Stochastic Models and Analysis for Resource Management in Server Farms

Varun Gupta

CMU-CS-11-114

May 2011

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Mor Harchol-Balter, Chair
David G. Andersen
Anupam Gupta
Alan Scheller-Wolf
Devavrat Shah, MIT
Don Towsley, UMass (Amherst)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Queueing theory, Multi-server systems, Load balancing, Scheduling, $M/G/k$, Time-varying load, Energy management, Stochastic modeling, Heavy-traffic analysis

Abstract

Server farms are popular architectures for computing infrastructures such as supercomputing centers, data centers and web server farms. As server farms become larger and their workloads more complex, designing efficient policies for managing the resources in server farms via trial-and-error becomes intractable. In this thesis, we employ stochastic modeling and analysis techniques to understand the performance of such complex systems and to guide design of policies to optimize the performance.

There is a rich literature on applying stochastic modeling to diverse application areas such as telecommunication networks, inventory management, production systems, and call centers, but there are numerous *disconnects* between the workloads and architectures of these traditional applications of stochastic modeling and how compute server farms operate, necessitating new analytical tools. To cite a few:

(*i*) Unlike call durations, supercomputing jobs and file sizes have high variance in service requirements and this critically affects the optimality and performance of scheduling policies.

(*ii*) Most existing analysis of server farms focuses on the First-Come-First-Served (FCFS) scheduling discipline, while time sharing servers (e.g., web and database servers) are better modeled by the Processor-Sharing (PS) scheduling discipline.

(*iii*) Time sharing systems typically exhibit thrashing (resource contention) which limits the achievable concurrency level, but traditional models of time sharing systems ignore this fundamental phenomenon.

(*iv*) Recently, minimizing energy consumption has become an important metric in managing server farms. State-of-the-art servers come with multiple knobs to control energy consumption, but traditional queueing models don't take the metric of energy consumption into account.

In this thesis we attempt to bridge some of these disconnects by bringing the stochastic modeling and analysis literature closer to the realities of today's compute server farms. We introduce new queueing models for computing server farms, develop new stochastic analysis techniques to evaluate and understand these queueing models, and use the analysis to propose resource management algorithms to optimize their performance.

Acknowledgments

This thesis is merely the ticket stub of a long roller coaster ride, and among my many companions, there are first among equals without whom I might not have made it to the end. Kumar Avijit, Hetunandan Kamisetty, Balakrishnan Narayanaswamy, Swapnil Patil, Amar Phanishayee, Vyas Sekar, Gaurav Veda: I owe you much more than you realize.

This thesis would not be possible without my advisor Mor Harchol-Balter who initiated me into the beautiful world of stochastic processes, gave valuable advice and support at every stage, and above all, remained patient. I hope that some of her optimism and enthusiasm has rubbed off on me during my stay at Carnegie Mellon.

I have also been extremely lucky to have had exceptional mentors whose faith in me made me feel like a valuable part of the research community: Alan Scheller-Wolf, who was always forthcoming with advice on research and career and was a second advisor to me; Sem Borst, who was both a mentor and a friend during my stint at Bell Labs; Jim Dai, whose appreciation of my work gave me the self-confidence a young researcher needs and who has been an inspiration; Milan Vojnovic, from whom I learned much about problem solving and research during the summer at Microsoft Research. I have also learned a lot from my co-authors, colleagues and teachers whose words would have inevitably found their way into this thesis: Ana Bušić, Paul Enders, Peter Harrison, Michael Kozuch, Takayuki Osogami, Kavita Ramanan, Karl Sigman, Ward Whitt, Adam Wierman, Bert Zwart.

I would like to thank the staff of the Computer Science Department for looking after the graduate students. Sharon Burks, Deborah Cavlovich, Catherine Copetas and Sophie Park deserve special mention for making problems disappear at little or no notice.

A special note of thanks is also due to the operators of the Carnegie Mellon escort service who gave me a ride home after many late nights in the office.

I also wish to thank Prasad Chebolu, Deepak Garg, Vineet Goyal, Himanshu Jain, Vijay Krishnamurthy, Viswanath Nagarajan, Sandeep Pandey, and Mohit Singh who

made me feel as if I had never left home.

Finally, words fail me in expressing my gratitude to my family. It is said that you don't choose your family; they are God's gift to you. Even if it weren't the case, I would not have chosen any differently. My parents have always given me the freedom to follow my heart, even if it means sacrificing their happiness, and have had faith in me when I myself had none. They have also been a constant source of strength – just the thought of them is enough to guide me in moments of self-doubt.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Server farms are becoming an increasingly popular paradigm of computation since they use low-cost commodity hardware to provide computing power exceeding that of any single device. In addition, the server farm architecture allows the design of fault-tolerant and scalable systems – failures of a few servers don't bring down the entire server farm, and adding capacity is as easy as adding more servers. Server farms also lead to the design of energy-efficient computing systems by combining slower but more power-efficient processors to reduce the peak power consumption (for example, the IBM Blue Gene supercomputer [1] and the FAWN cluster architecture [14]). Another benefit of server farms is that by allowing the consolidation of multiple workloads, server farms lead to efficient resource utilization, such as in cloud computing centers.

Figure 1.1 shows the components of a simple server farm. New requests or jobs arrive at the front-end dispatcher, or load balancer. The load balancer routes the incoming requests to the back-end servers, which serve them. Requests leave the server farm once they complete processing at the back end servers.[1] Even in this simplified setup, there are two fundamental design questions:

Question 1: Load balancing: Which back-end server should process the incom-

[1]We present a subset of the request types, i.e., a job is dispatched to a single server and is processed at that one server until completion. There are certainly more complicated scenarios. For example, requests that *fork* into multiple smaller requests and are completed once all the sub-requests are complete (*join*), or map-reduce type tasks which involve multiple of these *fork-join* stages. In this thesis we focus on the simple subset.

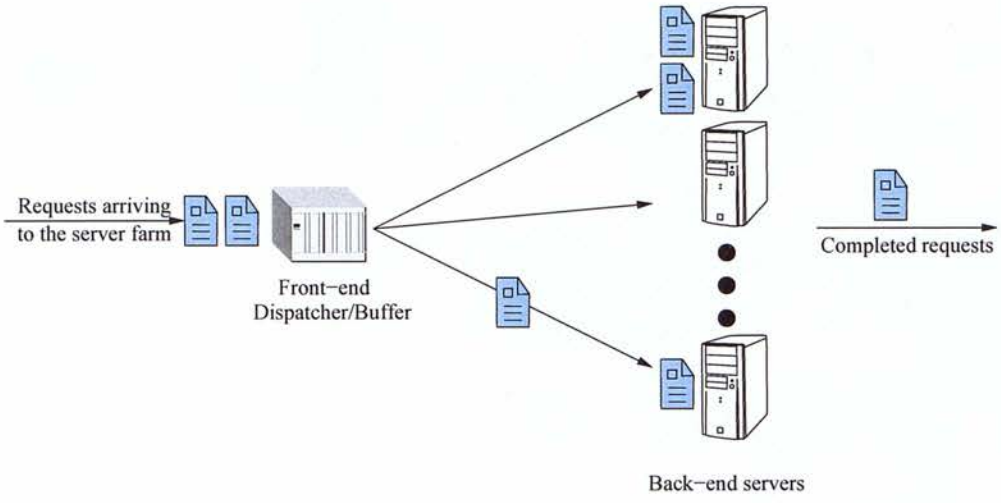


Figure 1.1: The server farm architecture.

ing request? Should the dispatch be immediate, or can the front-end dispatcher defer the decision? Should the dispatching policy be dynamic (depending on the current load of back-end servers), or static?

Question 2: Scheduling policy: How should the back-end servers schedule the tasks dispatched to them?

Today's server farms and data centers cater to a wide spectrum of workloads – from database queries, to computationally intensive jobs, to file streaming and web requests, to map-reduce tasks. Each of the aforementioned workloads imposes different constraints on which dispatching policies and scheduling policies are feasible. For example, a database query may only be dispatched to a server that stores the required data (locality) and hence dispatching policies are static, while a job only requiring processing may be dispatched to any server, allowing dynamic dispatching policies. Further, while a database query or a CPU intensive job may be queued for later processing, web and file download requests are latency-sensitive and must begin processing immediately to prevent dropped connections caused by time-outs. As another example, a supercomputing job runs alone on its back-end server in a non-preemptive fashion, whereas web and database requests typically timeshare their back-end server with other requests. It is clear that different applications compel different answers to the scheduling/dispatching questions raised above.

In addition to the problem of matching the dispatching and scheduling policies to the

workload and application, server farms are also required to satisfy multiple conflicting performance goals – low mean response times, efficient utilization of resources, flexibility to adapt to varying and unpredictable demands, performance isolation for high priority jobs, minimizing energy, to name a few. This adds at least two more problem dimensions:

Question 3: Dynamic Capacity Scaling/Server Management: When and which back-end servers should be turned off, or hibernate, to save energy? When should servers be turned on to increase capacity?

Question 4: Provisioning/Dimensioning: Given a cost budget, how should one design a server farm (in terms of number of servers and server speeds) to maximize performance?

In this thesis we will use stochastic (queueing theoretic) modeling and analysis techniques to guide these design decisions by modeling server farms as multi-server queues. Queueing theoretic modeling abstracts out the important features of the scheduling policies governing the performance of the system under consideration, and by imposing structured probabilistic assumptions on the sequence of request arrivals allows answering questions of the kind: What is the average response time of the requests? What fraction of requests experience response time larger than T_{max}? How sensitive are these performance metrics to parameter X? In addition to providing these answers for server farms of arbitrary size, stochastic analysis provides insights into the qualitative effect of various system parameters on the performance which may then be combined with other techniques, such as control theory and feedback systems, for operating the server farms.

Queueing theory started with the work of Erlang [55] (also see [138]) who was motivated by applications in telecommunications. Modern queueing theory has been shaped by applications to production systems, inventory management, and call centers [100]. The workloads and architectures of these application areas are very different from compute server farms and hence existing stochastic modeling and analysis results do not directly apply to problems faced by computer systems designers:

1. **Assumption of FCFS back-end servers:** Most analysis of dispatching policies for multi-server queues assumes that the servers follow a non-preemptive First-Come-First-Served (FCFS) scheduling policy. While this model fits problem domains which gave rise to the traditional multi-server models, such as telephone networks, call centers, hospital emergency rooms, queues in supermarkets etc., computer systems such as *web servers* are better modeled as

time-sharing systems. Designing and analyzing dispatching policies for time-sharing back-end servers is still an open question.

2. **Assumption of ideal time-sharing:** While there is a large body of work on analyzing a single time-sharing server, all existing analytical work models this time-sharing server as operating under an ideal Processor Sharing (PS) scheduling policy. Under PS, the server's capacity is independent of the number of concurrently running tasks. However time sharing systems, such as *database servers* and thread-based web servers exhibit *thrashing* which causes loss of server's capacity due to resource contention when too many tasks are concurrently active.

3. **Assumption of low job-size variability:** Even in application scenarios where traditional models of multi-server are a good fit, the existing analyses and approximations are severely lacking because they are derived under the assumption that the service requirements of the jobs have low variance. One example of such an application scenario is *supercomputing centers* where jobs are typically scheduled in a non-preemptive FCFS fashion and thus the traditional call center model fits. However, existing approximations which are reasonably accurate for the problem domain of call centers can be off by unacceptable margins when the variance in service requirements is significant, which is the case in supercomputing workloads.

4. **Lack of evaluation of energy-performance trade-offs:** Traditional models of server farms have not dealt with the metric of *energy*. This question becomes even more important because demands faced by today's server farms vary substantially over time. Provisioning for peak demand is extremely wasteful of energy and thus it is imperative to develop algorithms to power down servers when demand is low and turn them back on when demand increases. However, the existing models and analyses do not deal with the associated setup costs and delays of powering servers up and down in a server farm.

The goal of the thesis is to develop stochastic modeling and analysis techniques to bridge the disconnects between prior work on the analysis of multi-server systems and the problems faced in management of compute server farms today. Before summarizing the research questions addressed in the thesis, we introduce the notation used in the thesis. In some chapters we will need additional notation or will deviate from the notation mentioned below, and hence notation will be reintroduced within the chapters as well.

1.2 Notation and Preliminaries

The arrival and service processes constitute the most important aspects of a stochastic model of a queueing system. The arrival process specifies the instants at which jobs arrive into the system (server farm), and the service process specifies the size or processing requirements of the jobs. Unless otherwise stated, we assume that the arrival process is a renewal process, by which we mean that the times between consecutive arrivals are independent and identically distributed (*i.i.d.*) random variables. We use A to denote a generic interarrival time. We denote the mean of A by $\mathbf{E}[A] = \frac{1}{\lambda}$. Thus λ denotes the mean arrival rate. We also assume that job sizes form a sequence of independent and identically distributed (*i.i.d.*) random variables, where the random variable S (for service time distribution) denotes a generic job size. We will be dealing with models where the server speed may be heterogeneous or state-dependent. Consequently, we will use job sizes to denote the amount of work (for example the number of cycles for a CPU job, or file size for an I/O job). We denote the mean of S by $\mathbf{E}[S]$, and assume $\mathbf{E}[S] = 1$ without loss of generality. In empirical computing workloads it is often the case that A and S exhibit high variability. One of the most widely used metrics for characterizing this variability is the *the squared coefficient of variation* (SCV) of the interarrival and service distributions, C_A^2 and C_S^2, respectively:

$$C_A^2 = \frac{var(A)}{\mathbf{E}[A]^2}; \quad C_S^2 = \frac{var(S)}{\mathbf{E}[S]^2},$$

where $var(X)$ denotes the variance of random variable X. For a substantial part of the thesis we will assume that arrival process is Poisson, that is, A obeys the exponential distribution with mean $1/\lambda$, abbreviated as $A \sim \text{Exp}(\lambda)$, and will denote the arrival process by Poisson(λ). We will use the symbol μ to denote the service rate, or capacity of each server. When we talk about homogeneous server farms, where the capacity of each server is the same, we will use $\rho = \lambda\mathbf{E}[S]/\mu$ to denote the 'load' which represents the amount of work coming into the system per unit of time. We use T to denote the random variable for the response time, defined as the time between a job's arrival to the system and its departure from the system. We use W to denote the random variable for waiting time, defined as the total time spent in the system waiting to receive service.

Since the queueing models considered in the thesis are very different from the classical models, we will extend Kendall's notation to abbreviate them. In Kendall's notation, a queueing system shorthanded as $A/B/C/D$ denotes a system with arrival process A, service time distribution B, number of servers C, and scheduling

policy D.[2] Typical values for the arrival process A that we will use are: M (for Markovian) for a Poisson arrival process, M_t for a doubly stochastic Poisson process (that is, the mean arrival rate at time t is given by some function $\lambda(t)$), BPP for a Batch Poisson Process (Poisson process with a random number of arrivals at a time), and GI for general *i.i.d.* interarrival times. Typical values for the service time distribution B we will use are: M for exponentially distributed, D for degenerate (non-random), H_2 for 2-phase hyperexponential (a mixture of two exponential distributions with different means), H_2^* for degenerate hyperexponential (mixture of an exponential distribution and a point mass at 0), and G for generally distributed *i.i.d.* service times. Typical values for the scheduling policy are FCFS for First-Come-First-Served, and PS for Processor Sharing (if there are n jobs queued at the server, each job gets $\frac{1}{n}$th of the server's capacity). However, we will sometimes combine the dispatching and scheduling policies in D. For example, we will use $M/G/k$/JSQ/PS to denote the multi-server system with Poisson arrivals, general service distribution, and k servers where each server follows the processor sharing (PS) scheduling policy and new requests join the shortest queue (JSQ) immediately on arrival.

1.3 Summary of research questions

We now give a formal summary of the disconnects between existing analytical stochastic work on multi-server systems and the needs of practitioners that the thesis aims to bridge. The goal of the thesis is not to bridge all the disconnects at once, but to analyze each individually in depth to show how and where traditional policies and analysis fail. Each disconnect guides the answer to one or more of the four design decisions presented in Section 1.1. We motivate each disconnect with a computing application, and develop frameworks for optimizing and analyzing the performance under the unique constraints/opportunities presented by the motivating application. Table 1.1 provides a brief summary of the various pieces of the thesis.

1: High job-size variance of computer systems workloads invalidates existing approximations:
We begin with a scenario where the difference in workloads encountered in computing applications and traditional applications of queueing models compel us to develop new techniques to analyze queueing systems, because existing analytical approximations are insufficient. We illustrate this point by considering the $M/G/k$ First-Come-First-Served (FCFS) multi-server system. The $M/G/k$ FCFS system has traditionally been used as a model of systems such

[2]Note the absence of dispatching policy in Kendall's notation.

	Disconnect	Motivating Application	Design Decisions Influenced	Contributions	Chpt.
1.	High job-size variability	Supercomputing	Provisioning	new analysis	2, 3
2.	Thrashing/Resource Contention	Database concurrency control, thread-pool management	Scheduling	new model + analysis + algorithm	4
3.	Server farms with time-sharing servers	Web server farms	Dispatching	new analysis	5
4.	Energy-performance trade-offs	Cloud computing, data centers	Capacity Scaling/ Server Management	new model + algorithm	6
5.	Time-varying demands	DB servers, Cloud computing, data centers	Capacity Scaling/ Server Management	new algorithms	4.3, 5.6.3, 6.6

Table 1.1: A tabular summary of the questions addressed in the thesis.

as call centers, manufacturing, hospital emergency rooms, and supercomputing centers. The $M/G/k$ FCFS system is notoriously hard to analyze, and despite being one of the oldest multi-server models to be studied, expressions for even the mean response time are not available for general service distributions. In the absence of such results, the following approximation proposed by Lee and Longton [108] which only involves the first two moments of the service distribution (S) is widely used:

$$\mathbf{E}\left[W^{M/G/k/FCFS}\right] \approx \frac{C_S^2 + 1}{2}\mathbf{E}\left[W^{M/M/k/FCFS}\right]$$

where $W^{M/M/k/FCFS}$ denotes the delay in an $M/M/k$ FCFS system with the same mean job size, arrival rate and service rate as the $M/G/k$ FCFS system ($W^{M/M/k}$ has an exact and explicit expression).

We challenge the status quo in Chapter 2 by proving an **inapproximability result**: any approximation based only on the first two moments of the service distribution S must be inaccurate for some service distribution when C_S^2 is large. This is significant because many computer systems workloads such as supercomputing jobs and sizes of files transferred over the Internet exhibit $C_S^2 > 40$ (e.g., [20]), unlike call centers and manufacturing systems where C_S^2 is small (e.g., [142]).

Motivated by this result, in Chapter 3, we pursue approximations for $W^{M/G/k}$ utilizing higher moments of S. In fact, our goal is more ambitious: Given the

moments of S, we want to identify service distributions that maximize/minimize the mean waiting time. Thus our goal is to find **sharp bounds** on $\mathbf{E}\left[W^{M/G/k}\right]$ given the moments of S. By analyzing the $M/G/k$ system in the limit where the arrival rate approaches 0, we identify a link with the classical areas of moment problem and Tchebycheff systems, and are able to show that these extremal distributions are what are known as the principal representations. In fact we go further: we find that the same service distributions are extremal for two seemingly very different, and as yet unsolved, queueing systems.

Next we turn to scenarios where new architectures of computing applications force us to develop new stochastic models and new analysis tools.

2: Effects of thrashing are ignored while analyzing PS-like systems: We consider the problem of concurrency control in database servers, and managing the thread pool in web servers. Processor sharing (PS) is an idealized scheduling policy commonly used to model time-sharing systems such as the CPU, bandwidth sharing systems, web and database servers. Under PS, a server shares its capacity equally among all the jobs in its queue. However almost all analytical results on the analysis of PS ignore the effects of *thrashing* [78]. Thrashing, or resource contention, causes the net capacity of a resource to decrease as the number of jobs concurrently sharing the resource increases, and, in the absence of any concurrency control mechanism, can bring the system to a halt. To get around this problem a Multi-Programming-Limit (MPL) is placed on the maximum number of jobs allowed to share the resource simultaneously and is almost always chosen to be the point of maximum efficiency (capacity). Existing work on analysis of PS with an MPL either ignores the effect of variability of service distribution (C_S^2), or assumes that μ (service rate/capacity) is independent of the state (number of jobs) of the system.

In Chapter 4, we present an approximate analysis of PS-MPL queueing systems to find the optimal MPL for minimizing the mean response time as a function of C_S^2, λ and the μ-vs.-MPL function. The optimal MPL depends crucially on the arrival rate λ, which may not always be known at the time of system design, or may fluctuate at small time scales. As a second contribution, we develop **traffic-oblivious dynamic MPL control policies**, which adapt the MPL based on the instantaneous queue length, rather than by attempting to learn the instantaneous arrival rate. Finally, we propose the first **heavy-traffic scaling** for analysis of 'non-work-conserving' time-sharing systems (i.e., systems where, depending on the current state, the service rate can be smaller than the peak

service rate) and present a preliminary heavy-traffic approximation for the stationary distribution of number of jobs in $GI/G/$PS-MPL systems.

3: No analysis of load balancing policies for PS server farms:
The next problem we address focuses on developing smart load balancing policies for server farms. Motivated by supercomputing applications, there is a large body of work on analyzing dispatching policies for server farms where the servers operate under FCFS scheduling where the relative performance of different load balancing policies is well understood. However, in many application scenarios, such as web servers and file downloads, the scheduling policy employed by servers is PS (in other words, preemptive scheduling policies are feasible). Unfortunately, policies which perform well for FCFS server farms may not perform well for PS server farms.

In Chapter 5, we show via simulations that Least-Work-Left and Size-Based dispatching, which perform well under FCFS scheduling, are far from optimal under PS scheduling. By contrast Join-the-Shortest-Queue is **near optimal**, while being oblivious to the job sizes or the service distribution. We also find that JSQ/PS systems exhibit a **near-insensitivity** property: moments of S larger than the mean have minimal impact on the mean response time. Armed with the above observation, we propose **sharper approximations** for JSQ/PS systems via a novel Single-Queue-Approximation technique. Finally, we propose a careful **many-servers heavy-traffic scaling**. We utilize our scaling to present another closed-form approximation for the JSQ load balancing policy that provides new insights into the behavior of JSQ, and also allows us to study the **impact of heterogeneity** in server capacities on the performance of JSQ-type dispatching policies.

4: Limited understanding of energy-performance trade-offs:
Energy consumption has recently emerged as a key metric for the evaluation of scheduling policies and server management policies. Naturally there are trade-offs involved between minimizing the energy consumed and guaranteeing low response times. While one wants to turn off idle servers, or put them into some sleep state, the penalties to boot up the servers may be prohibitive.

In Chapter 6, we consider the metric of the product of the mean response time and mean power consumed (Energy-Response time-Product, ERP) to capture the trade-offs involved in minimizing energy consumptions and maximizing performance and analyze server management policies with respect to the ERP

metric. We prove that optimal or near-optimal policies can be found within a substantially small set of policies, and provide rules of thumb to choose the right policy from among this set. Finally we propose two heuristic policies for energy management when the demand is non-stationary.

5: Time-varying arrival patterns:
Most of the work on stochastic analysis of multi-server queues has focused on Poisson or renewal arrival processes. This implies that the mean traffic demand remains constant over time. This assumption is violated in the real world, as the arrival patterns at web server farms and data centers, for example, show strong diurnal and seasonal variations. Designing robust server management policies which may self-adapt the capacity of the server farm to unpredictable arrival patterns is one of the holy grails of capacity provisioning. Unfortunately, systems with time-varying arrival patterns are not well understood analytically. In previous work [69], we proved that the answer to the question, *'Is a system with time-varying arrival pattern worse than a system with constant mean arrival rate,'* is not always yes. While there is existing analytical work on designing server management policies for time-varying arrival patterns, the proposed policies either involve repeated static provisioning, or assume that the arrival pattern is known beforehand. Additionally, it is often assumed that server farm capacity may be increased instantaneously – an assumption that is not always justified.

For each of the three application scenarios discussed above (PS servers with thrashing; load balancing; energy management), we have striven to present **traffic-oblivious policies** in addition to optimal/near-optimal policies for a stationary arrival process. In Section 4.3, we present two traffic-oblivious heuristics for the problem of concurrency (MPL) control in time sharing systems. In Section 5.6.3, we prove that in the presence of heterogeneous servers, JSQ minimizes the mean response time while being traffic-oblivious in the many-server regime, which is a good approximation for today's large server farms. We also propose another policy HYBRID, that is also optimal in many-servers limit when the traffic intensity is very close to capacity, and provides favorable performance when the number of servers is smaller. Finally, in Section 6.6, we design two traffic-oblivious server management policies with the goal of optimizing energy-performance tradeoff and show that they can be extended to application scenarios beyond the simple abstract model of Chapter 6.

Organization

Broadly, each disconnect corresponds to a chapter, and the aim while writing has been to make the chapters self-contained. The reader is encouraged to familiarize oneself with the ideas from Chapter 2 before reading of Chapter 3, although this is not necessary for understanding the analysis.

Chapter 2

Towards a New Theory of Moments-based Bounds I: An Inapproximability Result for the $M/G/k$ Multi-server Queue

> "But it is also worth asking whether anything can be done about some of the simple unsolved problems which have, perhaps wisely, been left to one side of the mainstream of research. For example, $M/G/1$ was solved by Pollaczek, and $M/D/k$ for general k by Erlang, but what about $M/G/k$? This is surely an important system, with the Poisson arrivals that are still the most useful input process, and independent service times having a given but arbitrary distribution."
>
> - J.F.C. Kingman [100]

We begin this thesis with a problem that is at the heart of the century old field of queueing theory – analyzing the mean waiting time in an $M/G/k$ multi-server queue (Figure 2.1). The $M/G/k$ queue is widely used as a model for call centers and telecommunication systems, inventory management systems, hospital emergency rooms, and supercomputing systems, and new applications of this model are continually being discovered. The wide applicability of the $M/G/k$ system makes it a prime target for developing and testing new stochastic analysis techniques, which provide insights into queueing systems far beyond the $M/G/k$ queue itself.

While exact expressions for the mean waiting time of the $M/G/k$ queue are only available for very special cases, numerous bounds and expressions have been proposed

in the literature. These bounds and approximations are usually functions of the first or first two moments of the service distribution. The present chapter is devoted to proving the insufficiency of existing approaches to approximate the mean waiting time of the $M/G/k$ queue by proving that no single approximation based on only the first two moments can be accurate for all service distributions. In Chapter 3, we build on the lessons from this chapter to propose a new framework for establishing tight bounds on the mean sojourn time of queueing systems by utilizing higher moments.

2.1 Introduction

The $M/G/k$ queue is one of the oldest and most classical models of a multi-server system and has been used as a model in a wide range of applications, including call centers, manufacturing systems, and computer systems.

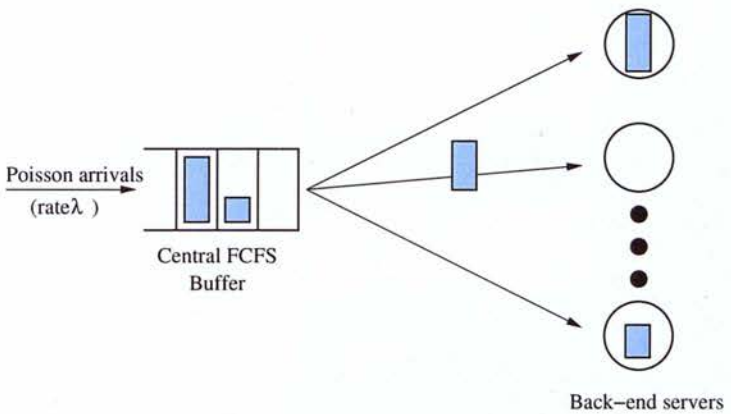


Figure 2.1: The $M/G/k$ queueing system.

