlation that what we have assumed thus far, general available parallelism, and for a generalized exponential (GE) distribution of demand.

[4] Note that we have also measured the coefficient of variation in service times on our local CM-5 to be ranging from 2.8 to about 5, with the higher end being more typical.

A random variable has a GE distribution if it is either zero with a certain probability or exponentially distributed otherwise. The GE distribution is completely parameterized by the mean and coefficient of variation and is thus suitable for analyzing the behavior of a policy with respect to $C_D$. The GE distribution has been used in several previous studies to analyze systems such as FIFO and/or Preemptive Resume G/G/1 or G/G/c queues [34, 35, 36, 78]. It has also been used to validate approximations in (non-product form) closed queueing networks with FIFO queues under non-exponential service demands [86]. In the context of parallel processor system models, we have already seen that Towsley et al. [80] have used a GE distribution to model variation in task service times.

In this section we show that $\overline{R}_{EQ}$ is insensitive to $C_D$ when job demand has a GE distribution under very general assumptions for the other workload parameters. We also show that for the same workload assumptions $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_D^2$ which corroborates our results from approximate analysis. The result for EQ derived in this section generalizes to all policies that provide instant service to arriving jobs, e.g., PSCDF, RRP and LCFS-PR. The result for FCFS and PSAPF generalizes to all policies that do not always provide instant service to arriving jobs and where jobs with zero demand do not affect the performance of jobs with nonzero demand.

Section 7.2.1 provides more details of the workload assumptions for the results in this section. Section 7.2.2 proves the sensitivity result for EQ using sample path analysis and also presents the sensitivity result for FCFS and PSAPF. The proof for the FCFS and PSAPF result is given in Appendix A.3.3.

### 7.2.1 Workload Assumptions

Assume that a stream of parallel jobs, $i = 1, 2, \ldots,$ arrive at the system. Each job $i$ has the follow characteristics:

(1) Arrival time $A_i$,

(2) Total service demand (execution time on one processor) $D_i$,

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

(4) Execution rate function $E_i : [0, P] \to [0, P]$, which is nondecreasing and has the following properties:

$$E_i(x) \begin{cases} = x, & 0 \le x \le 1, \\ \le x, & 1 < x \le N, \\ = E_i(N_i), & N_i < x \le P. \end{cases}$$

(5) External class $C_i$.

We assume the following about the workload:

- The job arrival process is a Poisson process with rate $\lambda$.
- $D_i$ has a generalized exponential distribution with parameters $\mu_i$ and $\alpha$, for $i = 1, 2, \ldots$. That is,
$$D_i = \begin{cases} \exp(\mu_i), & \text{with probability } \alpha, \\ 0, & \text{otherwise.} \end{cases}$$
We further assume that $E[D_i] = f_i(N_i, C_i)$ is independent of $\alpha$, where $f_i$ is a positive real valued function. Thus, expected demand of job $i$ is allowed to depend on $N_i$ and $C_i$ but not on $\alpha$.

These workload assumptions lead to the following relationships between demand parameters:

$$E[D_i] = \frac{\alpha}{\mu_i} = f_i(N_i, C_i).$$

Thus

$$\mu_i = \frac{\alpha}{f_i(N_i, C_i)}. \tag{7.10}$$

The coefficient of variation of $D_i$, i.e., $C_{D_i}$, is related to $\alpha$ as follows:

$$C_{D_i}^2 = \alpha \frac{2/\mu_i^2}{E[D_i]^2} - 1 = \frac{2}{\alpha} - 1.$$

Thus $C_{D_i}$ is independent of $i$, and we let $C_v^2 \equiv 2/\alpha - 1$, i.e.,

$$\alpha = (1 + C_v^2)/2. \tag{7.11}$$

As per our workload assumptions, each job has a demand drawn from a GE distribution that has a job dependent mean but the same coefficient of variation, $C_v$, for all jobs. Furthermore, the expected of value of job demand ($E[D_i]$) is independent of the probability, $\alpha$, that a job has nonzero demand.

### 7.2.2 Sensitivity of EQ, FCFS, and PSAPF to $C_v$

We first prove that that for the workload assumptions of Section 7.2.1 $\overline{R}_{EQ}$ is insensitive to $C_v$. The proof uses sample path analysis and is given here rather than in the appendix because it reveals the kind of policies for which the result can be extended. We then state the result that $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_v^2$ for the workload assumptions of Section 7.2.1.

#### 7.2.2.1 $\overline{R}_{EQ}$ is insensitive to $C_v$

**Theorem 7.2.1** *Under the workload assumptions of Section 7.2.1, $\overline{R}_{EQ}$ is independent of $C_v$.*

**Proof.** Let $\Gamma$ denote a system with the EQ policy and workload assumptions as in Section 7.2.1. Let a job be of type 1 if its demand is nonzero and of type 2 if its demand is zero. Type 1 jobs arrive according to a Poisson process with rate $\lambda\alpha$ and type 2 jobs arrive according to a Poisson process with rate $\lambda(1-\alpha)$. Let $\overline{R}_{\Gamma,i}$ denote the mean response time of type $i$ jobs in system $\Gamma$, $i = 1, 2$. Clearly, $\overline{R}_{\Gamma,2} \equiv 0$ since type 2 jobs receive instant service in $\Gamma$. It therefore follows that the overall mean response time of the EQ policy in system $\Gamma$ is given by

$$\overline{R}_{EQ} = \alpha \overline{R}_{\Gamma,1}. \tag{7.12}$$

We wish to show that $\overline{R}_{EQ}$ is independent of $C_v$ or equivalently of $\alpha$ (by virtue of (7.11)). From (7.12) it follows that we need to show that $\overline{R}_{\Gamma,1}$ is inversely proportional to $\alpha$. We show this to be true by constructing an equivalent system, $\Gamma_I$, for type 1 jobs and showing how it is related to a "faster" system $\Gamma_{II}$ which does not depend on $\alpha$.

Type 2 jobs exit instantaneously upon arrival in system $\Gamma$ and thus they do not affect the processor allocation to type 1 jobs. As a result $\overline{R}_{\Gamma,1}$ is the same as the mean response time of a system $\Gamma_I$ where only type 1 jobs arrive with rate $\lambda\alpha$ and have exponential demands given by $D_i \sim \exp(\mu_i)$. Denote the mean response time of $\Gamma_I$ by $\overline{R}_{\Gamma_I}$. Thus $\overline{R}_{\Gamma,1} = \overline{R}_{\Gamma_I}$. Now consider a system $\Gamma_{II}$ in which only type 1 jobs arrive with rate $\lambda$ and have exponential demands given by $D_i \sim \exp(\mu_i/\alpha)$. Thus in system $\Gamma_{II}$ job arrival and service rates are $1/\alpha$ times faster than in $\Gamma_I$. Denote the mean response time of $\Gamma_{II}$ by $\overline{R}_{\Gamma_{II}}$. We show using sample path analysis that $\overline{R}_{\Gamma_I} = 1/\alpha \overline{R}_{\Gamma_{II}}$, and thus $\overline{R}_{\Gamma,1} = 1/\alpha \overline{R}_{\Gamma_{II}}$ which is inversely proportional to $\alpha$ as required. We first suitably couple sample paths and then show that $\overline{R}_{\Gamma_I} = 1/\alpha \overline{R}_{\Gamma_{II}}$ over each pair of coupled sample paths.

*Coupling of Sample Paths in gI and gII*

Fix $\{N_i, E_i, C_i\}_{i=1}^{\infty}$ as the same for both $\Gamma_I$ and $\Gamma_{II}$. (For notational convenience we index type 1 jobs as $1, 2, \ldots$.) For $\Gamma_I$ fix arrival times and job demands as $\{A_i, D_i\}_{i=1}^{\infty}$, where $A_i$ are generated at jumps of a Poisson process with rate $\lambda\alpha$, and $D_i$ is a sample from an exponential distribution with rate $\mu_i$. For system $\Gamma_{II}$ the arrival process is Poisson with rate $\lambda$ and the demand of the $i^{th}$ job is exponentially distributed with rate $\mu_i/\alpha$. As a result, fix the arrival instant of the $i^{th}$ job in $\Gamma_{II}$ to be $A_i\alpha$, and the demand of the $i^{th}$ job to be $D_i\alpha$, where $A_i$ and $D_i$ are the arrival time and demand, respectively, of job $i$ in $\Gamma_I$.

*Sample Path Analysis*

For the above coupling of sample paths it follows that under the EQ policy $\Gamma_{II}$ is simply a time compressed version of $\Gamma_I$. Thus if $\mathcal{S}$ is the state of system $\Gamma_I$ at time $t$ then $\mathcal{S}$ is the state of system $\Gamma_{II}$ at time $\alpha t$. Let $Q_{\Gamma_I}(t)$ denote the number of jobs in $\Gamma_I$ at time $t$, and let

$Q_{\Gamma_{II}}(t)$ denote the number of jobs in $\Gamma_{II}$ at time $t$. We have $Q_{\Gamma_I}(t) = Q_{\Gamma_{II}}(\alpha t)$. Taking time averages, we get

$$\begin{aligned}
\overline{Q}_{\Gamma_I} &= \lim_{t\to\infty} \frac{1}{t}\int_0^t Q_{\Gamma_I}(u)du \\
&= \lim_{t\to\infty} \frac{1}{t}\int_0^t Q_{\Gamma_{II}}(\alpha u)du \\
&= \lim_{t\to\infty} \frac{1}{\alpha t}\int_0^{\alpha t} Q_{\Gamma_{II}}(s)ds, \quad \text{(where } s = \alpha u\text{)} \\
&= \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} Q_{\Gamma_{II}}(s)ds, \quad \text{(where } \tau = \alpha t\text{)} \\
&= \overline{Q}_{\Gamma_{II}}.
\end{aligned}$$

By Little's Law [76] it follows that over each pair of coupled sample paths,

$$\overline{R}_{\Gamma_I} = \frac{\overline{Q}_{\Gamma_I}}{\lambda\alpha} = \frac{\overline{Q}_{\Gamma_{II}}}{\lambda\alpha} = \frac{1}{\alpha}\overline{R}_{\Gamma_{II}},$$

which completes the sample path analysis. Now uncondition on $\{A_i, D_i, N_i, E_i, C_i\}_{i=1}^{\infty}$.

Relating back to equation (7.12) we have $\overline{R}_{\Gamma,1} = \overline{R}_{\Gamma_I} = 1/\alpha\overline{R}_{\Gamma_{II}}$ and thus for system $\Gamma$

$$\overline{R}_{EQ} = \alpha \times \frac{1}{\alpha}\overline{R}_{\Gamma_{II}} = \overline{R}_{\Gamma_{II}}.$$

To complete the proof it remains to show that $\overline{R}_{\Gamma_{II}}$ is independent of $C_v$ or equivalently is independent of $\alpha$. This follows because $\Gamma_{II}$ is a system where jobs arrive with rate $\lambda$ and have exponential demands with mean $\alpha/\mu_i$, both of which are independent of $\alpha$. We have $\alpha/\mu_i$ independent of $\alpha$ because from (7.10)

$$\frac{\alpha}{\mu_i} = \frac{\alpha}{\alpha/f_i(N_i, E_i, C_i)} = f_i(N_i, E_i, C_i),$$

and $f_i$ was assumed to be independent of $\alpha$ in Section 7.2.1. ∎

It is clear that the above proof also holds for policies other than EQ that provide instant service to type 2 jobs and do not make use absolute values of job arrival times and/or demands in scheduling jobs. That is, they may use the relative ordering of arrival times and/or demands as in PSCDF but not the values of arrival times and demands themselves. This assumption is needed because in the sample path analysis we used the fact that $\Gamma_{II}$ is simply a time compressed version of $\Gamma_I$, which would not hold if processor allocation was based on absolute values of arrival times and/or demands.

#### 7.2.2.2 $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_v^2$

**Theorem 7.2.2** *Under the workload assumptions of Section 7.2.1, $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_v^2$.*

**Proof.** See Appendix A.3.3. ∎

The result for FCFS and PSAPF extends to policies that do not always provide instant service to jobs, do not differentiate between type 1 and type 2 jobs (for the GE distribution), in which type 1 jobs are not affected by type 2 jobs and as before in which absolute values of job arrival times and/or demands is not used for scheduling decisions. Policies other than FCFS and PSAPF that satisfy these requirements include SAPF and AP/DA Fixed Priority policies.

## 7.3 Generalization and Unification of Previous Work

The policy comparisons that we have shown for the ASP, EQS, FCFS, and PSAPF policies in Section 7.1.1-7.1.3 enable us to delineate regions of the parameter space under which each policy performs best. Using the results derived from analytic comparisons in Sections 7.1.1-7.1.3 assuming the linear ERF, and extending the results for sublinear ERFs we obtain the delineation shown in Figure 7.7. The key determinants of relative policy performance are labelled along the axes of the figure. $C_v$ is a key parameter in all comparisons in this chapter, correlation between mean demand and available parallelism determines relative orderings between FCFS and PSAPF and between PSAPF and EQS, and the ERF sublinearity also affects relative policy performance since at very sublinear ERFs EQS will be perform best for all $C_v \geq 0$ and all $0 < r < 1$. The parameter $S_n$ is not shown in Figure 7.7, even though it is a key determinant of absolute performance for ASP and EQS, because the performance of EQS is better than that of ASP throughout the range of $S_n$.

We analytically derived results only for the topmost plane where the ERF is linear and the comparison is favorable to ASP, FCFS, and PSAPF with respect to EQS. In particular, the orderings between PSAPF and EQS, between FCFS and EQS at $r = 0$, and between FCFS and PSAPF at $r = 0$ were shown analytically. To supplement the results from analysis and complete the topmost plane of Figure 7.7 we assume the following specific results (1) FCFS performs as well or better than ASP when $C_D = 0$ (i.e., in the narrow regions where ASP performs marginally better than EQS), (2) $\overline{R}_{ASP}(C_D = 1)$ is a lower bound for its performance when $C_D > 1$, and (3) the simulation results for PSAPF versus FCFS when $r = 1$ (Section 7.1.3) hold generally for all distributions of demand and parallelism. The exact crossover of FCFS and PSAPF in the topmost plane where $C_v \leq 1$ and $r > 0$ has not been derived in this chapter and is thus indicated by the line break along the boundary. Extending the results from the topmost plane to sublinear ERFs makes use of Theorem 7.1.1 which shows that EQS should perform relatively better as the ERF sublinearity increases. The crossover points of best performance for $C_v \leq 1$ and sublinear $\gamma$ are again shown by the line breaks and the exact crossover points may depend not only on the specific ERF but also on the available parallelism distribution. However, for $C_v \geq 1$ and $r = 0$ and $C_v \geq 2$ and $r = 1$ the result that EQS performs best holds for all distributions of $N$.

Due to the general workload assumptions in this chapter the delineation generalizes and unifies previous results, as follows. First consider the line for $r = 0$ and $\gamma = \gamma^l$, and variable $C_v$.

- A previous study shows that PSAPF, FCFS, and EQ have almost the same performance at $(C_v = 1, r = 0, \gamma^l)$ [41]. This is shown in Figure 7.7 for FCFS and EQS[5] and approximation (7.4) shows that $\overline{R}_{PSAPF} \approx \overline{R}_{FCFS}$ under these conditions.

- For an uncorrelated workload ($r = 0$) with specific hyperexponential demand distributions ($C_v > 1$), specific distributions of $N$, and linear speedups, [41] shows that $\overline{R}_{EQ} < \overline{R}_{PSAPF}, \overline{R}_{FCFS}$. Figure 7.7 shows the same result for all distributions of demand and parallelism.

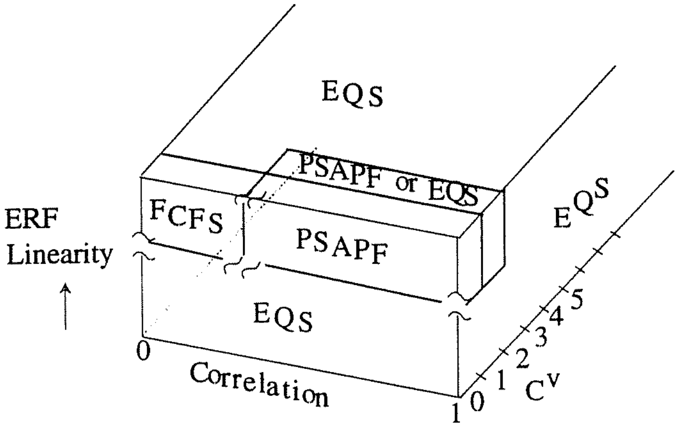


Figure 7.7: Summary of Policy Comparison Results

Now consider the line $r = 1$, $\gamma = \gamma^l$, and variable $C_v$.

- Leutenegger and Nelson [40] compare PSAPF, FCFS, EQ and several other policies for a workload with a fixed number of jobs having i.i.d. *exponential* task service times (for which $C_v < 1$ and $r = 1$) and linear task execution rates. They show PSAPF to be the optimal policy for the workload. This is consistent with Figure 7.7 which shows that *in*

[5] Note that when $\gamma$ is linear, all EQ policies have the same performance under the assumption of $E(j) = \gamma(j)$.

*general* for $C_v < 1$, $r = 1$, and $\gamma = \gamma^l$, PSAPF is optimal among ASP, FCFS, EQ, and PSAPF.

- Leutenegger and Vernon [41] show that for specific hyperexponential demands, specific parallelism distributions, full correlation ($r = 1$), and linear speedups, $\overline{R}_{EQ} < \overline{R}_{PSAPF}$. Leutenegger [39] gives additional simulation data for specific distributions of parallelism, other assumptions being the same, which show PSAPF to perform better than EQ if $C_v < 2$ and worse if $C_v > 2$. [39] also shows cases with $C_v$ between 1 and 2 where PSAPF performs worse than EQ. Again these results are consistent with Figure 7.7.

Now consider $0 < r < 1$, which is the case in [69, 48, 49] where no quantitative measures of workload correlation are given. These studies show that for particular workloads with sublinear ERFs, EQS outperforms ASP under exponential per class demands ($C_v = 1$) [69] and under a specific mix of applications with $C_v > 1$ [48, 49]. The same result is shown in Section 7.1.1 and Figure 7.7 for all distributions of demand and parallelism.

Other results in the literature show that $\overline{R}_{PSAPF} < \overline{R}_{FCFS}$ for hyperexponential demands ($C_v > 1$), specific distributions of parallelism, and both $r = 0$ and $r = 1$ [43, 41]. We have shown the same result (Section 7.1.2) for all distributions of demand with $C_v > 1$ and all distributions of $N$. Finally, [80] shows FCFS to outperform Round Robin Process and Processor Sharing for i.i.d. generalized exponential task service times with coefficient of variation < 4. Note that for this model $r = 1$ and Figure 7.7 shows that if the $C_v$ of the sum of task service times is low, then PSAPF performs better than FCFS whereas if $C_v$ is high then EQS performs better.

## 7.4 Conclusions

We have compared the performance of four parallel processor allocation policies, ASP, EQS, FCFS, and PSAPF that were shown in previous literature to have high performance under specific workloads. The comparisons were made over a general workload model that includes controlled correlation between demand and parallelism, general distribution of available parallelism, general demands for jobs with no correlation, and a general deterministic job execution rate function for all jobs. Under the assumption that jobs can dynamically and efficiently redistribute their work across the processors allocated to them, the mean response time of each policy was estimated using interpolation approximations for various regions of the parameter space. We showed the mean response time formulas to be accurate and used them to obtain key determinants of policy performance. By using the key parameters to explore the design space we generalized and unified previous policy comparison results. The regions of the parameter space under which each policy performs best are delineated in Figure 7.7.