An $M/G/k$ queue consists of k identical servers and a First-Come-First-Serve (FCFS) queue (see Figure 2.1). The jobs (or customers) arrive according to a Poisson process with rate λ and their service requirements (job sizes) are assumed to be independent and identically distributed random variables having a general distribution; we use S to denote such a generic random variable. If an arriving job finds a free server, it immediately enters service, otherwise it waits in the FCFS queue. When a server becomes free, it chooses the next job to process from the head of the FCFS queue. We denote the load of this $M/G/k$ system as $\rho = \lambda \mathbf{E}[S]$, and assume $\rho < k$ so that a steady-state distribution exists [92, 93]. We will focus on the metric of mean waiting

	$C_S^2 = 19$	$C_S^2 = 99$
	$\mathbf{E}[W]$	$\mathbf{E}[W]$
2-moment approximation (Eqn. 2.1)	6.6873	33.4366
Weibull	6.0691±0.0138	25.9896±0.1773
Truncated Pareto ($\alpha = 1.1$)	5.5277±0.0216	24.6049±0.2837
Lognormal	4.9937±0.0249	19.5430±0.4203
Truncated Pareto ($\alpha = 1.3$)	4.8788±0.0249	18.7738±0.3612
Truncated Pareto ($\alpha = 1.5$)	3.9466±0.0321	10.6487±0.5373

Table 2.1: Simulation results for the mean waiting time for an $M/G/k$ with $k = 10$ and $\rho = 9$. The first line shows the mean waiting time given by the analytical 2-moment approximation in Equation (2.1). All service distributions throughout the chapter have $\mathbf{E}[S] = 1$.

time in this chapter, denoted as $\mathbf{E}\left[W^{M/G/k}\right]$, and defined to be the expected time from the arrival of a customer to the time it enters service.

Even though the $M/G/k$ queue has received a lot of attention in the queueing literature, an exact analysis for even the simplest metric of mean waiting time for the case $k \geq 2$ still eludes researchers. To the best of our knowledge, the first approximation for the mean waiting time for an $M/G/k$ queue was given by Lee and Longton [108] nearly half a century ago:

$$\mathbf{E}\left[W^{M/G/k}\right] \approx \left(\frac{C_S^2 + 1}{2}\right) \mathbf{E}\left[W^{M/M/k}\right] \tag{2.1}$$

where $\mathbf{E}\left[W^{M/M/k}\right]$ is the mean waiting time with exponentially distributed job sizes with the same mean, $\mathbf{E}[S]$, as in the $M/G/k$ system, and C_S^2 is the squared coefficient of variation (SCV) of S. Many other authors have also proposed simple approximations for the mean waiting time, [34, 79, 80, 106, 120, 159], but all these closed-form approximations involve only the first two moments of the service distribution.

Whitt [153], while referring to (2.1) as "usually an excellent approximation, even given extra information about the service-time distribution," hints that approximations based on the first two moments of S may be inaccurate when C_S^2 is large. Similar suggestions have been made by many authors, but there are very limited numerical experiments to support this. While a high C_S^2 may not be of major concern in applications such as manufacturing or customer contact centers, the invalidity of the approximation (2.1) *is* a major problem in computer and communication systems. In Table 2.1, we consider a range of distributions (Weibull, lognormal, truncated

Pareto[1]) used in the literature to model computer systems workloads. We compare the mean waiting time obtained via simulations to the mean waiting time predicted by the approximation (2.1) for two values of C_S^2, $C_S^2 = 19$ and $C_S^2 = 99$. Such high values of C_S^2 are typical for workloads encountered in computer systems, such as the sizes of files transferred over the Internet [20], and the CPU requests of UNIX jobs [49] and supercomputing jobs [74]. As can be seen, there is a huge disagreement between the simulated mean waiting time and the 2-moment approximation (2.1). Further, the simulated mean waiting times are consistently smaller than the analytical approximation. Also observe that different distributions with the same mean and C_S^2 yield very different mean waiting times.

Goal

The goals of this chapter are two-fold:

1. Investigate the (in)sufficiency of first two moments of the service distribution for approximating $\mathbf{E}\left[W^{M/G/k}\right]$,
2. Investigate how characteristics of the service distribution other than the first two moments affect $\mathbf{E}\left[W^{M/G/k}\right]$.

Ideally, to address our first goal, we should consider the set $\{G|C_S^2\}$ of *all* positive distributions with a given mean and second moment. Each distribution in this set when chosen as the service distribution for the $M/G/k$ queue yields a value for the mean waiting time. We want to establish the set of values (an interval of $\mathbb{R}_0^+$) that are attained as the mean waiting time. We refer to this interval as "the span". To define the span, set

$$W_h^{C_S^2} = \sup\left\{\mathbf{E}\left[W^{M/G/k}\right] \;\middle|\; \mathbf{E}[S] = 1, \mathbf{E}\left[S^2\right] = C_S^2 + 1\right\}, \tag{2.2}$$

and

$$W_l^{C_S^2} = \inf\left\{\mathbf{E}\left[W^{M/G/k}\right] \;\middle|\; \mathbf{E}[S] = 1, \mathbf{E}\left[S^2\right] = C_S^2 + 1\right\}. \tag{2.3}$$

[1]The cumulative distribution function of a truncated Pareto distribution with support $[x_{min}, x_{max}]$ and parameter α is given by:

$$F(x) = \frac{x_{min}^{-\alpha} - x^{-\alpha}}{x_{min}^{-\alpha} - x_{max}^{-\alpha}} \qquad x_{min} \leq x \leq x_{max}$$

Therefore, specifying the first two moments and the α parameter uniquely defines a truncated Pareto distribution.

The span ranges $(W_l^{C_S^2}, W_h^{C_S^2})$. One of the contributions of this chapter is a lower bound on the span for the case $\rho < k-1$ in Theorem 2.1, and for the case $\rho > k-1$ in Theorem 2.2. We believe that the bounds presented in Theorem 2.1 for the case $\rho < k-1$ are tight, and conjecture tight bounds for the case $\rho > k-1$ in Conjecture 2.1 (see Section 2.3).

Theorem 2.1 *For any* $\mathbf{E}[S] = 1$ *finite* C_S^2 *and* $\rho < k-1$,

$$W_h^{C_S^2} \geq (C_S^2 + 1)\mathbf{E}\left[W^{M/D/k}\right]$$
$$W_l^{C_S^2} \leq \mathbf{E}\left[W^{M/D/k}\right]$$

and thus,

$$\frac{W_h^{C_S^2}}{W_l^{C_S^2}} \geq C_S^2 + 1$$

where $\mathbf{E}\left[W^{M/D/k}\right]$ *is the mean waiting time when the service distribution is deterministic* 1.

Theorem 2.2 *For* $\mathbf{E}[S] = 1$ *any finite* C_S^2 *and* $\rho > k-1$,

$$W_h^{C_S^2} \geq \left(\frac{C_S^2 + 1}{2}\right)\mathbf{E}\left[W^{M/M/k}\right]$$
$$W_l^{C_S^2} \leq \mathbf{E}\left[W^{M/M/k}\right] + \left\lceil\frac{\rho - (k-1)}{k-\rho}\right\rceil\frac{C_S^2 - 1}{2}$$

and thus,

$$\frac{W_h^{C_S^2}}{W_l^{C_S^2}} \geq \frac{\left(\frac{C_S^2+1}{2}\right)\mathbf{E}\left[W^{M/M/k}\right]}{\mathbf{E}\left[W^{M/M/k}\right] + \left\lceil\frac{\rho-(k-1)}{k-\rho}\right\rceil\frac{C_S^2-1}{2}}$$

where $\mathbf{E}\left[W^{M/M/k}\right]$ *is the mean waiting time when the service distribution is exponential with mean* 1.

Theorem 2.1 will be proved in Section 2.4 and follows by combining a result of Daley [44] with some new observations. Theorem 2.2 is far more intricate to prove, and forms the bulk of the chapter (Section 2.5).

We now make a few important observations on the span:

- Since we prove a lower bound for $W_h^{C_S^2}$ and an upper bound for $W_l^{C_S^2}$, Theorems 2.1 and 2.2 provide a lower bound on the span for general distributions.
- The span can be quite large if C_S^2 is high. In particular, when $\rho < k-1$, Theorem 2.1 states that the maximum possible mean waiting time is at least (C_S^2+1) times the minimum possible mean waiting time. Thus, Theorems 2.1 and 2.2 prove that *any* approximation based only on the first two moments of S will be inaccurate for some service distribution.
- The lower bound on $W_h^{C_S^2}$ in Theorem 2.2 is the same as the 2-moment approximation in (2.1). (The lower bound on $W_h^{C_S^2}$ in Theorem 2.1 is very close but slightly higher than the 2-moment approximation.)

Another interesting point is that the lower bound on the span depends on the load, ρ. The case $\rho \geq k-1$ is commonly known in the queueing literature as *0-spare servers* and the case $\rho < k-1$ is known as *at least 1 spare server*. The presence of spare servers is known to play a crucial role in determining whether the mean waiting time is infinite given that the second moment of the service distribution is infinite (see [133] and references therein), and on the tail of the waiting time distribution (see [60]). Observe that the number of spare servers (zero or at least one) affects whether C_S^2 shows up in the lower bound of the span in our results. When there is even just one spare server, the lower bound is independent of C_S^2, which suggests that having even one spare server might potentially reduce most of the effect of C_S^2 on the mean waiting time.

The **key insight** in proving Theorem 2.1 ($\rho < k-1$) is to consider two extreme two-point service distributions and find the mean waiting time under these extremal distributions. To prove Theorem 2.2 ($\rho \geq k-1$), we consider two extreme distributions in the class of 2-phase hyperexponential distributions and obtain the mean waiting time under those service distributions. We believe that it is not hard to tighten the bound in Theorem 2.2 by extending our proof technique to work with two-point distributions (mixtures of two mass points), and thus establish a wider "span" than we do in this chapter. However, presently, we focus on 2-phase hyperexponential distributions for ease of exposition and to elucidate the basic steps in obtaining the bound. Clearly the span described by Theorem 2.1 is non-empty for all $C_S^2 > 0$. The span described by Theorem 2.2 is non-empty only when $C_S^2 > 1$ even though the theorem is true for all values of C_S^2. In fact, Proposition 2.1 (Appendix 2.A) shows that our lower bound on the span is strictly non-empty when $k \geq 2$ and $C_S^2 > 1$.

The bounds on $W_h^{C_S^2}$ and $W_l^{C_S^2}$ in Theorem 2.2 are identical for $k = 1$, and in fact in this case agree with the well-known Pollaczek Khintchine formula

$$\mathbf{E}\left[W^{M/G/1}\right] = \left(\frac{C_S^2 + 1}{2}\right) \mathbf{E}\left[W^{M/M/1}\right], \tag{2.4}$$

which shows that the mean waiting time is completely determined by C_S^2 and $\mathbf{E}[S]$.

Similar results on stationary waiting time and queue length distributions in a $GI/M/k$ queue were derived by Eckberg [50] and further developed by Whitt [152] by considering extremal interarrival time distributions. For the $GI/M/k$ queue, proving such theorems is simplified due to the availability of rather explicit expressions for the queue length and waiting time distributions in terms of the Laplace transform of the inter-arrival time distribution.

Outline

Section 2.2 reviews existing work on obtaining closed-form, numerical and heavy-traffic approximations for $\mathbf{E}\left[W^{M/G/k}\right]$. In Section 2.3 we seek insights into why the first two moments of the service distribution are insufficient for approximating the mean delay. We also seek answer to the question: "Which characteristics of the service distribution, outside of the first two moments, are important in determining the mean waiting time?" Our insights stem from numerical experiments based on the 2-phase hyperexponential class of service distributions. These insights help us later in proving Theorem 2.2. Sections 2.4 and 2.5 are devoted to proving Theorems 2.1 and 2.2, respectively. In Section 2.6, we address the question of the effect of higher moments of service distribution on the mean waiting time.

2.2 Prior Work

While there is a large body of work on approximating the mean waiting time of an $M/G/k$ system, all the closed-form approximations only involve at most the first two moments of the service distribution. As mentioned earlier, to the best of our knowledge, the first approximation for the mean waiting time for an $M/G/k$ queue was given in (2.1) by Lee and Longton [108]. This approximation is very simple, is exact for $k = 1$ and was shown to be asymptotically exact in heavy traffic by Köllerström [106]. The same expression is obtained by Nozaki and Ross [120] by making approximating assumptions about the $M/G/k$ system and solving for exact state probabilities of the approximating system, and by Hokstad [79] by

starting with the exact equations and making approximations in the solution phase. Boxma et al. [34] obtain a closed-form approximation for the mean waiting time in an $M/D/k$ system, extending the heavy traffic approximation of Cosmetatos [40]. Takahashi [144] obtains expressions for mean waiting time by assuming a parametric formula. Kimura [95] uses the method of system interpolation to derive a closed-form approximation for the mean waiting time that combines analytical solutions of simpler systems.

There is also a large literature on numerical methods for approximating the mean waiting time by making much weaker assumptions and solving for state probabilities. For example, Tijms et al. [76] assume that if a departure from the system leaves behind i jobs where $1 \leq i < k$, then the time until the next departure is distributed as the minimum of i independent random variables, each of which is distributed according to the equilibrium distribution of S. If, however, the departure leaves behind $i \geq k$ jobs, then the time until the next departure is distributed as S/k. Similar approaches are followed in [79, 80, 113, 115, 140]. Miyazawa [115] uses "basic equations" to provide a unified view of approximating assumptions made in [120], [79] and [76], and to derive new approximation formulas. Boxma et al. [34] also provide a numerical approximation for $M/G/k$ which is reasonably accurate for service distributions with low variability ($C_S^2 \leq 1$) by assuming a parametric form and matching the heavy traffic and light traffic behaviors. Other numerical algorithms include [45, 46, 47]. While these numerical methods are accurate and usually give an approximation for the entire waiting time distribution, the final expressions do not give any structural insight into the behavior of the queueing system and the effect of $M/G/k$ parameters on waiting time.

Heavy traffic, light traffic and diffusion approximations for the $M/G/k$ system have been studied in [37, 71, 94, 106, 153, 154, 159]. The diffusion approximations used in [154] are based on many-server diffusion limits. Motivated by call center applications, there is now a huge body of literature for multiserver systems with a large number of exponential servers; see the survey paper [62] and references therein.

Bounds on the mean waiting time for $M/G/k$ queues (and more generally, for $GI/G/k$ queues) have mainly been obtained via two approaches (e.g., see Section 11-7 from Wolff [158]). The first approach is by assuming various orderings (stochastic ordering, increasing convex ordering) on the service distributions (see [43, 116, 141, 150, 151]), but these tend to be very loose as approximations. Moreover, one does not always have the required strong orderings on the service distribution. The second, and more practical, approach that started with the work of Kingman [99] is obtaining bounds on mean waiting time in terms of the first two moments of the inter-arrival and service distributions. The best known bounds of this type for $\mathbf{E}\left[W^{GI/G/k}\right]$ are presented by Daley [44]. Scheller-Wolf and Sigman

[132] derive bounds on for the case $\rho < \left\lfloor \frac{k}{2} \right\rfloor$ by reducing the $GI/G/k$ waiting time recursion into an equivalent single-server recursion with dependent service times. Foss and Korshunov [60] and Scheller-Wolf and Vesilo [133] use dependent $D/GI/1$ queues to bound a $GI/G/k$ system, and obtain necessary and sufficient conditions under which higher (even fractional) moments of delay are finite.

Daley [44] also conjectures tight upper and lower bounds on $GI/GI/k$ mean waiting time in terms of the first two moments of interarrival and service distributions, and proves a tight lower bound

$$\inf \mathbf{E}\left[W^{GI/GI/k}\right] = 0, \quad \text{when } \rho < k - 1.$$

While bounds for $GI/GI/k$ mean waiting time are more general, they can also be loose when applied to $M/G/k$.

Recently, Bertsimas and Natarajan [24] have proposed a computational approach based on semidefinite optimization to obtain bounds on the moments of waiting time in $GI/GI/k$ queues given the information of moments of the job size and the interarrival time distributions. We, however, prove $\mathbf{E}\left[W^{M/G/k}\right]$ is inapproximable within a certain factor based on just the knowledge of the first two moments of the service distribution.

2.3 Insights into why two-moment approximations are not enough

Our goal in this section is to illustrate the inadequacy of the first two moments of the service distribution for approximating $\mathbf{E}\left[W^{M/G/k}\right]$. As we said earlier, to achieve this goal, ideally we would have to look at the set of all distributions with given first two moments and establish the span of mean waiting time. As expected, this turns out not to be feasible. However, to illustrate inadequacy of first two moment, it suffices to establish this fact for an analytically tractable subset of distributions. We choose this tractable subset as the class of two-phase hyperexponential distributions, denoted by H_2 (see Definition 2.1 below). Distributions in the H_2 class are mixtures of two exponential distributions and thus have three degrees of freedom. Having three degrees of freedom provides us a method to create a set of distributions with any given first two moments ($C_S^2 > 1$ in the case of H_2) and analyze the effect of some other characteristic. A natural choice for this third characteristic is the *third moment*

of the distribution[2]. The H_2 distribution is also convenient because it allows us to capture the effect of *small vs. large jobs* (the two phases of the hyperexponential) – an insight which will be very useful to us later in proving our theorems.

Definition 2.1 *Let $\mu_1 > \mu_2 \ldots > \mu_n > 0$. Let $p_i > 0$, $i = 1, \ldots, n$, be such that $\sum_{i=1}^{n} p_i = 1$. We define the $n-$phase hyperexponential distribution, H_n, with parameters μ_i, p_i, $i = 1, \ldots, n$, as:*

$$H_n \sim \begin{cases} \text{Exp}(\mu_1) & \textit{with probability } p_1 \\ \text{Exp}(\mu_2) & \textit{with probability } p_2 \\ \vdots & \\ \text{Exp}(\mu_n) & \textit{with probability } p_n \end{cases}$$

where $\text{Exp}(\mu_i)$, $i = 1, \ldots, n$, *are n independent exponential random variables with mean* $\frac{1}{\mu_i}$, $i = 1, \ldots, n$.

Definition 2.2 *Let $\mu_1 > \mu_2 \ldots > \mu_{n-1} > 0$. Let $p_i > 0$, $i = 0, \ldots, n-1$, be such that $\sum_{i=0}^{n-1} p_i = 1$. We define the $n-$phase degenerate hyperexponential distribution, H_n^*, with parameters p_0, μ_i, p_i, $i = 1, \ldots, n-1$, as:*

$$H_n^* \sim \begin{cases} 0 & \textit{with probability } p_0 \\ \text{Exp}(\mu_1) & \textit{with probability } p_1 \\ \vdots & \\ \text{Exp}(\mu_{n-1}) & \textit{with probability } p_{n-1} \end{cases}$$

where $\text{Exp}(\mu_i)$, $i = 1, \ldots, n-1$, *are $n-1$ independent exponential random variables with mean* $\frac{1}{\mu_i}$, $i = 1, \ldots, n-1$.

Figure 2.2 shows the mean waiting time for an $M/H_2/k$ system evaluated numerically using matrix analytic methods. The dashed line shows the standard two moment approximation of (2.1). Note that the $x-$axis is actually not showing $\mathbf{E}[S^3]$ but rather a normalized version of the third moment, θ_3, which we define as:

$$\theta_3 = \frac{\mathbf{E}[S^3]\mathbf{E}[S]}{\mathbf{E}[S^2]^2}. \tag{2.5}$$

[2]In [45, 153], the authors use the quantity r, which denotes the fraction of load contributed by the branch with the smaller mean, as the third parameter to specify the H_2 distribution. We choose the third moment because it is more universal and better understood than r. Further, r is an increasing function of the third moment and thus one can go back and forth between the two parametrizations.

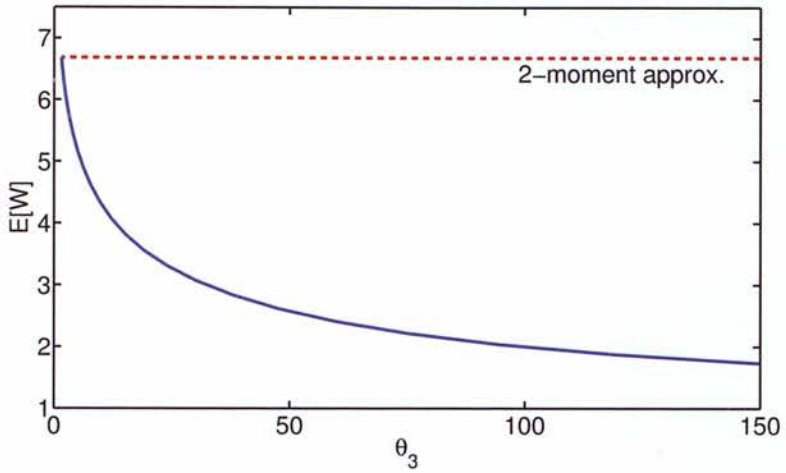


Figure 2.2: Illustration of the inadequacy of two-moment approximations for mean delay $\mathbf{E}\left[W^{M/G/k}\right]$. As shown, the normalized 3rd moment, θ_3, of the service distribution has a big effect on mean waiting time of an $M/H_2/10$ system (solid line). The parameters of the service distribution were held constant at $\mathbf{E}[S] = 1$ and $C_S^2 = 19$ with load $\rho = 9$. The dashed line shows the standard two-moment approximation of (2.1). The values on the $x-$axis are the normalized third moment (2.5).

The above normalization for the third moment with respect to the first two moments is analogous to the definition of the squared coefficient of variation, $C_S^2 = \frac{\mathbf{E}\left[S^2\right]}{\mathbf{E}[S]^2} - 1$, which is the scale-invariant normalization of the second moment with respect to the first moment. For positive distributions, θ_3 takes values in the range $[1, \infty)$, and our ongoing work on approximations for $\mathbf{E}\left[W^{M/G/k}\right]$ based on higher moments of service distribution suggests that θ_3 is the right variable to look at. We will use the normalized third moment, θ_3, throughout the chapter.

Our first interesting observation is that the $M/H_2/k$ mean waiting time actually *decreases as the third moment of S increases.* We also observe that the existing two moment approximation is insufficient as it sits at one end of the spectrum of possible values for $\mathbf{E}\left[W^{M/H_2/k}\right]$. For lower values of the third moment the approximation is good, but it is very inaccurate for high values. Moreover, *any* approximation based only on the first two moments will be inaccurate for some distribution because the span of possible values of mean waiting time for the same first two moments of the service distribution is large.

While the drop in mean waiting time with increasing θ_3 seems very counter-intuitive, this phenomenon can partially be explained by looking at how increasing θ_3 alters the distribution of load among the small and large jobs. Let $\rho(x)$ represent the fraction of load made up by jobs of size smaller than x. If $f(x)$ represents the probability density function of the service distribution, then,

$$\rho(x) = \frac{1}{\mathbf{E}[S]} \int_0^x uf(u)du.$$

In Figure 2.3, we show $\rho(x)$ for distributions in the H_2 class with mean 1, $C_S^2 = 19$ and different values of θ_3. As a reference, we also show $\rho(x)$ for the exponential distribution with mean 1. As can be seen from Figure 2.3, increasing θ_3 while holding the first two moments of the H_2 distribution constant, causes the load to (almost monotonically) shift towards smaller jobs. While the large jobs also become larger, they become rarer at an even faster rate so that in the limit as $\theta_3 \to \infty$, the $\rho(x)$ curve for the H_2 distribution converges to the $\rho(x)$ curve for the exponential distribution with the same mean. Thus as θ_3 increases, the fraction of smaller jobs arriving into the $M/H_2/k$ queue increases, thereby causing a smaller mean waiting time. In fact, this behavior would hold for any $M/G/k$ system where the service distribution is a mixture of two scaled versions of an arbitrary distribution.

Based on the numerical evidence of the huge variation in $\mathbf{E}\left[W^{M/H_2/k}\right]$, a natural question that arises is: Can this span of possible values of $\mathbf{E}\left[W^{M/H_2/k}\right]$ be quantified? Lemmas 2.3 and 2.4 in Section 2.5 answer this question. Lemma 2.3 is obtained by considering the case of a distribution in the H_2 class with a small value of θ_3. In particular, we consider the case of an H_2^* distribution (see Definition 2.2) which we can prove has the lowest possible third moment of all distributions in the H_2 family (with any given first two moments), and we derive the exact mean waiting time under the H_2^* jobs size distribution. Likewise, Lemma 2.4 is derived by considering the case of an H_2 distribution where θ_3 goes to ∞ and we derive the asymptotic mean waiting time for that situation. Since we restrict our attention to a subset of the entire space of distributions with given first two moments, our results provide a lower bound on the exact span of $\mathbf{E}\left[W^{M/G/k}\right]$. However, all known tight bounds for $GI/GI/1$ involving the first two moments of the service distribution are obtained by considering two-point distributions. We too conjecture that the bounds in Theorem 2.1 are tight, whereas the bounds in Theorem 2.2 can be tightened as described in the conjecture below:

Conjecture 2.1 *For* $\mathbf{E}[S] = 1$ *any finite* C_S^2,

$$W_h^{C_S^2} = \left(C_S^2 + 1\right) \mathbf{E}\left[W^{M/D/k}\right] \quad \textit{for all } \rho < k$$

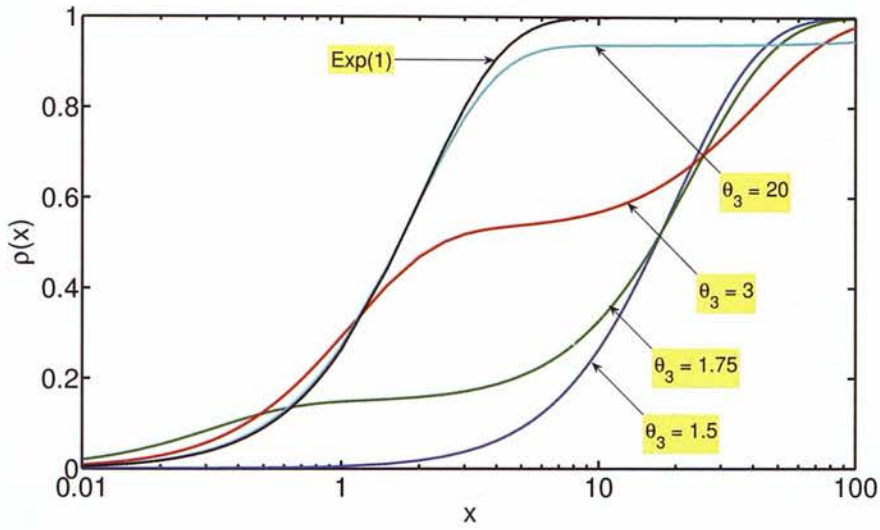


Figure 2.3: Illustration of the effect of the normalized 3rd moment, θ_3, on the distribution of load as a function of job size for the H_2 class of distributions. The first two moments were held constant at $\mathbf{E}[S] = 1$ and $C_S^2 = 19$. The distribution of the load for exponential distribution with mean 1, labeled Exp(1), is shown for reference.

and,

$$W_l^{C_S^2} = \begin{cases} \mathbf{E}\left[W^{M/D/k}\right] & \text{if } \rho < k-1 \\ \mathbf{E}\left[W^{M/D/k}\right] + \left[\frac{\rho-(k-1)}{k-\rho}\right] \frac{C_S^2}{2} & \text{if } \frac{k-1}{k} \leq \rho < 1 \end{cases}$$

where $\mathbf{E}\left[W^{M/D/k}\right]$ *is the mean waiting time when all the jobs have a constant size* 1.

2.4 Proof of Theorem 2.1

To obtain the bounds on $W_h^{C_S^2}$ and $W_l^{C_S^2}$ in Theorem 2.1, it suffices to show the existence of service distributions with SCV C_S^2 which give the desired expressions for mean waiting times. To obtain an upper bound on $W_l^{C_S^2}$, we use a corollary of [44], Proposition 3.15:

Lemma 2.1 (Daley [44, Proposition 3.15]) *For* $\mathbf{E}[S] = 1$, *any* $C_S^2 > 0$ *and* $0 <$

$\epsilon < \sqrt{\frac{1}{C_S^2}}$, *define the following random variable with a two-point distribution:*

$$D^{(\epsilon)} \sim \begin{cases} 1 - \epsilon\sqrt{C_S^2} & \textit{with probability } \frac{1}{1+\epsilon^2} \\ 1 + \frac{\sqrt{C_S^2}}{\epsilon} & \textit{with probability } \frac{\epsilon^2}{1+\epsilon^2}. \end{cases}$$

For $\rho < k - 1$ and any given GI arrival process,

$$\lim_{\epsilon \to 0} \mathbf{E}\left[W^{GI/D^{(\epsilon)}/k}\right] = \mathbf{E}\left[W^{GI/D/k}\right]$$

where $\mathbf{E}\left[W^{GI/D/k}\right]$ is the mean waiting time when the service distribution is deterministic 1.

By definition, each distribution in the $D^{(\epsilon)}$ family has mean 1 and SCV C_S^2. The bound on $W_l^{C_S^2}$ follows by setting the inter-arrival time distribution to be Exponential ($GI \equiv M$).

To obtain a lower bound on $W_h^{C_S^2}$, we consider the following two-point distribution:

$$D_2^* \sim \begin{cases} 0 & \text{with probability } \frac{C_S^2}{C_S^2+1} \\ C_S^2 + 1 & \text{with probability } \frac{1}{C_S^2+1}. \end{cases}$$

It is easy to verify that the above distribution has mean 1, squared coefficient of variation C_S^2, and $\theta_3 = 1$. We denote the $M/G/k$ system with D_2^* service distribution as $M/D_2^*/k$.

The bound on $W_h^{C_S^2}$ follows from the following lemma:

Lemma 2.2 *For any $\rho < k$ and $C_S^2 > 0$,*

$$\mathbf{E}\left[W^{M/D_2^*/k}\right] = (C_S^2 + 1)\mathbf{E}\left[W^{M/D/k}\right].$$

Proof: Since the scheduling discipline is size independent, the distributions of the waiting times experienced by zero-sized jobs and non-zero jobs are identical. Further, to find the waiting time distribution experienced by non-zero sized jobs, we can ignore the presence of zero-sized jobs. The waiting time distribution of the non-zero sized jobs is thus equivalent to the waiting time distribution in an $M/D/k$ system with arrival rate $\frac{\lambda}{C_S^2+1}$ and mean job size $(C_S^2 + 1)$. The latter system, however, is just an $M/D/k$ system with arrival rate λ and mean job size 1 seen on a slower time scale, slowed by a factor $(C_S^2 + 1)$. Hence, the mean waiting time of the original system is also $(C_S^2 + 1)$ times the mean waiting time of an $M/D/k$ system with arrival rate λ and mean job size 1. ■

2.5 Proof of Theorem 2.2

As in the proof of Theorem 2.1, to obtain the bounds on $W_h^{C_S^2}$ and $W_l^{C_S^2}$ in Theorem 2.2, it suffices to show the existence of service distributions with SCV C_S^2 which give the desired mean waiting times. To handle the case $\rho > k - 1$, we resort to service distributions in the class of 2-phase hyperexponentials.

To obtain a lower bound on $W_h^{C_S^2}$, we consider the following degenerate hyperexponential distribution:

$$H_2^* \sim \begin{cases} 0 & \text{with probability } \frac{C_S^2-1}{C_S^2+1} \\ \text{Exp}\left(\frac{2}{C_S^2+1}\right) & \text{with probability } \frac{2}{C_S^2+1}. \end{cases}$$

It is easy to verify that the above distribution has mean 1, squared coefficient of variation C_S^2, and $\theta_3 = \frac{3}{2}$. The H_2^* distribution as defined above has the lowest third moment among all the H_n distributions with mean 1 and SCV C_S^2:

Claim 2.1 *Let $\cup_{n>1}\{H_n|C_S^2\}$ be the set of all hyperexponential distributions with finite number of phases, mean 1 and squared coefficient of variation C_S^2 ($C_S^2 > 1$). The H_2^* distribution lying in this set has the smallest third moment among all the distributions in $\cup_{n>1}\{H_n|C_S^2\}$.*

The bound on $W_h^{C_S^2}$ in Theorem 2.2 follows from the following lemma which can be proved along the lines of Lemma 2.2:

Lemma 2.3 *For any $\rho < k$ and $C_S^2 > 1$,*

$$\mathbf{E}\left[W^{M/H_2^*/k}\right] = \left(\frac{C_S^2+1}{2}\right)\mathbf{E}\left[W^{M/M/k}\right].$$

Note that the bound obtained from Lemma 2.3 is weaker than the bound from Lemma 2.2 since $\mathbf{E}\left[W^{M/M/k}\right] < 2 \cdot \mathbf{E}\left[W^{M/D/k}\right]$. We present Lemma 2.3 here for comparison with the corresponding upper bound on $W_l^{C_S^2}$ in Lemma 2.4 and the 2-moment approximation (2.1), which involve $\mathbf{E}\left[W^{M/M/k}\right]$.

To obtain a bound on $W_l^{C_S^2}$, we consider a sequence of systems parametrized by a parameter ϵ in which we *fix the first two moments of the service distribution* analogous to Lemma 2.1. The parameter ϵ allows for increasing the third moment as ϵ goes to 0. More precisely, we consider the sequence of queues $M/H_2^{(\epsilon)}/k$ (see Section 2.5.2, Definition 2.3) as $\epsilon \to 0$ and prove the following limit theorem:

Lemma 2.4 *For* $\mathbf{E}[S] = 1$ *and any finite* C_S^2,

$$\lim_{\epsilon \to 0} \mathbf{E}\left[W^{M/H_2^{(\epsilon)}/k}\right] = \begin{cases} \mathbf{E}\left[W^{M/M/k}\right] & \text{if } \rho < k-1 \\ \mathbf{E}\left[W^{M/M/k}\right] + \left[\frac{\rho-(k-1)}{k-\rho}\right] \frac{C_S^2-1}{2} & \text{if } \rho \geq k-1 \end{cases}$$

where $\mathbf{E}\left[W^{M/M/k}\right]$ *is the mean waiting time when the service distribution is exponential with mean* 1.

The rest of this section is devoted to proving Lemma 2.4. Since the proof of Lemma 2.4 involves a new technique, we begin in Section 2.5.1 with a high level proof idea. Subsequent subsections will provide the rigorous lemmas.

2.5.1 Proof idea

The key steps involved in the analysis are as follows:

1. We first observe that the $H_2^{(\epsilon)}$ service distribution is made up of two classes of jobs – small jobs and large jobs. We use N_s and N_ℓ to denote the number of small and large jobs in system, respectively.

2. We show that the expected number of large jobs, $\mathbf{E}\left[N_\ell^{M/H_2^{(\epsilon)}/k}\right]$, vanishes as ϵ goes to zero; therefore it suffices to consider only small jobs (see Section 2.5.3).

3. For each $M/H_2^{(\epsilon)}/k$ system, we construct another system, $U^{(\epsilon)}$, which stochastically upper bounds the number of small jobs in the corresponding $M/H_2^{(\epsilon)}/k$ system. That is,

   $$N_s^{M/H_2^{(\epsilon)}/k} \leq_{st} N_s^{U^{(\epsilon)}}$$

   (see Section 2.5.4).

4. To analyze $N_s^{U^{(\epsilon)}}$, we consider two kinds of periods: **good** periods – when there are no large jobs in the system, and **bad** periods – when there is at least one large job in the system. Our approach is to obtain upper bounds on the mean number of small jobs during the good and bad periods, $\mathbf{E}\left[N_s^{U^{(\epsilon)}} \mid \text{good period}\right]$ and $\mathbf{E}\left[N_s^{U^{(\epsilon)}} \mid \text{bad period}\right]$, respectively, and obtain an upper bound on $\mathbf{E}\left[N_s^{U^{(\epsilon)}}\right]$ using the law of total probability:

   $$\begin{aligned} \mathbf{E}\left[N_s^{U^{(\epsilon)}}\right] = \mathbf{E}\left[N_s^{U^{(\epsilon)}} \mid \text{good period}\right] &\mathbf{Pr}[\text{good period}] \\ + \mathbf{E}\left[N_s^{U^{(\epsilon)}} \mid \text{bad period}\right] &\mathbf{Pr}[\text{bad period}] \end{aligned}$$

We obtain upper bounds on the mean number of small jobs during the good and bad periods using the following steps (see Section 2.5.5):

(a) We first look at the number of small jobs only at *switching points*. That is, we consider the number of small jobs only at the instants when the system switches from a good period to a bad period and vice versa.

(b) To obtain bounds on the number of small jobs at the switching points, we define a random variable Δ, which upper bounds the *increment* in the number of small jobs during a bad period. Further, by our definition, the upper bound Δ is independent of the number of small jobs at the beginning of the bad period. To keep the analysis simple, this independence turns out to be crucial.

(c) Next we obtain a stochastic upper bound on the number of small jobs at the end of a good period by solving a fixed point equation of the form

$$A \stackrel{d}{=} \Phi(A + \Delta)$$

where A is the random variable for (the stochastic upper bound on) the number of small jobs at the end of a good period, and Φ is a function that maps the number of small jobs at the beginning of a good period to the number of small jobs at the end of the good period.

(d) Finally, we obtain the mean number of small jobs *during* the good and bad periods from the mean number of small jobs at the switching points.

5. Similar to $U^{(\epsilon)}$, for each $M/H_2^{(\epsilon)}/k$ system, we also construct a system, $L^{(\epsilon)}$, which stochastically lower bounds the number of small jobs in the corresponding $M/H_2^{(\epsilon)}/k$ system. That is,

$$N_s^{M/H_2^{(\epsilon)}/k} \geq_{st} N_s^{L^{(\epsilon)}}$$

(see Section 2.5.6). We omit the analysis of $L^{(\epsilon)}$ since it is similar to analysis of $U^{(\epsilon)}$. Note, that we indeed obtain

$$\mathbf{E}\left[N_s^{U^{(\epsilon)}}\right] = \mathbf{E}\left[N_s^{L^{(\epsilon)}}\right] + o(1)$$

Convergence of $\mathbf{E}\left[N^{M/H_2^{(\epsilon)}/k}\right]$ follows from convergence of its upper and lower bounds.

6. Finally, we use Little's law to obtain mean waiting time, $\mathbf{E}\left[W^{M/H_2^{(\epsilon)}/k}\right]$, from the mean number of waiting jobs, $\mathbf{E}\left[N^{M/H_2^{(\epsilon)}/k}\right] - \rho$.

2.5.2 Preliminaries

Below we give a formal definition of the $H_2^{(\epsilon)}$ class of service distributions.

Definition 2.3 *We define a family of distributions parametrized by ϵ as follows:*

$$H_2^{(\epsilon)} = \begin{cases} \text{Exp}\left(\mu_s^{(\epsilon)}\right) & \text{with probability } p^{(\epsilon)} \\ \text{Exp}\left(\mu_\ell^{(\epsilon)}\right) & \text{with probability } 1 - p^{(\epsilon)} \end{cases}$$

$\mu_s^{(\epsilon)} > \mu_\ell^{(\epsilon)}$, where $\mu_s^{(\epsilon)}$, $\mu_\ell^{(\epsilon)}$ and $p^{(\epsilon)}$ satisfy,

$$\frac{p^{(\epsilon)}}{\mu_s^{(\epsilon)}} + \frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}} = \mathbf{E}\left[S^{(\epsilon)}\right] = 1$$

$$2\frac{p^{(\epsilon)}}{\left(\mu_s^{(\epsilon)}\right)^2} + 2\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^2} = \mathbf{E}\left[\left(S^{(\epsilon)}\right)^2\right] = C_S^2 + 1$$

$$6\frac{p^{(\epsilon)}}{\left(\mu_s^{(\epsilon)}\right)^3} + 6\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^3} = \mathbf{E}\left[\left(S^{(\epsilon)}\right)^3\right] = \frac{1}{\epsilon}$$

For proving the upper bound on the lower bound $W_l^{C_S^2}$ of $\mathbf{E}[W]$, we look at $\mathbf{E}\left[W^{M/H_2^{(\epsilon)}/k}\right]$ as $\epsilon \to 0$. That is, the third moment of service time goes to ∞. Below we present some elementary results on the asymptotic behavior[3] of the parameters of the $H_2^{(\epsilon)}$

[3]We will use the following asymptotic notation frequently in this chapter: We say a function $h(\epsilon)$ is:

1. $\Theta(g(\epsilon))$ if

$$0 < \liminf_{\epsilon \to 0} \left|\frac{h(\epsilon)}{g(\epsilon)}\right| \leq \limsup_{\epsilon \to 0} \left|\frac{h(\epsilon)}{g(\epsilon)}\right| < \infty$$

   Intuitively, this means that the functions h and g grow at the same rate, asymptotically, as $\epsilon \to 0$.

2. $o(g(\epsilon))$ if

$$\lim_{\epsilon \to 0} \left|\frac{h(\epsilon)}{g(\epsilon)}\right| = 0$$

   Intuitively, h becomes insignificant when compared with g, asymptotically, as $\epsilon \to 0$.

3. $O(g(\epsilon))$ if

$$\limsup_{\epsilon \to 0} \left|\frac{h(\epsilon)}{g(\epsilon)}\right| < \infty$$

distribution, which will be used in the analysis in Section 2.5.5.

Lemma 2.5 *The $\mu_s^{(\epsilon)}$, $\mu_\ell^{(\epsilon)}$ and $p^{(\epsilon)}$ can be expressed in terms of ϵ as :*

$$\mu_s^{(\epsilon)} = 1 + \frac{3}{2}(C_S^2 - 1)^2\epsilon + \Theta(\epsilon^2)$$
$$\mu_\ell^{(\epsilon)} = 3(C_S^2 - 1)\epsilon + 18C_S^2(C_S^2 - 1)\epsilon^2 + \Theta(\epsilon^3)$$
$$p^{(\epsilon)} = 1 - \frac{9}{2}(C_S^2 - 1)^3\epsilon^2 + \Theta(\epsilon^3)$$

Proof in Appendix 2.A.

Corollary 2.1 *As $\epsilon \to 0$,*

$$p^{(\epsilon)} \to 1 \quad , \quad \mu_s^{(\epsilon)} \to 1$$
$$\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}} \to 0 \quad , \quad \frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^2} \to \frac{C_S^2-1}{2}$$

Corollary 2.1 formalizes the observation we made from Figure 2.3: As the third moment grows, asymptotically, all the load is made up *only* by the small jobs, whose mean approaches 1. While the mean size of the large jobs also grows linearly in the third moment (asymptotically), the probability that a large job arrives vanishes at a faster rate. Thus, intuitively, our $M/H_2^{(\epsilon)}/k$ system rarely encounters a large job in the limit as $\epsilon \to 0$.

It is important to point out that, as $\epsilon \to 0$, the $H_2^{(\epsilon)}$ distribution converges in distribution to the Exp(1) distribution. Thus, the stationary queue length and waiting time distributions of the sequence of $M/H_2^{(\epsilon)}/k$ systems also converge in distribution to the queue length and waiting time distributions of the corresponding $M/M/k$ system [30, 139]. However, convergence in distribution of the waiting time *does not* imply convergence of the mean waiting time; namely, it is possible that

$$\lim_{\epsilon \to 0} \mathbf{E}\left[W^{M/H_2^{(\epsilon)}/k}\right] \neq \mathbf{E}\left[W^{M/M/k}\right]. \tag{2.6}$$

Indeed, (2.6) can be verified for $k = 1$ where the mean waiting time is given by the Pollaczek-Khintchine formula (2.4). Lemma 2.4 proves that the non-convergence (2.6) holds more generally for the $M/H_2^{(\epsilon)}/k$ system when $\rho > k - 1$.

That is, h is either $\Theta(g(\epsilon))$ or $o(g(\epsilon))$.

Daley [44] proved an analogous non-convergence result by considering a class of service distributions, $S^{(\epsilon)}$, which includes $H_2^{(\epsilon)}$ service distributions. He further conjectured [44, Conjecture 3.19] an expression for the difference,

$$\lim_{\epsilon \to 0} \mathbf{E}\left[W^{GI/S^{(\epsilon)}/k}\right] - \mathbf{E}\left[W^{GI/S/k}\right],$$

where S denotes the limiting service distribution. The proof of Lemma 2.4 verifies Daley's conjecture for the case of Poisson arrival process and H_2 service distribution.

2.5.3 Bounding the number of large jobs

The following lemma proves that to bound the mean number of jobs in an $M/H_2^{(\epsilon)}/k$ system within $o(1)$, it suffices to consider only the small jobs.

Lemma 2.6 $\mathbf{E}\left[N_\ell^{M/H_2^{(\epsilon)}/k}\right] = o(1)$

Proof: We will upper bound the expected number of large customers in the system by (a) giving high priority to the small customers and letting the large jobs receive service only when there are no small jobs in the system, and (b) by allowing the large customers to be served by at most one server at any time. Further, we increase the arrival rate of small customers to λ and increase the mean size of the small customers to 1. By not being work conserving, increasing the arrival rate, and making small jobs stochastically larger, the modified system can become overloaded. However, since we are only interested in the asymptotic behavior as $\epsilon \to 0$, it suffices to find an ϵ' such that the above system is stable for all $\epsilon < \epsilon'$. This is indeed true for $\epsilon' = \frac{1}{6}\left[\frac{\rho(C_S^2+1)^2}{4(1-\rho/k)} + 1\right]^{-1}$ (See proof of Lemma 2.9).

For brevity, we use $M(a)/M(b)/k$ to denote an $M/M/k$ queue with arrival rate a and service rate b. Let $\overline{N_\ell}^{(\epsilon)}$ be the steady-state number of customers in an $M\left(\lambda(1-p^{(\epsilon)})\right)/M\left(\mu_\ell^{(\epsilon)}\right)/1$ queue with service interruptions, where the server is interrupted for the duration of the busy period of an $M(\lambda)/M(1)/k$ queue. It is easy to see that

$$\mathbf{E}\left[N_\ell^{M/H_2^{(\epsilon)}/k}\right] \leq \mathbf{E}\left[\overline{N_\ell}^{(\epsilon)}\right].$$

The proof is completed by the following lemma:

Lemma 2.7 $\mathbf{E}\left[\overline{N_\ell}^{(\epsilon)}\right] = o(1)$

Proof in Appendix 2.A. ∎

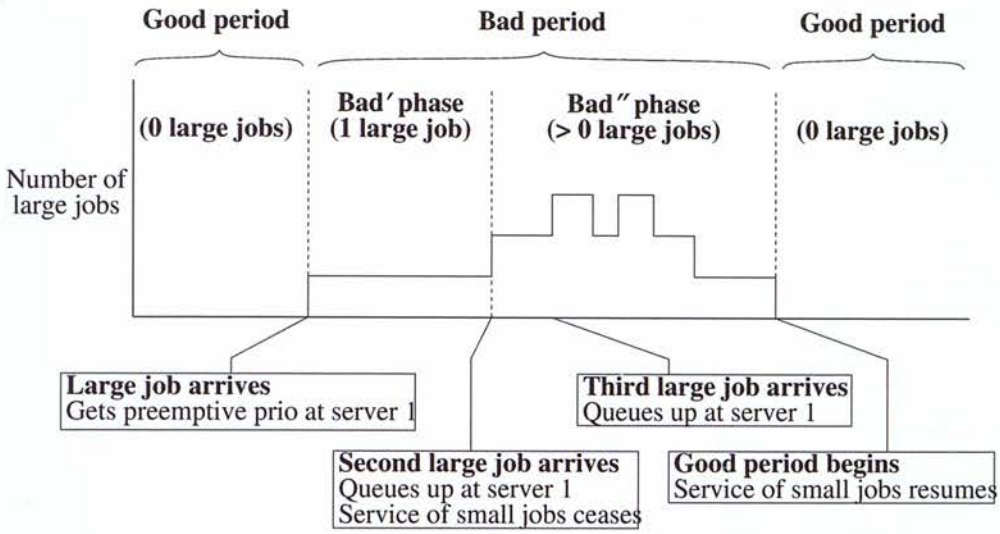


Figure 2.4: Construction of system $U^{(\epsilon)}$ which upper bounds the number of jobs in an $M/H_2^{(\epsilon)}/k$

2.5.4 Construction of $U^{(\epsilon)}$: the upper bounding system for $N_s^{M/H_2^{(\epsilon)}/k}$

Figure 2.4 illustrates the behavior of system $U^{(\epsilon)}$, which upper bounds the number of small jobs in an $M/H_2^{(\epsilon)}/k$. Denote periods where there are no large jobs (including when the system is idle) as *good* periods, and periods when there is at least 1 large job as a *bad* period. During a good period, the small jobs receive service according to a normal k server FIFO system. As soon as a large job arrives, we say that a bad period begins. The bad period consists of up to 2 phases, called *bad′* and *bad″*. A bad′ phase spans the time from when a large job first arrives until either it leaves or a second large job arrives (whichever happens earlier). A bad″ phase occurs if a second large job arrives while the first large job is still in the system, and covers the period from when this 2nd large job arrives (if it does) until there are no more large jobs in the system.

The large job starting a bad period preempts the small job at server 1 (if any) and starts receiving service. The small jobs are served by the remaining $(k-1)$ servers. If a second large job arrives during a bad period while the first large job is still in system, starting a bad″ phase, we cease serving the small jobs and continue serving the large jobs by *only* server 1 until this busy period of large jobs ends (there are no more large jobs). When the last large job leaves, we resume the service of small jobs

according to a normal k server FIFO system.

Analyzing system $U^{(\epsilon)}$ is simpler than analyzing the corresponding $M/H_2^{(\epsilon)}/k$ system because in $U^{(\epsilon)}$, the large jobs form an $M/M/1$ system independent of the small jobs, due to preemptive priority and service by only one server. The small jobs operate in a random environment where they have either k, $(k-1)$ or 0 servers.

Lemma 2.8 *The number of small jobs in an $M/H_2^{(\epsilon)}/k$ system, $N_s^{M/H_2^{(\epsilon)}/k}$, is stochastically upper bounded by the number of small jobs in the corresponding system $U^{(\epsilon)}$, $N_s^{U^{(\epsilon)}}$.*

Proof: Straightforward using stochastic coupling. ∎

Stability of system $U^{(\epsilon)}$: Since system $U^{(\epsilon)}$ is not work conserving, there are values of ϵ for which it is unstable, even when $\rho < k$. Therefore we restrict our attention to the following range of ϵ:

Lemma 2.9 *The upper bounding system, $U^{(\epsilon)}$, is stable for $\epsilon < \epsilon'$ where*

$$\epsilon' = \frac{1}{6}\left[\frac{\rho(C_S^2+1)^2}{4(1-\rho/k)} + 1\right]^{-1}.$$

Proof in Appendix 2.A.

2.5.5 Analysis of system $U^{(\epsilon)}$

Figure 2.5 introduces the notation we will use in this section. Since in this section we focus only on the analysis of system $U^{(\epsilon)}$, we will omit superscripting the random variables used in analysis by $U^{(\epsilon)}$ for readability. Unless explicitly superscripted, random variables correspond to the $U^{(\epsilon)}$ system. We define the following random variables:

- $N_{s,g}^* \equiv$ the number of small jobs *at the end* of a good period, that is, when the system switches from a good to a bad period
- $N_{s,b}^* \equiv$ the number of small jobs *at the end* of a bad period, that is, when the system switches from a bad to a good period
- $N_{s,g} \equiv$ the time stationary number of small jobs *during* a good period
- $N_{s,b} \equiv$ the time stationary number of small jobs *during* a bad period

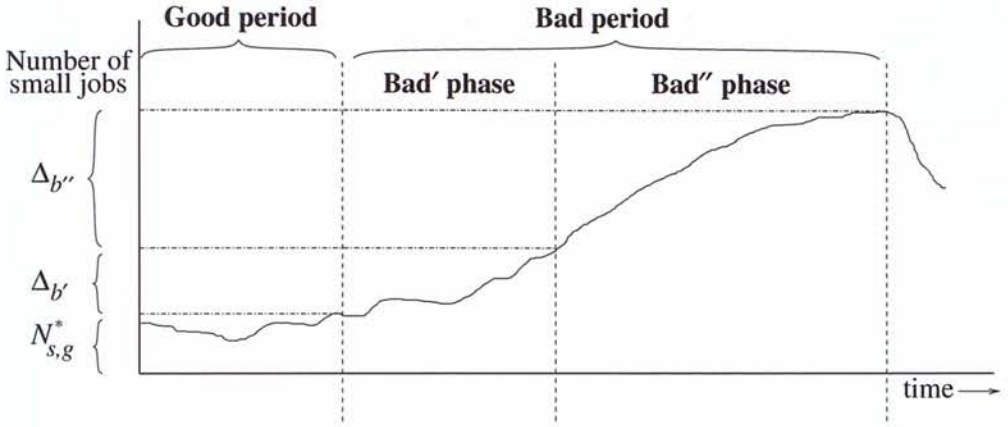


Figure 2.5: Notation used for analysis of system $U^{(\epsilon)}$

- $\Delta_{b'} \equiv$ the *increment* in the number of small jobs during a bad′ period (when small jobs have $(k-1)$ servers available)
- $\Delta_{b'}(n) \equiv$ the *increment* in the number of small jobs during a bad′ period given that the bad′ period begins with n small jobs
- $\Delta_{b''} \equiv$ the *increment* in the number of small jobs during a bad″ period (where the service of small jobs has been blocked)
- $\Delta_b = \Delta_{b'}(0) + \Delta_{b''}$

We denote the fraction of time spent in a good, bad, bad′ and bad″ phase by $\mathbf{Pr}[g]$, $\mathbf{Pr}[b]$, $\mathbf{Pr}[b']$ and $\mathbf{Pr}[b'']$ respectively.

By the law of total probability,

$$\mathbf{E}[N_s] = \mathbf{E}[N_{s,g}]\mathbf{Pr}[g] + \mathbf{E}[N_{s,b}]\mathbf{Pr}[b] \tag{2.7}$$

In Section 2.5.5, we derive stochastic upper bounds on $N_{s,g}$ and $N_{s,b}$, which give us an upper bound, (2.9), on $\mathbf{E}[N_s]$. In Sections 2.5.5 and 2.5.5, we derive expressions for the quantities appearing in (2.9). These are used to obtain the final upper bound on $\mathbf{E}[N_s]$ in Section 2.5.5.