The main results of this chapter are as follows:

- Coefficient of variation in demand ($C_D$) can be critical in determining *relative* policy performance. This might be obvious from uniprocessor scheduling results, but most previous analyses and comparisons of parallel processor policies have assumed exponential demands or exponential task service times. This result shows that it is important to consider the implications of the exponential assumption on the conclusions reached.

- Sublinearity of speedup curves and correlation between demand and parallelism are also influential parameters for relative policy performance. While speedup curves have been explicitly specified in previous studies, in many cases no indication of correlation is provided. This result shows that it is important to provide correlation information about the workload and consider its implications on relative policy performance.

- The EQS policy has superior processor allocation characteristics in terms of processor efficiency. More specifically, for a fixed set of jobs with a common nondecreasing and concave execution rate function the EQS policy achieves optimal processor utilization over all allocation policies.

- EQS substantially outperforms ASP for both uncorrelated as well as correlated workloads. This result is shown for more general demand and parallelism distributions than in previous studies.

- EQS outperforms PSAPF when $C_D$ is moderate to high even when workload correlation is high. PSAPF has lower mean response time tha EQ only for highly correlated workloads at low to moderate values of $C_D$, and when execution rates are (close to) linear. These results are hold for all distributions of available parallelism and general distributions of demand. PSAPF outperforms FCFS for most of the parameter space. (FCFS has slightly lower mean response time when correlation is low, $C_D < 1$, and job execution rate is linear.)

- Since general purpose computer systems are likely to have high variation in job processing requirements, the EQS policy seems to be the best candidate for implementation among the policies considered in this thesis.

The policies examined in this study have been idealizations of practical processor allocation policies since we have assumed zero scheduling and preemption overheads. Our results should continue to hold for practical (approximate) implementations of these policies that ensure that the overheads are small compared to job service times. In this thesis we assume that applications can dynamically and efficiently redistribute their work among allocated processors. In environments where this is not possible, based on the results of this chapter, a natural candidate policy to consider is temporal equiallocation.

# Chapter 8

# Further Analysis of the EQS Policy

Chapter 7 showed that among ASP, EQS, FCFS, and PSAPF, the EQS policy has highest performance for most of the practical regions of the design space under our workload model. The key properties that causes EQS to outperform the other three policies are the insensitivity of $\overline{R}_{EQS}$ to coefficient of variation, $C_D$, in job demand and the higher efficiency in processor allocation.

We have not yet examined thoroughly how the EQS policy performs with respect to workload parameters other than $C_D$[1]. More specifically, we need to explain the following.

- How does $\overline{R}_{EQS}$ behave as a function of mean demand $\overline{D}$?
- What measures of available parallelism, job execution rates, and correlation between demand and parallelism are key determinants of the performance of the EQS policy?
- How does $\overline{R}_{EQS}$ behave as a function of available parallelism? For example, how does its performance change with changes in the distribution of workload parallelism?
- How do communication and synchronization overheads, and load imbalance of parallel programs affect the performance of EQS?
- How does $\overline{R}_{EQS}$ behave as a function of workload correlation?

The above questions have not been thoroughly studied in the literature for equipartitioning policies. In this chapter we answer all these questions using sample path analysis as well as the

[1] We have used the property that $\overline{R}_{EQS}$ is (almost) uniquely determined by $S_n$ and in this chapter we will explain how $S_n$ turns out to be the key parameter.

approximate analysis of Chapter 5. Using sample path analysis we derive bounds on $\overline{R}_{EQS}$ that show that under exponential job processing requirements (demands) and any concave nondecreasing job execution rate function for all jobs $\overline{R}_{EQS}$ is minimum when the all jobs are fully parallel and is maximum when all jobs are fully sequential. From Theorem 7.2.1 in Chapter 7 it follows that this lower bound also holds for generalized exponential distributions. Further proofs show that the upper bound holds under more general workload conditions that include general interarrival times, general demands, general available parallelism, and general nondecreasing execution rates, with arbitrary dependencies among these workload variables.

Using approximate analysis we derive the key parallelism parameter for $\overline{R}_{EQS}$ and use it to study the behavior of EQS as a function of available parallelism, ERF sublinearity, and correlation.

In Section 8.1 we derive mean response time bounds for EQS. Section 8.2 studies the qualitative behavior of $\overline{R}_{EQS}$ for uncorrelated workloads as a function of workload parameters. In Section 8.3 we generalize our analysis to correlated workloads. Finally, in Section 8.4 we summarize the results derived in this chapter and relate them to results that have appeared in the literature.

## 8.1 Mean Response Time Bounds for the EQS Policy

In this section we first derive lower and upper bounds on $\overline{R}_{EQS}$ for the workload ($\mathcal{F}_N$, $\exp(1/\overline{D})$, $r = 0$, $\gamma \in \mathcal{E}^c$), where $\mathcal{E}^c$ is the class of concave and nondecreasing ERFs. These bounds show that the mean response time is minimum when all jobs are fully parallel (i.e., $N = P$) and is maximum when all jobs are fully sequential (i.e., $N = 1$), all else being equal. We then show that the upper bound for $\overline{R}_{EQS}$ holds under more general workload assumptions, which include general job arrival times, job demands, available parallelisms, and execution rates, with arbitrary dependencies among these workload variables. The lower and upper bounds in this section are generalizations of the bounds in [1] for the EQS policy, and are obtained as corollaries of more general bounds, which show that the performance of EQS improves with "increase" in available parallelism. In [1] it was shown that the mean response time of any processor conserving policy[2] under *exponential* job demands and *linear* job execution rates is minimum when $N = P$ and maximum when $N = 1$. Note that the generalizations below are only with respect to the EQS policy and do not hold for all processor conserving policies.

[2] A processor conserving policy does not allocate more processors to a job than the job can productively make use of, and it does not leave a processor idle if any job can make use of that processor.

### 8.1.1 Lower and Upper Bounds: $\mathcal{F}_D^u = \exp,\ r = 0,\ \gamma \in \mathcal{E}^c$

We show that under the workload $(\mathcal{F}_N, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c)$, the performance of EQS is optimal when all jobs are fully parallel and is pessimal when all jobs are fully sequential. Note that these bounds are derived assuming $N$ and $D$ are independent, $D$ is exponential, the workload ERF is concave and nondecreasing, and each job can dynamically redistribute its work across its processor allocation. The assumption of exponential job demand is probably not a serious limitation in this case, since the approximate analysis in this thesis as well as the simulation experiments reported in this and previous papers indicate that $\overline{R}_{EQS}$ depends only on mean demand and not on distribution of demand.[3]

The bounds follow as an immediate consequence of the following theorem.

**Theorem 8.1.1** *If $\ell$ and $m$ are constants such that $\ell \le m$, then under the workload assumptions* $(\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c)$,

$$\overline{R}_{EQS}(m \le N \le P) \quad \le \quad \overline{R}_{EQS}(1 \le N \le \ell).$$

**Proof.** See Appendix B. ■

The intuition for Theorem 8.1.1 is that whenever the number of jobs in each system is equal, the total job completion rate in the system with higher available parallelism is greater than or equal to the job completion rate in the other system.

Setting $\ell = m = P$ in Theorem 8.1.1 we obtain the following lower bound on $\overline{R}_{EQS}$:

**Corollary 8.1.1** *Under the workload assumptions* $(\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c)$,

$$\overline{R}_{EQS}(\mathcal{F}_N) \quad \ge \quad \overline{R}_{EQS}(N = P).$$

In [1] a corresponding bound was given for all processor conserving policies assuming exponential demands and the linear ERF. As in [1] we note that a tighter lower bound can be obtained when $\overline{N} \ne P$, by using the fact that $\overline{R}_{EQS} \ge \overline{S}$. This yields the following bound, which we henceforth refer to as the $N = P$ lower bound:

$$\overline{R}_{EQS}(\mathcal{F}_N) \quad \ge \quad \max\left\{\overline{S},\ \overline{R}_{EQS}(N = P)\right\}, \quad \text{under } (\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c). \tag{8.1}$$

Setting $m = \ell = 1$ in Theorem 8.1.1 we obtain following the bound on $\overline{R}_{EQS}$, which we henceforth refer to as the $N = 1$ upper bound:

**Corollary 8.1.2**

$$\overline{R}_{EQS}(\mathcal{F}_N) \quad \le \quad \overline{R}_{EQS}(N = 1), \quad \textit{under } (\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c). \tag{8.2}$$

[3] The bounds also hold for the generalized exponential distribution since from Theorem 7.2.1 $\overline{R}_{EQ}$ is the same under exponential and generalized exponential demands.

For the linear ERF, the bounds in (8.1) and (8.2) can be shown to reduce to the following [1]:

$$\max\left(\overline{D}E[1/N], \frac{1}{1-\rho}\frac{\overline{D}}{P}\right) \le \overline{R}_{EQS}(\mathcal{F}_N, \exp(1/\overline{D}), r=0, \gamma^l) \le \overline{R}_{M/M/P}.$$

### 8.1.2 Experimental Evaluation of the $N = P$ and $N = 1$ Bounds

For the workload $(\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c)$ and for many distributions of $N$, the mean system response time will lie closer to the $N = P$ lower bound than the $N = 1$ upper bound, primarily because the lower bound is the maximum of the mean service time and the mean reponse time when $N = P$. For example, for a given distribution of $N$, $N \not\equiv 1$, the $N = P$ bound is exact when $\rho \to 0$ but this is not true of the $N = 1$ bound. We further illustrate this point by comparing simulation estimates of $\overline{R}_{EQS}$ against the bounds for a 100 processor system, the H and L distributions of $N$ given in Table 3.2, exponential job demand $D$ with mean $\overline{D} = P = 100$, and ERF $\gamma(k) = k^{0.8}$, $k = 1, 2, \ldots, P$.[4] As seen from Figure 8.1a and b the $N = 1$ upper bound is rather loose for workloads with high average available parallelism, but is much tighter when average available parallelism is low. Conversely, $\overline{R}_{EQS}(N = P)$ is tighter for the H workload, but looser for the L workload. Taking the maximum of $\overline{S}$ and $\overline{R}_{EQS}(N = P)$, i.e., the $N = P$ bound, results in a tight bound for both high and low average available parallelism.

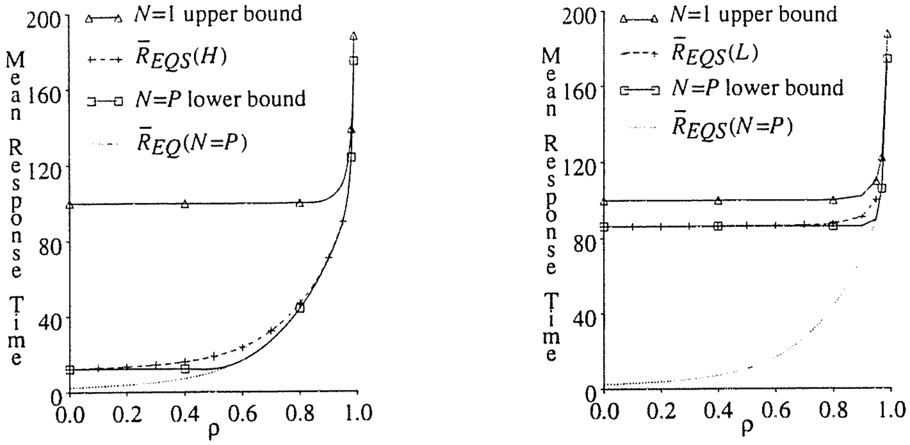
### 8.1.3 Upper Bound under General Workloads

We now show that the $N = 1$ upper bound holds under more general workload assumptions, i.e., general arrivals, general available parallelism, general demands, and general nondecreasing execution rates, with arbitrary dependencies among these workload variables. The upper bound follows as a direct consequence of the following theorem.

**Theorem 8.1.2** *Let $\Gamma_I$ be a system with the EQS policy and primitive workload variables $\{(A_i, D_i, N_i \ge k, E_i), i = 1, 2, \ldots\}$, where $A_i$ is job i's arrival time, $D_i$ its total demand, $N_i$ its available parallelism, and $E_i$ its execution rate function. Let these primitive variables have arbitrary marginals (given that $N_i \ge k$, and the other variables make sense, e.g., $D_i \ge 0$) with arbitrary dependencies among them. Let $\Gamma_{II}$ be a system with the EQS policy and the same workload as $\Gamma_I$ except that $N_i = k$ for all $i = 1, 2, \ldots$. Then*

$$\overline{R}_{\Gamma_I} \le \overline{R}_{\Gamma_{II}}, \quad k = 1, 2, \ldots, P.$$

[4] All simulation experiments in this chapter have 95% confidence intervals with less than 5% half-widths in almost all cases, and less than 10% otherwise. The confidence intervals were generated using the regenerative method whenever feasible and otherwise the method of batch means.



(a) $H$ workload

(b) $L$ workload

Figure 8.1: Tightness of N=1 and N=P bounds for $\overline{R}_{EQ}$: $D \sim exp$, $r = 0$

ERF: $\gamma(k) = k^{0.8}$,
$\overline{D} = P = 100$

**Proof.** See Appendix B. ∎

The intuition for Theorem 8.1.2 is that system $\Gamma_I$ allocates at least as much processing power to each unfinished job as $\Gamma_{II}$ does.

Setting $k = 1$ into Theorem 8.1.2 we obtain the following result.

**Corollary 8.1.3** *Let $\Gamma_I$ be a system with the EQS policy and primitive workload variables $\{(A_i, D_i, N_i, E_i), i = 1, 2, \ldots\}$. Let these primitive variables have arbitrary marginals with arbitrary dependencies among them. Let $\Gamma_{II}$ be a system with the EQS policy and the same workload as $\Gamma_I$ except that $N_i = 1$ for all $i = 1, 2, \ldots$. Then*

$$\overline{R}_{\Gamma_I} \leq \overline{R}_{\Gamma_{II}}, \quad k = 1, 2, \ldots, P.$$

*More specifically,*

$$\overline{R}_{EQS}(\mathcal{F}_N) \quad \leq \quad \overline{R}_{EQS}(N = 1), \quad \textit{under } (\cdot, \mathcal{F}^u_D, r, \gamma).$$

If we consider only constant values of $N$ in Theorem 8.1.2 we get the following corollary.

**Corollary 8.1.4** *Consider a system with the EQS policy with general $\{(A_i, D_i, E_i), i = 1, 2, \ldots\}$ (i.e., these primitive variables have arbitrary marginals with arbitrary dependencies among them). Then*

$$\overline{R}_{EQS}(N = P) \leq \ldots \leq \overline{R}_{EQS}(N = k) \leq \ldots \leq \overline{R}_{EQS}(N = 1), \quad k = P, \ldots, 2.$$

*where $N = k$ denotes $N_i = k$, for all $i = 1, 2, \ldots$.*

This corollary shows that for workloads with constant available parallelism the performance of EQS improves as available parallelism increases. This result is generalized in Section 8.2.2.

## 8.2 Behavior of $\overline{R}_{EQS}$ with respect to Key Parameters: Uncorrelated Workloads ($r = 0$)

For the sake of simplicity we focus on uncorrelated workloads in this section and then generalize the results for correlated workloads in Section  8.3. To determine the functional dependence of EQS on key parameters we first need to identify which workload parameters are key determinants of policy performance. Section 8.2.1 points out that it is quite straightforward to determine the key parameter of job demand and discusses how $\overline{R}_{EQS}$ varies with the key demand parameter. In Section 8.2.2 the behavior of $\overline{R}_{EQS}$ is examined as a function of several different parameters of available parallelism and the key parallelism parameter is identified. Section 8.2.3 presents insights into the behavior of EQS as a function of sublinearity in the workload ERF and compares the performance of EQS versus EQT, and finally, Section 8.2.4 presents a summary of the results.

### 8.2.1 $\overline{R}_{EQS}$ as a function of job demand

All interpolation approximations that we have derived for $\overline{R}_{EQS}$ and all exact reductions for EQS (see Chapter 5) show that when $r = 0$, $\overline{R}_{EQS}$ depends only on $\overline{D}$ and not on higher moments of demand. Simulation experiments have also verified this result for specific demand distributions such as deterministic, $Er_2$, exponential, $H_2$, and Gamma. Furthermore, we also saw from Theorem 7.2.1 that $\overline{R}_{EQS}$ is insensitive to $C_D$ for a GE demand distribution.

The dependence of $\overline{R}_{EQS}$ on $\overline{D}$ can be readily obtained from the interpolation approximations. For a given $\rho$, $\overline{R}_{EQS}$ is directly proportional to $\overline{D}$. See for example approximation (5.5). Also note that $\overline{R}_{EQS}(N = k)$ given by equation (5.1) in Section 5.1.1 is directly proportional to $\overline{D}$ for a given $\rho$ (because $\lambda = \rho P/\overline{D}$). Hence approximations (5.6) and (5.7) are also directly proportional to $\overline{D}$. Thus $\overline{R}_{EQS}$ increases linearly in $\overline{D}$ given that all other workload variables remain unchanged.

### 8.2.2 $\overline{R}_{EQS}$ as a function of available parallelism

To understand the behavior of $\overline{R}_{EQS}$ as a function of available parallelism, $N$, we need to know which parameters of $N$ are principal determinants of $\overline{R}_{EQS}$. Natural candidates are $\overline{N}$ and $C_N$.

$$\begin{array}{ll}
\textbf{minimize} \displaystyle\sum_{k=1}^{P} R_k \, p_k & \textbf{maximize} \displaystyle\sum_{k=1}^{P} R_k \, p_k \\
\text{subject to:} & \text{subject to:} \\
\text{(i) } \underline{p} \geq 0 & \text{(i) } \underline{p} \geq 0 \\
\text{(ii) } \displaystyle\sum_{k=1}^{P} p_k \ = \ 1 & \text{(ii) } \displaystyle\sum_{k=1}^{P} p_k \ = \ 1 \\
\text{(iii) } \displaystyle\sum_{k=1}^{P} f(k) \, p_k \ = \ E[f(N)] = a & \text{(iii) } \displaystyle\sum_{k=1}^{P} f(k) \, p_k \ = \ E[f(N)] = a
\end{array}$$

Figure 8.2: Linear Programs for Min and Max of $\overline{R}_{EQS}$

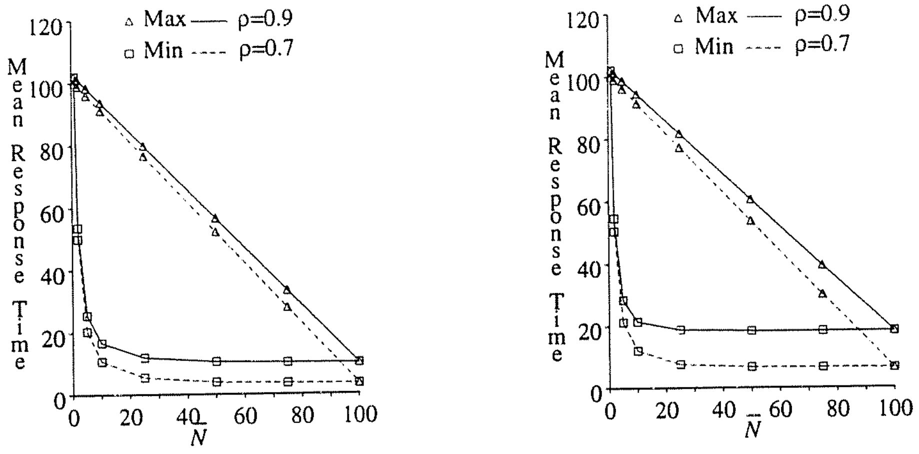
Another measure of $N$ that could be a key determinant when $r = 0$ is $E[1/\gamma(N)]$ since the mean service time is $\overline{D}E[1/\gamma(N)]$.

A possible approach to determining if a given parameter of $N$, say $E[f(N)]$, uniquely determines $\overline{R}_{EQS}$ is to test whether $\overline{R}_{EQS}$ remains unchanged across *all* distributions of $N$ that have a given $E[f(N)]$, for each possible value of $E[f(N)]$. In other words, if $\min\{\overline{R}_{EQS}(\mathcal{F}_N) : E[f(N)] = x\} = \max\{\overline{R}_{EQS}(\mathcal{F}_N) : E[f(N)] = x\}$ for all feasible $x$, then $E[f(N)]$ is a parameter that uniquely determines $\overline{R}_{EQS}$. To use this approach we must obtain the minimum and maximum of $\overline{R}_{EQS}(\mathcal{F}_N, r = 0)$ over all distributions of $N$ for each value of $E[f(N)]$. A key observation about the accurate approximation (5.7) is that $\overline{R}_{EQS}(N = k, r = 0)$ does not depend on the pmf, $\underline{p}$ (see equation (5.1)). Thus, for given fixed values for $\lambda$, $\overline{D}$, and $\gamma$, $\hat{R}^{\underline{p}}_{EQ}$ can be viewed as a linear combination of the $p_k$'s and we can use linear programming [17] to obtain the minimum and maximum mean response times. The generic form of the linear program is given in Figure 8.2, where $R_k$ denotes $\overline{R}_{EQS}(N = k)$.

Below, the linear programs of Figure 8.2 are used to determine whether $\overline{N}$, $C_N$, or $E[1/\gamma(N)]$ uniquely determine $\overline{R}_{EQS}$.

#### 8.2.2.1 $\overline{R}_{EQS}$ versus $\overline{N}$

Setting $f(N)] = N$ in Figure 8.2 we obtain linear programs that minimize and maximize the estimator $\hat{R}^{\underline{p}}_{EQ}$ for a given $\overline{N}$, $\lambda$, $\overline{D}$, and $\gamma$ over all possible pmfs $\underline{p}$ such that the expected value of $N$ is $\overline{N}$. For $P = 100$ and specific values of $\lambda$, $\overline{D}$, and $\gamma$, the linear programs were solved for $\overline{N} = 1, 2, 5, 10, 25, 50, 75, and 100$ using the Simplex Method of linear programming [17]. Figures 8.3a and b plot the envelopes obtained by the minimum and maximum values of $\hat{R}^{\underline{p}}_{EQ}$ versus $\overline{N}$ for $\overline{D} = P$, two different ERFs, and two different values of $\rho = \lambda\overline{D}/P = \lambda$. The minimum value of $\hat{R}^{\underline{p}}_{EQ}$ for a given $\overline{N}$ was obtained for a distribution of $N$ with low $C_N$ (typically $K_2(\lfloor\overline{N}\rfloor, \lceil\overline{N}\rceil, \lceil\overline{N}\rceil - \overline{N})$). The maximum value was obtained for the $K_2(1, P, \frac{P-\overline{N}}{P-1})$ distribution of $N$.



(a) Linear ERF

(b) $\gamma(k) = 101k/(k + 100)$

Figure 8.3: Envelopes of $\overline{R}_{EQS}$ versus $\overline{N}$

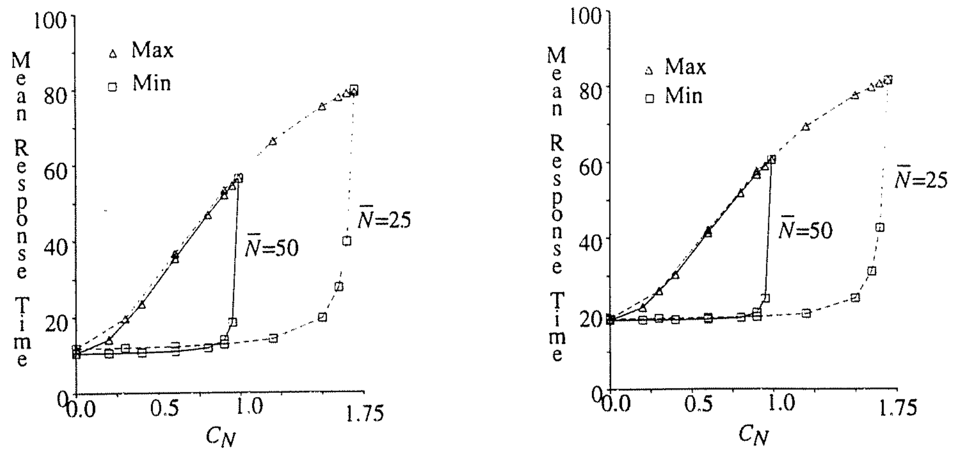
$\overline{D} = P = 100$

Figure 8.3 clearly shows that for uncorrelated workloads, $\overline{N}$ alone does not adequately capture the influence of $\mathcal{F}_N$ on the behavior of $\overline{R}_{EQS}$. For example, at $\rho = 0.9$ and $\overline{N} = 25$ in Figure 8.3a, $\hat{R}^{\underline{p}}_{EQ}$ ranges from a minimum of 11.88 when $N = 25$, to a maximum of 79.83 when $N$ has the $K_2(1, 100, \frac{75}{99})$ distribution.

Although $\overline{N}$ does not in general uniquely determine $\overline{R}_{EQS}$, the envelopes in Figure 8.3 provide useful bounds on $\overline{R}_{EQS}$ and lead to two useful observations. First, for each of the given parameter settings and across all distributions of $N$, $\hat{R}^{\underline{p}}_{EQ}$ is maximum when $N = 1$ and minimum when $N = P$. This is consistent with the bounds for $\overline{R}_{EQS}$ that were derived in Section 2.4, where the upper bound was derived for general demands and the lower bound was derived for exponential demands. For the envelopes in Figure 8.3 job demand has a general distribution. Second, the plots for the maximum value of $\hat{R}^{\underline{p}}_{EQ}$ versus $\overline{N}$ in Figures 8.3 reveal an interesting property of the $K_2(1, P, \frac{P-\overline{N}}{P-1})$ distribution of $N$ – namely, that the response time for this distribution decreases linearly as the mean available parallelism increases (i.e. as the fraction of fully parallel jobs increases). This observation is only for a specific distribution of $N$; results below show that the result also holds for other distributions of $N$.

### 8.2.2.2 $\overline{R}_{EQS}$ versus $C_N$

We next examine whether $C_N$ and $\overline{N}$ together uniquely determine $\overline{R}_{EQS}$ for a given $\overline{D}$, $\lambda$, and $\gamma$. Figures 8.4a and b plot envelopes of $\hat{R}^{\underline{p}}_{EQ}$ versus $C_N$ for two values of $\overline{N}$ and two different ERFs, for systems with $P = 100$, $\overline{D} = P$, and $\rho = 0.9$. As before, the envelopes were obtained



(a) Linear ERF

(b) $\gamma(k) = 101k/(k + 100)$

Figure 8.4: Envelopes of $\overline{R}_{EQS}$ versus $C_N$

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

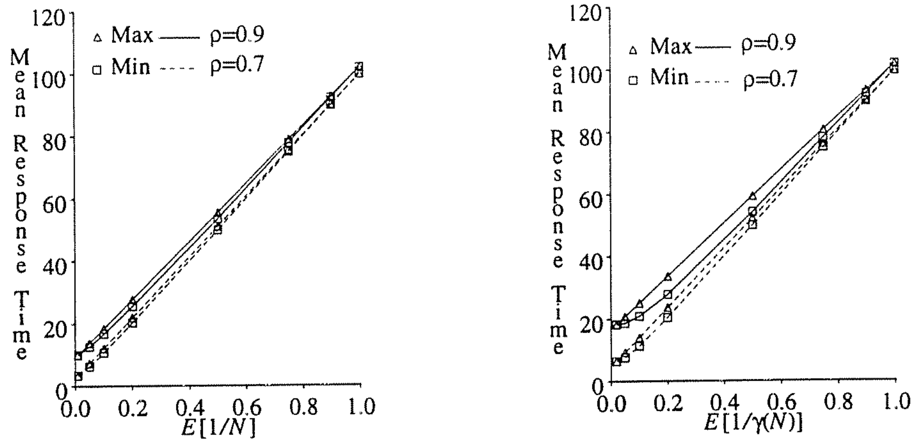
the linear programming. Note that for each value of $\overline{N}$, the range of $C_N$ is constrained as specified in Theorem 3.5.1. As was the case in Figure 8.3 the envelopes of $\hat{R}^{\underline{p}}_{EQ}$ versus $C_N$ are very similar for both types of ERFs. The envelopes also have similar shape and orientation for both values of $\overline{N}$ and for different values of $\rho$ (not shown). However, unlike the envelopes for $\overline{N}$ there is no particular pattern to the distributions of $N$ that yield the minimum or maximum value of $\hat{R}^{\underline{p}}_{EQ}$ at different values of $C_N$.

The plots in Figure 8.4 show that $C_N$ and $\overline{N}$ together are not sufficient to determine the behavior of $\overline{R}_{EQS}$ as a function of workload parallelism. However, the envelopes show that, for the parameter values examined, $\hat{R}^{\underline{p}}_{EQ}$ is minimum when $C_N$ is minimum and is maximum when $C_N$ is maximum, and that the range of possible mean response times is low for low $C_N$.

#### 8.2.2.3 $\overline{R}_{EQS}$ versus $E[1/\gamma(N)]$

The linear programs in Figure 8.2 with $f(N) = 1/\gamma(N)$ is used next to obtain envelopes of $\hat{R}^{\underline{p}}_{EQ}$ versus $E[1/\gamma(N)]$ for given values of $\overline{D}$ and $\lambda$, and a given function $\gamma$. Note that $E[1/\gamma(N)]$ can vary from $1/\gamma(P)$ (when $N = P$) to 1 (when $N = 1$). Figure 8.5a and b plots the envelopes for two different ERFs and two different values for $\rho$, given that $P = 100$ and $\overline{D} = P$.

For the linear ERF in Figure 8.5a we observe that there is very little spread between the minimum and maximum values of $\overline{R}_{EQS}$ for a fixed value of $E[1/N]$. For sublinear ERFs as in



(a) Linear ERF

(b) $\gamma(k) = 101k/(k+100)$

Figure 8.5: Envelopes of $\overline{R}_{EQS}$ vs $E[1/\gamma(N)]$

$\overline{D} = P = 100$

Figure 8.5b the spread is somewhat larger but is still quite small. These results indicate that $E[1/\gamma(N)]$ almost uniquely determines $\overline{R}_{EQS}(r=0)$ and is thus the key parameter of available parallelism for uncorrelated workloads.

The qualitative behavior of $\overline{R}_{EQS}$ versus the key parameter $E[1/\gamma(N)] = \overline{S}/\overline{D} \equiv S_n$ yields the following insights into the performance of EQS as a function of $N$ when $r = 0$.

(1) Since $\overline{R}_{EQS}$ increases nearly linearly in $E[1/\gamma(N)]$, a workload with a lower value of $E[1/\gamma(N)]$ has a smaller mean response time.

(2) Since $1/\gamma(P) \leq E[1/\gamma(N)] \leq 1/\gamma(1)$ it follows that $\overline{R}_{EQS}(N=P) \leq \overline{R}_{EQS}(\mathcal{F}_N) \leq \overline{R}_{EQS}(N=1)$. That is, $N = P$ *is optimal and* $N = 1$ *is pessimal* for the workload ($\lambda$, $\mathcal{F}_N$, $\mathcal{F}^u_D$, $r = 0$, $\gamma$), which is generalization of the mean response time bounds of Section 2.4.

(3) We next consider distributions of $N$ between the two extremes of $N = 1$ and $N = P$. In particular, consider two distributions $\mathcal{F}_{N_1}$ and $\mathcal{F}_{N_2}$ such that $N_2 \leq_{st} N_1$ (i.e., $P[N_2 \leq n] \geq P[N_1 \leq n]$, $1 \leq n \leq P$). Under this condition it is shown in [62] that $E[f(N_1)] \leq E[f(N_2)]$ for any nonincreasing function $f$. Setting $f = 1/\gamma$ it follows that $E[1/\gamma(N_1)] \leq E[1/\gamma(N_2)]$ and thus $\overline{R}_{EQS}(\mathcal{F}_{N_1}) \leq \overline{R}_{EQS}(\mathcal{F}_{N_2})$. Thus, *a stochastic increase in available parallelism leads to a decrease in mean response time* for the EQS policy. Hence the EQS policy does not discourage and may encourage users to increase program parallelism (up to the point where the ERF is nondecreasing).

(4) A stochastic increase in parallelism also increases the mean parallelism. What if the mean parallelism is the same but the variability in parallelism changes? More precisely, consider $\overline{N}_1 = \overline{N}_2$ and $N_1 \leq_v N_2$, which means that $E[f(N_1)] \leq E[f(N_2)]$ for all convex functions $f$ [62]. If $\gamma$ is concave then $1/\gamma$ is convex and it follows that $\overline{R}_{EQS}(N_1) \leq \overline{R}_{EQS}(N_2)$ if $N_1 \leq_v N_2$. Thus the mean response time of EQS decreases with a decrease in variability of $N$ if $\gamma$ is concave and $\overline{N}$ remains fixed. Note that for the bounded distributions considered in this thesis the highest variability in $N$ for a fixed $\overline{N}$ is when $N$ has a $K_2(1, P, .)$ distribution and the least variability in $N$ is when $N$ is constant. Also recall (from 3.5.1) that for a given $\overline{N}$, the $K_2(1, P, \cdot)$ has the highest $C_N$ and the constant distribution has the lowest $C_N$. Thus for a given $\overline{N}$, $\overline{R}_{EQS}$ is maximum when $C_N$ is highest and is minimum when $C_N$ is lowest. This results generalizes the corresponding results for specific workloads in Figures 8.3 and 8.4.

Note that results (1) and (2) above contrast with studies of fork-join queueing systems that have shown parallelism to be harmful for other scheduling disciplines, particularly at high loads [45, 67, 13].

Another consequence of the (nearly) linear increase in $\overline{R}_{EQS}$ as a function of $E[1/\gamma(N)] = \overline{S}/\overline{D}$ is that the interpolation approximation on $E[1/\gamma(N)]$ for $\overline{R}_{EQS}$ (see (5.8) is likely to be accurate, which was verified in the validations for the approximation. (More than 95% of the validations had relative errors between -5% and 15%, and the maximum relative error was about 30%.)

### 8.2.3 $\overline{R}_{EQS}$ as function of ERF sublinearity

Intuition suggests that system performance should improve with a decrease in synchronization and communication overheads. This is also shown analytically, since an increase in $\gamma$ decreases $E[1/\gamma(N)]$ which in turn decreases $\overline{R}_{EQS}$. This section addresses the following further questions about the behavior of $\overline{R}_{EQS}$ as a function of ERF sublinearity:

- How stable is the system as a function of ERF sublinearity?
- Precisely how does $\overline{R}_{EQS}$ behave as the ERF sublinearity increases for given functional forms of $\gamma$
- How does the behavior of $\overline{R}_{EQS}$ change with the functional form of $\gamma$?
- Under the linear ERF, spatial and temporal equipartitioning have the same performance assuming an identical allocation of *processing power*. How does the behavior of spatial equipartitioning differ from that of temporal equipartitioning when the ERF is sublinear?

(i) **System stability versus degree of sublinearity**

Under the assumptions of negligible preemption and scheduling overhead, and $E(x) = x$ for $0 \leq x \leq c$ where $c$ is a constant greater than zero, the stability condition for a system with the EQS scheduling policy for any ERF $\gamma$ is the same as for the linear ERF, that is $\lambda < P/\overline{D}$ or $\rho < 1$. This stability property of the EQS policy is not shared by several other processor scheduling policies for parallel systems. For example, consider the FCFS policy with a workload having $N = P$ and ERF $\gamma$. This system behaves like an $M/G/1$ system with mean service time $\overline{x} = \overline{D}/\gamma(P)$ and thus the stability condition is $\lambda < \gamma(P)/\overline{D}$. That is, the upper bound on arrival rate for stable operation of the FCFS system depends on $\gamma(P)$ and degrades as the sublinearity of $\gamma$ increases. If $\gamma(P) = P/2$ then the upper bound on $\lambda$ is half that of the EQS system.

(ii) **Sensitivity of $\overline{R}_{EQS}$ to ERF sublinearity and type**

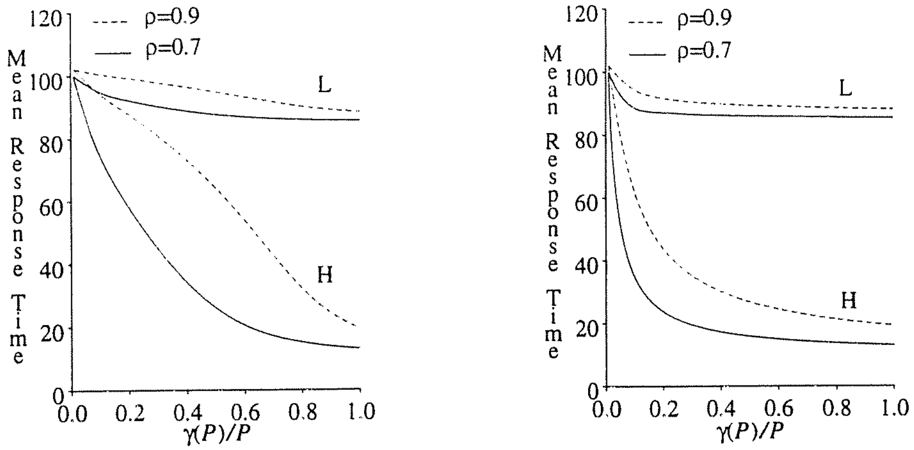
The sensitivity of $\overline{R}_{EQS}$ to the degree of ERF sublinearity is examined for the following two specific ERF functions. In each function the degree of sublinearity is controlled by a single parameter.

(a) $\gamma(k) = k^c$, $k = 1, 2, \ldots, N$, $0 \leq c \leq 1$. When $c = 0$ we obtain the constant ERF $\gamma(k) = 1$, and when $c = 1$ we obtain the linear ERF $\gamma(k) = k$. Thus we control the degree of sublinearity by varying $c$ from 0 to 1. This ERF is plotted in Figure 3.2a for different values of $c$.

(b) $\gamma(k) = \dfrac{(1+\beta)k}{k+\beta}$, $k = 1, 2, \ldots, N$, $0 \leq \beta < \infty$. When $\beta = 0$, we obtain $\gamma(k) = 1$, and when $\beta = \infty$ we obtain the linear ERF. Thus we control the degree of sublinearity by varying $\beta$ from 0 to $\infty$. This ERF is plotted in Figure 3.2b for several values of $\beta$.