Stochastic Bounds

Obtaining a stochastic upper bound on $N_{s,g}$: Let $\Phi(A)$ be a mapping between non-negative random variables where $\Phi(A)$ gives the random variable for the number

of small jobs at the end of a good period, given that the number at the beginning of the good period is given by A. Let $\bar{N}^*_{s,g}$ be the solution to the following fixed point equation:

$$\bar{N}^*_{s,g} \stackrel{d}{=} \Phi(\bar{N}^*_{s,g} + \Delta_b) \tag{2.8}$$

Lemma 2.10

$$N_{s,g} \stackrel{d}{=} N^*_{s,g} \leq_{st} \bar{N}^*_{s,g}$$

Proof sketch: The first relation follows since the length of a good period is exponential and its termination is independent of the number of small jobs. Hence, by *conditional* PASTA [148] (see also [69] for a similar use of conditional PASTA),

$$N_{s,g} \stackrel{d}{=} N^*_{s,g}$$

Intuitively, Δ_b stochastically upper bounds the increment in the number of small jobs during a bad period since it assumes there were zero small jobs at the beginning of the bad period and hence ignores the departures of those small jobs. Therefore, solving the fixed point equation (2.8) gives a stochastic upper bound on $N^*_{s,g}$. A formal proof of the stochastic inequality is in Appendix 2.A. ■

Obtaining a stochastic upper bound on $N_{s,b}$: The required upper bound is given by the following lemma.

Lemma 2.11

$$N_{s,b} \leq_{st} \bar{N}^*_{s,g} + \Delta_{b'}(0) + \mathbf{I}_{b''|b} A_\lambda\left(T_{b''e}\right)$$

where $A_\lambda\left(T_{b''e}\right)$ is the number of arrivals of a Poisson process (with rate λ) during a random time interval $T_{b''e}$ denoting the excess *of the length of a bad'' period, and where $\mathbf{I}_{b''|b}$ denotes an indicator random variable which is* 1 *with probability* $\mathbf{Pr}[b'']/\mathbf{Pr}[b]$.

Proof sketch: Observe that the first term in the upper bound is a stochastic upper bound on the number of small jobs at the beginning of a bad period. The second term denotes a stochastic upper bound on the increment in the number of small jobs during a bad' phase. Finally, the third term denotes the "average increment" in the number of small jobs during a bad'' phase. See Appendix 2.A for the complete proof. ■

Combining the bounds on $N_{s,g}$ and $N_{s,b}$, we get an upper bound on $\mathbf{E}[N_s]$:

$$\mathbf{E}[N_s] \leq \mathbf{E}\left[\bar{N}^*_{s,g}\right]\mathbf{Pr}[g] + \mathbf{E}\left[\bar{N}^*_{s,g} + \Delta_{b'}(0) + \mathbf{I}_{b''|b} A_\lambda\left(T_{b''e}\right)\right]\mathbf{Pr}[b] \tag{2.9}$$

To complete the proof, we need expressions for each of the quantities in equation (2.9). In Section 2.5.5 we will obtain expressions for $\mathbf{E}[\Delta_{b'}(0)]$ for the cases $\rho < k-1$ and $\rho \geq k-1$. In Section 2.5.5 we will obtain $\mathbf{E}\left[\bar{N}^*_{s,g}\right]$. However, to do this, we will need the first two moments of Δ_b, $\mathbf{E}[\Delta_b]$ and $\mathbf{E}[\Delta_b^2]$, which are also derived in Section 2.5.5.

To obtain $\mathbf{Pr}[b]$, recall that the large jobs form an $M/M/1$ system. Hence (see Lemma 2.5 for expressions for p and μ_ℓ),

$$\begin{aligned}\mathbf{Pr}[b] = \mathbf{Pr}[\geq 1 \text{ large job}] &= \frac{\lambda(1-p^{(\epsilon)})}{\mu_\ell^{(\epsilon)}} \\ &= \frac{3\rho(C_S^2-1)^2\epsilon}{2} + \Theta(\epsilon^2)\end{aligned} \tag{2.10}$$

The following asymptotic behavior of $\frac{\mathbf{Pr}[b'']}{\mathbf{Pr}[b]}\mathbf{E}[A_\lambda\left(T_{b''e}\right)]$ is proved in the proof of Lemma 2.14:

$$\frac{\mathbf{Pr}[b'']}{\mathbf{Pr}[b]}\mathbf{E}[A_\lambda\left(T_{b''e}\right)] = \Theta(1) \tag{2.11}$$

In Section 2.5.5, we perform the final calculations by substituting the above quantities into (2.9).

Obtaining $\mathbf{E}[\Delta_b]$ and $\mathbf{E}[\Delta_b^2]$

Recall that we defined,

$$\Delta_b = \Delta_{b'}(0) + \Delta_{b''}$$

where $\Delta_{b'}(0)$ is the random variable for the number small jobs at the end of a bad$'$ phase given that it starts with 0 small jobs and $\Delta_{b''}$ is the number of small of jobs that arrive during a bad$''$ phase.

Lemma 2.12 gives the expressions for $\mathbf{E}[\Delta_{b'}(0)]$ and $\mathbf{E}[\Delta_{b'}^2(0)]$. Lemma 2.14 gives the asymptotic expressions for $\mathbf{E}[\Delta_{b''}]$ and $\mathbf{E}[\Delta_{b''}^2]$ which will be sufficient for our purposes of obtaining $\mathbf{E}[N_s]$ within $o(1)$.

Lemma 2.12
Case: $\rho < k-1$

$$\mathbf{E}[\Delta_{b'}(0)] = O(1)$$
$$\mathbf{E}\left[\Delta_{b'}^2(0)\right] = O(1)$$

Case: $\rho > k-1$

$$\mathbf{E}[\Delta_{b'}(0)] = \frac{(\rho-(k-1))}{3(C_S^2-1)\epsilon} + \Theta(1)$$
$$\mathbf{E}\left[\Delta_{b'}^2(0)\right] = \frac{2}{9}\frac{(\rho-(k-1))^2}{(C_S^2-1)^2\epsilon^2} + \Theta\left(\frac{1}{\epsilon}\right)$$

Proof: We can think of $\Delta_{b'}(0)$ as the number of jobs in an $M/M/k-1$ with arrival rate $\lambda_s = \lambda p$ and service rate μ_s at time $T \sim \text{Exp}(\beta)$ ($\beta = \lambda(1-p) + \mu_\ell$) given that it starts empty. Let us call this $N^{M(\lambda_s)/M(\mu_s)/k-1}(T)$. Let $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)$ be the number of jobs in an $M/M/1$ with arrival rate λ_s and service rate $(k-1)\mu_s$ at time T given that it starts empty. Then,

$$N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T) \leq_{st} N^{M(\lambda_s)/M(\mu_s)/k-1}(T) \leq_{st} \quad N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T) + (k-1) \tag{2.12}$$

To see why (2.12) is true, first note that using coupling, $N^{M(\lambda_s)/M(\mu_s)/k-1}(T)$ can be (stochastically) sandwiched between $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)$ and the number of jobs in an $M/M/k-1$ where the service is stopped when the number of jobs goes below $k-1$. Finally, again using coupling, the number of jobs in this latter system can be stochastically upper bounded by $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T) + (k-1)$.

Therefore, using (2.12), we only need to evaluate the first and second moments of $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)$ to obtain $\mathbf{E}[\Delta_{b'}(0)]$ and $\mathbf{E}[\Delta_{b'}^2(0)]$ within an error of $\Theta(1)$ and $\Theta(\mathbf{E}[\Delta_{b'}(0)])$, respectively. We do this next.

Case: $\rho < k-1$
For this case the $M/M/k-1$ system is stable during bad′ phases, and hence

$$\mathbf{E}[\Delta_{b'}(0)] = O(1)$$
$$\mathbf{E}\left[\Delta_{b'}^2(0)\right] = O(1).$$

Case: $\rho > k-1$

The following lemma gives the expressions for the first and second moments of $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)$ for the case $\rho > k-1$.

Lemma 2.13 *Let $T \sim \text{Exp}(\beta)$ and $\lambda_s > (k-1)\mu_s$. Then,*

$$\mathbf{E}\Big[N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)\Big] = \frac{\lambda_s - (k-1)\mu_s}{\beta} + \Theta(1)$$

$$\mathbf{E}\Big[(N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T))^2\Big] = 2\left(\frac{\lambda_s - (k-1)\mu_s}{\beta}\right)^2 + \Theta\left(\frac{1}{\beta}\right).$$

Proof of Lemma 2.13: See Appendix 2.A.

Now, using the inequality (2.12) and Lemma 2.13, and substituting in the expressions for μ_s, λ_s and μ_ℓ from Lemma 2.5 :

$$\begin{aligned}
\mathbf{E}[\Delta_{b'}(0)] &= \mathbf{E}\Big[N^{M(\lambda_s)/M(\mu_s)/k-1}(T)\Big] \\
&\leq \mathbf{E}\Big[N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)\Big] + O(1) \\
&= \frac{\lambda_s - (k-1)\mu_s}{\beta} + \Theta(1) \\
&= \frac{\lambda p - (k-1)\mu_s}{\lambda(1-p) + \mu_\ell} + \Theta(1) \\
&= \frac{\lambda(1-\Theta(\epsilon^2)) - (k-1)(1+\Theta(\epsilon))}{\lambda\Theta(\epsilon^2) + (3(C_S^2-1)\epsilon + \Theta(\epsilon^2))} + \Theta(1) \\
&= \frac{(\rho - (k-1))}{3(C_S^2-1)\epsilon} + \Theta(1)
\end{aligned}$$

and,

$$\begin{aligned}
\mathbf{E}\Big[\Delta_{b'}^2(0)\Big] &= \mathbf{E}\Big[\Big(N^{M(\lambda_s)/M(\mu_s)/k-1}(T)\Big)^2\Big] \\
&\leq \mathbf{E}\Big[\Big(N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)\Big)^2\Big] + O\left(\frac{1}{\epsilon}\right) \\
&= 2\left(\frac{\lambda_s - (k-1)\mu_s}{\beta}\right)^2 + \Theta\left(\frac{1}{\beta}\right) + O\left(\frac{1}{\epsilon}\right) \\
&= 2\left(\frac{(\rho - (k-1))}{3(C_S^2-1)\epsilon}\right)^2 + \Theta\left(\frac{1}{\epsilon}\right)
\end{aligned}$$

■

Lemma 2.14 *The asymptotics for the first and second moments of $\Delta_{b''}$ are given by:*

$$\mathbf{E}[\Delta_{b''}] = O(1)$$

$$\mathbf{E}\left[\Delta_{b'}^2\right] = \Theta\left(\frac{1}{\epsilon}\right)$$

Proof: See Appendix 2.A. ∎

Obtaining $\mathbf{E}\left[\bar{N}_{s,g}^*\right]$

We will use the following lemma to obtain $\mathbf{E}\left[\bar{N}_{s,g}^*\right]$.

Lemma 2.15 *Consider an $M/M/k$ system with arrival rate λ and mean job size μ^{-1}. We interrupt this $M/M/k$ system according to a Poisson process with rate α, and at every interruption, a random number of jobs are added to the system. The number of jobs injected are i.i.d. random variables which are equal in distribution to some non-negative random variable Δ. Let $N^{(Int)}$ denote the number of jobs in this $M/M/k$ system. If $\mathbf{E}[\Delta] = o\left(\frac{1}{\alpha}\right)$, we have,*

$$\mathbf{E}\left[N^{(Int)}\right] = \mathbf{E}\left[N^{M/M/k}\right] + \frac{\frac{\alpha}{2}\mathbf{E}[\Delta^2]}{k\mu - \lambda} + o(1).$$

Proof in Appendix 2.A.

To use the above lemma, we will consider an $M/M/k$ with arrival rate $\lambda p^{(\epsilon)}$, mean job size $\frac{1}{\mu_1^{(\epsilon)}}$, $\alpha = \lambda(1 - p^{(\epsilon)})$ and $\Delta \stackrel{d}{=} \Delta_b$. Using the expression for $\mathbf{E}[\Delta_b]$ derived in Section 2.5.5, one can check that the condition of Lemma 2.15 is met. Therefore,

$$\mathbf{E}\left[\bar{N}_{s,g}^*\right] = \mathbf{E}\left[N^{M/M/k}\right] + \frac{1}{2}\frac{\lambda(1-p)\mathbf{E}[\Delta_b^2]}{k\mu - \lambda} + o(1) \tag{2.13}$$

Substituting $\mathbf{E}[\Delta_b^2]$ from Section 2.5.5 and using Lemma 2.5,
Case: $\rho < k - 1$

$$\begin{aligned}\mathbf{E}\left[\bar{N}_{s,g}^*\right] &= \mathbf{E}\left[N^{M/M/k}\right] + \frac{1}{2}\frac{\lambda\left(\frac{9}{2}(C_S^2-1)^3\epsilon^2\right)\Theta\left(\frac{1}{\epsilon}\right)}{k\mu - \lambda} + o(1)\\ &= \mathbf{E}\left[N^{M/M/k}\right] + o(1)\end{aligned}$$

Case: $\rho > k - 1$

$$\begin{aligned}&\mathbf{E}\left[\bar{N}_{s,g}^*\right]\\ =&\mathbf{E}\left[N^{M/M/k}\right] + \frac{1}{2}\frac{\lambda\left(\frac{9}{2}(C_S^2-1)^3\epsilon^2\right)\left(\frac{2}{9}\frac{(\rho-(k-1))^2}{(C_S^2-1)^2\epsilon^2}\right)}{k\mu - \lambda} + o(1)\\ =&\mathbf{E}\left[N^{M/M/k}\right] + \frac{\rho}{k-\rho}\left[\rho - (k-1)\right]^2\frac{C_S^2-1}{2} + o(1)\end{aligned}$$

Putting it together: Upper bound on $\mathbf{E}[N_s]$

Recall the expression for upper bound on $\mathbf{E}[N_s]$ from equation (2.9):

$$\mathbf{E}[N_s] \leq \mathbf{E}\left[\bar{N}^*_{s,g}\right](1 - \mathbf{Pr}[b]) + \mathbf{E}\left[\bar{N}^*_{s,g} + \Delta_{b'}(0) + \mathbf{I}_{b''|b} A_\lambda\left(T_{b''e}\right)\right]\mathbf{Pr}[b]$$

Substituting the expressions for $\mathbf{E}\left[\bar{N}^*_{s,g}\right]$ from Section 2.5.5, $\mathbf{E}[\Delta_{b'}(0)]$ from Lemma 2.12, $\mathbf{Pr}[b]$ from Equation (2.10) and $\frac{\mathbf{Pr}[b'']}{\mathbf{Pr}[b]}\mathbf{E}[A_\lambda\left(T_{b''e}\right)]$ from Equation (2.11) into the above equation, we get:

Case: $\rho < k - 1$

$$\mathbf{E}[N_s] \leq \mathbf{E}\left[N^{M/M/k}\right] + o(1)$$

Case: $\rho > k - 1$

$$\begin{aligned}
\mathbf{E}[N_s] &\leq \left(\mathbf{E}\left[N^{M/M/k}\right] + \frac{\rho}{k-\rho}\left[\rho - (k-1)\right]^2 \frac{C_S^2 - 1}{2}\right) \\
&\quad + \left(\frac{(\rho - (k-1))}{3(C_S^2 - 1)\epsilon} + \Theta(1)\right)\left(\frac{3\rho(C_S^2 - 1)^2\epsilon}{2}\right) + o(1) \\
&= \mathbf{E}\left[N^{M/M/k}\right] + \frac{\rho}{k-\rho}\left[\rho - (k-1)\right]^2 \frac{C_S^2 - 1}{2} + \rho\left[\rho - (k-1)\right]\frac{C_S^2 - 1}{2} + o(1) \\
&= \mathbf{E}\left[N^{M/M/k}\right] + \frac{\rho}{k-\rho}\left[\rho - (k-1)\right]\frac{C_S^2 - 1}{2} + o(1)
\end{aligned}$$

Case: $\rho = k - 1$
The critical case $\rho = k - 1$ is difficult to handle directly. However, we can infer the limit

$$\lim_{\epsilon \to 0} \mathbf{E}\left[N^{M/H_2^{(\epsilon)}/k}\right] = \mathbf{E}\left[N^{M/M/k}\right]$$

from the preceding analysis to obtain upper bounds for the cases $\rho < k - 1$ and $\rho > k - 1$, and the matching lower bounds obtained via analysis of the system described in Section 2.5.6 as follows. For each ϵ, let $f^{(\epsilon)} : [0, k) \to \mathbb{R}_0^+$ denote the function mapping the load ρ to the mean number of jobs in an $M/H_2^{(\epsilon)}/k$ system, $\mathbf{E}\left[N^{M/H^{(\epsilon)}/k}\right]$. Let $f(\cdot)$ be the point-wise limit of $f^{(\epsilon)}(\cdot)$ as $\epsilon \to 0$. Since each $f^{(\epsilon)}$ is a monotonic function, f is also monotonic. Further,

$$\lim_{\rho \uparrow k-1} f(\rho) = \lim_{\rho \downarrow k-1} f(\rho) = \mathbf{E}\left[N^{M/M/k}\right].$$

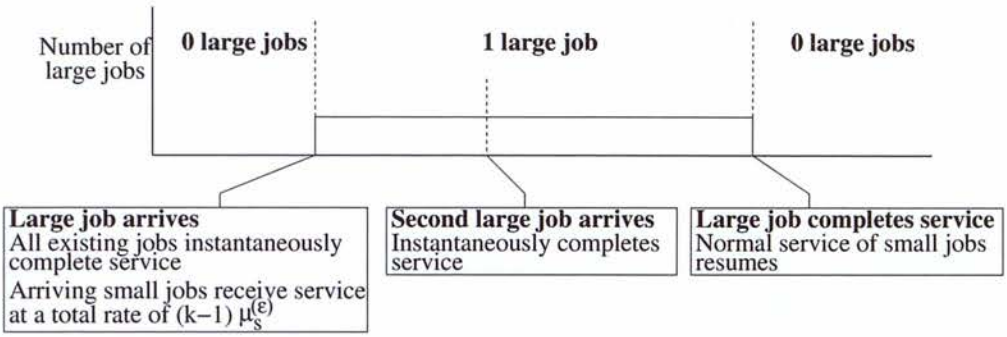


Figure 2.6: Construction of system $L^{(\epsilon)}$ which lower bounds the number of jobs in an $M/H_2^{(\epsilon)}/k$

Thus we conclude,

$$f(k-1) = \lim_{\epsilon \to 0} \mathbf{E}\Big[N^{M/H_2^{(\epsilon)}/k}\Big]\Big|_{\rho=k-1} = \mathbf{E}\Big[N^{M/M/k}\Big].$$

2.5.6 Construction of $L^{(\epsilon)}$: the lower bounding system

Case: $\rho > k-1$

Figure 2.6 shows the behavior of system $L^{(\epsilon)}$ for this case. As before, denote the periods where there are no large jobs in the system as *good* periods, and periods when there is at least 1 large job as *bad* periods. During a good period, the small jobs receive service according to a normal k server FIFO system. As soon as a large job arrives to begin the bad period, all the small jobs currently in the system instantaneously complete service. That is, the system restarts with 1 large job. Any large jobs that arrive during this bad period complete service instantaneously. Further, whenever there are fewer than $(k-1)$ small jobs in the system during a bad period, they are collectively served at a total rate of $(k-1)\mu_s^{(\epsilon)}$.

Case: $\rho \leq k-1$

For this case we can consider an alternate lower bounding system which simplifies the analysis. In the lower bounding system, $L^{(\epsilon)}$, all large jobs instantaneously complete service on arrival. Thus the number of large jobs is always 0 and the number of small jobs behaves as in an $M/M/k$ with arrival rate $\lambda p^{(\epsilon)}$ and mean job size $\frac{1}{\mu_s^{(\epsilon)}}$.

Lemma 2.16 *The number of small jobs in an $M/H_2^{(\epsilon)}/k$ system, $N_s^{M/H_2^{(\epsilon)}/k}$, is stochastically lower bounded by the number of small jobs in the corresponding system*

$L^{(\epsilon)}$, $N_s^{L^{(\epsilon)}}$.

Proof: Straightforward using stochastic coupling. ∎

Sketch of Analysis of $L^{(\epsilon)}$

Case: $\rho > k-1$

The analysis of system $L^{(\epsilon)}$ is simplified because the large jobs form an $M/M/1/1$ system independent of the small jobs. The length of a bad period is distributed as $\text{Exp}\left(\mu_\ell^{(\epsilon)}\right)$ and the length of a good period is distributed as $\text{Exp}\left(\lambda(1-p^{(\epsilon)})\right)$. Further, during a bad period, the number of small jobs behaves as in an $M/M/1$ queue with arrival rate $\lambda p^{(\epsilon)}$ and service rate $(k-1)\mu_s^{(\epsilon)}$ starting with an empty system. Therefore, the distribution of the number of small jobs at the end of bad periods (and hence, by conditional PASTA, the distribution of the time average number of small jobs during the bad periods) in system $L^{(\epsilon)}$ can be derived along the lines of proof of Lemma 2.12. To complete the proof we need to find the stationary mean number of small jobs at the end of good periods (and hence, by conditional PASTA, the stationary mean number of small jobs during the good periods). This is equivalent to finding the mean number of jobs in an $M/M/k$ at time $T \sim \text{Exp}\,(\lambda(1-p))$, starting at $t=0$ with number of jobs sampled from the distribution of the number of small jobs at the end of bad periods. To do this, we start with Eqn. (2.50), proceed as in the proof of Lemma 2.13 by finding the root of the denominator in the interval $[0,1)$ and equating the numerator to zero at this root. We then follow the proof of Lemma 2.15 to obtain the mean number of jobs at time T.

Case: $\rho \leq k-1$

As stated earlier, in constructing the lower bound system $L^{(\epsilon)}$, we assume that the large jobs complete service instantaneously on arrival. Therefore, the number of large jobs in the system is 0 with probability 1. The distribution of the time average number of small jobs in the system is given by the stationary distribution in an $M/M/k$ FCFS system.

2.6 Effect of higher moments

In Theorems 2.1 and 2.2, we proved that the first two moments of the service distribution alone are insufficient to approximate the mean waiting time accurately. In Section 2.3, by means of numerical experiments, we observed that within the H_2 class of distributions, the normalized third moment of the service distribution has a significant impact on the mean waiting time. Further, we observed that for H_2 service distributions, increasing the normalized third moment causes the mean waiting

time to drop. It is, therefore, only natural to ask the following questions: Are three moments of the service distribution sufficient to accurately approximate the mean waiting time, or do even higher moments have an equally significant impact? Is the qualitative effect of 4th and higher moments similar to the effect of the 3rd moment or is it the opposite? In this section, we touch upon these interesting and largely open questions.

	$C_S^2 = 19$		$C_S^2 = 99$	
	$\mathbf{E}[W]$	θ_3	$\mathbf{E}[W]$	θ_3
2-moment approx. (Eqn. 2.1)	6.6873	-	33.4366	-
Weibull	6.0691	4.2	25.9896	8.18
Truncated Pareto ($\alpha = 1.1$)	5.5241	4.24	24.5788	6.30
Lognormal	4.9937	20	19.5548	100
Truncated Pareto ($\alpha = 1.3$)	4.8770	7.59	18.8933	16.85
Truncated Pareto ($\alpha = 1.5$)	3.9504	20	10.5404	100

Table 2.2: Results from simulating an $M/G/k$ with $k = 10$ and $\rho = 9$ (confidence intervals omitted). All service distributions have $\mathbf{E}[S] = 1$.

	$C_S^2 = 19$		$C_S^2 = 99$	
	$\mathbf{E}[W]$	θ_3	$\mathbf{E}[W]$	θ_3
2-moment approx. (Eqn. 2.1)	0.2532	-	1.2662	-
Weibull	0.1374	4.2	0.4638	8.18
Truncated Pareto ($\alpha = 1.1$)	0.0815	4.24	0.2057	6.30
Lognormal	0.0854	20	0.2154	100
Truncated Pareto ($\alpha = 1.3$)	0.0538	7.59	0.0816	16.85
Truncated Pareto ($\alpha = 1.5$)	0.0355	20	0.0377	100

Table 2.3: Results from simulating an $M/G/k$ with $k = 10$ and $\rho = 6$ (confidence intervals omitted). All service distributions have $\mathbf{E}[S] = 1$.

We first revisit the simulation results of Table 2.1. Table 2.2 shows the simulation results of Table 2.1 again, but with an additional column – the normalized third moment of the service distribution. We have omitted the confidence intervals in Table 2.2. Observe that the lognormal distribution and the Pareto distribution with $\alpha = 1.5$ have *identical first three moments*, yet exhibit very different mean waiting times. This behavior is compounded when the system load is reduced to $\rho = 6$ (Table 2.3). As we saw in Section 2.3, the disagreement in the mean waiting time for the lognormal and the truncated Pareto distribution can be partly explained by the

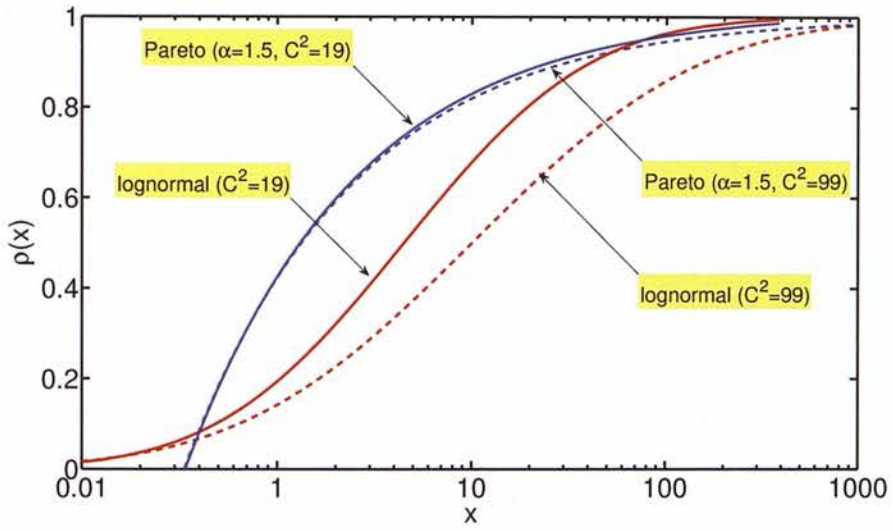
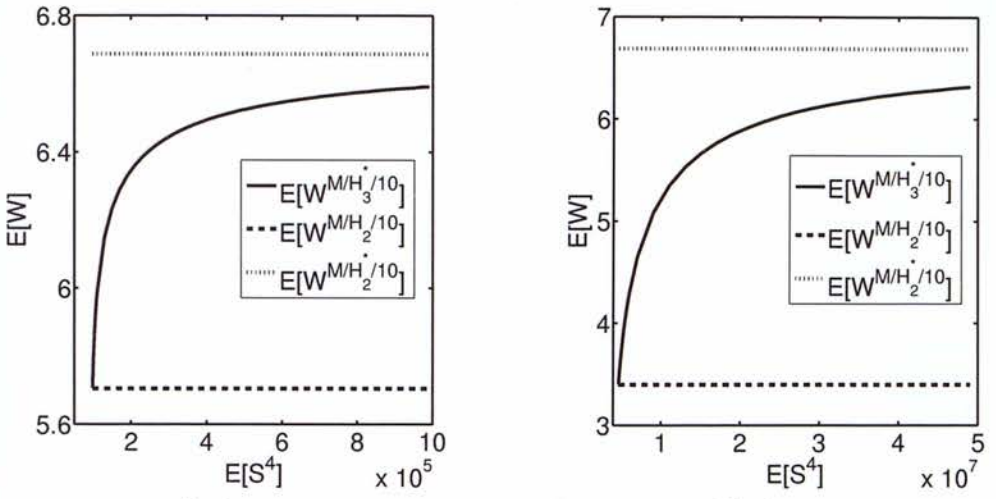


Figure 2.7: The distribution of load as a function of job size for the lognormal and bounded Pareto ($\alpha = 1.5$) distributions for two values of squared coefficient of variation. Although the lognormal and Pareto distributions have identical first three moments, the distribution of load among different job sizes is drastically different.

very different looking $\rho(x)$ curves for these distributions, shown in Figure 2.7. The bulk of the load in the lognormal distribution is comprised of larger jobs as compared to the truncated Pareto distribution.