Figure Figure 8.6 plots $\overline{R}_{EQS}$, estimated from approximation (5.12), versus ERF sublinearity, $\gamma(P)/P$, for each of the above ERFs, the H and L workloads of Table 3.2, and two different values of $\rho$. For each curve, $P = 100$ and $\overline{D} = P$. For both ERF types we observe that sublinearity has a fairly small impact on overall mean response time for the L workload, since a significant fraction of the jobs are sequential and the service time for sequential jobs is independent of ERF sublinearity. On the other hand, for the H workload the ERF sublinearity has a significant impact on mean response time. Furthermore, the precise behavior of $\overline{R}_{EQS}$ as a function of ERF sublinearity differs for the two different types of ERFs, and the difference increases as $\rho$ increases.

For the ERF $\gamma(k) = (1+\beta)k/(k+\beta)$ the mean response time of EQS decreases dramatically when $\gamma(P)$ increases from 1 to $0.5P$, and then decreases very gradually as $\gamma(P)$ varies from



(a) $\gamma(k) = k^c$

(b) $\gamma(k) = (1+\beta)k/(k+\beta)$

Figure 8.6: $\overline{R}_{EQS}$ vs ERF sublinearity

$\overline{D} = P = 100$

$0.5P$ to $P$. [5] For the ERF $\gamma(k) = k^c$, as $\rho$ increases the mean response time decreases more gradually for $\gamma(P)$ in the range of 1 to $0.5P$. $\overline{R}_{EQS}$ behaves differently under these two ERF types because of the different behavior of these ERFs when processor allocation is low (say in the region of 0-0.20P, see Figure 3.2). At higher load, jobs are allocated fewer processors, and for any fixed average allocation of processors $k < P$, say k=10, the curves that correspond to fixed increases in $\gamma(P)$ more rapidly approach rate $k$ for the ERF controlled by $\beta$ than for the ERF controlled by $c$. (Note that for the $H$ workload, the mean number of jobs in the system as obtained from the interpolation on $\underline{p}$ is greater than 10 under all ERFs for both values of $\rho$ in Figure 8.6.)

One conclusion of this sensistivity study is that, as intuition might suggest, the EQS policy provides better performance to workloads that have the initial part of their ERFs close to linear (say the first 10-20%). Another conclusion is that if the ERF has this property, $\overline{R}_{EQS}$ is relatively insensitive to ERF sublinearity in the range of $\gamma(P) > 0.5$, particularly if the workload is not fully parallel and $\rho$ is less than 0.9.

[5] Note that the degree of insensitivity of $\overline{R}_{EQS}$ to ERF sublinearity when $\gamma(P) > 0.5P$ is due to the fact that the H workload contains a fraction of sequential jobs, whose service times dominate in the mean service time estimates. For a fully parallel workload, the decrease in $\overline{R}_{EQS}$ as $\gamma(P)$ increases is still gradual for $\gamma(P) > 0.5$, but has somewhat more negative slope than the H workload.

(iii) **Performance of spatial EQS versus temporal EQS**

The EQS policy spatially allocates the integral part of a job's processor allocation and allocates only the fractional part temporally. If the allocation was purely temporal then the performance would be likely to degrade for sublinear ERFs as shown in the measurement study of [47]. This is true assuming that under EQS jobs can dynamically redistribute their work among the processors allocated to them. To illustrate the comparison between spatial and temporal EQS consider the ERF $\gamma(k) = (1+\beta)k/(k+\beta)$ and constant available parallelism $N = k$ for all jobs. The temporal EQS policy allocates $k$ processors at a time to a job and time slices jobs if there are more than $P/k$ jobs in the system. If $k$ evenly divides $P$ then it is easy to verify that $\overline{R}_{temporal-EQS} = \overline{R}_{M/G/c\ PS}$, where $c = P/k$ and each server in the $M/G/c$ queue is of power $\gamma(k)$. Figure 8.7b plots the ratio of $\overline{R}_{temporal-EQS}$ to $\overline{R}_{spatial-EQS}$ versus $\beta$, where an increase in $\beta$ indicates an increase in speedups as shown in Figure 8.7a. For the linear ERF, i.e., $\beta = \infty$, temporal and spatial EQS have identical performance. However, as ERF sublinearity increases ($\beta$ decreases) $\overline{R}_{temporal-EQS}$ diverges away from $\overline{R}_{spatial-EQS}$. For small values of $k$ (i.e., low parallelism in the workload) the difference between temporal and spatial EQS is quite small in the practical range of $\beta$, but as $k$ increases temporal EQS performs worse due to inefficient utilization of processors. This degradation is particularly noticeable at $\rho = 0.9$ and $k = 20$, and also at $\rho = 0.7$ and $k = 50$. For $k = 50$ and $\rho = 0.9$ (not shown), even at close to linear speedups such as $\beta = 500$ the value of $\overline{R}_{temporal-EQS}$ is 7.6 times that of $\overline{R}_{spatial-EQS}$.

### 8.2.4 Summary of insights for $r = 0$

In this section the following properties of the EQS policy for uncorrelated workloads with $E = \gamma$, i.e., jobs can dynamically redistribute their work on the processors allocated to them, were derived from the interpolation approximation in (5.7).

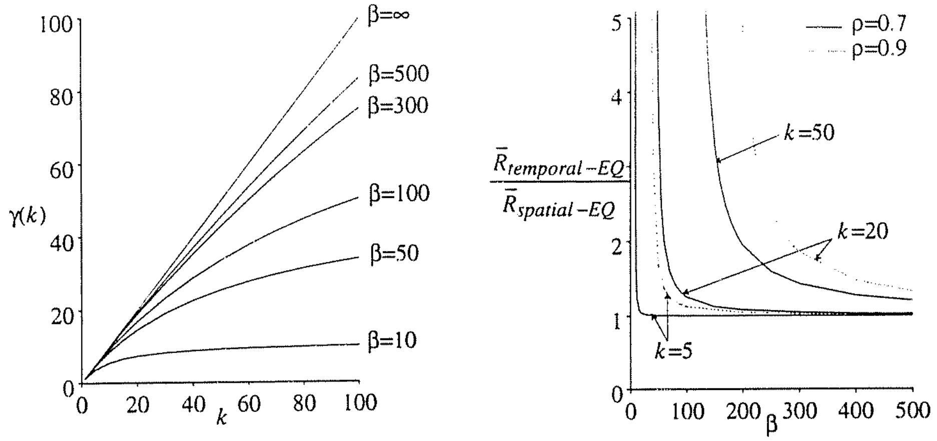
(i) *$\overline{D}$ and $S_n \equiv E[1/\gamma(N)]$ are the key determinants of $\overline{R}_{EQS}$* (given $\rho$ fixed)

(ii) *$\overline{R}_{EQS}$ increases linearly with $\overline{D}$ for a given value of $\rho$ and is insensitive to higher moments of D (e.g., $C_D$).*

(iii) *Parallelism Considered Useful:* $\overline{R}_{EQS}$ decreases with a stochastic increase in available parallelism. In particular, $N = P$ is optimal and $N = 1$ is pessimal for the EQS policy.

(iv) *For concave ERFs, $\overline{R}_{EQS}$ decreases with a decrease in the variability of available parallelism.*

(v) *Graceful degradation with ERF sublinearity:* In the absence of preemption and scheduling overhead, the EQS system is stable as long as $\lambda < P/\overline{D}$, regardless of the degree of



(a) ERF $\gamma(k) = (1+\beta)k/(k+\beta)$ (b) $\overline{R}_{temporal-EQ}/\overline{R}_{spatial-EQ}$ vs $\beta$: $N = k$

Figure 8.7: Performance of Spatial versus Temporal EQ

$$\overline{D} = P = 100$$

sublinearity in the ERF. Furthermore if the workload ERF is close to linear when processor allocation is 0 – 20% of P, $\overline{R}_{EQS}$ is relatively insensitive to ERF sublinearity at higher processor allocations, given that the applications have at least 50-60% efficiency on P processors.

(vi) *Spatial EQS performs significantly better than temporal EQS for sublinear ERFs* assuming that work can be efficiently and dynamically redistributed among the processors allocated to a job.

## 8.3 Behavior of $\overline{R}_{EQS}$ for Correlated Workloads

In Section 8.2 the behavior of EQS was studied for uncorrelated workloads using approximation (5.7) which has the form

$$\overline{R}_{EQS}(\mathcal{F}_N, r=0) \approx \sum_{k=1}^{P} p_k \overline{R}_{EQS}(N=k, r=0), \quad \text{under } (\cdot, \mathcal{F}_D^u, \cdot, \gamma). \tag{8.3}$$

To study the behavior of $\overline{R}_{EQS}$ for correlated workloads, we make a key observation about the following approximation for $\overline{R}_{EQS}$ under general workload conditions, which was derived in

Section 5.3 of Chapter 5.

$$\overline{R}_{EQS}(\mathcal{F}_N, r) \approx \sum_{k=1}^{P} p'_k \overline{R}_{EQS}(N = k, r = 0), \quad p'_k = p_k \frac{\overline{D}_k}{\overline{D}} = p_k \left(1 - r^2 + r^2 \frac{k}{\overline{N}}\right). \qquad (8.4)$$

Comparing (8.3) with (8.4) shows that $\overline{R}_{EQS}(\mathcal{F}_N, r)$ is obtained by replacing $p_k$ in (8.3) by $p'_k$. We note that $p'_k \geq 0$ and $\sum_{k=1}^{P} p'_k = 1$. Hence $\underline{p}' = (p'_1, \ldots, p'_P)$ is a pmf for a random variable $N' \in \{1, \ldots, P\}$. This implies that if we use the random variable $N'$ instead of $N$ in approximation (8.3) we will obtain an estimate for $\overline{R}_{EQS}(\mathcal{F}_N, r)$. Thus we can view the behavior of $\overline{R}_{EQS}$ under a correlated workloads as equivalent to the behavior of $\overline{R}_{EQS}$ under an uncorrelated workload with a different distribution of available parallelism. Formally,

$$\overline{R}_{EQS}(\mathcal{F}_N, r) \approx \overline{R}_{EQS}(\mathcal{F}_{N'}, r = 0), \qquad \text{under } (\cdot, \mathcal{F}^u_D, \cdot, \gamma). \qquad (8.5)$$

### 8.3.1 $\overline{R}_{EQS}$ as a function of job demand and parallelism

When $r > 0$, as in the case of $r = 0$, $\overline{D}$ is the only determinant of $\overline{R}_{EQS}$ with respect to job demand. This is true because approximation (8.4) is a weighted sum of the mean response times of EQS under constant available parallelism, which depend only on the first moment of demand and not on higher moments. (Note that the weights $p'_k$ do not depend on $\overline{D}$ since the ratio of $\overline{D}_k/\overline{D}$ is independent of $\overline{D}$, for $k = 1, 2, \ldots, P$.)

Regarding the key determinant of $\overline{R}_{EQS}$ with respect to the distribution of available parallelism for correlated workloads, in Section 8.2 we showed that $E[1/\gamma(N)]$ is the key determinant of $\overline{R}_{EQS}(\mathcal{F}_N, r = 0)$ (given that $\lambda$, $\overline{D}$, and $\gamma$ are fixed). This, together with approximation (8.5), implies that $E[1/\gamma(N')]$ is the key determinant for $\overline{R}_{EQS}(r)$. Simplifying $E[1/\gamma(N')]$ we get

$$\begin{aligned} E\left[\frac{1}{\gamma(N')}\right] &= \sum_{k=1}^{P} p'_k \frac{1}{\gamma(k)} \\ &= \sum_{k=1}^{P} p_k \frac{\overline{D}_k}{\overline{D}} \frac{1}{\gamma(k)} \\ &= \frac{1}{\overline{D}} \overline{S} \equiv S_n. \end{aligned}$$

Thus $S_n$ *is the key parameter for job parallelism, workload correlation, and job execution rate function.* Moreover, the result that $\overline{R}_{EQS}$ increases (nearly) linearly as a function of $S_n$, as per Figure 8.5, holds for correlated workloads since it holds for all distributions of $N$ in uncorrelated workloads. we can show that under nondecreasing $\gamma$ $S_n(r)$ is minimum when $N = P$ and maximum when $N = 1$, and a stochastic increase in $N$ causes $S_n$ to decrease. Thus for correlated workloads $N = P$ workload has optimal performance and the $N = 1$ workload has pessimal performance. Unlike the case of $r = 0$ for a fixed $\overline{N}$, $S_n$ does not necessarily decrease with decrease in variability of $N$. For example, when $r = 1$ it follows from Theorem 3.5.4 in Appendix A that for concave $\gamma$ and concave $N/\gamma(N)$, $S_n$ is minimum when variability in $N$ is maximum and is maximum when variability in $N$ is minimum. Thus, since property (v) in Section 8.2.4 is expected to hold generally for uncorrelated workloads, all of the properties of $\overline{R}_{EQS}$ summarized apply to workload with $r > 0$, except property (iv).

### 8.3.2 $\overline{R}_{EQS}$ as a function of $r$

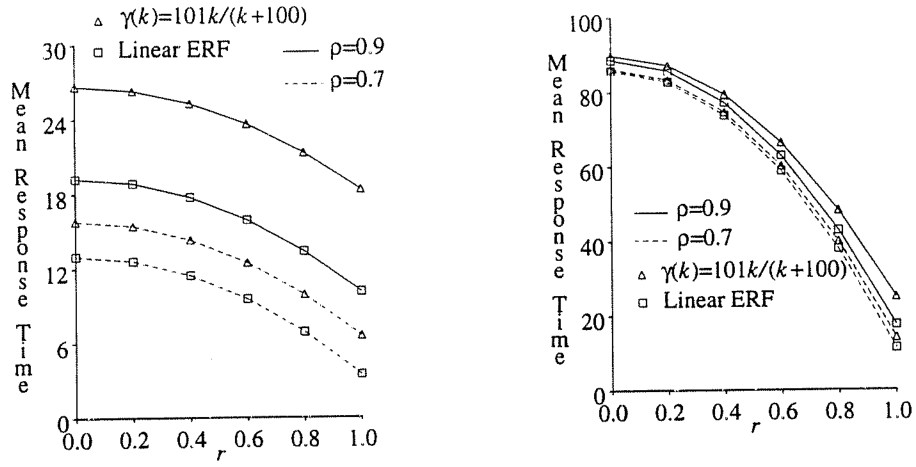
We now study the behavior of $\overline{R}_{EQS}$ when workload correlation increases. Recall from (8.5) that the behavior of EQS under correlated workloads and a distribution of available parallelism $\mathcal{F}_N$ is the same as the behavior of EQS under no correlation and a distribution of available parallelism $\mathcal{F}_{N'}$. The pmf of $N'$, $\underline{p}'$, is related to the pmf of $N$, $\underline{p}$, as follows:

$$p'_k = p_k \frac{\overline{D}_k}{\overline{D}} = p_k \left(1 - r^2 + r^2 \frac{k}{\overline{N}}\right).$$

Thus, $p'_k < p_k$ for $k < \overline{N}$ and $p'_k > p_k$ for $k > \overline{N}$. As a result the random variable $N'$, has stochastically higher available parallelism than $N$ (i.e., $N' \geq_{st} N$). The intuitive reason for the increase in effective available parallelism is that under correlated workloads, jobs that have small demands and exit the system quickly have on average smaller parallelism and leave behind jobs that have larger available parallelisms on average.

As seen in the previous section a stochastic increase in parallelism causes $\overline{R}_{EQS}(r = 0)$ to decrease and hence $\overline{R}_{EQS}(r)$ decreases with correlation, under the given model of workload correlation and given that $\overline{D}$ remains unchanged. The intuition for this result is that as $r$ increases, larger demand jobs have larger available parallelisms, and this causes them to complete faster than if they had lower parallelisms as in uncorrelated workloads. (Consider for example the case where a sequential job in an uncorrelated workload runs on one processor but the remaining processors are idle.)

Concerning the quantitative behavior of $\overline{R}_{EQS}$ as a function of $r$, Figure 8.8 depicts $\overline{R}_{EQS}$ (as obtained from approximation (8.4) versus $r$ for the H and L workloads, two types of ERFs (one linear and one sublinear), and two values of $\rho$. The trends are stronger for the L workload than for the $H$ workload, but the general behavior of $\overline{R}_{EQS}$ versus $r$ is the same for both types of ERF and for both values of $\rho$. In particular, there is a significant decrease in mean response time as $r$ increases, and the decrease is greatest as $r$ approaches 1. This latter property is due to the quadratic dependence of $\overline{R}_{EQS}$ on $r$ (under the given workload model), which is shown in the interpolation on $r$ (5.10). The results show that EQS is a high performance policy under highly correlated workloads. Note that again that this property is not shared by all scheduling



(a) H workload

(b) L workload

Figure 8.8: $\overline{R}_{EQS}$ versus Workload Correlation

$\overline{D} = P = 100$

disciplines.

We therefore learn that an increase in workload correlation between demand and parallelism can cause mean response time to substantially decrease under the EQS policy. This shows that EQS is a high performance policy even under highly correlated workloads, which is why EQS outperforms PSAPF even at low to moderate $C_v$ as shown in Chapter 7.

## 8.4 Conclusion

We first summarize the results of this chapter and then examine how they are related to previous work.

### 8.4.1 Summary of Results

In this chapter we have studied the qualitative behavior of the EQS policy as a function of workload parameters. First, we used sample path analysis to derive mean response time bounds for EQS. These bounds show that under exponential demands and the same concave ERF for all jobs EQS has optimal performance when all jobs are fully parallel, and under general demand, available parallelism, execution rates, and correlation, EQS has pessimal performance when all jobs are fully sequential. Second, we used approximate analysis to understand the behavior

$\overline{R}_{EQS}$ under the general assumptions of the workload model.

The approximate analysis yielded the insights that within the accuracy of the model (1) mean workload demand $\overline{D}$ and normalized mean service time $S_n := \overline{S}/\overline{D}$ are the key determinants of $\overline{R}_{EQS}$; $\overline{R}_{EQS}$ increases linearly in each of these determinants, (2) $\overline{R}_{EQS}$ decreases with a stochastic increase in available parallelism, in particular, it is optimal when all jobs are fully parallel and pessimal when all jobs are fully sequential, (3) for uncorrelated workloads $\overline{R}_{EQS}$ decreases with decrease in variability of available parallelism, (4) $\overline{R}_{EQS}$ decreases with increase in workload correlation ($\overline{D}$ remaining fixed), and (5) in the absence of preemption and scheduling overhead the EQS system has the same stability condition for sublinear ERFs as it does for the linear ERF.

Although the above results were derived assuming that all jobs have the same ERF, $\gamma$, it seems likely that the results will hold more generally as long as job ERFs are (nondecreasing and) uncorrelated with parallelism. Thus, the key properties of the system that lead to the nice performance behavior are (a) the equiallocation of processing power, (b) job ERFs are uncorrelated with parallelism, and (c) job execution efficiencies improve for smaller processor allocations (yet highly parallel jobs can make use of larger processor allocations when contention is low). The results explain why several previous studies that fit assumptions (a) and (c) have observed the EQS policy to have high performance. If property (c) does not hold for a given equiallocation system (e.g., an EQ policy under workloads where $E(j) = (j/N)\gamma(N)$) then some of the results should continue to hold and other results do not hold. For example, insensitivity of mean response time to coefficient of variation in demand, $C_D$, should continue to hold, but the result that mean response time decreases with "increase" in available parallelism will not necessarily hold, and system performance can be expected to be more sensitive to ERF sublinearity. Thus, for high-performance multiprogrammed parallel systems, the development of architectural and software support that allows jobs to dynamically and efficiently redistribute their work across their processor allocation is highly desirable.

### 8.4.2 Related Work

Related work in the literature that has studied the qualitative behavior of EQ policies includes [41, 39, 47, 1]. For a workload with equal division of demand among tasks Leutenegger and Vernon [41] use simulation to show that $\overline{R}_{RRJ}$ is independent of $C_D$ for specific $H_2$ distributions for demand and specific bounded-exponential distributions for parallelism. For the same workload Leutenegger [39] also shows $\overline{R}_{RRJ}$ to increase linearly with $\overline{D}$ and to decrease with workload parallelism for specific distributions of demand and parallelism. We have shown the same results for general distributions of demand and parallelism.

Agrawal et al. [1] derive a general lower bound for parallel processor policies that holds under general workload conditions. They give examples to show that for uncorrelated workloads with specific distributions of demand and parallelism and a linear ERF the mean response time of EQS is within twice of the best achievable performance. They also show that for exponential demands and linear ERFs the mean response time of any processor conserving policy (EQS is included in this class) is minimum when $N = P$ and maximum when $N = 1$. We have generalized this result for the EQS policy by showing that $\overline{R}_{EQS}$ is maximum when $N = 1$ under general demands and ERFs, and that $\overline{R}_{EQS}$ is minimum when $N = P$ under exponential demands and concave ERFs.

There have been no results in the literature for the qualitative behavior of EQS as a function of workload ERF and workload correlation. The study by McCann et al. [47] show spatial EQ to outperform temporal EQ using measurements for a specific mix of parallel programs. We have corroborated their result for our workload model.

# Chapter 9

# Conclusions

## 9.1 Summary

Under a general workload model we have developed a new approach of interpolation approximations to model parallel processor policies and used it to evaluate and compare the performance of scheduling policies known from previous studies to have high performance under specific workloads. These policies include the dynamic spatial equipartitioning (EQS) policy, the Preemptive Smallest Available Parallelism First (PSAPF) policy, the dynamic First Come First Serve (FCFS) policy, and a run-to-completion policy called Adaptive Static Partitioning (ASP).

The main results of this thesis are as follows:

- The interpolation approximation approach is a promising method for efficient analysis of parallel processor scheduling policies, in that, it yields ready insight by means of closed form formulas of mean response time which are easy to evaluate for systems with hundreds of processors.
- Coefficient of variation in demand ($C_D$) can be critical in determining *relative* policy performance. This might be obvious from uniprocessor scheduling results, but unfortunately most previous analyses and comparisons of parallel processor policies have assumed exponential demands or exponential task service times. This result shows that it is important to consider the implications of the exponential assumption on the conclusions reached.
- Sublinearity of speedup curves and correlation between demand and parallelism are also influential parameters for relative policy performance. While speedup curves have been explicitly specified in previous studies, in many cases no indication of correlation is provided. This result shows that it is important to provide correlation information about the

workload and consider its implications on the conclusions.

- We have unified and generalized previous policy comparison results for the ASP, FCFS, EQ, and PSAPF policies by showing how previous results map to different regions of the parameter space and generalizing the results over broader regions of the design space. In particular, we have shown that the EQS policy has highest performance over most of the parameter space, that is, all of the parameter space except where coefficient of variation in demand is low (less than or equal to 1 for uncorrelated workloads and less than or equal to 2 for correlated workloads) and the execution rates are close to linear. For $C_D < 1$ the FCFS policy has highest performance if correlation is low, whereas PSAPF dominates if correlation is high. We note that $C_D < 1$ is probably a less likely region of the design space for general workloads, as in uniprocessor scheduling environments [65, page 16], [81].

- The EQS policy has superior processor allocation characteristics in terms of processor efficiency. More specifically, for a fixed set of jobs with a common nondecreasing and concave execution rate function the EQS policy achieves optimal processor utilization over all allocation policies.

- Under our workload model job arrival rate, mean demand, and mean service time are the key determinants of $\overline{R}_{EQS}$. Within the accuracy of the model
  - $\overline{R}_{EQS}$ increases in each of mean demand and mean service time, given a fixed system offered load,
  - $\overline{R}_{EQS}$ decreases with stochastic increase in available parallelism,
  - $\overline{R}_{EQS}$ decreases with decrease in variability of available parallelism for a concave workload ERF and uncorrelated workloads,
  - $\overline{R}_{EQS}$ decreases with increase in workload correlation, and
  - when preemption and scheduling overheads are negligible the EQS system has the same stability condition for sublinear ERFs as it does for the linear ERF and its mean response time is relatively insensitive to parallel program overheads if the workload is not fully parallel and the ERF is nearly linear for small processor allocations.

To achieve the main goal of studying qualitative policy behavior and comparing scheduling policies over a general workload model this thesis proceeded through a series of stages. Chapter 2 first reviewed parallel processor scheduling results in the literature to show what needed to be done to obtain a better understanding of scheduling policy performance. In Chapter 3 we developed a general workload model that captures the essential features of parallel applications and made judicious assumptions about the workload parameters to permit broad applicability as well as ease of analysis. To permit broad applicability we assumed a general distribution of demand by means of which we saw that $C_D$ is a key parameter that influences relative policy performance. This could not have been shown had we assumed exponential or some other specific distribution of job demand. To permit ease of analysis we assumed the linear ERF to model the PSAPF, FCFS, and ASP policies. However, we showed in Chapter 7 that this assumption did not limit the applicability of our policy comparison results since the EQS policy performed better than these three policies under this assumption even though the assumption favors these three policies relative to the EQS policy.

Chapter 4 proposed the approach of interpolation approximations where we found points in the parameter space for which the parallel system (under a given scheduling policy) reduces to a queueing system that has a known solution. We then interpolated among the "known" values to derive estimates for the "unknown" regions of the parameter space. The purpose of developing the new approach was to enable us to readily see the dependence of policy performance on workload parameters by means of closed form expressions, and to easily evaluate mean response times for large systems. Chapters 5 and 6 showed that the interpolation approximation approach resulted in accurate approximations for the mean response times of the EQS, ASP, FCFS, and PSAPF policies. In Chapter 7 we used the interpolation approximations for these policies to achieve our purpose of evaluating and comparing policy performance over the parameter space. We used the interpolation approximations to obtain key determinants of relative policy performance. We then used the approximate mean response time formulas to delineate the design space *without having to recourse to specific experimental workload settings* in many cases, as seen by the comparisons between FCFS and PSAPF for uncorrelated workloads, and EQS and PSAPF for both uncorrelated as well as correlated workloads. We could therefore compare policy performance over general distributions of available parallelism and a general range of $C_D$.

We finally point out the potential of analytic modeling that has been exploited in this thesis. In our analysis of policy performance and in the comparisons of policy performance we made use of general demands, general available parallelism and also in case of the EQS policy a general ERF. These generalities were captured in the form of mean response time expressions. On the contrary an experimental approach such as simulation would have forced us to assume specific distributions for workload parameters, and it would be unknown whether the experimental results would apply to other settings for workload parameters. Since there are an uncountable number of settings it would be impossible to explore the entire design space as done using analytic modeling. For example, in Chapter 8 we used linear programming to obtain minimum and maximum mean response times for the EQS policy and this in turn yielded a key workload parameter for EQS. For the linear programs we made no assumptions about the distribution of available parallelism and thus obtained results than span across an uncountable number of distributions. Clearly, this would have been impossible to do using simulation even if we could estimate mean response times in a few seconds per data point. Similarly, in Chapter 7 we used nonlinear programming to obtain minimum and maximum values of the mean response time of PSAPF over all distributions of available parallelism which would again not be possible using an experimental method such as simulation.

Thus analytic modeling is particularly important in evaluating parallel processor policies since in addition to processing demand one also has to consider job parallelism which needs to be modeled generally because it is unknown to date as to what is a realistic distribution for program parallelism. We feel that we have developed a promising analytic approach in this thesis and used it to show that under general workload conditions equipartitioning policies have highest performance over known scheduling policies that do not use information about job demand.

## 9.2 Future Research Directions

A number of research directions are open for the future. These include:

1. Use of Application Characteristics for Scheduling

   Further work needs to be done to determine how useful are application characteristics for scheduling. For example, modifying the EQS policy to use average parallelism in addition to available parallelism may lead to improved performance. Similarly, limiting the maximum allocation under ASP to equal a job's average parallelism or processor working set may lead to an AP/RTC policy with higher performance than ASP.

2. Guidelines for Interpolation Approximations

   This thesis examined several types of interpolation approximations, e.g., interpolation on system utilization, $\rho$, interpolation on average available parallelism, $\overline{N}$, and interpolation on pmf, $\underline{p}$. Other types of interpolation approximations are also possible, e.g., interpolation on coefficient of variation in demand, $C_D$. In general, how does one decide which interpolation to use? As an example consider the RRP policy. When available parallelism $N$ is constant, RRP is identical to EQS in terms of allocation of processing power. As a result a simple interpolation on $\overline{N}$ or $\underline{p}$ will show that under the linear ERF RRP and EQS have identical performance. However, Leutenegger and Vernon [41] give experimental data to show that for the linear ERF RRP can perform significantly worse than RRJ (temporal EQS). This difference can be shown by using an interpolation on $\rho$. Using the DPS bound for RRP in [1] we can derive a heavy traffic limit for $\overline{R}_{RRP}$ which will be different from the heavy traffic limit for EQS, and thus an interpolation on $\rho$ will show the difference between

$\overline{R}_{RRP}$ and $\overline{R}_{EQS}$. Thus, deciding which interpolation to use is sensitive to the policy being modeled, and for a given policy or a given class of policies it will be useful if guidelines can be developed as to which interpolation approximations will produce accurate results.

3. EQS with Multiple Job Classes

   This thesis examined the EQS policy for a single class of jobs having a general distribution for demand and parallelism and a common workload ERF $\gamma$. A more realistic workload model will include multiple job classes based on ERFs. Mean response times per class can provide further insight into the EQS policy, e.g., how does increasing the sublinearity of one class affect mean response time of other classes under EQS. Developing mean response time approximations on a per class basis may be more complex than the approximations in this thesis.

4. ASP with General Demands and Sublinear ERFs

   We obtained an approximation for $\overline{R}_{ASP}$ for exponential demands and linear ERFs. Although these two assumptions are restrictive they did not limit the policy comparison results in this thesis since they are more favorable to ASP with respect to EQS and yet EQS was shown to perform better. However, when ASP is compared with other AP/RTC policies in the future it may not be reasonable to assume exponential demands and/or linear ERFs. To obtain an approximation for $\overline{R}_{ASP}$ for general demands and sublinear ERFs we can use an interpolation on $\rho$ since as $\rho \rightarrow 1$ ASP tends to allocate one processor per job and hence its mean response time will approach that of an M/G/c queue. To get an accurate approximation it may be necessary to take light traffic derivatives as in [83]. Although this approach is cumbersome as the number of derivatives increases it can nevertheless lead to estimates of mean response time that can be computed quickly.

5. General Distribution of Inter-Arrival Times

   We have only focused on Poisson arrivals in this thesis. The interpolation approximation approach should extend to more general distributions of inter-arrival times by using results from GI/G/c queues in the literature. For example, Sakasegawa's approximation [64] for the mean number of customers in a GI/G/c queue is a closed form expression that uses the first two moments of inter-arrival and service times. This can be used for the approximations for $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$. Approximations for the EQS policy were derived using results from PS queues and symmetric queues. For general inter-arrival times we have not come across mean response time solutions for PS queues or for symmetric queues in the literature, but it might be possible to derive simple approximations for their mean response times along the lines of Sakasegawa's approximation.

6. Closed System Models

   (a) For a closed system model where customers alternately think and submit jobs for execution we can get exact mean response time estimates using recurrence relations. The main idea is to develop recurrences for the mean time for the system to return to a regenerative state (say all customers thinking) as well as develop recurrences for the mean number of completions during that time. Using these two mean values we can derive the throughput of the system and from that the overall mean response time. The recurrence relations approach, however, will be limited by the number of jobs in the system. For a large customer population asymptotic job bounds [38] can be derived as in uniprocessor system models.

   (b) Since recurrence relations may be limited to small system sizes and will not permit general distributions of demand, the interpolation approximation approach can be used to solve the above closed system model. In addition, the interpolation approximation approach can also be used to solve for a closed system with an I/O subsystem, where where jobs enter the I/O system only after completing service at the parallel system and then loop back to the processor system. For example, for the EQS policy one can first assume that all jobs have constant available parallelism ($N = k$). Since the EQS policy reduces to a symmetric queue that satisfies product form, we can solve the overall network if the I/O subsystem also satisfies product form. After obtaining estimates for $\overline{R}_{EQS}(N = k)$ we can interpolate on $\underline{p}$ to obtain $\overline{R}_{EQS}$. The accuracy of this approximation will, however, need to be validated extensively against simulation.

7. Including Memory Constraints into Scheduling Algorithms

   We have only examined processor scheduling algorithms in this thesis. Practical policies also need to consider memory constraints of jobs. Given the level of difficulty in modeling processor scheduling policies alone it seems rather complex to include memory constraints into models of scheduling algorithms. It may, however, be the case that memory constraints will automatically limit the maximum number of the jobs in the system leading to analyzable models whose underlying Markov chains are either in matrix-geometric form or have a finite number of states. Whether or not interpolation approximations will be useful in this context is unknown at this point in time.

# Appendix A

# Proofs and Derivations

In this appendix we provide proofs and derivations that were skipped in the main text. This appendix is organized by chapter number. Section A.1 provides the proof for Theorem 3.5.2 in Chapter 3. Section A.2 provides the derivation of the approximation for $\overline{R}_{PSAPF}(r = 1)$ in Chapter 6. In Section A.3 we provide proofs for two theorems and nonlinear programming details used in Chapter 7, and finally in Section A.4 we provide proofs of upper and lower bounds for $\overline{R}_{EQS}$ that were derived in Chapter 8.

## A.1 Proofs for Chapter 3

**Theorem 3.5.2** *Let $N$ have a bounded-geometric distribution with parameters $P_{\max}$ and $p$. For a given $\overline{N}$, $C_N$ is maximum when $p = 1$ and $C_N$ is minimum when $P_{\max} = 0$.*

**Proof.** We proved the first part (maximum $C_N$ when $p = 1$) of this theorem in Chapter 3.5.1. We now prove that over all bounded-geometric distributions with the same $\overline{N}$, $C_N$ is minimum when $P_{\max} = 0$.

Consider two random variables, $N_1$ =Bounded-Geometric($P_{\max} > 0, p$) and $N_2$ = Bounded-Geometric(0,$u$), such that $\overline{N}_1 = \overline{N}_2$. We show that $C_{N_2} \leq C_{N_1}$ which means that $C_N$ is minimum for bounded-geometric distributions when $P_{\max} = 0$. The pmf of $N_1$ is given by

$$p_{1i} = \begin{cases} (1 - P_{\max})p(1 - p)^{i-1}, & 1 \leq i \leq P - 1 \\ (1 - P_{\max})(1 - p)^P + P_{\max}, & i = P, \end{cases} \tag{A.1}$$

and the pmf of $N_2$ is given by

$$p_{2i} = \begin{cases} u(1 - u)^{i-1}, & 1 \leq i \leq P - 1 \\ (1 - u)^P, & i = P. \end{cases} \tag{A.2}$$

Let $\Delta_i \equiv p_{1i} - p_{2i}$. Since $\sum_{i=1}^{P} p_{1i} = \sum_{i=1}^{P} p_{2i} = 1$, we have

$$\sum_{i=1}^{P} \Delta_i = 0. \tag{A.3}$$

Since $\overline{N}_1 = \overline{N}_2$, we have $\sum_{i=1}^{P} ip_{1i} = \sum_{i=1}^{P} ip_{2i}$, which means that

$$\sum_{i=1}^{P} i\Delta_i = 0. \tag{A.4}$$

To prove that $C_{N_1} \geq C_{N_2}$ we show that $\overline{N_1^2} \geq \overline{N_2^2}$, that is, $\sum_{i=1}^{P} i^2\Delta_i \geq 0$.

We divide our analysis into two parts: (1) $p_{11} \geq p_{21}$, and (2) $p_{11} < p_{21}$. In the first case we prove that $\sum_{i=1}^{P} i^2\Delta_i \geq 0$. We then show that the second case is impossible.

(i) $\underline{p_{11} \geq p_{21}}$:

We first show that there exists an $n \leq P-1$ such that

$$p_{1i} \geq p_{2i}, \quad 1 \leq i < n, \qquad \text{and} \quad p_{1i} \leq p_{2i}, \quad n \leq i \leq P-1, \tag{A.5}$$

and then use this property to prove that $C_{N_1} \geq C_{N_2}$.

To prove (A.5) we show that

(a) it is impossible that $p_{1i} \geq p_{2i}$ for all $1 \leq i \leq P-1$, and

(b) after the first $i$ such that $p_{1i} \leq p_{2i}$, we have $p_{1j} \leq p_{2j}$ for $i \leq j \leq P-1$.

Property (a) is proved as follows. Assume that $p_{1i} \geq p_{2i}$, for all $1 \leq i \leq P-1$. Hence $\Delta_i \geq 0$ for all $1 \leq i \leq P-1$. From (A.4) it follows that $\sum_{i=1}^{P-1} i\Delta_i = -P\Delta_P = P\sum_{i=1}^{P-1} \Delta_i$. Since $\Delta_i \geq 0$ this equality means that $\Delta_i = 0$, $i = 1, 2, \ldots, P-1$ and hence $\Delta_P$ is also zero. But $\Delta_i = 0$ for $i = 1, 2, \ldots, P$ means that $p_{1i} = p_{2i}$ and therefore $P_{\max} = 0$ which is a contradiction since we had assumed that $P_{\max} > 0$ for $N_1$. To prove property (b) we proceed as follows. Since $p_{11} \geq p_{21}$, it follows from (A.1) and (A.2) that $(1 - P_{\max})p \geq u$. Let $k$ be the first $i$ for which $p_{1i} \leq p_{2i}$. As seen from property (a) it must be that $k \leq P-1$. We therefore have,

$$\begin{aligned} (1 - P_{\max})p &\geq u \\ (1 - P_{\max})p(1-p)^k &\leq u(1-u)^k. \end{aligned}$$

These two inequalities imply that $(1-p)^k \leq (1-u)^k$, that is, $(1-p) \leq (1-u)$. As a result, for $k \leq i \leq P-1$,

$$p_{1i} = (1 - P_{\max})p(1-p)^k(1-p)^{i-k} \leq u(1-u)^k(1-u)^{i-k} = p_{2i}, \quad k \leq i \leq P-1,$$

which proves property (b).