The example of lognormal and Pareto ($\alpha = 1.5$) distributions suggests that even knowledge of three moments of the service distribution may not be sufficient for accurately approximating the mean waiting time. *So what is the effect of higher moments on the mean waiting time?* To begin answering this question, we will follow a similar approach as in Section 2.3 where we looked at the H_2 service distribution. However, we first need to expand the class of service distributions to allow us control over the 4th moment. For this purpose, we choose the *3-phase degenerate hyperexponential* class of distribution, denoted by H_3^*. Analogous to the H_2^* distribution, H_3^* is the class of mixtures of three exponential distributions where the mean of one of the phases is 0 (see Definition 2.2). Compared to the H_2 class, the H_3^* class has one more parameter and thus four degrees of freedom, which allows us control over the 4th moment while holding the first three moments fixed.

We now extend the numerical results of Figure 2.2 by considering service distributions in the H_3^* class with the same mean and SCV as the example illustrated in Figure 2.2.



(a) $\theta_3 = 3$ (b) $\theta_3 = 20$

Figure 2.8: Illustration of the effect of 4th moment of the service distribution on mean waiting time of an $M/H_3^*/10$ system for two values of the normalized third moment. Dashed line shows the mean waiting time under an H_2 service distribution with the same first three moments and the light dotted line shows the mean waiting time under an H_2^* service distribution with the same first two moments as the H_3^* distribution. The mean and squared coefficient of variation of the service distribution were held constant at $\mathbf{E}[S] = 1$ and $C_S^2 = 19$ with load $\rho = 9$ (same as Figure 2.2).

To demonstrate the effect of the 4th moment, we choose two values of θ_3 and plot the $\mathbf{E}[W]$ curves as a function of the 4th moment in Figure 2.8. As a frame of reference, we also show the mean waiting time under the H_2 service distribution (with the same first three moments as H_3^*) and that under H_2^* distribution (with the same first two moments as H_3^*).

As is evident from Figure 2.8, the fourth moment can have as significant an impact on the mean waiting time as the third moment. As the 4th moment is increased, the mean waiting time increases from $\mathbf{E}\left[W^{M/H_2/k}\right]$ to $\mathbf{E}\left[W^{M/H_2^*/k}\right]$. Therefore, the qualitative effect of the 4th moment is *opposite* to that of the third moment.

The effect of the fourth moment also helps explain the disagreement between the mean waiting time for the lognormal, the truncated Pareto ($\alpha = 1.5$) and the H_2 distributions. For the case $C_S^2 = 19$, the lognormal distribution has a much higher 4th moment ($\mathbf{E}\left[S^4\right] = 64 \times 10^6$) than the Pareto ($\mathbf{E}\left[S^4\right] = 5.66 \times 10^6$) and the H_2 ($\mathbf{E}\left[S^4\right] = 4.67 \times 10^6$) distribution with $\theta_3 = 20$. While this is a possible cause for

a higher mean waiting time under the lognormal distribution, there is still disagreement between the mean waiting time under the lognormal distribution and the H_3^* distribution (see Figure 2.8) with the same first 4 moments, indicating that even higher moments are playing an important role as well! In the next chapter, we will conjecture and provide analytical and simulation evidence for sharp bounds on the mean waiting time in terms of moments of the service distribution.

In conclusion, by looking at a range of distributions including hyperexponential, Pareto and lognormal distributions, we see that the moments of the service distribution may not be sufficient to accurately predict the mean waiting time. Further, for distributions such as the lognormal distribution which are not uniquely determined by their moments, no finite number of moments may suffice. Other characteristics, such as the distribution of load among the small and large job sizes, may lead to more accurate approximations.

2.7 Summary and Open Questions

In this chapter, we addressed the classical problem of approximating the mean waiting time of an $M/G/k$ queueing system, which has been at the heart of queueing theory since its beginnings, and is the key hurdle in answering the question: What is the minimum number of servers necessary for a given QoS guarantee. In the absence of exact analysis, most work in literature on this problem has proposed approximations which are functions of at most the first two moments of the service distribution. As the major contribution of this chapter, we proved that it is impossible to develop any approximation based on only the first two moments of the service distribution that is accurate for all service distributions. Specifically, we proved that specifying the first two moments of the service distribution insufficiently limits the range of possible values of mean waiting time: The maximum value of this range can be as much as $(C_S^2 + 1)$ times the minimum value. We also conjecture that the bounds derived in this chapter are sharp given the first two moments.

We will continue to explore the question of moments-based bounds in Chapter 3.

Impact: The major contribution of this chapter has been to deepen the understanding of the $M/G/k$ queueing system which we believe would lead to tighter analysis, and hence efficiently provisioned server farms. In addition, we also advocate a new perspective for developing approximations for queueing systems: by exploring the worst case deviation of the performance of a queueing system from the proposed approximation – a perspective that is ubiquitous in the theoretical computer science community but not well-assimilated in the stochastic analysis community.

Open Problems: What partial characterizations of the service distribution (i.e., independent of k and ρ) are representative for the purpose of approximating $M/G/k$ mean waiting time? Our experiments suggest that *moments* are not the ideal job size characteristic on which to base approximations for mean waiting time. The moment sequence *can* be useful if one of the moments (appropriately normalized) is small. As an example, if the service distribution has a small normalized third moment, then an approximation based on only the first two moments is likely to be accurate. However, there are also many service distributions, e.g., the lognormal distribution (whose moments are all high), for which moments are not useful in accurately predicting mean waiting time.

2.A Proofs

Proposition 2.1 *Let* $\mathbf{E}\left[W^{M/M/k}\right]$ *be the mean waiting time in an* $M/M/k$ *with mean job size* 1. *For all values of* $k \geq 2$, $\rho \in [k-1, k)$ *and* $C_S^2 > 1$,

$$\left(\frac{C_S^2+1}{2}\right)\mathbf{E}\left[W^{M/M/k}\right] > \mathbf{E}\left[W^{M/M/k}\right] + \left[\frac{\rho-(k-1)}{k-\rho}\right]\frac{C_S^2-1}{2}.$$

Proof: Our aim is to prove that for $k \geq 2$, $\rho \geq k-1$ and $C_S^2 > 1$

$$\frac{C_S^2-1}{2}\mathbf{E}\left[W^{M/M/k}\right] > \left[\frac{\rho-(k-1)}{k-\rho}\right]\frac{C_S^2-1}{2}. \tag{2.14}$$

Recall that we take $\mathbf{E}[S] = 1$ without loss of generality so that $\rho \geq k-1$ is equivalent to $\lambda \geq k-1$. Let $C(k,\lambda)$ be the probability of wait in an $M/M/k$. It is easily shown that

$$\mathbf{E}\left[W^{M/M/k}\right] = \frac{C(k,\lambda)}{k-\lambda}. \tag{2.15}$$

Therefore, using $\rho = \lambda$, (2.14) holds if (we have assumed $\frac{C_S^2-1}{2} > 0$)

$$C(k,\lambda) > [\lambda - (k-1)]. \tag{2.16}$$

It is known that $C(k,\lambda)$ is a strictly convex function in λ on $[0,k]$ (see [109]). Since (2.16) trivially holds for $\lambda = k-1$, and since the right hand side of (2.16) has derivative (w.r.t. λ) 1, it suffices to show that

$$\frac{d}{d\lambda}C(k,\lambda)\bigg|_{\lambda\to k} < 1. \tag{2.17}$$

Let A_λ be a random variable that is Poisson with mean λ. It is well known ([103], page 103) that

$$C(k,\lambda) = \frac{1}{\frac{\rho}{k} + \left(1 - \frac{\rho}{k}\right)\frac{P(A_\lambda \leq k)}{P(A_\lambda = k)}}. \tag{2.18}$$

Using this expression, we find that

$$\begin{aligned}
\frac{d}{d\lambda}C(k,\lambda)\bigg|_{\lambda\to k} &= \frac{d}{d\lambda}\frac{1}{\frac{\lambda}{k} + \left(1-\frac{\lambda}{k}\right)\frac{P(A_\lambda\leq k)}{P(A_\lambda = k)}}\bigg|_{\lambda\to k} \\
&= -\frac{\frac{1}{k} - \frac{1}{k}\frac{P(A_\lambda\leq k)}{P(A_\lambda=k)} + \left(1-\frac{\lambda}{k}\right)\frac{d}{d\lambda}\frac{P(A_\lambda\leq k)}{P(A_\lambda=k)}}{\left(\frac{\lambda}{k} + \left(1-\frac{\lambda}{k}\right)\frac{P(A_\lambda\leq k)}{P(A_\lambda=k)}\right)^2}\bigg|_{\lambda\to k} \\
&= \frac{1}{k}\frac{P(A_k \leq k-1)}{P(A_k = k)} \\
&= \frac{1}{k}\sum_{i=0}^{k-1}\frac{P(A_k = i)}{P(A_k = k)}.
\end{aligned}$$

Now, note that at $\lambda = k$

$$\frac{P(A_k = k-1)}{P(A_k = k)} = \frac{k^{k-1}/(k-1)!}{k^k/k!} = 1.$$

If $i < k-1$ we find that

$$\frac{P(A_k = i)}{P(A_k = i+1)} = \frac{i+1}{k} < 1,$$

which implies that

$$\frac{P(A_k = i)}{P(A_k = i+1)} < 1, \quad i < k-1.$$

Consequently, for $k \geq 2$, we see that

$$\frac{d}{d\lambda}C(k,\lambda)\bigg|_{\lambda\to k} = \frac{1}{k}\sum_{i=0}^{k-1}\frac{k^i/i!}{k^k/k!} < 1, \tag{2.19}$$

which completes the proof of the proposition. ∎

Proof of Lemma 2.5: Suppressing the superscript, we have the following equations from Definition 2.3:

$$\frac{p}{\mu_s} + \frac{1-p}{\mu_\ell} = 1 \tag{2.20}$$

$$\frac{p}{\mu_s^2} + \frac{1-p}{\mu_\ell^2} = \frac{C_S^2+1}{2} \tag{2.21}$$

$$\frac{p}{\mu_s^3} + \frac{1-p}{\mu_\ell^3} = \frac{1}{6\epsilon} \tag{2.22}$$

Performing (2.21) − (2.20) × (2.20):

$$p(1-p)\left(\frac{1}{\mu_s} - \frac{1}{\mu_\ell}\right)^2 = \frac{C_S^2-1}{2} \tag{2.23}$$

Performing (2.22) × (2.20) − (2.21) × (2.21):

$$\frac{p(1-p)}{\mu_s\mu_\ell}\left(\frac{1}{\mu_s} - \frac{1}{\mu_\ell}\right)^2 = \frac{1}{6\epsilon} - \frac{(C_S^2+1)^2}{4} \tag{2.24}$$

The above two equations give:

$$\mu_s\mu_l = \frac{\frac{C_S^2-1}{2}}{\frac{1}{6\epsilon} - \frac{(C_S^2+1)^2}{4}} \tag{2.25}$$

From equations (2.20) and (2.21),

$$p(\mu_\ell - \mu_s) = \mu_s\mu_\ell - \mu_s$$

$$p(\mu_\ell^2 - \mu_s^2) + \mu_s^2 = \frac{C_S^2+1}{2}(\mu_s\mu_\ell)^2$$

Substituting $p(\mu_\ell - \mu_s)$ as $\mu_s\mu_\ell - \mu_s$ in the second equation gives:

$$\begin{aligned}\mu_s + \mu_\ell &= 1 + \frac{C_S^2+1}{2}\mu_s\mu_\ell \\ &= 1 + \frac{\frac{C_S^2+1}{2}\cdot\frac{C_S^2-1}{2}}{\frac{1}{6\epsilon} - \frac{(C_S^2+1)^2}{4}}\end{aligned}$$

Finally,

$$\begin{aligned}\mu_s\mu_\ell &= \frac{\frac{C_S^2-1}{2}}{\frac{1}{6\epsilon} - \frac{(C_S^2+1)^2}{4}} = 3(C_S^2-1)\epsilon\left(1 - \frac{3(C_S^2+1)^2}{2}\epsilon\right)^{-1} \\ &= 3(C_S^2-1)\epsilon\left[1 + \frac{3}{2}(C_S^2+1)^2\epsilon + \frac{9}{4}(C_S^2+1)^4\epsilon^2 + \Theta(\epsilon^3)\right] \\ \mu_s + \mu_\ell &= 1 + \frac{\frac{C_S^2+1}{2}\cdot\frac{C_S^2-1}{2}}{\frac{1}{6\epsilon} - \frac{(C_S^2+1)^2}{4}} = 1 + \frac{3}{2}(C_S^2+1)(C_S^2-1)\epsilon\left(1 - \frac{3(C_S^2+1)^2}{2}\epsilon\right)^{-1}\end{aligned}$$

$$= 1 + \frac{3}{2}(C_S^2+1)(C_S^2-1)\epsilon\left[1 + \frac{3}{2}(C_S^2+1)^2\epsilon + \frac{9}{4}(C_S^2+1)^4\epsilon^2 + \Theta(\epsilon^3)\right]$$

It is straightforward to verify that the expressions for μ_s and μ_ℓ in Lemma 2.5 satisfy the above equations. The expression for p then follows from $p = 1 - \mu_\ell \frac{\mu_s - 1}{\mu_s - \mu_l}$. ∎

Proof of Lemma 2.7: Recall that $\overline{N_\ell}^{(\epsilon)}$ is defined to be the steady-state number of customers in an $M\left(\lambda(1-p^{(\epsilon)})\right)/M\left(\mu_\ell^{(\epsilon)}\right)/1$ queue with service interruptions where the server is interrupted for the duration of the busy period of an $M(\lambda)/M(1)/k$ queue. The busy period of an $M(\lambda)/M(1)/k$ queue has finite second moment [145], and hence the second moment of the service interruptions is also finite. Let $B_{\lambda,1,k}$ be the busy period of this queue. Define $\rho_\ell^{(\epsilon)} = \lambda(1-p^{(\epsilon)})/\mu_\ell^{(\epsilon)}$.

Our aim is to prove:

$$\mathbf{E}\left[\overline{N_\ell}^{(\epsilon)}\right] = o(1)$$

The lemma follows by specializing results for the $M/G/1$ queue with server breakdowns to the special case considered here, see e.g. Adan & Resing [11, page 101]. For completeness, we provide a new proof of the $M/G/1$ queue with breakdowns by viewing it as a special case of an $M/G/1$ with setup times [143, page 130]. Let G be a so-called *generalized* service time, which is the service time of a large customer plus the total duration of service interruptions while that customer was in service. Let $\alpha = \lambda(1-p^{(\epsilon)})$ denote the arrival rate of the customers. The breakdowns (busy periods of the $M(\lambda)/M(1)/k$ queue) arrive at a rate λ when the system is "up", and let $\widetilde{B}_{\lambda,1,k}(s)$ denote the Laplace transform of the duration of these breakdowns. We can now view the $M(\alpha)/M(\mu_\ell^{(\epsilon)})/1$ queue with breakdowns as an $M/G/1$ queue with service distribution given by the generalized service time, G, and a setup time I at the beginning of each busy period, where the Laplace transform of I, $\widetilde{I}(s)$, satisfies:

$$\widetilde{I}(s) = \frac{\alpha}{\alpha+\lambda} + \frac{\lambda}{\alpha+\lambda}\cdot\widetilde{B}_{\lambda,1,k}(\alpha)\cdot\widetilde{I}(s) + \frac{\lambda}{\alpha+\lambda}\cdot\frac{\alpha}{\alpha-s}\left(\widetilde{B}_{\lambda,1,k}(s) - \widetilde{B}_{\lambda,1,k}(\alpha)\right) \tag{2.26}$$

In the above equation, the first term denotes the event that the customer arrives before the breakdown, the second term denotes the event that the breakdown arrives before the customer, but no customers arrive during this breakdown, and the third term denotes the event that the breakdown arrives before the customer and a customer arrives during this breakdown. By differentiating (2.26) with respect to s once and twice, and evaluating at $s = 0$, the first two moments of I are obtained,

respectively, as:

$$\mathbf{E}[I] = \left(\frac{\lambda}{\alpha+\lambda}\right) \cdot \frac{\mathbf{E}[B_{1,\lambda,k}] - \frac{1-\widetilde{B}_{1,\lambda,k}(\alpha)}{\alpha}}{1 - \frac{\lambda}{\alpha+\lambda} \cdot \widetilde{B}_{1,\lambda,k}(\alpha)} \tag{2.27}$$

$$\mathbf{E}\left[I^2\right] = \left(\frac{\lambda}{\alpha+\lambda}\right) \frac{\mathbf{E}\left[B^2_{1,\lambda,k}\right] - 2\frac{\mathbf{E}\left[B_{1,\lambda,k}\right]}{\alpha} + 2\frac{1-\widetilde{B}_{1,\lambda,k}(\alpha)}{\alpha^2}}{1 - \frac{\lambda}{\alpha+\lambda} \cdot \widetilde{B}_{1,\lambda,k}(\alpha)} \tag{2.28}$$

Define $\overline{V_\ell}^{(\epsilon)}$ to be the system time (response time) of large customers in the modified queue. From [143, page 130], we get

$$\begin{aligned}\mathbf{E}\left[\overline{V_\ell}^{(\epsilon)}\right] &= \mathbf{E}[G] + \left(\frac{\rho_G}{1-\rho_G}\right)\frac{\mathbf{E}[G^2]}{2\mathbf{E}[G]} + \frac{2\mathbf{E}[I] + \alpha\mathbf{E}[I^2]}{2(1+\alpha\mathbf{E}[I])} \\ &= \mathbf{E}[G] + \left(\frac{\rho_G}{1-\rho_G}\right)\frac{\mathbf{E}[G^2]}{2\mathbf{E}[G]} + \left(\frac{\lambda\mathbf{E}[B_{\lambda,1,k}]}{1+\lambda\mathbf{E}[B_{\lambda,1,k}]}\right)\frac{\mathbf{E}\left[B^2_{\lambda,1,k}\right]}{2\mathbf{E}[B_{\lambda,1,k}]}.\end{aligned} \tag{2.29}$$

Here $\rho_G = \rho_\ell^{(\epsilon)}(1 + \mathbf{E}[B_{\lambda,1,k}]/\lambda)$. The first two moments of G are given by

$$\mathbf{E}[G] = \frac{1}{\mu_\ell^{(\epsilon)}}\left(1 + \frac{\mathbf{E}[B_{\lambda,1,k}]}{\lambda}\right) \tag{2.30}$$

and that

$$\mathbf{E}\left[G^2\right] = \frac{2}{\left(\mu_\ell^{(\epsilon)}\right)^2}\left(1 + \frac{\mathbf{E}[B_{\lambda,1,k}]}{\lambda}\right)^2 + \frac{1}{\mu_\ell^{(\epsilon)}}\lambda\mathbf{E}\left[B^2_{\lambda,1,k}\right]. \tag{2.31}$$

From these equations, it follows that $\mathbf{E}[G] = \Theta(1/\epsilon)$ and $\mathbf{E}[G^2] = \Theta(1/\epsilon^2)$. This implies $\mathbf{E}\left[\overline{V_\ell}^{(\epsilon)}\right] = \Theta(1/\epsilon)$. By Little's law, $\mathbf{E}\left[\overline{N_\ell}^{(\epsilon)}\right] = \lambda(1-p^{(\epsilon)})\mathbf{E}\left[\overline{V_\ell}^{(\epsilon)}\right]$, which implies $\mathbf{E}\left[\overline{N_\ell}^{(\epsilon)}\right] = \Theta(\epsilon)$. ∎

Proof of Lemma 2.9: Consider a further modification of system $U^{(\epsilon)}$ where the small jobs are not served during the entire bad period. That is, even when there is only a single large job in the system, we stop serving small jobs. The fraction of time this modified system $U^{(\epsilon)}$ is busy with large jobs is given by $\lambda\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}} = \rho\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}}$. The load of the small jobs is less than ρ/k. Thus, system $U^{(\epsilon)}$ will be stable if $\frac{\rho}{k} < 1 - \rho\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}}$.

Since $p^{(\epsilon)} \leq 1$ and $\mu_s^{(\epsilon)} \geq 1$, we have

$$\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^2} \leq \frac{C_S^2+1}{2}$$

$$\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^3} \geq \frac{1}{6\epsilon} - 1$$

Now,

$$\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}} = \frac{\left(\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^2}\right)^2}{\frac{1-p^{(\epsilon)}}{\left(\mu_\ell^{(\epsilon)}\right)^3}} \leq \frac{\left(\frac{C_S^2+1}{2}\right)^2}{\frac{1}{6\epsilon}-1}$$

Thus,

$$\epsilon < \frac{1}{6}\left[\frac{\rho(C_S^2+1)^2}{4(1-\rho/k)} + 1\right]^{-1}$$

$$\implies \rho\frac{\left(\frac{C_S^2+1}{2}\right)^2}{\frac{1}{6\epsilon}-1} < 1 - \frac{\rho}{k}$$

$$\implies \rho\frac{1-p^{(\epsilon)}}{\mu_\ell^{(\epsilon)}} < 1 - \frac{\rho}{k}$$

■

Proof of Lemma 2.10: Recall that $\Phi(A)$ was defined as the mapping between non-negative random variables where $\Phi(A)$ gives the random variable for the number of jobs at the end of a good period given that the number at the beginning of the good period is A. Let $\Psi(A)$ be another mapping between random variables defined by:

$$\Psi(A) = \Delta_{b''} + \sum_{i=0}^{\infty}(i + \Delta_{b'}(i))\mathbf{I}_{\{A=i\}}$$

That is, $\Psi(A)$ gives the number of small jobs at the end of a bad period given that the number at the start is A. Further, the following facts can be easily verified via coupling:

1. $A_1 \leq_{st} A_2 \implies \Phi(A_1) \leq_{st} \Phi(A_2)$
2. $\Delta_{b'}(0) \geq_{st} \Delta_{b'}(1) \geq_{st} \ldots \Delta_{b'}(i) \geq \Delta_{b'}(i+1) \geq \ldots$

The last fact implies $\Psi(A) \leq_{st} A + \Delta_{b'}(0) + \Delta_{b''} \stackrel{def}{=} A + \Delta_b$. This gives us a way to stochastically upper bound $N_{s,g}^*$. We defined $\bar{N}_{s,g}^*$ to be the solution to the following

fixed point equation:

$$\bar{N}^*_{s,g} \stackrel{d}{=} \Phi(\bar{N}^*_{s,g} + \Delta_b)$$

Also,

$$N^*_{s,g} \stackrel{d}{=} \Phi(\Psi(N^*_{s,g}))$$

Let $Y(0) = \bar{Y}(0) = 0$. Further, let $Y(n+1) = \Phi(\Psi(Y(n)))$ and $\bar{Y}(n+1) = \Phi(\bar{Y}(n) + \Delta_b)$. Since the Markov chains defined by the transition functions $\Phi(\Psi(\cdot))$ and $\Phi(\cdot + \Delta_b)$ are positive recurrent (we proved system $U^{(\epsilon)}$ stable for $\epsilon < \epsilon'$ but the proof implies the stability of this system as well) and irreducible,

$$N^*_{s,g} = \lim_{n\to\infty} Y(n)$$
$$\bar{N}^*_{s,g} = \lim_{n\to\infty} \bar{Y}(n)$$

Since $Y(n) \leq_{st} \bar{Y}(n)$ for all n by induction, $N^*_{s,g} \leq_{st} \bar{N}^*_{s,g}$. ∎

Proof of Lemma 2.11: Let $N_{s,b'}$ denote the number of small jobs during the bad′ phase and $N_{s,b''}$ denote the number of jobs during the bad″ phase. We will stochastically bound $N_{s,b'}$ and $N_{s,b''}$ separately using stochastic coupling.

Bound for $N_{s,b'}$: We know that the lengths of bad′ phases of system $U^{(\epsilon)}$ are i.i.d. random variables. Let $T_{b'}$ denote a random variable which is equal in distribution to these. It is easy to see that $N_{s,b'}$ is equal in distribution to the number of small jobs in the following regenerative process. The system regenerates after i.i.d. periods whose lengths are equal in distribution to $T_{b'}$. At each regeneration the system starts with a random number of small jobs sampled from the distribution of $N^*_{s,g}$ and then the system evolves as an $M/M/k-1$ with arrival rate λp and service rate μ_s until the next renewal.

Now, $N_{s,b'}$ can be stochastically upper bounded by the number in system in another regenerative process where the renewals happen in the same manner but at every renewal the system starts with a random number of jobs sampled from the distribution of $\bar{N}^*_{s,g}$. These jobs never receive service. However, we also start another $M/M/k-1$ from origin (initially empty) with arrival rate λp and service rate μ_s and look at the total number of small jobs.