We have thus proved (A.5). Let $n$ be the first $i$ for which $p_{1i} \leq p_{2i}$, $n \leq P-1$. From (A.5) it follows that

$$\Delta_i \geq 0, \quad i = 1, \ldots, n-1, \qquad \text{and} \quad \Delta_i \leq 0, \quad i = n, \ldots, P-1. \tag{A.6}$$

We also have

$$\sum_{i=1}^{n-1}(P-i)\Delta_i = -\sum_{i=n}^{P-1}(P-i)\Delta_i.$$

This follows because $\sum_{i=1}^{P}(P-i)\Delta_i = 0$ (see (A.3) and (A.4)), which is equivalent to $\sum_{i=1}^{P-1}(P-i)\Delta_i = 0$ or $\sum_{i=1}^{n-1}(P-i)\Delta_i = -\sum_{i=n}^{P-1}(P-i)\Delta_i$. We therefore have

$$\sum_{i=1}^{n-1}(P-i)i\Delta_i \leq -\sum_{i=n}^{P-1}(P-i)n\Delta_i \leq -\sum_{i=n}^{P-1}(P-i)i\Delta_i,$$

or

$$\sum_{i=1}^{P-1}(P-i)i\Delta_i \leq 0. \tag{A.7}$$

Multiplying (A.4) by $P$ we have,

$$\sum_{i=1}^{P-1} Pi\Delta_i + P^2\Delta_P = 0. \tag{A.8}$$

Subtracting (A.7) from (A.8) we get

$$\sum_{i=1}^{P} i^2\Delta_i \geq 0,$$

that is $\sum_{i=1}^{P} i^2 p_{1i} \geq \sum_{i=1}^{P} i^2 p_{2i}$, which shows that $\overline{N_1^2} \geq \overline{N_2^2}$ or $C_{N_1} \geq C_{N_2}$ as required.

(ii) $\underline{p_{11} < p_{21}}$

We show that this case is impossible. The proof proceeds as follows. Let $n$ be the first $i$ such that $p_{1i} \geq p_{2i}$. If $n = P$ then $\Delta_i < 0$ for $i = 1, \ldots, P-1$ and therefore $P\Delta_P = -P\sum_{i=1}^{P-1}\Delta_i > -\sum_{i=1}^{P-1} i\Delta_i$. Hence $\sum_{i=1}^{P} i\Delta_i > 0$, which contradicts (A.4). Therefore, $n \leq P-1$. We now have,

$$\begin{aligned}(1-P_{\max})p &< u \qquad (\text{since } p_{11} < p_{21})\\ (1-P_{\max})p(1-p)^{n-1} &\geq u(1-u)^{n-1}.\end{aligned}$$

From these two inequalities it follows that $(1-p)^{n-1} > (1-u)^{n-1}$ or $(1-p) > (1-u)$. Therefore,

$$p_{1i} = (1-P_{\max})p(1-p)^{n-1}(1-p)^{i-n+1} \leq u(1-u)^{n-1}(1-u)^{i-n+1} = p_{2i}, \quad n \leq i \leq P-1.$$

Furthermore,

$$p_{2P} = (1-u)^P < (1-p)^P < P_{\max} + (1-P_{\max})(1-p)^P = p_{1P}.$$

We have therefore shown that

$$\Delta_i < 0, \quad i = 1, \ldots, n-1, \qquad \text{and} \quad \Delta_i \geq 0, \quad i = n, \ldots, P.$$

From (A.3) it follows that $\sum_{i=n}^{P} \Delta_i = -\sum_{i=1}^{n-1} \Delta_i$. Hence,

$$\sum_{i=n}^{P} i\Delta_i > -n\sum_{i=1}^{n-1} \Delta_i > -\sum_{i=1}^{n-1} i\Delta_i.$$

Combining the leftmost and rightmost terms, we get $\sum_{i=1}^{P} i\Delta_i > 0$, which contradicts (A.4).

■

## A.2 Derivations for Chapter 6

In this section we derive (6.15) using the per class mean response time approximation for an M/G/c PR queue given in [78]. That is, we show that,

$$\overline{R}_{\Gamma_k,C_k} \approx c\overline{x}_k + \frac{1}{p_k}\left(\sum_{i=1}^{k-1} g_i\right)\left(\sigma_k^{\sqrt{2(c+1)}-2} - \sigma_{k-1}^{\sqrt{2(c+1)}-2}\right) + \frac{1}{p_k} g_k \sigma_k^{\sqrt{2(c+1)}-2}, \quad p_k > 0, \tag{A.9}$$

where $c = P/k$, $\overline{x}_i = (\overline{D}i)/(\overline{N}P)$, $\sigma_i = \lambda \sum_{j=1}^{i} p_i \overline{x}_i$, and

$$g_i = p_i \frac{\sigma_{i-1}}{1-\sigma_{i-1}}\overline{x}_i + \frac{\lambda p_i \sum_{j=1}^{i}\{p_j(1+C_v^2)\overline{x}_j^2\}}{2(1-\sigma_{i-1})(1-\sigma_i)}.$$

Recall that $\overline{R}_{\Gamma_k,C_k}$ in Section 6.3.4 was shown equal to the mean response time of the $k^{th}$ priority class in an *M/G/c PR* queue with $k$ priority classes. To estimate $\overline{R}_{\Gamma_k,C_k}$ we use Tabetaeoul and Kouvatsos' heuristic [78] for per class mean response times of an *M/G/c PR* queue. To begin with, $\overline{R}_{\Gamma_k,C_k}$ can be expressed in terms of the overall mean response times for the first $i$ classes, $\overline{T}^i_{M/G/c\ PR}$, for $i = k-1, k$. That is,

$$\overline{R}_{\Gamma_k,C_k} = \frac{\Lambda_k \overline{T}^k_{M/G/c\ PR} - \Lambda_{k-1}\overline{T}^{k-1}_{M/G/c\ PR}}{\lambda_k}, \tag{A.10}$$

where $\lambda_k = \lambda p_k$, and $\Lambda_k = \sum_{i=1}^{k} \lambda_i$. Let $\overline{z}_i$ denote the service time of class $i$ in the M/G/c PR queue, for $i = 1, \ldots, k$, and let $\overline{y}_i$ denote the overall mean service time of the first $i$ classes, that is $\overline{y}_i = \frac{1}{\Lambda_i}\sum_{j=1}^{i}\lambda_i\overline{z}_i$ for $i = 1, \ldots, k$. Define $\overline{X}^k_{M/G/c\ PR} \equiv \overline{T}^k_{M/G/c\ PR} - \overline{y}_k$. Tabetaeoul and Kouvatsos [78] proposed the following heuristic to estimate $\overline{X}^k_{M/G/c\ PR}$.

$$\overline{X}^k_{M/G/c\ PR} \approx \overline{X}^k_{M/G/1_c\ PR} \cdot \frac{\overline{X}^k_{M/G/c}}{\overline{X}^k_{M/G/1_c}}, \tag{A.11}$$

where $\overline{X}^k_{M/G/1_c\ PR} \equiv \overline{T}^k_{M/G/1_c\ PR} - \overline{y}_k/c$, $\overline{T}^k_{M/G/1_c\ PR}$ being the overall mean response time of the first $k$ classes in an $M/G/1_c$ $PR$ queue that is obtained by replacing the c servers of the M/G/c queue by a single server c times faster. (The $M/G/1_c$ $PR$ system has the same job priorities, service demands, and arrival rates as the M/G/c PR system.) Similarly, $\overline{X}^k_{M/G/c}$ is the overall mean waiting time in an M/G/c system that has the same workload as the M/G/c PR system (i.e., k classes) and $\overline{X}^k_{M/G/1_c}$ is the overall mean waiting time in an equivalent $M/G/1_c$ system.

We first provide closed form expressions for $\overline{X}_{M/G/c}$ and $\overline{X}_{M/G/1_c}$ and then for $\overline{X}_{M/G/1_c\ PR}$ so that we can get an expression for $\overline{X}^k_{M/G/c\ PR}$ using (A.11). Using Sakasegawa's approximation for GI/G/c FCFS queues [64], we obtain

$$\overline{X}^k_{M/G/c} \approx \frac{\sigma_k^{\sqrt{2(c+1)}}(1 + CV_k^2)}{2\Lambda_k(1 - \sigma_k)},$$

where $CV_k$ is the overall coefficient of variation in the service requirement of the $k$ classes, and $\sigma_k = \sum_{i=1}^{k}\lambda_i\overline{z}_i/c$. Using the analysis in [32] for the M/G/1 queue we get,

$$\overline{X}^k_{M/G/1_c} = \frac{\sigma_k^2(1 + CV_k^2)}{2\Lambda_k(1 - \sigma_k)},$$

and as a result,

$$\frac{\overline{X}^k_{M/G/c}}{\overline{X}^k_{M/G/1_c}} \approx \sigma_k^{\sqrt{2(c+1)}-2}. \tag{A.12}$$

From the analysis in [33] we have the following expression for $\overline{X}^k_{M/G/1_c\ PR} \equiv \overline{T}^k_{M/G/1_c\ PR} - \overline{y}_k/c$.

$$\overline{X}^k_{M/G/1_c\ PR} = \frac{1}{\Lambda_k}\sum_{i=1}^{k}\lambda_i\left\{\frac{\sigma_{i-1}}{(1-\sigma_{i-1})}\cdot\frac{\overline{z}_i}{c} + \frac{\sum_{j=1}^{i}\lambda_j(1 + C_v^2)\frac{\overline{z}_j^2}{c^2}}{2(1-\sigma_{i-1})(1-\sigma_i)}\right\},$$

where $C_v$ is the coefficient of variation of service requirement of class $j$, for $j = 1, \ldots, k$.

Substituting the above expression for $\overline{X}^k_{M/G/1_c\ PR}$ along with (A.12) into (A.11) we obtain

$$\overline{X}^k_{M/G/c\ PR} \approx \frac{1}{\Lambda_k}\left[\sum_{i=1}^{k}\lambda_i\left\{\frac{\sigma_{i-1}}{1-\sigma_{i-1}}\cdot\frac{\overline{z}_i}{c} + \frac{\sum_{j=1}^{i}\lambda_j(1 + C_v^2)\frac{\overline{z}_j^2}{c^2}}{2(1-\sigma_{i-1})(1-\sigma_i)}\right\}\right]\sigma_k^{\sqrt{2(c+1)}-2}.$$

Using $\overline{T}^{k}_{M/G/c\ PR} = \overline{X}^{k}_{M/G/c\ PR} + \overline{y}_k/c$, substituting into (A.10), and simplifying we get

$$\overline{R}_{\Gamma_k,C_k} \approx \overline{z}_k + \frac{1}{\lambda_k}\lambda\left(\sum_{i=1}^{k-1} g_i\right)\left(\sigma_k^{\sqrt{2(c+1)}-2} - \sigma_{k-1}^{\sqrt{2(c+1)}-2}\right) + \frac{1}{p_k}\lambda\, g_k\sigma_k^{\sqrt{2(c+1)}-2}, \quad p_k > 0,$$

where

$$g_i = p_i\frac{\sigma_{i-1}}{1-\sigma_{i-1}}\overline{z}_i/c \ + \ \frac{\lambda p_i \sum_{j=1}^{i}\{p_j(1+C_v^2)\overline{z}_j^2/c^2\}}{2(1-\sigma_{i-1})(1-\sigma_i)}.$$

We used $\lambda_i = \lambda p_i$ to obtain $g_i$. Substituting $\overline{x}_i = \overline{z}_i/c$ into the above expressions for $\overline{R}_{\Gamma_k,C_k}$ and $g_i$ we obtain (A.9) as required. Note that $\overline{z}_i = \overline{D}_i/k = (\overline{D}i)/(\overline{N}k)$, and since $c = P/k$, we get $\overline{x}_i = (\overline{D}i)/(\overline{N}P)$, for $i = 1, \ldots, k$.

## A.3 Proofs and Derivations for Chapter 7

### A.3.1 Proof of Theorem 7.1.1

**Theorem 7.1.1** *Consider a set of $K$ jobs with available parallelisms $(n_1, \ldots, n_K)$. Let $\Psi$ be a processor allocation policy that allocates $a_i^{\Psi}$ processors to job $i$, for $i = 1, \ldots, K$, and let $u_i^{\Psi} \equiv \min(a_i^{\Psi}, n_i)$ be the useful processor allocation to job $i$ under policy $\Psi$, for $i = 1, \ldots, K$. For a workload ERF $\gamma$ that is concave and nondecreasing and assuming that $E(j) = \gamma(j)$, i.e., jobs can dynamically and efficiently redistribute their work on the processsors allocated to them, we have*

$$\sum_{i=1}^{K}\gamma(u_i^{EQS}) \ \geq \ \sum_{i=1}^{K}\gamma(u_i^{\Psi}), \quad \textit{for any allocation policy } \Psi. \tag{A.13}$$

**Proof.** Without loss of generality assume that $n_1 \leq n_2 \leq \cdots \leq n_K$. We divide the proof into three cases. First, when there are enough processors so that each job gets as many processors under EQS as it can make use of. Second, when all jobs get the same allocation under EQS, and third when all jobs do not get the same allocation under EQS. Throughout the proof we use the following two properties:

- $\sum_{i=1}^{K} a_i^{\Psi} \leq P$.
- $u_i^{EQS} = a_i^{EQS}$, and $u_i^{\Psi} \leq a_i^{\Psi}$, $i = 1, \ldots, K$.

Case 1: $\sum_{i=1}^{K} n_i \leq P$.
We have $u_i^{EQS} = a_i^{EQS} = n_i \geq u_i^{\Psi}$, $i = 1, \ldots, K$. Hence,

$$\sum_{i=1}^{K}\gamma(u_i^{EQS}) = \sum_{i=1}^{K}\gamma(n_i) \geq \sum_{i=1}^{K}\gamma(u_i^{\Psi}),$$

where the last inequality follows because $\gamma$ is nondecreasing.

Case 2: $n_1 \geq P/K$, i.e., all jobs under EQS get P/K processors each.
By definition, $u_i^{EQS} = a_i^{EQS} = P/K$. Therefore,

$$\sum_{i=1}^{K} \gamma(u_i^{EQS}) = K\,\gamma\left(\frac{P}{K}\right) \geq K\gamma\left(\frac{\sum_{i=1}^{K} a_i^{\Psi}}{K}\right) \geq \sum_{i=1}^{K} \gamma(a_i^{\Psi}) \geq \sum_{i=1}^{K} \gamma(u_i^{\Psi}),$$

where we have used the fact that $\gamma$ is concave and nondecreasing.

Case 3: $n_1 < P/K$.
Under EQS jobs with low available parallelism get as many processors as their available parallelisms and the remaining jobs get the resultant equipartition number. Let the first J jobs under EQS get as many processors as their available parallelisms. That is, $a_i^{EQS} = n_i$, for $i = 1, \ldots, J$ and $a_i^{EQS} = (P - \sum_{\ell=J+1}^{K} n_\ell)/(K-J)$ for $i = J+1, \ldots, K$, where $a_j^{EQS} \leq a_k^{EQS}$ for $j \in \{1, \ldots, J\}$ and $k \in \{J+1, \ldots, K\}$. We now have

$$\sum_{i=1}^{K} \gamma(u_i^{EQS}) = \sum_{i=1}^{J} \gamma(n_i) + (K-J)\,\gamma\left(\frac{P - \sum_{i=1}^{J} n_i}{K-J}\right). \tag{A.14}$$

For any other policy $\Psi$ we have

$$\sum_{i=1}^{K} \gamma(u_i^{\Psi}) = \sum_{i=1}^{J} \gamma(u_i^{\Psi}) + \sum_{i=J+1}^{K} \gamma(u_i^{\Psi}) \leq \sum_{i=1}^{J} \gamma(u_i^{\Psi}) + (K-J)\,\gamma\left(\frac{P - \sum_{i=1}^{J} u_i^{\Psi}}{K-J}\right), \tag{A.15}$$

where the last inequality follows due to concavity of $\gamma$. On account of (A.14) and (A.15), to prove the theorem it suffices to show that

$$\sum_{i=1}^{J} \gamma(n_i) + (K-J)\,\gamma\left(\frac{P - \sum_{i=1}^{J} n_i}{K-J}\right) \geq \sum_{i=1}^{J} \gamma(u_i^{\Psi}) + (K-J)\,\gamma\left(\frac{P - \sum_{i=1}^{J} u_i^{\Psi}}{K-J}\right). \tag{A.16}$$

To complete the proof we make use of the following property of concave functions. For a concave function $f$

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} \geq \frac{f(x_4) - f(x_3)}{x_4 - x_3}, \quad \text{where } x_1 \leq x_2 \leq x_3 \leq x_4. \tag{A.17}$$

This property is illustrated by Figure A.1 where slope of line AB $\geq$ slope of line CD. Since $u_i^{\Psi} \leq n_i$, for $i = 1, \ldots, J$ we have $(P - \sum_{i=1}^{J} u_i^{\Psi})/(K-J) \geq (P - \sum_{i=1}^{J} n_i)/(K-J)$. Therefore,

$$u_i^{\Psi} \leq n_i \leq \frac{P - \sum_{i=1}^{J} n_i}{K-J} \leq \frac{P - \sum_{i=1}^{J} u_i^{\Psi}}{K-J}, \tag{A.18}$$

where the second inequality is a consequence of $a_j^{EQS} \leq a_k^{EQS}$ for $j \in \{1, \ldots, J\}$ and $k \in \{J+1, \ldots, K\}$. Applying (A.17) by using $x_1, x_2, x_3$, and $x_4$ from (A.18) we get,

$$\frac{\gamma(n_i) - \gamma(u_i^{\Psi})}{n_i - u_i^{\Psi}} \geq \frac{\gamma\left(\frac{P - \sum_{\ell=1}^{J} u_\ell^{\Psi}}{K-J}\right) - \gamma\left(\frac{P - \sum_{\ell=1}^{J} n_\ell}{K-J}\right)}{\left(\frac{P - \sum_{\ell=1}^{J} u_\ell^{\Psi}}{K-J}\right) - \left(\frac{P - \sum_{\ell=1}^{J} n_\ell}{K-J}\right)}$$

$$= \frac{\gamma\left(\frac{P-\sum_{\ell=1}^{J} u_\ell^\Psi}{K-J}\right) - \gamma\left(\frac{P-\sum_{\ell=1}^{J} n_\ell}{K-J}\right)}{\frac{\sum_{\ell=1}^{J}(n_\ell - u_\ell^\Psi)}{K-J}}, \quad i = 1, \ldots, J.$$

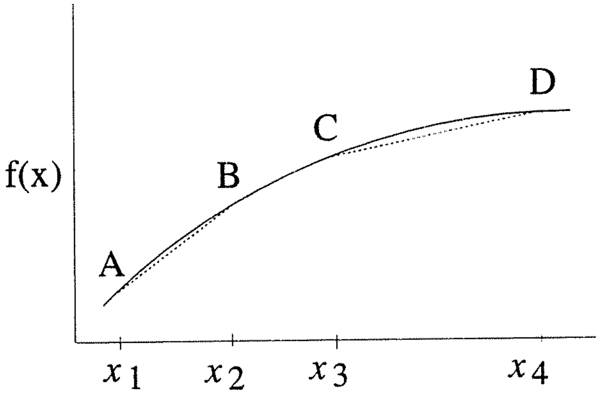


Figure A.1: A Property for Concave Functions

As a result, for $i = 1, \ldots, J$,

$$\gamma(n_i) - \gamma(u_i^\Psi) \geq (n_i - u_i^\Psi) \cdot \left[(K-J) \cdot \frac{\gamma\left(\frac{P-\sum_{\ell=1}^{J} u_\ell^\Psi}{K-J}\right) - \gamma\left(\frac{P-\sum_{\ell=1}^{J} n_\ell}{K-J}\right)}{\sum_{\ell=1}^{J}(n_\ell - u_\ell^\Psi)}\right].$$

Summing both sides from $i = 1$ to $J$ we get,

$$\sum_{i=1}^{J} \gamma(n_i) - \sum_{i=1}^{J} \gamma(u_i^\Psi) \geq (K-J) \cdot \left\{\gamma\left(\frac{P-\sum_{\ell=1}^{J} u_\ell^\Psi}{K-J}\right) - \gamma\left(\frac{P-\sum_{\ell=1}^{J} n_\ell}{K-J}\right)\right\}.$$

Rearranging terms,

$$\sum_{i=1}^{J} \gamma(n_i) + (K-J)\gamma\left(\frac{P-\sum_{i=1}^{J} n_i}{K-J}\right) \geq \sum_{i=1}^{J} \gamma(u_i^\Psi) + (K-J)\gamma\left(\frac{P-\sum_{i=1}^{J} u_i^\Psi}{K-J}\right),$$

which is what we set out to prove (compare with (A.16)).