Finally, since $T_{b'}$ is an exponential random variable, by PASTA, the distribution of number of jobs at a randomly chosen time (or as $t \to \infty$) is the same as the number of jobs at a random chosen renewal. Therefore,

$$N_{s,b'} \leq_{st} \bar{N}^*_{s,g} + \Delta_{b'}(0) \tag{2.32}$$

Bound for $N_{s,b''}$: To obtain a stochastic upper bound on $N_{s,b''}$, we follow the same procedure as above. It is easy to see that $N_{s,b''}$ is stochastically upper bounded by the number of jobs in the following regenerative system. The renewals happen after i.i.d. intervals which are equal in distribution to $T_{b''}$, the random variable for the length of a bad'' phase in system $U^{(\epsilon)}$. At every renewal, the system starts with a random number of jobs sampled from the distribution of $\bar{N}^*_{s,g} + \Delta_{b'}(0)$ and external arrivals happen at a rate λ (there are no departures) until the next renewal. Let $T_{b''e}$ denote the age (and equal in distribution to the excess) of $T_{b''}$ and $A_\lambda(T)$ denote the number of arrivals in time T of a Poisson process with rate λ. This gives us the following stochastic bound on $N_{s,b''}$,

$$N_{s,b''} \leq_{st} \bar{N}^*_{s,g} + \Delta_{b'}(0) + A_\lambda\left(T_{b''e}\right) \tag{2.33}$$

The excess of $T_{b''}$ comes into the picture because we need the number of jobs at a randomly chosen instant of time during the bad'' phase. The time elapsed since the starting of a bad'' phase until this randomly chosen instant of time is distributed as $T_{b''e}$, the excess of $T_{b''}$. Finally, combining (2.32) and (2.33),

$$N_{s,b} \leq_{st} \bar{N}^*_{s,g} + \Delta_{b'}(0) + \mathbf{I}_{b''|b} A_\lambda\left(T_{b''e}\right) \tag{2.34}$$

■

Proof of Lemma 2.13: The z-transform of $N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)$ is given by [69, Theorem 4]:

$$\widehat{N}^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)(z) = \frac{\beta z - (k-1)\mu_s(1-z)p_0}{\beta z - ((k-1)\mu_s - \lambda_s z)(1-z)} \tag{2.35}$$

where,

$$p_0 = \frac{\beta\xi}{(k-1)\mu_s(1-\xi)}$$

and ξ is the root of the polynomial in the denominator of $\widehat{N}^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)(z)$ in the interval $(0,1)$. Let η be the other root (lying in $(1,\infty)$). Therefore, we can write (2.35) as,

$$\begin{aligned}\widehat{N}^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)(z) &= \frac{\beta z - (k-1)\mu_s(1-z)\frac{\beta\xi}{(k-1)\mu_s(1-\xi)}}{-\lambda_s(z-\xi)(z-\eta)} \\ &= \frac{\beta}{-\lambda_s(1-\xi)(z-\eta)}\end{aligned}$$

$$= \frac{1-\eta}{z-\eta} \tag{2.36}$$

The last step follows since $\widehat{N}^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)(z)|_{z=1} = 1$. By differentiating the transform in (2.36) and evaluating the derivatives at $z = 1$, we have

$$\mathbf{E}\left[N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T)\right] = \frac{1}{\eta - 1}$$
$$\mathbf{E}\left[(N^{M(\lambda_s)/M((k-1)\mu_s)/1}(T))^2\right] = \frac{2}{(\eta-1)^2} + \frac{1}{\eta-1}$$

Factoring the denominator of (2.35), we can write η as the larger root of the quadratic equation:

$$z^2\lambda_s - z(\lambda_s + \beta + (k-1)\mu_s) + (k-1)\mu_s$$

That is,

$$\begin{aligned}
\eta &= \frac{\lambda_s + \beta + (k-1)\mu_s + \sqrt{(\lambda_s + \beta + (k-1)\mu_s)^2 - 4\lambda_s(k-1)\mu_s}}{2\lambda_s} \\
&= \frac{\lambda_s + \beta + (k-1)\mu_s + \sqrt{(\lambda_s + \beta - (k-1)\mu_s)^2 + 4\beta(k-1)\mu_s}}{2\lambda_s} \\
&= \frac{\lambda_s + \beta + (k-1)\mu_s + (\lambda_s + \beta - (k-1)\mu_s)\sqrt{1 + 4\frac{\beta(k-1)\mu_s}{(\lambda_s+\beta-(k-1)\mu_s)^2}}}{2\lambda_s} \\
&= \frac{\lambda_s + \beta + (k-1)\mu_s + (\lambda_s + \beta - (k-1)\mu_s)\left(1 + 2\frac{\beta(k-1)\mu_s}{(\lambda_s+\beta-(k-1)\mu_s)^2} + \Theta(\beta^2)\right)}{2\lambda_s} \\
&= 1 + \frac{\beta}{\lambda_s} \cdot \left(1 + \frac{(k-1)\mu_s}{(\lambda_s + \beta - (k-1)\mu_s)}\right) + \Theta(\beta^2) \\
&= 1 + \frac{\beta}{\lambda_s - (k-1)\mu_s} + \Theta(\beta^2)
\end{aligned}$$

which results in the expressions in the lemma. ∎

Proof of Lemma 2.14: Recall that $\Delta_{b''}$ is the random variable denoting the number of small jobs that arrive during time $T_{b''}$, where $T_{b''}$ is the random variable for the length of the bad'' phase of a bad period. Using $A_\lambda(T)$ to denote the number of Poisson (with rate λ) arrivals in a random time interval T, we have $\Delta_{b''}$ is equal in distribution to $A_{\lambda_p}(T_{b''})$. The following equalities are easy to prove:

$$\mathbf{E}[A_\lambda(T)] = \lambda\mathbf{E}[T] \tag{2.37}$$

$$\mathbf{E}\left[(A_\lambda(T))^2\right] = \lambda^2\mathbf{E}\left[T^2\right] + \lambda\mathbf{E}[T] \tag{2.38}$$

Thus we need the first two moments of $T_{b''}$ to obtain the first two moments of $\Delta_{b''}$. The Laplace transform of $T_{b''}$, $\widetilde{T_{b''}}(s)$, is given by:

$$\widetilde{T_{b''}}(s) = \frac{\mu_\ell}{\mu_\ell + \lambda(1-p)} + \frac{\lambda(1-p)}{\mu_\ell + \lambda(1-p)}\widetilde{B}^2(s) \tag{2.39}$$

where $\widetilde{B}(s)$ is the Laplace transform for the length of busy periods of an $M/M/1$ with arrival rate $\lambda(1-p)$ and service rate μ_ℓ. To see this, note that with probability $\frac{\mu_\ell}{\mu_\ell+\lambda(1-p)}$, the large job starting the bad phase leaves before another large job arrives and thus bad$''$ phase has length 0. With probability $\frac{\lambda(1-p)}{\mu_\ell+\lambda(1-p)}$, a large job arrives and starts the bad$''$ phase. In this case, the length of the bad$''$ phase is the time for an $M/M/1$ with arrival rate $\lambda(1-p)$ and service rate μ_ℓ to become empty starting with 2 jobs in the system. This is just the sum of two independent $M/M/1$ busy periods.

By differentiating the transform in (2.39) and evaluating at $s = 0$, we obtain:

$$\mathbf{E}[T_{b''}] = \frac{\lambda(1-p)}{\mu_\ell + \lambda(1-p)}\left(\frac{2}{\mu_\ell - \lambda(1-p)}\right) = \Theta(1) \tag{2.40}$$

$$\mathbf{E}\left[T_{b''}^2\right] = \frac{\lambda(1-p)}{\mu_\ell + \lambda(1-p)}\left(\frac{4\mu_\ell}{(\mu_\ell - \lambda(1-p))^3}\right) = \Theta\left(\frac{1}{\epsilon}\right) \tag{2.41}$$

Obtaining $\mathbf{E}[\Delta_{b''}]$ and $\mathbf{E}[\Delta_{b''}^2]$: Substituting $\lambda \equiv \lambda p$ and $T \equiv T_{b''}$ in (2.37)-(2.38) and using (2.40)-(2.41), we get the following asymptotics which will be sufficient for our purposes:

$$\mathbf{E}[\Delta_{b''}] = \lambda p\mathbf{E}[T_{b''}] = \Theta(1) \tag{2.42}$$

$$\mathbf{E}\left[\Delta_{b''}^2\right] = \lambda^2 p^2\mathbf{E}\left[T_{b''}^2\right] + \lambda p\mathbf{E}[T_{b''}] = \Theta\left(\frac{1}{\epsilon}\right) \tag{2.43}$$

Obtaining $\mathbf{E}[A_\lambda(T_{b''e})]$: $A_\lambda(T_{b''e})$ denotes the number of Poisson (with rate λ) arrivals in a random time interval given by $T_{b''e}$ – the stationary age (equivalently excess) of a renewal process where renewals intervals are i.i.d. according to $T_{b''}$. Note that $A(T_{b''e})$ is not equal in distribution to $\Delta_{b''}$ since $T_{b''}$ is not an exponential random variable. From (2.37),

$$\mathbf{E}[A_\lambda(T_{b''e})] = \lambda\mathbf{E}[T_{b''e}]$$

From the formula for stationary age (equivalently excess) of a renewal process [131],

$$\mathbf{E}[T_{b''e}] = \frac{\mathbf{E}[T_{b''}^2]}{2\mathbf{E}[T_{b''}]} = \Theta\left(\frac{1}{\epsilon}\right)$$

Combining, we get the following asymptotics for $\mathbf{E}[A_\lambda(T_{b''e})]$ which will be sufficient for our purposes:

$$\mathbf{E}[A_\lambda(T_{b''e})] = \Theta\left(\frac{1}{\epsilon}\right) \tag{2.44}$$

$$\frac{\mathbf{Pr}[b'']}{\mathbf{Pr}[b]}\mathbf{E}[A_\lambda(T_{b''e})] = \frac{\mathbf{E}[T_{b''}]}{\left(\frac{1}{\mu_\ell - \lambda(1-p)}\right)}\mathbf{E}[A_\lambda(T_{b''e})] = \Theta(1) \tag{2.45}$$

■

Proof of Lemma 2.15: Recall that $N^{(Int)}$ denotes the number of jobs in the interrupted $M/M/k$ system. Let $\widehat{N^{(Int)}}(z)$ be the z-transform of $N^{(Int)}$ and let $\hat{\Delta}(z)$ be the z-transform of Δ. Since the interruptions happen according to a Poisson process, $N^{(Int)}$ also denotes the random variable for the number of jobs *just before* the interruptions. Let f map the z-transform of the distribution of number of jobs in an $M/M/k$ at time $t = 0$ to the z-transform of the distribution of number of jobs after the $M/M/k$ system has run (uninterrupted) for $T \sim \text{Exp}(\alpha)$ time. The solution for $\widehat{N^{(Int)}}(z)$ is given by the following fixed point equation:

$$\widehat{N^{(Int)}}(z) = f\left(\widehat{N^{(Int)}}(z)\hat{\Delta}(z)\right)$$

Our next goal is to derive the function $f(\cdot)$. Let $p_i(t)$ denote the probability that there are i jobs in the $M/M/k$ system at time t. We can write the following differential equations for $p_i(t)$:

$$\frac{d}{dt}p_0(t) = -\lambda p_0(t) + \mu p_1(t) \tag{2.46}$$

$$\frac{d}{dt}p_i(t) = \lambda p_{i-1}(t) - (\lambda + i\mu)p_i(t) + (i+1)\mu p_{i+1}(t) \quad \ldots 1 \le i \le k-1 \tag{2.47}$$

$$\frac{d}{dt}p_i(t) = \lambda p_{i-1}(t) - (\lambda + k\mu)p_i(t) + k\mu p_{i+1}(t) \quad \ldots i \ge k \tag{2.48}$$

Let $\hat{\Pi}(z,t) = \sum_{i=0}^{\infty} p_i(t)z^i$. Multiplying (2.46) by z^0 and the set of equations (2.47) and (2.48) by z^i and summing, we have:

$$\begin{aligned}\frac{\partial}{\partial t}\hat{\Pi}(z,t) = \hat{\Pi}(z,t)\left[k\mu\left(\frac{1}{z} - 1\right) + \lambda(z-1)\right] \\ + \mu\left(1 - \frac{1}{z}\right)\left[kp_0(t) + (k-1)zp_1(t) + \ldots + z^{k-1}p_{k-1}(t)\right]\end{aligned} \tag{2.49}$$

Let $\hat{\Pi}_\alpha(z) = \int_0^\infty \hat{\Pi}(z,t)\alpha e^{-\alpha t}dt$ and $p_{i,\alpha} = \int_0^\infty p_i(t)\alpha e^{-\alpha t}dt$. Integrating by parts, we get:

$$\hat{\Pi}_\alpha(z) = \int_0^\infty \hat{\Pi}(z,t)\alpha e^{-\alpha t}dt = \int_0^\infty \left(\hat{\Pi}(z,t)\right)\left(d\left(-e^{-\alpha t}\right)\right)$$

$$
\begin{aligned}
&= \left[-\widehat{\Pi}(z,t)e^{-\alpha t}\right]_{t=0}^{\infty} - \int_{t=0}^{\infty}\left(-e^{-\alpha t}\right)\left(d\widehat{\Pi}(z,t)\right) \\
&= \widehat{\Pi}(z,0) + \frac{1}{\alpha}\int_{t=0}^{\infty}\alpha e^{-\alpha t}\left(\widehat{\Pi}(z,t)\left[k\mu\left(\frac{1}{z}-1\right)+\lambda(z-1)\right]\right. \\
&\qquad \left. +\mu\left(1-\frac{1}{z}\right)\left[kp_0(t)+(k-1)zp_1(t)+\ldots+z^{k-1}p_{k-1}(t)\right]\right) \\
&= \widehat{\Pi}(z,0) + \frac{\widehat{\Pi}_\alpha(z)}{\alpha}\left[k\mu\left(\frac{1}{z}-1\right)+\lambda(z-1)\right] + \frac{\mu}{\alpha}\left(1-\frac{1}{z}\right)\left[kp_{0,\alpha}+\ldots+z^{k-1}p_{k-1,\alpha}\right]
\end{aligned}
\tag{2.50}
$$

To obtain $\widehat{N^{(Int)}}(z)$, we substitute $\widehat{\Pi}_\alpha(z) = \widehat{N^{(Int)}}(z)$, $\widehat{\Pi}(z,0) = \widehat{N^{(Int)}}(z)\widehat{\Delta}(z)$ and $p_{i,\alpha} = p_i = \mathbf{Pr}\left[N^{(Int)} = i\right]$. This gives:

$$
\widehat{N^{(Int)}}(z) = \frac{\mu\left[kp_0+(k-1)zp_1+\ldots+z^{k-1}p_{k-1}\right]}{(k\mu-\lambda z)-\alpha z\left(\frac{1-\widehat{\Delta}(z)}{1-z}\right)} \tag{2.51}
$$

Since $\widehat{N^{(Int)}}(1) = 1$, and $\lim_{z\to 1}\frac{1-\widehat{\Delta}(z)}{1-z} = \mathbf{E}[\Delta]$, we get

$$
kp_0+(k-1)p_1+\ldots+p_{k-1} = k - \frac{\lambda+\alpha\mathbf{E}[\Delta]}{\mu} \tag{2.52}
$$

The sum on the left is precisely the expected number of idle servers at $T \sim \text{Exp}(\alpha)$. Let

$$
C = 0\cdot k\cdot p_0 + (k-1)\cdot 1\cdot p_1 + (k-2)\cdot 2\cdot p_2 + \ldots + 1\cdot(k-1)\cdot p_{k-1}
$$

Then,

$$
\begin{aligned}
\mathbf{E}\left[N^{(Int)}\right] &= \frac{d}{dz}\widehat{N^{(Int)}}(z)\Big|_{z=1} \\
&= \left.\frac{\mu\frac{d}{dz}\left[kp_0+(k-1)zp_1+\ldots+z^{k-1}p_{k-1}\right]}{(k\mu-\lambda z)-\alpha z\left(\frac{1-\widehat{\Delta}(z)}{1-z}\right)}\right|_{z=1} \\
&\quad \left. - \frac{\mu\left[kp_0+(k-1)zp_1+\ldots+z^{k-1}p_{k-1}\right]}{\left((k\mu-\lambda z)-\alpha z\left(\frac{1-\widehat{\Delta}(z)}{1-z}\right)\right)^2}\frac{d}{dz}\left((k\mu-\lambda z)-\alpha z\left(\frac{1-\widehat{\Delta}(z)}{1-z}\right)\right)\right|_{z=1} \\
&= \frac{\mu C}{k\mu-\lambda-\alpha\mathbf{E}[\Delta]}
\end{aligned}
$$

$$- \frac{\widehat{N^{(Int)}}(1)}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} \left(-\lambda - \alpha \frac{1 - \widehat{\Delta}(z)}{1 - z} - \alpha z \left(\frac{1 - \widehat{\Delta}(z) - (1 - z)\frac{d\widehat{\Delta}(z)}{dz}}{(1 - z)^2} \right) \right) \Bigg|_{z=1}$$

and applying L'Hospital's rule to the last term,

$$\begin{aligned}
&= \frac{\mu C}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} \\
&\quad - \frac{1}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} \left(-\lambda - \alpha\mathbf{E}[\Delta] - \alpha \left(\frac{-\frac{d}{dz}\widehat{\Delta}(z) - (1 - z)\frac{d^2\widehat{\Delta}(z)}{dz^2} + \frac{d}{dz}\widehat{\Delta}(z)}{-2(1 - z)} \right) \Bigg|_{z=1} \right) \\
&= \frac{\mu C}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} - \frac{1}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} \left(-\lambda - \alpha\mathbf{E}[\Delta] - \alpha \frac{\mathbf{E}[\Delta^2] - \mathbf{E}[\Delta]}{2} \right) \\
&= \frac{\mu C}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} + \frac{\lambda + \frac{\alpha}{2}\left(\mathbf{E}[\Delta^2] + \mathbf{E}[\Delta]\right)}{k\mu - \lambda - \alpha\mathbf{E}[\Delta]} \qquad (2.53)
\end{aligned}$$

To calculate C we need the following relations obtained by matching the coefficients of z^i, $i = 0, \ldots, k-1$, from (2.50):

$$\begin{aligned}
-\lambda p_{0,\alpha} + \mu p_{1,\alpha} &= \alpha \left[p_{0,\alpha} - p_0(0) \right] \\
\lambda p_{i-1,\alpha} - (\lambda + i\mu) p_{i,\alpha} + (i+1)\mu p_{i+1,\alpha} &= \alpha \left[p_{i,\alpha} - p_i(0) \right] \qquad \ldots 1 \le i \le k-1
\end{aligned}$$

which yields $p_{i,\alpha} = p_{0,\alpha} \frac{1}{i!} \left(\frac{\lambda}{\mu}\right)^i + \Theta(\alpha)$. Let π_i be the stationary probabilities of an $M/M/k$ system with arrival rate λ and mean job size $\frac{1}{\mu}$. We can use (2.52) to write:

$$k\pi_0 + (k-1)\pi_1 + \ldots + \pi_{k-1} = k - \frac{\lambda}{\mu}$$

or equivalently,

$$\pi_0 \left(k \cdot 1 + (k-1) \cdot \frac{\lambda}{\mu} + \ldots + 1 \cdot \frac{1}{(k-1)!} \left(\frac{\lambda}{\mu}\right)^{k-1} \right) = k - \frac{\lambda}{\mu}$$

Rewriting (2.52) and using the facts $p_i = p_0 \frac{1}{i!} \left(\frac{\lambda}{\mu}\right)^i + \Theta(\alpha)$ and $\alpha\mathbf{E}[\Delta] = o(1)$:

$$p_0 \left(k \cdot 1 + (k-1) \cdot \frac{\lambda}{\mu} + \ldots + 1 \cdot \frac{1}{(k-1)!} \left(\frac{\lambda}{\mu}\right)^{k-1} \right) + \Theta(\alpha) = k - \frac{\lambda + o(1)}{\mu}$$

which gives $p_0 = \pi_0 + o(1)$, and hence $p_i = \pi_i + o(1)$ for $i \leq k-1$. Using this, we have:

$$\begin{aligned}
\frac{\mu C + \lambda}{k\mu - \lambda - \alpha \mathbf{E}[\Delta]} &= \frac{\mu \left(\sum_{i=0}^{k-1} i(k-i)p_i \right) + \lambda}{k\mu - \lambda + \alpha \mathbf{E}[\Delta]} \\
&= \frac{\mu \left(\sum_{i=0}^{k-1} i(k-i)\pi_i \right) + \lambda + o(1)}{k\mu - \lambda + o(1)} \\
&= \frac{\mu \left(\sum_{i=0}^{k-1} i(k-i)\pi_i \right) + \lambda}{k\mu - \lambda} + o(1) \\
&= \mathbf{E}\left[N^{M/M/k} \right] + o(1)
\end{aligned}$$

where $\mathbf{E}\left[N^{M/M/k} \right]$ is the mean number of jobs in a stationary $M/M/k$ queue with arrival rate λ and service rate μ. To see that $\mathbf{E}\left[N^{M/M/k} \right]$ can be written in the above form, note that when $\Delta \equiv 0$, $\mathbf{E}\left[N^{(Int)} \right] = \frac{\mu C + \lambda}{k\mu - \lambda}$ but $\mathbf{E}\left[N^{(Int)} \right] = \mathbf{E}\left[N^{M/M/k} \right]$. Finally,

$$\mathbf{E}\left[N^{(Int)} \right] = \mathbf{E}\left[N^{M/M/k} \right] + \frac{\frac{\alpha}{2}\mathbf{E}[\Delta^2]}{k\mu - \lambda} + o(1)$$

since $\alpha \mathbf{E}[\Delta] = o(1)$. ∎

Chapter 3

Towards a New Theory of Moments-based Bounds II: Markov-Krein Characterization of Mean Sojourn Time in Queueing Systems

In Chapter 2, we obtained an inapproximability gap for the mean waiting time in the $M/G/k$ model and further conjectured that the bounds we had obtained are sharp. Implicit in our sharpness conjecture was another stronger conjecture: given the first two moments of the service distribution, the mean waiting time is minimized and maximized by special two-point distributions. This conjecture echoes the results of the Stieltjes moment problem[90]: which probability distributions minimize or maximize the expectation of a function g given constraints (called moment constraints) on the distribution as expectations of a system $\{f_0, \ldots, f_n\}$ of functions. A classical theorem of Markov and Krein establishes sufficient conditions for the extremality properties of certain special point-mass distributions for the moment problem – these distributions are called the *principal representations* of the moment sequence. In this chapter we take a small step towards extending the classical literature on moment problem to three queueing systems which have so far defied exact analysis: (i) the $M/G/k$ multi-server system, (ii) queueing systems with fluctuating arrival and service rates, and (iii) the $M/G/1$ round-robin queue. We argue that rather than looking for exact expressions for the mean response time as a function of the entire service distribution, or approximations based on heavy traffic/diffusion asymptotics, a more fruitful approach is to identify distributions which minimize or maximize the

mean response time given the first n moments of the job size distribution.

By analyzing the queueing systems in appropriate "light-traffic" asymptotics, we prove that analogous to the classical Markov-Krein Theorem, these 'extremal' distributions are given by the *principal representations* of the moment sequence. We conjecture that (under some conditions) the property of *extremality* should be invariant to the system load, and thus our light traffic results should hold for general load as well, and propose potential strategies for a unified approach to finding moments-based bounds for queueing systems. By identifying the extremal distributions, our results allow one to numerically obtain **sharp bounds** on the performance of these queueing systems.

3.1 Introduction

Most results in queueing theory are concerned with obtaining explicit expressions for the performance metric of interest (e.g., mean response time) as a function of the distribution of some system parameter (e.g., job size distribution) under suitable assumptions to make the analysis tractable. However, there are many fundamental queueing systems for which such explicit results are not possible. We have already seen one example of this in Chapter 2: the $M/G/k$ First-Come-First-Serve multi-server model. We consider two more examples of such queueing systems in this chapter: the $M/G/1$ round-robin scheduling model, and systems with time-varying load, and present a fresh approach towards their analysis: via obtaining sharp bounds on the mean sojourn time, given a partial characterization of the system parameter in terms of the first n moments.

Motivation

Due to the abundance of work surrounding the $M/G/k$ multi-server model, we use it as an example to motivate our approach. We reviewed the prior work that studies the $M/G/k$ in Chapter 2, but for self-containment, we overview the literature relevant to this chapter again. For the $M/G/k$ system, our focus will be on the mean waiting time, denoted as $\mathbf{E}\left[W^{M/G/k}\right]$, and defined to be the expected time from the arrival of a customer to the time it begins service.

While the first approximation for $\mathbf{E}\left[W^{M/G/k}\right]$ dates back to 1959 [108], and only involves the first two moments of the service distribution, almost all modern closed-form approximations are motivated by diffusion analyses or other approximating assumptions and still involve only the first two moments of the service distribution.

Burman and Smith [37] proved a light-traffic approximation for $W^{M/G/k}$ which involves the entire job size distribution, and Boxma et al. [34] used it to obtain tighter approximations for $\mathbf{E}\left[W^{M/G/k}\right]$ for service distributions with low variance (SCV $<$ 1). This was achieved by interpolating the mean waiting time under deterministic jobs sizes, and under exponentially distributed job sizes, with the Burman-Smith approximation as the weighting function. However, extrapolating the Burman-Smith approximation yields inaccuracies when the job size distribution has high variance as is common in applications in computer science.

Bounds on the mean waiting time for $M/G/k$ queues (and more generally, for $GI/G/k$ queues) have mainly been obtained via two approaches (e.g., see Section 11-7 from Wolff [158]). The first approach is by assuming various orderings (stochastic ordering, increasing convex ordering) on the service distributions (see [43, 116, 141, 150, 151]), but these tend to be very loose as approximations. Moreover, one does not always have the required strong orderings on the service distribution. The second, and more practical, approach that started with the work of Kingman [99] is obtaining bounds on mean waiting time in terms of the first two moments of the inter-arrival and service distributions. The best known bounds of this type for $\mathbf{E}\left[W^{GI/G/k}\right]$ are presented by Daley [44]. Scheller-Wolf and Sigman [132] derive bounds on for the case $\rho < \left\lfloor \frac{k}{2} \right\rfloor$ by reducing the $GI/G/k$ waiting time recursion into an equivalent single-server recursion with dependent service times. Foss and Korshunov [60] and Scheller-Wolf and Vesilo [133] use dependent $D/GI/1$ queues to bound a $GI/G/k$ system, and obtain necessary and sufficient conditions under which higher (even fractional) moments of delay are finite.

Another approach used in the literature to establish bounds is by formulating a semidefinite program (SDP) with joint moments of service and inter-arrival time distribution forming the constraint set, and moments of waiting time as the objective function. With this approach, Bertsimas and Popescu [25] prove that the Markov inequality (using only the first moment) is tight, improve the Tchebycheff inequality (using the first two moments) to rediscover the corresponding tight inequality, and establish the analogous tight inequality that involves the first three moments. SDPs have also been used to obtain bounds on performance metrics for several queueing models. Recently, Bertsimas and Natarajan [24] have obtained numerical bounds on the moments of $W^{GI/G/K}$ given the information of moments of the service and the inter-arrival time distributions. Although most of the prior work obtains numerical bounds, Osogami and Raymond [123] use SDPs to establish closed-form bounds on the waiting time in a transient $GI/G/1$ queue. However, there are insufficient experimental results on the tightness of the resulting bounds.

Our Approach

Rather than try to obtain an explicit expression for the performance metric as a function of the job size distribution, or obtain approximations/bounds as functions of some moments of the job size distribution for which no tightness guarantees can be proved, we argue that a more fruitful approach is the following: We first obtain a partial characterization of the job size distribution, say, in terms of the first n moments. We then look at the set of all distributions which satisfy this partial characterization, and identify those distributions in this set that maximize or minimize the performance metric of interest. Once these extremal distributions are identified, numerical algorithms can be used to obtain provably tight bounds on performance. That is, **the bounds so obtained are the tightest achievable bounds given the partial characterization of the job size distribution**, not just arbitrary approximations or bounds. Our approach has the added benefit that many times the entire job size distribution is not available, while estimating the first few moments via sampling is a much easier task. By quantifying the gap between the upper and lower bounds given these first few moments, it can be determined if a more refined characterization, say, in terms of higher order moments, is necessary.

In this chapter, we take the first step towards obtaining tight bounds on the mean response time of the three queueing systems by analytically investigating suitable asymptotic regimes (to be made precise later). The asymptotic regimes are chosen so that the effect of the entire distribution of the system parameter of interest is evident (unlike heavy-traffic asymptotes where, usually, at most the first two moments are involved). Next, rather than using the asymptotic approximations to obtain quantitative behavior (by extrapolating to non-asymptotic regime), we extract **qualitative properties** by identifying distributions which minimize or maximize the performance metric in the asymptotic regime. We conjecture that the extremality property of the service distributions should be invariant to the arrival rate, and thus extremal distribution in the asymptotic regime should remain extremal in non-asymptotic regime as well (this is a non-trivial conjecture because there exist examples where the *relative performance* of two job size distributions is sensitive to the arrival rate for $M/G/k$).

The idea of obtaining tight bounds on the performance of a queueing system based on a partial characterization of the system parameters was first advocated by Eckberg [50] (and extended by Whitt [152]) for the $GI/M/k$ model. However, the explicit expressions for the waiting time distribution in the $GI/M/k$ model as a function of the Laplace transform of the inter-arrival time distribution greatly facilitates obtaining sharp bounds. The queueing systems we consider in this chapter do not have any such analyses available.

Summary of Results

We now briefly describe the three queueing systems, the "light traffic" asymptote we look at, and our results.

1. The $M/G/k$ multi-server system (Section 3.3)

Model: Recall that an $M/G/k$ system consists of k identical servers and a FCFS queue. The arrival process is Poisson with rate λ, and the job sizes are assumed to be $i.i.d$ random variables. We will use S to denote such a generic random variable. We are interested in obtaining bounds on the mean waiting time, $\mathbf{E}\left[W^{M/G/k}\right]$, as a function of the job size distribution S.

Asymptotic Regime: We let the arrival rate $\lambda \to 0$, and look at $\mathbf{E}\left[W^{M/G/k}\right]$ of a random arrival conditioned on the event that the arrival finds all servers busy. This can be seen as the first term in the Taylor series expansion of $\mathbf{E}\left[W^{M/G/k}\right]$ around $\lambda = 0$.

Results: We start with the Burman-Smith light-traffic approximation (Theorem 3.3), and prove the following:

1. Given the first $n = 2$ or 3 moments of the job size distribution, the extremal distributions are given by the principal representations of the moment sequence (defined in Section 3.2).
2. If we restrict the job size distribution to lie in the class of completely monotone (CM) distributions, then given the first n moments, the extremal distributions are given by the principal representations of the moment sequence within the hyperexponential class of distributions (mixtures of approximately $\frac{n}{2}$ exponential distributions; to be made formal in Section 3.2).

Finally, we illustrate the utility of our results by presenting numerical results that demonstrate that while two moments of the job size distribution are insufficient for approximating $\mathbf{E}\left[W^{M/G/k}\right]$ for real world heavy-tailed distributions, three moments usually suffice, especially if we add the knowledge of complete monotonicity.

2. The $M/G/1$ round-robin queue (Section 3.4)

Model: The $M/G/1$ round-robin queue consists of a single server and an infinite buffer. The arrival process is Poisson with rate λ, and new arrivals join the back of the buffer. Job sizes are assumed to be $i.i.d.$, with S used to denote a generic job size. Jobs are given q units of service at a time (called the quantum size), and if the job does not finish service, it joins the back of the buffer. For analytical simplicity we assume that service quanta are exponentially distributed random variables with mean $q = \frac{1}{\nu}$. That is, each time a job gets to the server, its service quantum is an $i.i.d.$ sample from an exponential distribution with rate ν. We will be interested in obtaining bounds on the mean response time, $\mathbf{E}\left[T^{M/G/1/RR}\right]$, in terms of moments of S.

Asymptotic Regime: We let the arrival rate $\lambda \to 0$, and look at the coefficient of $\Theta(\lambda)$ in the expression for $\mathbf{E}\left[T^{M/G/1/RR}\right]$.

Results: 1. We derive the light-traffic approximation for $\mathbf{E}\left[T^{M/G/1/RR}\right]$ when the job size distribution is hyperexponential with finite number of phases.
2. We use our light-traffic result to prove that if the job size distribution is restricted to lie in the class of CM distributions, then given the first n moments, the extremal distributions are given by the principal representations of the moment sequence within the hyperexponential class of distributions.

3. Systems with fluctuating arrival and service rates (Section 3.5)
Model: We analyze an $M/M/1$ system whose arrival and service rates are controlled by an exogenous environment process with two states: L and H. The job sizes are exponentially distributed. While in the H state, the arrival process is Poisson with rate λ_H, and server serves jobs at rate μ_H. During the L states, the arrival process is Poisson with rate λ_L, and the server's service rate is μ_L. The durations of stay in the L state during each visit are *i.i.d.* random variables with general distribution; we use τ_L to denote such a generic random variable. Similarly, we use τ_H to denote a generic random variable for the duration of stay in the H states during each visit. We will be interested in obtaining bounds on the mean number of jobs, $\mathbf{E}[N]$, in terms of moments of τ_L and τ_H. (As mentioned later, we expect our results to hold for systems where evolution during L and H states is governed by arbitrary Markov processes satisfying mild conditions.)
Asymptotic Regime: We consider the "fast-switching" asymptote. In particular, we index our time-varying load system with a parameter α, where in the αth system the durations of stay in L and H states are *i.i.d.* and given by $\alpha\tau_L$ and $\alpha\tau_H$, respectively. We then analyze $\mathbf{E}[N]$ in the limit $\alpha \to 0$. Note that as $\alpha \to 0$ and our systems switches very fast, the zeroth order behavior is given by an $M/M/1$ with the average arrival and service rates. We will be interested in the coefficients of higher order terms in the expansion of $\mathbf{E}[N]$ around $\alpha = 0$.
Results: 1. We derive the first fast-switching asymptote approximation for the time-varying load system when the distributions of τ_L and τ_H are hyperexponential with finite number of phases. In particular, we prove the following interesting result: The coefficient of α^i is a function of only the first $(i+1)$ moments of τ_L and τ_H. Further, this coefficient is linear in $\mathbf{E}\left[\tau_L^{i+1}\right]$ and $\mathbf{E}\left[\tau_H^{i+1}\right]$.
2. The above result immediately implies that if τ_L and τ_H are restricted to lie in the CM class, then given the first n moments, the number of jobs in the system (equivalently, the mean response time) in the fast-switching asymptote is extremized by CM distributions with extremal $(n+1)$st moment. These are again given by principal representations of the moment sequence in the class of hyperexponential distributions.

3. Our light-traffic result, and hence the result on extremal distributions, easily extend to general distributions, but we choose not to present them here since the analysis is almost identical but proof ideas are easy to illustrate for CM distributions. Finally, we illustrate the utility of our results in obtaining provable bounds on the performance of the N model for work-stealing (or the N-sharing system). While the N-sharing system can be modeled by a Markov chain, there are no exact numerical algorithms for solving it since the Markov chain is infinite in two dimensions.

A note on completely monotone class of distributions

A probability density function $f_X(\cdot)$ is said to belong to the class of completely monotone (CM) distributions if all derivatives of f_X exist and $(-1)^n f_X^{(n)}(x) \geq 0$ for all $x > 0$ and $n \geq 1$. It is well known that mixture of exponential distributions are dense in the CM class. That is, for any distribution function F in the CM class, there exist hyperexponential distributions $F^{(n)}$ with n phases such that $F^{(n)} \Rightarrow F$ as $n \to \infty$ [56, Theorem 3.2]. In fact, $F_X(\cdot)$ is a CM probability distribution function if and only if

$$F_X(x) = \int_0^\infty e^{-\mu x} dG(\mu),$$

where G is a proper probability distribution function, and commonly called the spectral distribution of F. Completely Monotone distributions are a subset of the Decreasing Failure Rate (DFR) class of distributions. It can be shown that this denseness is sufficient to approximate arbitrarily many moments of a CM distribution via mixture of exponential distributions. It has been established that many heavy-tailed distributions used to model computer systems workloads fall in the CM class, e.g., Pareto distributions, Weibull distributions with shape parameter less than 1 (heavier than exponential), and Gamma distributions with shape parameter less than 1 [56]. Further, there are several results on conditions under which the convergence of the inter-arrival and service-time distributions imply convergence in distribution of waiting time (see e.g. Borovkov [30, page 118], Stoyan [141]), although care must be exercised since convergence in distribution does not necessarily imply convergence of moments. To prove results about CM distributions, we will therefore restrict to looking for extremal distributions within hyperexponential distributions.

Outline

We introduce the concepts of Principal Representations and Tchebycheff systems of functions in Section 3.2. Section 3.2 also states the classical Markov-Krein Theorem which we use as a tool to prove our results for CM distributions. In Sections 3.3, 3.4

and 3.5, we present our results on tight moment-based bounds for (i) the $M/G/k$ multi-server model, (ii) $M/G/1$ round-robin scheduling, and (iii) systems with time-varying load, respectively, under "light-traffic" asymptote. We present conjectures on bounds under non-asymptotic regimes for these three queueing systems in Section 3.6. In Section 3.7, we present some approaches for proving our conjectures, and introduce a novel moment problem as a unified framework for analyzing the question of moment-based bounds for general queueing systems.

3.2 Principal Representations, Tchebycheff systems, and the Markov-Krein Theorem

The classic Tchebycheff inequality concerns bounds on the tail probability of a random variable X, given $\mathbf{E}[X]$ and $\mathbf{E}[X^2]$. In other words, given the expectations of functions $f_1(x) = x$ and $f_2(x) = x^2$, one asks for bounds on the expectation of $g(x) = \mathbf{1}_{|x-\mathbf{E}[X]|>a}$. The theory of Tchebycheff systems [90] generalizes this question by asking for bounds on the expectation of some given function $g(\cdot)$ of a random variable, given a partial characterization of the random variable in terms of generalized moment constraints expressed as expectations of some functions $f_1(\cdot), \ldots, f_n(\cdot)$. In this chapter, we will be concerned with the case $f_i(x) = x^i$. Below we present a special case of the results from this area. We will begin with the case where random variables are restricted to bounded support $[0, B]$ and where the results are easy to state. We then present results for the case where the support is $[0, \infty)$ and details are a little delicate. For a detailed treatment, we refer the reader to [90].

3.2.1 Random variables with support on $[0, B]$

We first introduce the notion of upper and lower principal representations as presented in [50]. Define the function $f_0(x) = 1, 0 \leq x \leq B$, and denote the moment space associated with $\{f_0, f_1, \ldots, f_n\}$ as

$$\mathcal{M}_B^{n+1} = \left\{ \mathbf{m} \in \mathbb{R}^{n+1} \middle| \exists \mu \in \mathcal{D}, m_i = \int_0^B f_i(u) d\mu(u), 0 \leq i \leq n \right\}$$

where $\mathcal{D}$ is the set of all non-decreasing right continuous functions for which the indicated integrals exist. For a point $\mathbf{m^0}$ in the interior of $\mathcal{M}_B^{n+1}$, we define the *lower and upper principal representation (pr)* to be distributions with a particular number of mass probabilities, some of which are restricted to be at 0 or B, in such a way that the first n moments of these distributions agree with $\mathbf{m^0}$. In particular the constraints on the support of principal representations are:

	Upper pr ($\bar{\mu}$)	Lower pr ($\underline{\mu}$)
n even	$\frac{n}{2}$ mass points in $(0, B)$, one at B	$\frac{n}{2}$ mass points in $(0, B)$, one at 0
n odd	$\frac{n-1}{2}$ mass points in $(0, B)$, one at 0, one at B	$\frac{n+1}{2}$ mass points in $(0, B)$

The upper and lower principal representations are *uniquely determined* when the functions $\{f_0, \ldots, f_n\}$ satisfy certain linear independence constraints mentioned later. To see why, consider the case of upper pr for n even. We have $n+1$ constraints, one each for m_i, $0 \leq i \leq n$. If we are allowed $\frac{n}{2}+1$ probability masses, then we have $n+2$ degrees of freedom: $\frac{n}{2}+1$ for the locations of the probability masses, and $\frac{n}{2}+1$ for the actual probability values. Since one probability mass is constrained to be at B, we lose one degree of freedom. Thus the degrees of freedom match the number of constraints, where these constraints are "linearly independent" in a sense made precise next.

Definition 3.1 *Functions $\{h_0, h_1, \ldots, h_n\}$ form a Tchebycheff system over $[a, b]$ provided the determinants*

$$U\begin{pmatrix} 0, 1, \cdots, n \\ x_0, x_1, \cdots, x_n \end{pmatrix} = \begin{vmatrix} h_0(x_0) & h_0(x_1) & \cdots & h_0(x_n) \\ h_1(x_0) & h_1(x_1) & \cdots & h_1(x_n) \\ \vdots & \vdots & & \vdots \\ h_n(x_0) & h_n(x_1) & \cdots & h_n(x_n) \end{vmatrix}$$

are strictly positive whenever $a \leq x_0 < x_1 < \cdots < x_n \leq b$.

In other words, any non-trivial linear combination of $h_0, \ldots, h_n$ must have at most n zeros in the interval $[0, B]$ (and then the signs of $\{h_i\}$ should be chosen appropriately to ensure that the determinant is positive). Systems of polynomials: $h_i(x) = x^{\alpha_i}$ $(0 \leq \alpha_0 < \alpha_1 < \ldots < \alpha_n)$ indeed form Tchebycheff systems.

The proof of the following theorem can be found in [90, Chpt. V, Sec. 5]:

Theorem 3.1 (Markov-Krein) *If $\{f_0, \ldots, f_n\}$ and $\{f_0, \ldots, f_n, g\}$ are Tchebycheff systems on $[0, B]$, then*

$$\beta_l \equiv \inf_{\mu_X \in \mathcal{D}} \{\mathbf{E}[g(X)] \mid \mathbf{Pr}[X \in [0, B]] = 1; \mathbf{E}[f_i(X)] = m_i,\ 0 \leq i \leq n\} = \int_0^B g(u) d\underline{\mu}(u) \ ,$$

$$\beta_u \equiv \sup_{\mu_X \in \mathcal{D}} \{\mathbf{E}[g(X)] \mid \mathbf{Pr}[X \in [0, B]] = 1; \mathbf{E}[f_i(X)] = m_i,\ 0 \leq i \leq n\} = \int_0^B g(u) d\bar{\mu}(u) \ ,$$

where $\underline{\mu}$ and $\bar{\mu}$ are the unique lower and upper pr's, respectively, of $\mathbf{m} = \{1, m_1, \ldots, m_n\}$, and μ_X denotes the measure induced by X on $\Re$.

3.2.2 Random variables with support on $[0, \infty)$

As before, denote the moment space associated with $\{f_0, f_1, \ldots, f_n\}$ as

$$\mathcal{M}_{\infty}^{n+1} = \left\{ \mathbf{m} \in \mathbb{R}^{n+1} \middle| \exists \mu \in \mathcal{D}, m_i = \int_0^{\infty} f_i(u) d\mu(u), 0 \leq i \leq n \right\}$$

where $\mathcal{D}$ is the set of non-negative regular measures of bounded variation for which the indicated integrals exist.

The definition of lower pr remains unchanged when we extend the support to $[0, \infty)$ as there are no atoms placed at the upper bound of the support. Hence, for a large enough B, the lower pr of $\mathbf{m^0}$ on $[0, B]$ will coincide with the lower pr on $[0, \infty)$. In particular, for n even, the lower pr will constitute of $\frac{n}{2}$ mass points in $(0, \infty)$ and one mass point at 0; for n odd, there will be $\frac{n+1}{2}$ mass points in $(0, \infty)$.

To define the upper pr, $\bar{\mu}$,we first need another definition.

Definition 3.2 *Functions $\{h_0, h_1, \ldots, h_n\}$ form a Tchebycheff system of Type II over $[0, \infty)$ provided:*
(i) $\{h_0, \ldots, h_{n-1}\}$ and $\{h_0, \ldots, h_n\}$ are Tchebycheff systems on $[0, \infty)$; and
(ii) there exists $A > 0$ such that $h_n(x) > 0$ for $x \geq A$, and

$$\lim_{x \to \infty} \frac{h_i(x)}{h_n(x)} = 0 \quad \text{for } i < n.$$

If $\{f_0, \ldots, f_n\}$ is a Tchebycheff system of Type II, then for $\mathbf{m^0}$ in the interior of $\mathcal{M}_{\infty}^{n+1}$, the upper pr puts one mass at ∞, $\left\lfloor \frac{n}{2} \right\rfloor$ mass points in $(0, B)$, and additionally one at 0 if n is odd. The following example might help readers uncomfortable with the idea of mass at infinity: Consider the case $f_i(x) = x^i$ and $n = 2$. In this case, the upper pr can be seen as the limit as $\epsilon \to 0$ of a sequence of distributions with support $[0, \frac{1}{\epsilon}]$ (indeed, the upper pr with support $\left[0, \frac{1}{\epsilon}\right]$) which put $\Theta(\epsilon^2)$ mass on $\frac{1}{\epsilon}$. Thus, this mass at ∞ is needed to satisfy the constraint corresponding to f_n, but does not contribute to constraints for $f_0, \ldots, f_{n-1}$ when $\{f_0, \ldots, f_n\}$ is a Tchebycheff system of Type II.

Theorem 3.2 (Markov-Krein) *[90, Theorem V5.1] If $\{f_0, \ldots, f_n\}$ and $\{f_0, \ldots, f_n, g\}$ are Tchebycheff systems on $[0, \infty)$, and $\mathbf{m^0}$ lies in the interior of $\mathcal{M}_{\infty}^{n+1}$, then there exists*

$$\beta_l \equiv \inf_{\mu_X \in \mathcal{D}} \{\mathbf{E}[g(X)] \mid \mathbf{Pr}[X \in [0, B]] = 1; \mathbf{E}[f_i(X)] = m_i, \ 0 \leq i \leq n\}$$

which is achieved uniquely for $\mu_X = \underline{\mu}$, *the lower pr of* $\mathbf{m^0}$.
The upper bound

$$\beta_u \equiv \sup_{\mu_X \in \mathcal{D}} \{\mathbf{E}[g(X)] \mid \mathbf{Pr}[X \in [0, B]] = 1; \mathbf{E}[f_i(X)] = m_i,\ 0 \leq i \leq n\}$$

in general may not be attained, or may be infinite. However, if $\{f_0, \ldots, f_n\}$ *is a Tchebycheff system of Type II, and*

$$\lim_{x \to \infty} \frac{g(x)}{f_n(x)} = \gamma < \infty,$$

then β_u *exists and is "achieved" by the upper pr of* $\mathbf{m^0}$.

In the last sentence of the theorem, we say "achieved" to emphasize the fact that the upper pr has a mass point at ∞ and thus it is not a finite measure. However, β_u exists and is achieved as a limit.

The Markov-Krein Theorem has been successfully applied in the context of queueing systems [50, 152]. In particular, for a $GI/M/1$ system, Theorem 3.1 proves that given the first n moments of the inter-arrival time distribution, the mean number of jobs in the system is extremized by inter-arrival time distributions which correspond to the upper and lower pr's. The proof follows by noting that the mean number of jobs in a $GI/M/1$ queue with *i.i.d.* inter-arrival times given by a random variable A is an increasing function of the Laplace-Stieltjes transform of the inter-arrival time distribution ($\tilde{A}(s) = \mathbf{E}\left[e^{-sA}\right]$), and the functions $g_s(x) = e^{-sx}$ form Tchebycheff system with $f_i(x) = x^i$.

Principal representations within Hyperexponential distributions Consider the following random variable with an n-phase hyperexponential distribution:

$$X \sim \begin{cases} \text{Exp}\left(\frac{1}{x_1}\right) & \text{with probability } q_1 \\ \vdots & \\ \text{Exp}\left(\frac{1}{x_n}\right) & \text{with probability } q_n \end{cases}$$

We can now define another random variable Y with distribution given by the *inverse spectrum* of X:

$$Y \sim \begin{cases} x_1 & \text{with probability } q_1 \\ \vdots & \\ x_n & \text{with probability } q_n \end{cases}$$

We have the following straightforward relationship between moments of X and Y: $\mathbf{E}[Y^i] = \frac{\mathbf{E}[X^i]}{i!}$. We define the upper and lower principal representation for a moment sequence $m_1, m_2, \ldots, m_n$ within the class of hyperexponential distributions as the distributions whose inverse spectrum are the upper and lower principal representations, respectively, for the moment sequence $m_1, \frac{m_2}{2!}, \ldots, \frac{m_n}{n!}$.

3.3 Bounds for the $M/G/k$ Multi-server Model

In this section we present our results on sharp moment-based bounds for the $M/G/k$ model in light traffic. Recall that the arrival process is Poisson with rate λ, and the job sizes are *i.i.d.* according to a random variable S. The load of the system is defined as $\rho = \lambda\mathbf{E}[S]$ and denotes the time average number of busy servers. The waiting time of a job is defined to be the time between when a job arrives to the system and when it begins service, and is denoted by $W^{M/G/k}$. We will analyze $\mathbf{E}\left[W^{M/G/k}\right]$ in the light traffic asymptote $\rho \to 0$ while holding S and k unchanged.

In Section 3.3.1, we present our results on bounds for general service distributions given the first 2 or 3 moments, and in Section 3.3.2 for completely monotone distributions given any number of moments. In Section 3.3.3, we present numerical results on bounds obtained using principal representations for common heavy-tailed service distributions.

3.3.1 Bounds for general distribution

We begin with a well-known result on the light traffic approximation for $W^{M/G/k}$.

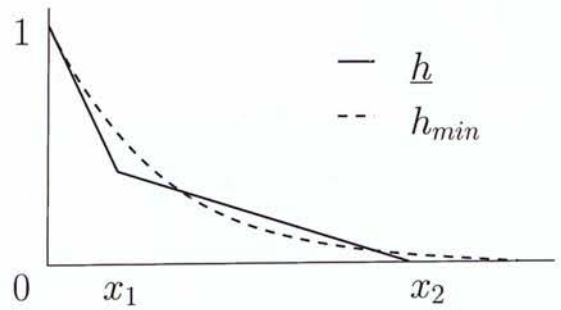
Theorem 3.3 (Burman Smith [37]) *Under the assumption that the service distribution is phase-type, as $\rho \to 0$, the probability that an arrival finds all servers busy is asymptotically given by $\frac{1}{k!}\left(\frac{\rho}{k}\right)^k$, and conditioning on this event, $W^{M/G/k}$ is distributed as the minimum of k independent copies of the stationary excess of S, denoted by S_e. The survival function of S_e is given by $\overline{F}_{S_e}(x) = \mathbf{Pr}[S_e > x] = \frac{\int_{u=x}^{\infty} \mathbf{Pr}[S > u]du}{\mathbf{E}[S]}$.*

Theorem 3.4 *Given the first n ($n = 2$ or 3) moments of the service distribution S, $\mathbf{E}\left[W^{M/G/k}\right]$ under light traffic is extremized by service distributions given by the lower and upper principal representations of the moment sequence.*

Proof: We present the proof for the case $n = 3$, where the lower pr minimizes, and upper pr maximizes $\mathbf{E}\left[W^{M/G/k}\right]$. Denote $\overline{F}_{S_e} = h$ for succinctness. Since $1 - h(x) = \frac{\int_0^x \mathbf{Pr}[S > u]du}{\mathbf{E}[S]}$ is the integral of a bounded, non-negative, decreasing function, $(1 - h(x))$ is a continuous, non-decreasing, concave function. The problem of extremizing $\mathbf{E}\left[W^{M/G/k}\right]$ in the light-traffic asymptote can thus be equivalently formulated as:

$$\begin{aligned}
&\min/\max \int_0^\infty h(u)^k du \\
&\text{subject to } h(\cdot) \text{ continuous, non-negative, non-increasing, convex ;} \\
&\qquad h(0) = 1 ; \\
&\qquad |h'(0^+)| = \frac{\mathbf{Pr}[S > 0]}{\mathbf{E}[S]} \leq \frac{1}{\mathbf{E}[S]} ; \\
&\qquad \int_0^\infty h(u)du = \frac{\mathbf{E}[S^2]}{2\mathbf{E}[S]} ; \\
&\qquad \int_0^\infty u \cdot h(u)du = \frac{\mathbf{E}[S^3]}{12\mathbf{E}[S]}.
\end{aligned}$$

(Note that a solution to the above problem exists because $0 \leq h(u)^k \leq h(u)$, and $\int h(u)du$ is finite.) Let $\underline{h}$ represent the survival functions of S_e corresponding to the lower pr of S for the given moment sequence. Now, suppose that $\underline{h}$ is not the solution to the minimization problem above, and the solution is instead given by h_{min}. For $n = 3$, we have that the lower pr has 2 point masses, say at $0 < x_1 < x_2 < \infty$, as shown below.



The absolute value of the slope of h_{min} at 0^+ must be at most that of $\underline{h}$ since the lower pr has no mass at 0 for $n = 3$, and because h_{min} is convex, it follows that $\delta = h_{min} - \underline{h}$ satisfies (i) $\int_0^\infty \delta(u)du = 0$ (i.e., areas under $\underline{h}$ and h_{min} are equal), (ii) $\int_0^\infty u \cdot \delta(u)du = 0$ (from moment conditions), and (iii) $\delta(\cdot)$ changes sign exactly twice, and the sequence of signs is $+-+$ (see the figure above). We obtain a contradiction:

$$\int h_{min}(u)^k du - \int \underline{h}(u)^k du$$

$$= \int \delta(u) \left[h_{min}(u)^{k-1} + h_{min}(u)^{k-2} \underline{h}(u) + \ldots + \underline{h}(u)^{k-1} \right] du$$
$$> 0$$

To see the last inequality, denote the function in the square brackets by $\ell(\cdot)$, and note that ℓ is convex. Now $\int_0^\infty \delta(u)\ell(u)du = [\delta(u)\ell'(u)]_0^\infty - \int_{u=0}^\infty \ell'(u) \int_{v=0}^u \delta(v)dvdu$. The first term is zero because $\delta(0) = \delta(\infty) = 0$. Now assuming derivatives exist (by approximating by smoothed versions), we find that $\ell'(u)$ is an increasing function, and $\int \delta(v)dv$ is a function that changes sign only once, from + to − and integrates to 0. Thus $\int_{u=0}^\infty \ell'(u) \int_{v=0}^u \delta(v)dvdu < 0$.

The proof for upper pr is identical except that the sequence of signs of δ in this case is $- + -$. For $n = 2$, δ changes sign once. ∎

3.3.2 Bounds for CM service distributions

Theorem 3.5 *If the service distribution is constrained to lie in the CM class, then given the first n moments of the service distribution S,* $\mathbf{E}\left[W^{M/G/k}\right]$ *under light traffic is extremized by the lower and upper principal representations of the moment sequence within the hyperexponential class of distributions.*

Proof: The first step of the proof is to restrict our attention to hyperexponential distributions with finite number of phases as they are dense in the CM class. We will now use the Markov-Krein Theorem to prove the result. However, Theorem 3.1 does not apply directly to our problem because as Theorem 3.3 shows, the mean waiting time in light traffic can not be written as $\mathbf{E}[f(S)]$ for any function $f(\cdot)$. Instead, we prove a stronger result.