∎

### A.3.2 Derivation of Min. and Max. $\overline{R}_{PSAPF}$ at $r = 1$: $\gamma^l$

In this section we provide details of how to minimize and maximize $\overline{R}_{PSAPF}$ at $r = 1$ and $\gamma = \gamma^l$ across all distributions of $N$ that have the same $\overline{N}$. $\overline{R}_{PSAPF}$ is given by (6.16) for these parameter settings. Our objective is to minimize or maximize $\overline{R}_{PSAPF}$ over all pmf's $\underline{p}$ subject to the constraint that $\sum_{i=1}^{P} ip_i = \overline{N}$. It is easy to verify that the set of pmf's that satisfies this equality constraint is a convex set[1]. We henceforth denote this convex set by $\Omega$. From [4] we note the following:

(1) Any local minimum of a convex function over a convex set is also a global minimum.

(2) If a convex function has a maximum over a convex set S, then the maximum is achieved at an extreme point of S, where an extreme point is a point that does not lie strictly within the line segment connecting two other points of the set.

If we can show that $\overline{R}_{PSAPF}$ is convex in $\underline{p}$ over the set $\Omega$, our task of finding the global minimum and maximum values of $\overline{R}_{PSAPF}$ over $\Omega$ will be considerably simplified. (Note that in general there is no known algorithm that obtains the global minimum and maximum for an arbitrary nonlinear function.) We have been unable to rigorously prove that $\overline{R}_{PSAPF}$ is convex as desired, but we have empirically verified this property by selecting random pairs of points in $\Omega$ and verifying that the line segment connecting the mean response time values between each pair lies above the mean response time function. We have also plotted the shape of $\overline{R}_{PSAPF}$ for 2 and 3 dimensional problem sizes and verified it to be convex in $\Omega$. We shall therefore assume that $\overline{R}_{PSAPF}$ is convex in $\Omega$ and use properties (1) and (2) from above.

We obtained the minimum values of $\overline{R}_{PSAPF}$ by writing a nonlinear program in GAMS [7]. By running the program over various values of $\lambda$, $C_v$, and $\overline{N}$ as inputs we obtained the curves shown in Figure 7.6. We specified different initial feasible points $\underline{p} \in \Omega$ to the GAMS program and always obtained the same value for the minimum, thus strengthening our belief that $\overline{R}_{PSAPF}$ is convex in $\Omega$.

To obtain the maximum values of $\overline{R}_{PSAPF}$ we first computed the extreme points in $\Omega$. It can be verified that an extreme point in $\Omega$ is obtained by considering only two nonzero values in the pmf $\underline{p}$ and attaching suitable weights to them so that the mean is $\overline{N}$.[2] (That only two nonzero values are needed results from the fact that there are only two equality constraints, the first $\sum p_i = 1$ and the second $\sum ip_i = \overline{N}$.) Once the extreme points in $\Omega$ were obtained we then computed $\overline{R}_{PSAPF}$ at these points and selected the maximum value (see property (2)).

### A.3.3 Proof of Theorem 7.2.2

**Theorem 7.2.2** *Under the workload assumptions of Section 7.2.1, $\overline{R}_{FCFS}$ and $\overline{R}_{PSAPF}$ increase linearly in $C_v^2$.*

[1] A nonempty set S in $I\!R^n$ is said to be convex if the line segment joining any two points of the set also belongs to the set, i.e., if $\overline{x}_1$ and $\overline{x}_2$ are in S, then $\lambda\overline{x}_1 + (1-\lambda)\overline{x}_2$ is also in S for all $\lambda$ between 0 and 1 [4].

[2] In the special case where $\overline{N}$ is an integer, the point specified by ($p_{\overline{N}} = 1$, $p_i = 0$ for $i \neq \overline{N}$) will also be an extreme point.

**Proof.** Let $\Gamma$ denote the system under consideration, with scheduling policy $\Psi$ which is either FCFS or PSAPF. Let a job be of type 1 if its demand is nonzero and of type 2 if its demand is zero. Type 1 jobs arrive according to a Poisson process with rate $\lambda\alpha$ and type 2 jobs arrive according to a Poisson process with rate $\lambda(1-\alpha)$. Let $\overline{R}_{\Gamma,i}$ denote the mean response time and $\overline{W}_{\Gamma,i}$ the mean waiting time (until first service) of type $i$ jobs in system $\Gamma$, $i = 1, 2$. Since $\Psi$ does not differentiate between type 1 and type 2 jobs and as a result $\overline{W}_{\Gamma,1} = \overline{W}_{\Gamma,2}$. Therefore, the mean response time of policy $\Psi$ in system $\Gamma$ is given by

$$\begin{aligned} \overline{R}_{\Psi} &= \alpha\overline{R}_{\Gamma,1} + (1-\alpha)\overline{R}_{\Gamma,2} \\ &= \alpha\overline{R}_{\Gamma,1} + (1-\alpha)\overline{W}_{\Gamma,2} \\ &= \alpha\overline{R}_{\Gamma,1} + (1-\alpha)\overline{W}_{\Gamma,1}. \end{aligned} \tag{A.19}$$

In system $\Gamma$ type 2 jobs do not delay type 1 jobs. As a result $\overline{R}_{\Gamma,1}$ is the same as the mean response time of a system $\Gamma_I$ where only type 1 jobs arrive with rate $\lambda\alpha$ and have exponential demands given by $D_i \sim \exp(\mu_i)$. Thus $\overline{R}_{\Gamma,1} = \overline{R}_{\Gamma_I}$ and $\overline{W}_{\Gamma,1} = \overline{W}_{\Gamma_I}$. Now consider a "faster" system $\Gamma_{II}$ in which type 1 jobs arrive with rate $\lambda$ and have exponential demands given by $D_i \sim \exp(\mu_i/\alpha)$. Then as in the proof of Theorem 7.2.1 it follows that $\overline{R}_{\Gamma_I} = 1/\alpha\overline{R}_{\Gamma_{II}}$ and $\overline{W}_{\Gamma_I} = 1/\alpha\overline{W}_{\Gamma_{II}}$, where $\overline{R}_{\Gamma_{II}}$ and $\overline{W}_{\Gamma_{II}}$ are independent of $\alpha$. We now have $\overline{R}_{\Gamma,1} = 1/\alpha\overline{R}_{\Gamma_{II}}$ and $\overline{W}_{\Gamma,1} = 1/\alpha\overline{W}_{\Gamma_{II}}$. Using this in (A.19) we have the mean response time of policy $\Psi$ in system $\Gamma$ is given by

$$\begin{aligned} \overline{R}_{\Psi} &= \alpha \times \frac{1}{\alpha}\overline{R}_{\Gamma_{II}} + (1-\alpha) \times \frac{1}{\alpha}\overline{W}_{\Gamma_{II}} \\ &= \overline{R}_{\Gamma_{II}} - \overline{W}_{\Gamma_{II}} + \frac{1}{\alpha}\overline{W}_{\Gamma_{II}} \\ &= \overline{R}_{\Gamma_{II}} - \overline{W}_{\Gamma_{II}} + \frac{1+C_v^2}{2}\overline{W}_{\Gamma_{II}} \\ &= b_{\Psi} + c_{\Psi}C_v^2, \end{aligned}$$

where $b_{\Psi} = \overline{R}_{\Gamma_{II}} - \frac{1}{2}\overline{W}_{\Gamma_{II}}$ and $c_{\Psi} = \frac{1}{2}\overline{W}_{\Gamma_{II}}$. Note that $b_{\Psi} > 0$ since $\overline{R}_{\Gamma_{II}} \geq \overline{W}_{\Gamma_{II}}$, and $c_{\Psi} > 0$ since in $\Gamma_{II}$ jobs do not always receive instant service. ∎

## A.4 Proofs for Chapter 8

### A.4.1 Proof of Theorem 8.1.1

**Theorem 8.1.1** *If $\ell$ and $m$ are constants such that $\ell \leq m$, then under the workload assumptions* $(\cdot, \exp(1/\overline{D}), r = 0, \gamma \in \mathcal{E}^c)$,

$$\overline{R}_{EQS}(m \leq N \leq P) \quad \leq \quad \overline{R}_{EQS}(1 \leq N \leq \ell).$$

Let $\Gamma_I = (EQS,\ m \leq N \leq P,\ \exp(1/\overline{D}),\ r = 0,\ \gamma \in \mathcal{E}^c)$, and let $\Gamma_{II} = (EQS,\ 1 \leq N \leq \ell,\ \exp(1/\overline{D}),\ r = 0,\ \gamma \in \mathcal{E}^c)$. The following lemma is used in the proof of this theorem.

**Lemma A.4.1** *Suppose there are $K$ jobs in system $\Gamma_I$ such that the allocation of processing power to these jobs is $(a_1, a_2, \ldots, a_K)$, and suppose there are $M \geq K$ jobs in system $\Gamma_{II}$ such that the allocation of processing power to these jobs is $(b_1, b_2, \ldots, b_M)$. Then*

$$\sum_{i=1}^{K} \gamma(a_i) \geq \sum_{i=1}^{K} \gamma(b_i).$$

*(Note that the summation is from 1 to $K$ on both sides.)*

**Proof.** Since the ERF $\gamma$, which is the same for both systems, is *concave and nondecreasing*

$$\sum_{i=1}^{K} \gamma(b_i) \leq K\gamma\left(\frac{\sum_{i=1}^{K} b_i}{K}\right) \leq K\gamma\left(\frac{P}{K}\right) \tag{A.20}$$

Since $b_i \leq \ell$, $i = 1, 2, \ldots, K$, and $\gamma$ is nondecreasing

$$\sum_{i=1}^{K} \gamma(b_i) \leq K\gamma(\ell).$$

Using this along with (A.20) we get

$$\sum_{i=1}^{K} \gamma(b_i) \leq K \min(\gamma(\ell), \gamma(P/K)) \leq K \min(\gamma(m), \gamma(P/K)), \tag{A.21}$$

where the last inequality follows because $\ell \leq m$ and $\gamma$ is nondecreasing.
We now show that

$$K \min(\gamma(m), \gamma(P/K)) \leq \sum_{i=1}^{K} \gamma(a_i). \tag{A.22}$$

To see this consider the following two cases:

(i) $m \geq P/K$:
If $m \geq P/K$ then $a_i = P/K$, $i = 1, 2, \ldots, K$ (since $P/K$ is the equiallocation number and the available parallelism of each job in $\Gamma_I$ is at least $m$). Hence

$$K \min(\gamma(m), \gamma(P/K)) = K\gamma(P/K) = \sum_{i=1}^{K} \gamma(a_i).$$

(ii) $m < P/K$:

Since $m < P/K$ each job in $\Gamma_I$ gets at least $m$ processors. That is, $a_i \geq m$, $i = 1, 2, \ldots, K$. Hence

$$K \min(\gamma(m), \gamma(P/K)) = K\gamma(m) \leq \sum_{i=1}^{K} \gamma(a_i).$$

This proves inequality (A.22). The lemma follows from inequalities (A.21) and (A.22). ■

**Proof of Theorem 8.1.1.** We prove this theorem using sample path analysis. We make use of the following observations:

(i) If a job is allocated processing power $x$ then the residual life time of the job is exponentially distributed with rate $\gamma(x)\mu$.

(ii) If there are $k > 0$ jobs in system $\Gamma_i$, $i = I, II$, at time $t$ with the $j^{th}$ job having a processor allocation of $x_j$, $1 \leq j \leq k$, then the time to the next departure from $\Gamma_i$ is exponentially distributed with rate $\sum_{j=1}^{k} \gamma(x_j)\mu$.

Let $Q_i(t)$ be the number of jobs in system $\Gamma_i$ at time $t$, $i = I, II$. Let $\alpha_k^I(t) = \gamma(a_k)/P$, $k = 1, 2, \ldots, Q_I(t)$, where $a_k$ is the processor allocation to the $k^{th}$ job in $\Gamma_I$ at time $t$. Similarly let $\alpha_k^{II}(t) = \gamma(b_k)/P$, $k = 1, 2, \ldots, Q_{II}(t)$, where $b_k$ is the processor allocation to the $k^{th}$ job in $\Gamma_{II}$ at time $t$. Thus the $k^{th}$ job in $\Gamma_i$ departs with rate $\alpha_k^i(t)P\mu$, $i \in \{I, II\}$.

*Coupling of Sample Paths in $\Gamma_I$ and $\Gamma_{II}$*

Fix the arrival times of jobs to be the same in $\Gamma_I$ and $\Gamma_{II}$. Fix sequences of integers $\{N_i^I\}_{i=1}^{\infty}$ and $\{N_i^{II}\}_{i=1}^{\infty}$ for available job parallelisms in $\Gamma_I$ and $\Gamma_{II}$ respectively, where $m \leq N_i^I \leq P$ and $1 \leq N_i^{II} \leq \ell$, $i = 1, 2, \ldots$. Consider that *potential job completions* [85] occur in each of $\Gamma_I$ and $\Gamma_{II}$ at jumps of a Poisson process with rate $P\mu$. Fix the same potential completion instants $\{T_i\}_{i=1}^{\infty}$ in both $\Gamma_I$ and $\Gamma_{II}$. To generate *actual* job completion times in $\Gamma_I$ and $\Gamma_{II}$ let $\{U_i\}_{i=1}^{\infty}$ be i.i.d. Uniform[0,1) random variables. At the $r^{th}$ potential completion instant $T_r$, the $k^{th}$ job in $\Gamma_i$ departs if

$$U_r \in \left[\sum_{j=1}^{k-1} \alpha_j^i(T_r^-), \sum_{j=1}^{k} \alpha_j^i(T_r^-)\right), \quad k = 1, 2, \ldots, Q_i(t), \quad i \in \{I, II\}. \tag{A.23}$$

This ensures that the probability that the $k^{th}$ job departs from $\Gamma_i$ is $\alpha_k^i(T_r^-)$.

*Sample Path Analysis*

Using the above coupling of sample paths we show by an induction over time that for every sample path, for all $t \geq 0$

$$Q_I(t) \leq Q_{II}(t). \tag{A.24}$$

We carry out the induction only over arrival instants and potential completion instants since no jobs depart in between these event times. Let $\{t_i\}_{i=0}^{\infty}$ be the sequence of arrival and potential completion times arranged in increasing order. Let both $\Gamma_I$ and $\Gamma_{II}$ start out with zero jobs each. Then clearly (A.24) is satisfied at $t = t_0$. Assume that (A.24) is true for all $t \leq t_j$. Since no jobs arrive or depart in $(t_j, t_{j+1})$ (A.24) is also true for all $t_j < t < t_{j+1}$. We now prove that (A.24) is true at $t = t_{j+1}$. Consider all possible events at time $t_{j+1}$.

1. Job Arrival:

   By the induction hypothesis it follows that

   $$Q_I(t_{j+1}) = Q_I(t_j) + 1 \leq Q_{II}(t_j) + 1 = Q_{II}(t_{j+1}).$$

2. Potential Completion:

   (a) No departure from each of $\Gamma_I$ and $\Gamma_{II}$:

   $$Q_I(t_{j+1}) = Q_I(t_j) \leq Q_{II}(t_j) = Q_{II}(t_{j+1}).$$

   (b) Departure from $\Gamma_I$ but not from $\Gamma_{II}$:

   $$Q_I(t_{j+1}) = Q_I(t_j) - 1 \leq Q_{II}(t_j) - 1 = Q_{II}(t_{j+1}) - 1 < Q_{II}(t_{j+1}).$$

   (c) Departure from each of $\Gamma_I$ and $\Gamma_{II}$:

   $$Q_I(t_{j+1}) = Q_I(t_j) - 1 \leq Q_{II}(t_j) - 1 = Q_{II}(t_{j+1}).$$

   (d) Departure from $\Gamma_{II}$ but not from $\Gamma_I$:

   This implies that

   $$U_r \in \left[0, \sum_{i=1}^{Q_{II}(t_j)} \alpha_i^{II}(t_{j+1}^-)\right), \quad \text{and} \quad U_r \in \left[\sum_{i=1}^{Q_I(t_j)} \alpha_i^I(t_{j+1}^-), 1\right), \tag{A.25}$$

   where $t_{j+1} = T_r$, the $r^{th}$ potential completion instant, $1 \leq r \leq j+1$. Since these two intervals overlap, we have

   $$\sum_{i=1}^{Q_I(t_j)} \alpha_i^I(t_{j+1}^-) < \sum_{i=1}^{Q_{II}(t_j)} \alpha_i^{II}(t_{j+1}^-). \tag{A.26}$$

   Since $Q_I(t_j) \leq Q_{II}(t_j)$ (induction hypothesis) we have from Lemma A.4.1 that

   $$\sum_{i=1}^{Q_I(t_j)} \alpha_i^{II}(t_{j+1}^-) = \frac{1}{P}\sum_{i=1}^{Q_I(t_j)} \gamma(b_i) \leq \frac{1}{P}\sum_{i=1}^{Q_I(t_j)} \gamma(a_i) = \sum_{i=1}^{Q_I(t_j)} \alpha_i^I(t_{j+1}^-). \tag{A.27}$$

(A.26) and (A.27) together imply

$$\sum_{i=1}^{Q_I(t_j)} \alpha_i^{II}(t_{j+1}^-) \leq \sum_{i=1}^{Q_I(t_j)} \alpha_i^{I}(t_{j+1}^-) < \sum_{i=1}^{Q_{II}(t_j)} \alpha_i^{II}(t_{j+1}^-),$$

which shows that $Q_I(t_j) < Q_{II}(t_j)$. Hence

$$Q_I(t_{j+1}) = Q_I(t_j) \leq Q_{II}(t_j) - 1 = Q_{II}(t_{j+1}).$$

This completes the proof by induction. Thus, we have shown for every sample path that $Q_I(t) \leq Q_{II}(t)$, $\forall t \geq 0$. Hence for every sample path

$$\overline{Q}_I = \lim_{t\to\infty} \frac{1}{t} \int_0^t Q_I(s)ds \leq \lim_{t\to\infty} \frac{1}{t} \int_0^t Q_{II}(s)ds = \overline{Q}_{II},$$

from which it follows by Little's Law [76] that $\overline{R}_{\Gamma_I} \leq \overline{R}_{\Gamma_{II}}$ for every sample path. Now uncondition on arrival times, available parallelisms, and potential completion times. ∎

**Remark:** Note that the above proof does not require the assumption of Poisson arrivals. The arrival process can be any GI process.

## A.4.2 Proof of Theorem 8.1.2

**Theorem 8.1.2** *Let $\Gamma_I$ be a system with the EQS policy and primitive workload variables $\{(A_i, D_i, N_i \geq k, E_i), i = 1, 2, \ldots\}$, where $A_i$ is job i's arrival time, $D_i$ its total demand, $N_i$ its available parallelism, and $E_i$ its execution rate function. Let these primitive variables have arbitrary marginals (given that $N_i \geq k$, and the other variables make sense, e.g., $D_i \geq 0$) with arbitrary dependencies among them. Let $\Gamma_{II}$ be a system with the EQS policy and the same workload as $\Gamma_I$ except that $N_i = k$ for all $i = 1, 2, \ldots$. Then*

$$\overline{R}_{\Gamma_I} \leq \overline{R}_{\Gamma_{II}}, \quad k = 1, 2, \ldots, P.$$

**Proof.** Let $\mathcal{Q}_I(t)$ be the set of jobs in system $\Gamma_I$ at time $t$, and likewise, let $\mathcal{Q}_{II}(t)$ be the set of jobs in system $\Gamma_{II}$ at time $t$. We prove this theorem by suitably coupling sample paths for $\Gamma_I$ and $\Gamma_{II}$, and showing that for every sample path $\mathcal{Q}_I(t) \subseteq \mathcal{Q}_{II}(t)$, for all $t \geq 0$, from which it will follow that $\overline{R}_{\Gamma_I} \leq \overline{R}_{\Gamma_{II}}$.

*Coupling of Sample Paths in $\Gamma_I$ and $\Gamma_{II}$*

Fix $\{A_i, D_i\}_{i=1}^{\infty}$ as the same for both $\Gamma_I$ and $\Gamma_{II}$. For system $\Gamma_I$ choose a sequence $\{N_i^I\}_{i=1}^{\infty}$ such that $N_i^I \geq k$, $i = 1, 2, \ldots$. For system $\Gamma_{II}$ fix $N_i^{II} = k$ for all $i = 1, 2, \ldots$. Pick a sequence of execution rate functions $\{E_i^I\}_{i=1}^{\infty}$ for $\Gamma_I$ where $E_i^I$ is nondecreasing, $i = 1, 2, \ldots$

Fix the execution rate function for job $i$ in system $\Gamma_{II}$ as $E_i^{II}(x) = E_i^I(x)$, for $0 \leq x \leq k$, and $E_i^{II}(x) = E_i^I(k)$, $x \geq k$.

*Sample Path Analysis*

Under the above coupling of sample paths we show by induction over time that for every pair of coupled sample paths, for all $t \geq 0$,

$$Q_I(t) \subseteq Q_{II}(t). \tag{A.28}$$

Let $a_i^I(t)$ and $a_i^{II}(t)$ be the allocations of processing power to job $i$ in system $\Gamma_I$ and $\Gamma_{II}$, respectively, at time $t$. Note that $a_i^I(t) = 0$ if $i \notin Q_I(t)$, and $a_i^{II}(t) = 0$ if $i \notin Q^{II}(t)$. From (A.28) it follows that

$$a_i^I(t) \geq a_i^{II}(t), \quad i \in Q^I(t), \tag{A.29}$$

because

$$\begin{aligned} a_i^{II}(t) &= \min(k, P/|Q_{II}(t)|) \\ &\leq \min\left(N_i^I, P/|Q_{II}(t)|\right), \qquad \text{since } N_i^I \geq k \\ &\leq \min\left(N_i^I, P/|Q_I(t)|\right), \qquad \text{since } |Q_I(t)| \leq |Q_{II}(t)| \\ &\leq a_i^I(t). \end{aligned}$$

The last inequality holds because if job $i$ gets $N_i^I$ processors in $\Gamma_I$ then $a_i^I(t) = N_i^I$ and if job $i$ gets less than $N_i^I$ processors then it gets at least as many as the equiallocation number $P/|Q_I(t)|$, by definition of the EQS policy.

We carry out the induction over arrival and departure times in $\Gamma_I$ and $\Gamma_{II}$. Let $\{t_i\}_{i=0}^{\infty}$ be the sequence of arrival and departure times in $\Gamma_I$ and $\Gamma_{II}$ arranged in increasing order. Let both $\Gamma_I$ and $\Gamma_{II}$ start out with zero jobs each at $t = 0$. Then clearly (A.28) is satisfied at $t = t_0$. Assume that (A.28) is true for all $t \leq t_j$. Since no jobs arrive or depart in $(t_j, t_{j+1})$ it follows that (A.28) is true for all $t < t_{j+1}$. We now prove that (A.28) is true at $t = t_{j+1}$. Consider all possible events at time $t_{j+1}$.

1. Arrival of job k:

   By the induction hypothesis it follows that

   $$Q_I(t_{j+1}) = Q_I(t_j) \cup \{k\} \subseteq Q_{II}(t_j) \cup \{k\} = Q_{II}(t_{j+1}).$$

2. Departure from $\Gamma_I$ only:

   $$Q_I(t_{j+1}) \subset Q_I(t_j) \subseteq Q_{II}(t_j) = Q_{II}(t_{j+1}).$$

3. Departure from both $\Gamma_I$ and $\Gamma_{II}$:
   Suppose job $\ell$ departs from $\Gamma_I$ and job $m$ departs from $\Gamma_{II}$. Then we have the following cases depending on how $\ell$ is related to $m$:

   (a) $\ell = m$:

   $$Q_I(t_{j+1}) = Q_I(t_j) - \{\ell\} \subseteq Q_{II}(t_j) - \{m\} = Q_{II}(t_{j+1}).$$

   (b) $\ell \neq m$:
   Depending on whether or not $m$ and $\ell$ are present in $Q_I(t_j)$ and $Q_{II}(t_j)$, respectively, we have the following cases:

   (i) $m \in Q_I(t_j)$:
   This is impossible. The reason is that $m \in Q_I(t_j) \Rightarrow m \in Q_I(t_{j+1})$ because $\ell \neq m$. Since $Q_I(s) \subseteq Q_{II}(s)$, for all $0 \leq s < t_{j+1}$, it follows from (A.29) that $a_m^I(s) \geq a_m^{II}(s)$ for $A_m \leq s < t_{j+1}$. Since job $m$ has not departed from $\Gamma_I$ by time $t_{j+1}$, we have

   $$D_m > \int_0^{t_{j+1}} E_m^I(a_m^I(s))ds \geq \int_0^{t_{j+1}} E_m^{II}(a_m^{II}(s))ds.$$

   Hence job $m$ has not departed from $\Gamma_{II}$ by time $t_{j+1}$, which is a contradiction.

   (ii) $m \notin Q_I(t_j)$, $\ell \in Q_{II}(t_j)$:
   Since $m \notin Q_I(t_j)$, we have by the induction hypothesis that $Q_I(t_j) \subset Q_{II}(t_j)$ and since $\ell$ departs $\Gamma_I$ but not $\Gamma_{II}$ at time $t_{j+1}$ it follows that

   $$Q_I(t_{j+1}) \subset Q_{II}(t_{j+1}).$$

   (iii) $m \notin Q_I(t_j)$, $\ell \notin Q_{II}(t_j)$:
   Similar to case (i), it is impossible that $\ell \in Q_I(t_j)$ and $\ell \notin Q_{II}(t_j)$.

4. Departure from $\Gamma_{II}$ only:
   Suppose job $m$ departs from $\Gamma_{II}$. Then either $m \in Q_I(t_j)$ or $m \notin Q_I(t_j)$. The former case is impossible as in case 3(b)(i). In the latter case $Q_I(t_j) \subset Q_{II}(t_j)$ and $Q_I(t_{j+1}) \subseteq Q_{II}(t_{j+1})$.

This completes the proof by induction. We have shown that for sample path $Q_I(t) \subseteq Q_{II}(t)$. Therefore, job $i$ departs from $\Gamma_I$ at least as early as it does from from $\Gamma_{II}$, from which it follows that the response time of job $i$ in $\Gamma_I$ is less than or equal to its response time in $\Gamma_{II}$ for every sample path, $i = 1, 2, \ldots$. Therefore $\overline{R}_{\Gamma_I} \leq \overline{R}_{\Gamma_{II}}$ for every sample path. Now uncondition on $\{(A_i, D_i, N_i^I, E_i^I), i = 1, 2, \ldots\}$. ∎

# Bibliography

[1] R. Agrawal, R. Mansharamani, and M. Vernon. Response Time Bounds for Parallel Processor Allocation Policies. Technical Report #1152, Computer Sciences Department, University of Wisconsin-Madison, June 1993.

[2] F. Baccelli, and A. Makowski. Queueing Models for Systems with Synchronization Constraints. *Proceedings of the IEEE 77*, 1 (January 1989), 138-161.

[3] F. Baccelli, W. Massey, and D. Towsley. Acyclic Fork-Join Queueing Networks. *Journal of the ACM 36*, (1989), 615-642.

[4] M. Bazaraa, and C. Shetty. Nonlinear Programming: Theory and Algorithms. John Wiley & Sons, New York 1979.

[5] A. Bondi, and J. Buzen. The Response Times of Priority Classes under Preemptive Resume in M/G/m Queues. *Performance Evaluation Review 12*, 3 (August 1984), 195-201.

[6] A. Bricker, M. Litzkow, and M. Livny. Condor Technical Summary. Technical Report TR 1069, Computer Sciences Department, University of Wisconsin, Madison, WI, Jan. 1992.

[7] A. Brooke, D. Kendrick, and A. Meerhaus. GAMS, a User's Guide. Scientific Press, Redwood City, CA, 1988.

[8] S. Brumelle. Some Inequalities for Parallel-Server Queues. *Operations Research 19*, 2 (1971), 402–413.

[9] D. Burman, and D. Smith. Approximate Analysis of a Queueing Model with Bursty Traffic. *Bell System Technical Journal 62* (1983), 1433–1453.

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

[11] J. Buzen, and A. Bondi. The Response Time of Priority Classes under Preemptive Resume in M/M/m Queues. *Operations Research 31*, 2 (1983), 456-465.

[12] S. Cheng, and S. Dandamudi. Scheduling in Parallel Systems with a Hierarchical Organization of Tasks. *ACM International Conference on Supercomputing,* Washington, D.C., July 1992, 377–386.

[13] C. Chang, R. Nelson, and D. Yao. Optimal Task Scheduling on Distributed Parallel Processors. *Proceedings of Performance'93.*

[14] R., Conway, L. Maxwell, and L. Miller. *Theory of Scheduling.* Addison-Wesley, Reading, Massachusetts, 1967.

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

[16] D. Culler et al. LogP: Towards a Realistic Model of Parallel Computation. Technical Report. Computer Sciences Division, University of California, Berkeley, Jan. 1993.

[17] G. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, 1963.

[18] L. Dowdy. On the Partitioning of Multiprocessor Systems. Technical Report, Vanderbilt University, Nashville, TN, July 1988.

[19] D. Eager, J. Zahorjan, and E. Lazowska. Speedup Versus Efficiency in Parallel Systems. *IEEE Transactions on Computers, 38* 3 (Mar. 1989), 408-423.

[20] H. Flatt. A Simple Model for Parallel Processing. *IEEE Computer 17*, 11 (Nov. 1984), pg. 95.

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

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

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

[24] E. Gelenbe, D. Ghosal, and S. Tripathi. Analysis of Processor Allocation in Large Multiprocessor Systems. *Proceedings of the International Conference on the Performance of Distributed Systems and Integrated Communication Networks*, Kyoto, Japan, Sep. 1991.

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

[26] G. Grimmett, and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, 1989.

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

[28] A. Gupta, A. Tucker, and S. Urushibara. The Impact of Operating System Scheduling Policies and Synchronization Methods on the Performance of Parallel Applications. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 19*, 1 (May 1991), 120–132.

[29] J. Gustafson. Reevaluating Amdahl's Law. *Communications of the ACM*, May 1988, 532–533.

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

[31] L. Kleinrock. Time-shared Systems: A Theoretical Treatment. *Journal of the ACM 14*, 2 (Apr. 1967), 242–261.

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

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

[34] D. Kouvatsos. Maximum Entropy and the G/G/1/N Queue. *Acta Informatica 23*, 5 (1986), 545–565.

[35] D. Kouvatsos. A Maximum Entropy Analysis of the G/G/1 Queue at Equilibrium. *Journal of the Operational Research Society 39*, 2 (Feb. 1988), 183–200.

[36] D. Kouvatsos, and N. Tabet-Aouel. A Maximum Entropy Priority Approximation for a Stable G/G/1 Queue. *Acta Informatica 27*, 3 (1989), 247–286.

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

[38] E. Lazowska, J. Zahorjan, G. Graham, and K. Sevcik. *Quantitative System Performance: Computer System Analysis Using Queueing Network Models.* Prentice Hall 1984.

[39] S. Leutenegger. Issues in Multiprogrammed Multiprocessor Sharing. Ph.D. Thesis, Technical Report #954, Department of Computer Sciences, University of Wisconsin-Madison, Aug. 1990.

[40] 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.

[41] S. Leutenegger, and M. Vernon. The Performance of Multiprogrammed Multiprocessor Scheduling Policies. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 18*, 1 (May 1990), 226–236.

[42] Z. Liu and F. Baccelli. Generalized Precedence-Based Queueing Systems. *Mathematics of Operations Research 17*, 3 (Aug. 1992), 615–639.

[43] S. Majumdar, D. Eager, and R. Bunt. Scheduling in Multiprogrammed Parallel Systems. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 16*, 1 (May 1988), 104–113.

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

[45] A. Makowski, and R. Nelson. Distributed Parallelism Considered Harmful. Research Report RC 17448, IBM Research Division, 1991.

[46] A. Makowski, and R. Nelson. Optimal Scheduling for a Distributed Parallel Processing Model. Research Report RC 17449, IBM Research Division, 1991.

[47] 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.

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

[49] V.Naik, S. Setia, and M. Squillante. Performance Analysis of Job Scheduling Policies in Parallel Supercomputing Environments. *Proceedings of Supercomputing'93*, November 1993. IBM Research Report RC 19138, Sep. 1993.

[50] R. Nelson. A Performance Evaluation of a General Parallel Processing Model. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 18*, 1 (May 1990), 13–26.

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

[52] R. Nelson, and A. Tantawi. Approximate Analysis of Fork/Join Synchronization in Parallel Queues. *IEEE Transactions on Computers 37*, (Jun. 1988), 739–743.

[53] R. Nelson, and D. Towsley. A Performance Evaluation of Several Priority Policies for Parallel Processing Systems. COINS Technical Report 91-32, Computer and Information Sciences, University of Massachusetts, Amherst, MA, May 1991. (To appear in JACM.)

[54] R. Nelson, D. Towsley, and A. Tantawi. Performance Analysis of Parallel Processing Systems. *IEEE Transactions on Software Engineering 14*, 4 (Apr. 1988), 532–540.

[55] R. Nelson, D. Towsley, and A. Tantawi. The Order Statistics of the Sojourn Times of Customers that Form a Single Batch in the $M^X/M/c$ Queue. Research Report, IBM T.J. Watson Research Center, Aug. 1989.

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

[57] L. Ni, and C. Wu. Design Tradeoffs for Process Scheduling in Shared Memory Multiprocessor Systems. *IEEE Transactions on Software Engineering 15*, 3 (Mar. 1989), 327–334.

[58] J. Ousterhout. Scheduling Techniques for Concurrent Systems. *3rd International Conference on Distributed Computing Systems*, (1982) 22–30.

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

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

[61] A. Roberts, and D. Varberg. *Convex Functions.* Academic Press, New York, 1973.

[62] S. Ross. *Stochastic Processes.* New York, Wiley 1983.

[63] E. Rosti, E. Smirni, L. Dowdy, G. Serazzi, and B. Carlson. Robust Partitioning Policies of Multiprocessor Systems. Technical Report, Department of Computer Science, Vanderbilt University 1992. To appear, in *Performance Evaluation* (Special issue on the performance modeling of parallel processing systems).

[64] 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.

[65] C. Sauer, and K. M. Chandy. *Computer System Performance Modeling.* Prentice-Hall, 1981.

[66] M. Seager, and J. Stichnoth. Simulating the Scheduling of Parallel Supercomputer Applications. Technical Report, User Systems Division, Lawrence Livermore National Laboratory, Sep. 1989.

[67] S. Setia, M. Squillante, and S. Tripathi. Analysis of Processor Allocation in Multiprogrammed Parallel Processing Systems.Technical Report CS-TR-2840, University of Maryland, College Park, MD, Feb. 1992.

[68] S. Setia, M. Squillante, and S. Tripathi. Processor Scheduling on Multiprogrammed, Distributed Memory Parallel Systems. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 21*, 1 (May 1993), 158–170.

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

[70] S. Setia, and S. Tripathi. A Comparative Analysis of Static Processor Partitioning Policies for Parallel Computers. *Proceedings of the International Workshop on Modeling and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 1993.

[71] K. Sevcik. Characterization of Parallelism in Applications and Their Use in Scheduling. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 17*, 1 (1989), 171–180.

[72] K. Sevcik. Application Scheduling and Processor Allocation in Multiprogrammed Parallel Processing Systems. To appear, in *Performance Evaluation* (Special issue on the performance modeling of parallel processing systems).

[73] 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: Proceedings of the 18th Symposium on the Interface*, American Statistical Association (1986), 251–256.

[74] E. Smirni, E. Rosti, L. Dowdy, and G. Serazzi. Evaluation of Multiprocessor Allocation Policies. Technical Report, Vanderbilt University, Nashville, TN, 1993.

[75] M. Squillante. MAGIC: A Computer Performance Modeling Tool Based on Matrix-Geometric Techniques. *Proceedings of the* $5^{th}$ *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Feb. 1991.

[76] S. Stidham. A last word on $L = \lambda W$. *Operations Research 22*, 2 (1974), 417–421.

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

[78] N. Tabet-Aouel, and D. Kouvatsos. On an Approximation to the Mean Response Times of Priority Classes in a Stable G/G/c/PR Queue. *Journal of the Operational Research Society 43*, 3 (Mar 1992), 227–239.

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

[80] D. Towsley, C. Rommel, and J. Stankovic. Analysis of Fork-Join Program Response Times on Multiprocessors. *IEEE Transactions on Parallel and Distributed Systems 1*, 3 (July 1990), 286–303.

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

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

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

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

[85] J. Walrand. *Introduction to Queueing Networks.* Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[86] R. Walstra. Nonexponential Networks of Queues: a Maximum Entropy Analysis. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review* , 1985, 27–37.

[87] W. Whitt. An Interpolation Approximation for the Mean Workload in a GI/G/1 Queue. *Operations Research*, Vol. 37, No. 6, 1989, pp. 936-952.

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

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

[90] D. Yao. Some Results for the Queues $M^X/M/c$ and $GI^X/G/c$. *Operations Research Letters 4*, 2 (July 1985), 79–83.

[91] J. Zahorjan, and C. McCann. Processor Scheduling in Shared Memory Multiprocessors. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 18*, 1 (May 1990), 214–225.

[92] S. Zhou, and T. Brecht. Processor-pool-based Scheduling for Large-Scale NUMA Multiprocessors. *Proceedings of ACM SIGMETRICS Conference; Performance Evaluation Review 19*, 1 (May 1991), 133–142.