Consider a tagged arrival that finds all the servers busy. We fix the distribution of the job sizes at the first $k-1$ servers to be exponential with arbitrary parameters (say $\nu_1, \nu_2, \ldots, \nu_{k-1}$). We will now show that given the moments of the service distribution for the job at the kth server, the hyperexponential distributions that minimize or maximize the time until first departure, and hence the waiting time of the arrival, are given by the pr's *irrespective of the choice of* $\nu_1, \ldots, \nu_{k-1}$. Let the service distribution of the job at the kth server be:

$$S \sim \begin{cases} \text{Exp}\left(\frac{1}{x_1}\right) & \text{with probability } q_1, \\ \vdots & \\ \text{Exp}\left(\frac{1}{x_n}\right) & \text{with probability } q_n. \end{cases}$$

As defined in Section 3.2, let Y denote a random variable whose distribution is given by the inverse spectrum of S, and let $M = \sum_{j=1}^{k-1} \nu_j$. The mean waiting time of the tagged arrival, $\mathbf{E}[W^*]$, is then given by:

$$\begin{aligned}\mathbf{E}[W^*] &= \sum_{i=1}^{n} \frac{q_i x_i}{\mathbf{E}[S]} \cdot \frac{1}{M + \frac{1}{x_i}} \\ &= \frac{1}{\mathbf{E}[S]} \sum_{i=1}^{n} q_i \left(\frac{Mx_i - 1}{M^2} + \frac{1}{M^2(Mx_i + 1)} \right) \\ &= \frac{1}{M} - \frac{1}{M^2 \mathbf{E}[Y]} + \frac{1}{M^2 \mathbf{E}[Y]} \mathbf{E}\left[\frac{1}{MY + 1} \right]\end{aligned}$$

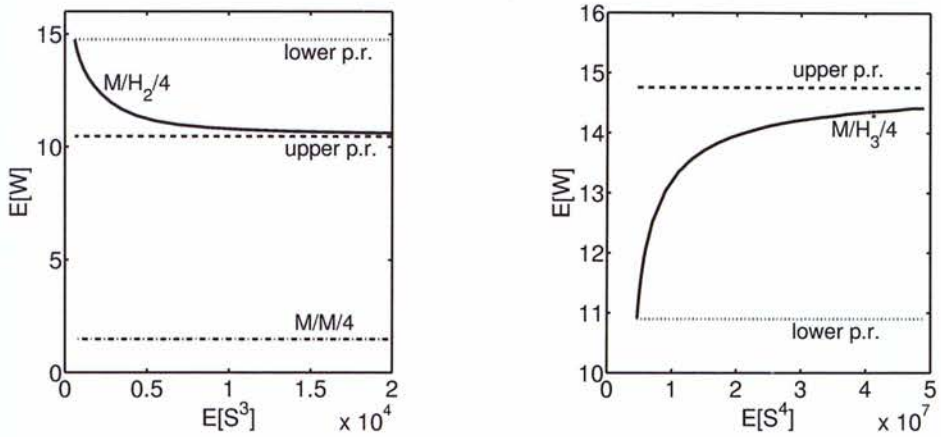
From Theorem 31 of [86], $\frac{1}{Mx+1}$ forms a Tchebycheff system with the functions $i!x^i$, and hence by Theorem 3.1, the result follows. ∎

Remark 1: The reader might wonder if we could use a similar proof outline as Theorem 3.5 to prove the result for general distributions. To be more precise, we can arbitrarily fix the residual sizes of jobs at the first $k-1$ servers as $u_1 \leq \ldots \leq u_{k-1}$. We may then ask the question: for given first n moments, what service distribution for S extremizes $\mathbf{E}[\min\{S_e, u_1\}]$. The latter expectation can indeed be written as $\mathbf{E}[f(S)]$, where $f(\cdot)$ is a piecewise polynomial function. However, even for $n = 3$, $f(x)$ **does not** form a Tchebycheff system with the moment functions x^0, x^1, x^2 and x^3. Thus, Theorem 3.4 can in some sense be seen as *breaking the Tchebycheff system barrier.*

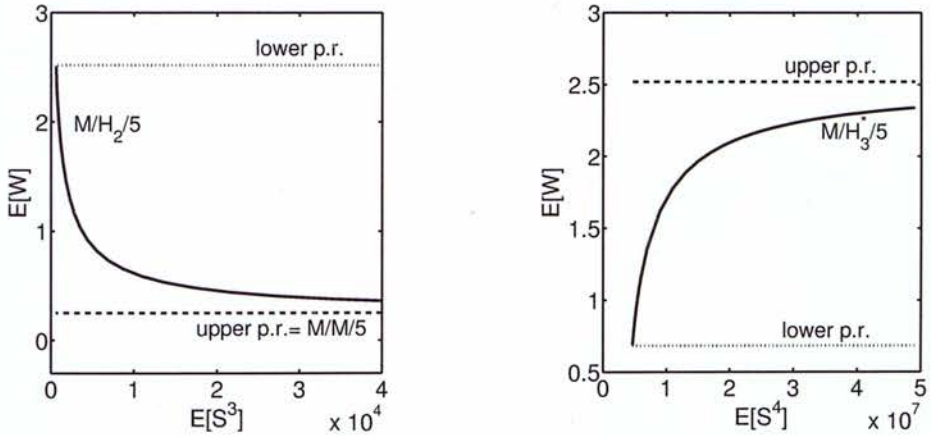
3.3.3 Simulation and numerical evaluation

We conjecture that Theorem 3.4 extends to any number, n, of moments and to general traffic, and Theorem 3.5 extends to arbitrary loads. See Section 3.6 for the specific properties that we conjecture to hold generally for moment-based tight bounds on $\mathbf{E}\left[W^{M/G/k}\right]$. In this section, we provide support for the conjectures and numerically study the quality of bounds obtained with principal representations.

Figure 3.1 provides numerical evidence in support of the validity of Theorem 3.5 for general loads. The solid curves in Figure 3.1(a) and Figure 3.1(c) show $\mathbf{E}\left[W^{M/G/k}\right]$ when the job size has a two-phase hyperexponential (H_2) distribution which allows us to vary $\mathbf{E}[S^3]$ while holding the first two moments fixed. The solid curves in Figure 3.1(b) and Figure 3.1(d) show $\mathbf{E}\left[W^{M/G/k}\right]$ when the job size has a degenerate three-phase hyperexponential ($H_3^\star$) distribution, which has two non-zero mean exponential phases and a phase with zero mean. Within $H_3^\star$ distributions, we can vary $\mathbf{E}\left[S^4\right]$, while holding the first three moments fixed. We set the number of servers, k, as indicated below each figure.



(a) $n = 2,\ k = 4$ (b) $n = 3,\ k = 4$



(c) $n = 2,\ k = 5$ (d) $n = 3,\ k = 5$

Figure 3.1: Mean waiting time in an $M/G/k$ system when the job sizes obey a hyperexponential distribution. In (a) and (c), we vary $\mathbf{E}[S^3]$, while keeping $\{\mathbf{E}[S] = 1,\ \mathbf{E}[S^2] = 20\}$. In (b) and (d), we vary $\mathbf{E}\left[S^4\right]$, while keeping $\{\mathbf{E}[S] = 1,\ \mathbf{E}[S^2] = 20,\ \mathbf{E}[S^2] = 8000\}$.

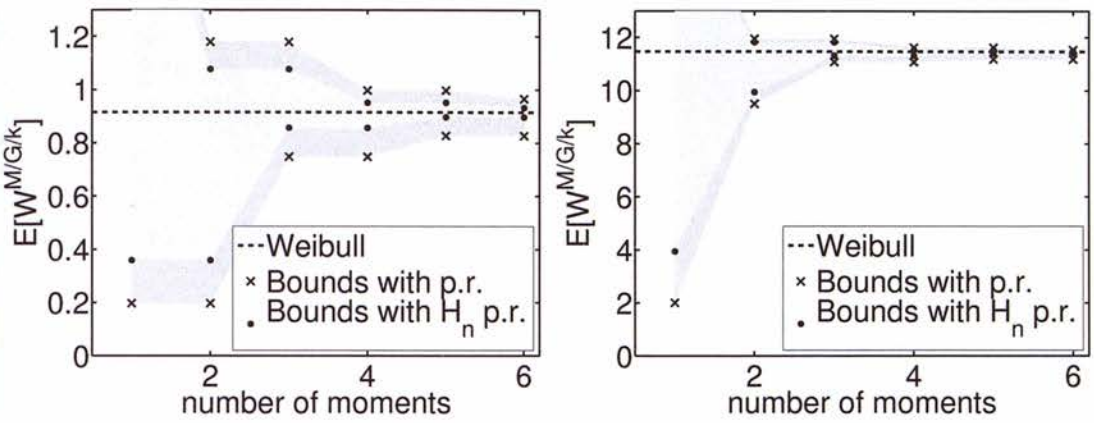
Observe that the solid curves lie between the mean waiting times attained when the service distributions are given by principal representations within hyperexponential distributions (dashed line and dotted line). The principal representations are determined by the first two moments of the H_2 distribution in Figure 3.1(a) and Figure 3.1(c) and the first three moments of the $H_3^\star$ distribution in Figure 3.1(b) and Figure 3.1(d). Also, observe that the solid curve is decreasing in $\mathbf{E}[S^3]$ and increasing

in $\mathbf{E}\left[S^4\right]$. A detail is that, in Figure 3.1(a), the upper principal representation refines the lower bound obtained from an exponential job-size distribution (line labeled with $M/M/4$). However, in Figure 3.1(c), the lower bound obtained with a principal representation coincides with the lower bound obtained from an exponential job-size distribution. See Conjecture 3.1 in Section 3.6 for the properties that we conjecture to hold generally for the bounds on $\mathbf{E}\left[W^{M/G/k}\right]$.

Figure 3.2 shows $\mathbf{E}\left[W^{M/G/k}\right]$ and its bounds obtained with principal representations, when the service distribution is a Weibull distribution. We fix the parameters of the Weibull distribution such that its probability density function is $f(x) = \frac{1}{2}x^{-1/2}\exp\left(-x^{1/2}\right)$ for $x \geq 0$. We also fix the number of servers, $k = 4$, and vary the load, $\rho \equiv \lambda\mathbf{E}[S]$, as indicated below each figure. The dashed line shows the exact value of $\mathbf{E}\left[W^{M/G/k}\right]$, and the crosses and the dots show bounds on $\mathbf{E}\left[W^{M/G/k}\right]$ obtained with principal representations. Specifically, a bound shown with a cross is the mean waiting time in the $M/G/k$ system whose service distribution has a principal representation that is determined by the moments of the Weibull distribution (see Theorem 3.4). A bound shown with a dot is obtained analogously with a principal representation within hyperexponential distributions (see Theorem 3.5). Notice that the Weibull distribution under consideration is completely monotonic (see [56]), so that principal representations within hyperexponential distributions give valid bounds. The horizontal axis indicates the number of moments used to determine the principal representations. The moments of the Weibull distribution under consideration are $\mathbf{E}[S^n] = (2n)!$ for $n = 1, 2, \ldots$.

In all cases, $\mathbf{E}\left[W^{M/G/k}\right]$ and the bounds shown with a cross are obtained via simulations; the bounds shown with a dot are calculated via matrix analytic methods. For each data point, the simulation is run 10 times and the average value of the 10 simulated mean waiting times is plotted. Each run of simulation is continued for 10,000,000 events, where an event is either an arrival or a departure of a job, and waiting times of the departed jobs are recorded (we ignore the first 100,000 departures). Confidence intervals are sufficiently small and not shown.

In Figure 3.2, notice that, except for $n = 2$, either lower bounds or upper bounds are shown for each n, where n is the number of moments used to determine the principal representation. This is because the lower (respectively, upper) bound obtained with an even (respectively, odd) number n of moments in general does not improve the corresponding lower (respectively, upper) bound obtained with $n - 1$ moments. An exception is that the lower bound obtained with $n = 2$ moments improves upon that with $n = 1$ for $\rho = 3.6$ (but not for $\rho = 2.4$, as predicted by our analysis in Chapter 2). The lower bound with $n = 2$ moments is given by a limiting distribution where one of the mass points, B, approaches infinity. This lower bound corresponds



(a) $\rho = 2.4$ (b) $\rho = 3.6$

Figure 3.2: Bounding mean waiting time in an $M/G/4$ queue when the job sizes obey a Weibull distribution.

to the principal representation with $B = 10^6$. It appears that the mean waiting time under the principal representation hardly changes in the interval between $B = 10^4$ and $B = 10^6$. For a $B > 10^6$, the analysis of the mean waiting time suffers from numerical errors.

Observe in Figure 3.2 that the principal representations within hyperexponential distributions (dot) can give bounds that are significantly better than the corresponding bounds obtained with the standard principal representations (cross). The principal representations within hyperexponential distributions provide bounds that are valid only for (service) distributions that are completely monotonic. The difference between a dot and a cross shows the refinement of the bound that we gain from the knowledge of complete monotonicity. Also observe that, as the number of moments used to determine principal representations grows, the upper and lower bounds approach each other quickly particularly at high load (Figure 3.2(b)). This makes intuitive sense, because $\mathbf{E}\left[W^{M/G/k}\right]$ becomes insensitive to third and higher moments in heavy-traffic.

3.3.4 A departure from Markov-Krein

The classical Markov-Krein theorem only enforces the condition that the moment constraint functions $\{f_0, \ldots, f_n\}$ be linearly independent (modulo signs of functions). As mentioned earlier, this condition holds for the power functions $f_i(x) = x^{\alpha_i}$,

$0 \leq \alpha_0 < \ldots \leq \alpha_n$. In particular, note that α_i need not be integral. However, here we see a departure in the behavior of $M/G/k$ from the classical Markov-Krein characterization – If the moment constraints involve fractional moments, the relative performance of upper and lower principal representations may flip as the arrival rate increases from light traffic to heavy traffic. Further, the upper and lower pr's may no longer provide bounds.

We will illustrate this point with an example. Consider the moment constraints $m_0 = \mathbf{E}[S^0] = 1$, $m_1 = \mathbf{E}[S^1] = 1$, and $m_{\frac{4}{3}} = \mathbf{E}\left[S^{\frac{4}{3}}\right] = 5$, and let us restrict ourselves to the class of hyperexponential distributions (since we do have the light traffic extremality results within the H_n class). The upper pr places almost the entire probability mass on the mean, and behaves as an exponential distribution in light traffic (this is true as long as the highest moment constraint is smaller than $1 + \frac{1}{k}$; $\frac{4}{3}$ for $k > 1$ suffices). Therefore in light traffic, the mean sojourn time of the upper pr is smaller than the mean sojourn time of the lower pr. However, due to the mass at ∞ in upper pr, the second moment is ∞ whereas the lower pr has all finite moments. Since the mean sojourn time in heavy traffic limit is completely determined by the first two moments, the mean sojourn time of the upper pr in heavy traffic is higher than the mean sojourn time of the lower pr. Further, the mean sojourn times of the upper and lower pr will cross at some arrival rate λ^*, where the mean sojourn time is T^*. Clearly, there are distributions with the given moment constraints with mean sojourn time different than T^* at λ^*. Thus the pr's do not provide bounds in this case. The same behavior is observed whenever the cardinality of moment constraints in the interval $(0, 2)$ is even.

The above discussion, while discomfiting, should be taken as an instructive caution. While we strive to prove a Markov-Krein characterization for $M/G/k$ mean sojourn time, conditions more than those in Theorems 3.1 and 3.2 would be needed. We conjecture that the knowledge of the integral moments suffices. However, fractional moments, in general, may not be admissible.

3.4 Bounds for $M/G/1$ Round-Robin

In this section we prove sharp moments-based bounds for the mean sojourn time, $\mathbf{E}\left[T^{M/G/1/RR}\right]$, of an $M/G/1$ round-robin queue with exponentially distributed quantum sizes and CM service distribution in the limit when the arrival rate $\lambda \to 0$. Formally, we consider round-robin scheduling where every time a job gets to the server, the server picks a quantum size i.i.d. from an Exp(ν) distribution..

Lemma 3.1 *Consider an $M/G/1$ Round-Robin system with i.i.d.* $\text{Exp}(\nu)$ *quanta, arrival rate* λ *and the following* H_n *service distribution:*

$$S \sim \begin{cases} \text{Exp}(\gamma_1) & \textit{with probability } q_1 \\ \vdots & \\ \text{Exp}(\gamma_n) & \textit{with probability } q_n \end{cases}$$

As the arrival rate $\lambda \to 0$*:*

$$\mathbf{E}\left[T^{M/G/1/RR}\right] = \mathbf{E}[S]\left(1 + \lambda \mathbf{E}[S]\right) + \frac{\lambda}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{q_i q_j (\gamma_i - \gamma_j)^2}{\gamma_i \gamma_j (\gamma_i \gamma_j + (\gamma_i + \gamma_j)\nu)} + o(\lambda).$$

Proof: As the arrival rate approaches 0, the coefficient of the $\Theta(\lambda)$ term will be dominated by events where (i) a job arrives to an empty system and is interrupted at most once during its stay, or (ii) a job arrives to a system with a yet uninterrupted job already in service, and there are no more arrivals during its sojourn.

Let us consider the case where an $\text{Exp}(\xi)$ job is interrupted by an $\text{Exp}(\chi)$ job. In this case, the mean residual sojourn time of the interrupted $\text{Exp}(\xi)$ job satisfies

$$\begin{aligned} A_{\xi,\chi} &= \frac{1}{\xi + \nu} + \frac{\nu}{\xi + \nu}\left(\frac{1}{\chi + \nu} + \frac{\nu}{\chi + \nu} A_{\xi,\chi} + \frac{\chi}{\chi + \nu}\frac{1}{\xi}\right) \\ &= \frac{1}{\xi}\left(1 + \frac{\xi\nu}{(\xi + \nu)(\chi + \nu) - \nu^2}\right) \end{aligned} \tag{3.1}$$

Similarly, the mean sojourn time of the interrupting $\text{Exp}(\chi)$ job is given by:

$$B_{\chi,\xi} = \frac{1}{\chi}\left(1 + \frac{\chi^2 + \chi\nu}{(\xi + \nu)(\chi + \nu) - \nu^2}\right) \tag{3.2}$$

Returning to our original round-robin system, a tagged class i job arrives to an empty system with probability $(1 - \lambda \mathbf{E}[S])$, and stays there for $\text{Exp}(\gamma_i + \lambda)$ time. With probability $\frac{\lambda}{\lambda + \gamma_i}$, the tagged class i job gets interrupted by another arrival which is of class j with probability q_j and spends additional time A_{γ_i,γ_j}. With probability $\lambda \mathbf{E}[S]$, the class i job arrives to a busy system and interrupts a class j job with probability $\frac{q_j}{\gamma_j \mathbf{E}[S]}$, in which case the sojourn time of the tagged class i job is B_{γ_i,γ_j}. Thus, the overall sojourn time of a class i job is given by:

$$\mathbf{E}[T_i] = (1 - \lambda \mathbf{E}[S])\left(\frac{1}{\gamma_i + \lambda} + \frac{\lambda}{\gamma_i + \lambda} \sum_j q_j A_{\gamma_i,\gamma_j}\right) + \lambda \mathbf{E}[S] \sum_j \frac{q_j}{\gamma_j \mathbf{E}[S]} B_{\gamma_i,\gamma_j} + O(\lambda^2)$$

$$= \frac{1+\lambda \mathbf{E}[S]}{\gamma_i} + \frac{1}{\gamma_i} \sum_{j=1}^{n} q_j \frac{\gamma_i - \gamma_j}{\gamma_j(\gamma_i\gamma_j + (\gamma_i+\gamma_j)\nu)} \tag{3.3}$$

Calculating $\mathbf{E}\left[T^{M/G/1/RR}\right] = \sum_i q_i \mathbf{E}[T_i]$, we get the expression in the theorem statement. ∎

Theorem 3.6 *Given the first n moments of the service distribution S in the CM class,* $\mathbf{E}\left[T^{M/G/1/RR}\right]$ *under light traffic is extremized by the lower and upper principal representations of the moment sequence within the class of hyperexponential distributions.*

Proof: We will follow similar steps as in the proof of Theorem 3.5. The first is to restrict our attention to hyperexponential distributions with finite number of phases as they are dense in the CM class. We will then use the Markov-Krein Theorem to show that $\mathbf{E}\left[T^{M/G/1/RR}\right]$ given in Lemma 3.1 is extremized by the principal representations within the hyperexponential class of distributions. Let Y denote a random variable with the same distribution as the inverse spectrum of the service distribution S, and let $x_i = \frac{1}{\gamma_i}$. From Lemma 3.1 (ignoring $o(\lambda)$ terms):

$$\begin{aligned}
&\mathbf{E}\left[T^{M/G/1/RR}\right] \\
=&\mathbf{E}[S]\,(1+\lambda\mathbf{E}[S]) + \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{q_i q_j(\gamma_i-\gamma_j)^2}{\gamma_i\gamma_j(\gamma_i\gamma_j+(\gamma_i+\gamma_j)\nu)} \\
=&\mathbf{E}[Y]\,(1+\lambda\mathbf{E}[Y]) + \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{q_i q_j(x_i-x_j)^2}{(1+(x_i+x_j)\nu)} \\
=&\mathbf{E}[Y]\,(1+\lambda\mathbf{E}[Y]) + \frac{\lambda}{2}\sum_{i=1}^{n} q_i \sum_{j=1}^{n} \frac{q_j}{\nu^2}\left(\nu x_j - 1 + \frac{\nu^2 x_i^2 + \nu x_i + 1}{\nu(x_i+x_j)+1}\right) \\
=&\mathbf{E}[Y]\,(1+\lambda\mathbf{E}[Y]) + \frac{\lambda}{2}\sum_{i=1}^{n} q_i \left[\frac{\nu\mathbf{E}[Y]-1}{\nu^2} - \frac{\nu^2 x_i^2 + \nu x_i + 1}{\nu^2}\mathbf{E}[f_i(Y)]\right]
\end{aligned}$$

where $f_k(x) = \frac{1}{\nu(x+x_k)+1}$. From Theorem 31 of [86], each $f_k(x)$ forms a Tchebycheff system with the functions $i!x^i$ (and the same pr, either upper or lower, minimizes each $\mathbf{E}[f_k(Y)]$, and similarly the other pr maximizes each $\mathbf{E}[f_k(Y)]$), and hence by Theorem 3.1, the result follows. ∎

Remark 2: Given $\mathbf{E}[S]$ and $\mathbf{E}[S^2]$, the lower bound within the CM distributions is attained by the upper pr, and is equal to the mean sojourn time under Exponential service distribution, $\mathbf{E}\left[T^{M/M/1/RR}\right]$, which also equals the mean sojourn time under

ideal Processor Sharing discipline, $\mathbf{E}\left[T^{PS}\right]$. The upper bound is attained by the lower pr, and can be shown to be [66]:

$$\mathbf{E}\left[T_h^{M/G/1/RR}\right] = \mathbf{E}\left[T^{M/M/1/RR}\right]\left[1 + \frac{C_S^2+1}{C_S^2+1} \cdot \frac{\lambda}{\nu + \frac{2}{\mathbf{E}[S](C_S^2+1)}}\right]. \tag{3.4}$$

As $C_S^2 \to \infty$, this upper bound converges to $\mathbf{E}\left[T^{M/M/1}\right]\left[1+\frac{\lambda}{\nu}\right]$. The $M/G/1/RR$ system therefore exhibits **bounded-sensitivity** – the effect of higher order characteristics beyond the mean is bounded if ν is bounded away from 0 (when $\nu \to 0$, the Round-Robin policy becomes FCFS).

3.5 Bounds for systems with time-varying load

In this section we prove tight moment-based bounds for an $M/M/1$ queue with arrival and service rates controlled by a 2-state environment process. However, we believe the results extend to much more general time-varying systems (see the remark after Theorem 3.8). The asymptotic regime we consider is what we call the "fast-switching asymptote": we let the duration of stay in the environment states on each visit approach 0. In Theorem 3.7, we prove the result when the distributions for the durations of environment states are CM, but our proof extends to generally distributed durations. In Section 3.5.2 we show an application of our results to the development of (conjectured) tight bounds on the performance of work-stealing, an exact analysis of which is impossible since the Markov chain for the work-stealing model is infinite in 2 dimensions.

Formally, we consider a system with an exogenous environment process with states L and H. The durations of the H states are i.i.d. according to a random variable τ_H, and those of L states are i.i.d. according to τ_L. The job sizes are i.i.d. exponential with mean 1. However, during the L state, the arrival process is Poisson with rate λ_L and the server's service rate is μ_L. Similarly, during the H states, the arrival process is Poisson with rate λ_H and the server's service rate is μ_H. We define $\mu_{avg} = \frac{\mu_L\mathbf{E}[\tau_L]+\mu_H\mathbf{E}[\tau_H]}{\mathbf{E}[\tau_L]+\mathbf{E}[\tau_H]}$, $\lambda_{avg} = \frac{\lambda_L\mathbf{E}[\tau_L]+\lambda_H\mathbf{E}[\tau_H]}{\mathbf{E}[\tau_L]+\mathbf{E}[\tau_H]}$, and $\rho = \frac{\lambda_{avg}}{\mu_{avg}}$. We will consider a sequence of systems indexed by a parameter α, where the durations of L and H states in the αth system are i.i.d. as $\alpha\tau_L$ and $\alpha\tau_H$, respectively. We will analyze the mean number of jobs in this sequence of systems, $\mathbf{E}[N_\alpha]$, as $\alpha \to 0$.

3.5.1 Fast-switching asymptote and bounds

Theorem 3.7 *Consider a time-varying load system with residence time in L and H states given by $\alpha\tau_L$ and $\alpha\tau_H$, respectively. Further, assume that the distributions of τ_L and τ_H are hyperexponential with finite number of phases. Then the mean number of jobs in the system as $\alpha \to 0$ is given by:*

$$\mathbf{E}[N_\alpha] = \frac{\rho}{1-\rho} + \sum_{i=1}^{\infty} \phi_i \alpha^i \tag{3.5}$$

where ϕ_i are functions of the first $i+1$ moments of τ_L and τ_H (and $\mu_L, \mu_H, \lambda_L, \lambda_H$), and are linear in $\mathbf{E}\left[\tau_H^{i+1}\right]$ and $\mathbf{E}\left[\tau_L^{i+1}\right]$.

Proof: We defer the proof to Appendix 3.A. We do not provide the full details but instead illustrate the main ideas behind the proof by looking at a finite buffer system with a buffer size of 1 (i.e., there can only be either 0 or 1 jobs in the system) with time-varying arrival and service rates. The proof easily extends to the infinite buffer case as well. ∎

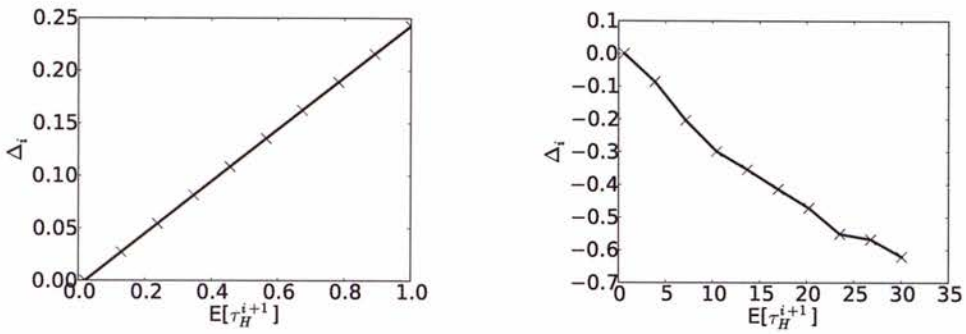
Theorem 3.8 *If τ_L and τ_H are constrained to lie in the CM class, then given the first n moments of τ_L and τ_H, the mean number of jobs, $\mathbf{E}[N]$, under the fast switching asymptote is extremized by the lower and upper principal representations of the moment sequence within the hyperexponential distribution.*

Proof: Given the first n moments of τ_L and τ_H, the coefficients of α^i for $0 \leq i \leq n-1$ are already fixed. The distributions which extremize the mean number of jobs will be those that extremize the coefficient of α^n. Since this is linear in the $(n+1)$st moments, and moment functions $f_i(x) = x^i$ form a Tchebycheff system, the theorem follows from Theorem 3.1. ∎

Remark 3: The result of this section easily extends to the case of general distributions for τ_L and τ_H. The only fact that is needed is that for any finite x, the probability of i arrivals or departures in duration αx is $c_i(\alpha x)^i - d_i(\alpha x)^{i+1} + o(\alpha^i)$ for some constants c_i and d_i – a simple consequence of the Poisson process.

Remark 4: The results of this section should also extend to more general time-varying systems. For example, during the L and H, the system could evolve according to arbitrary finite-state Markov processes with generators Q_L and Q_H, respectively, as long as the characteristic polynomials of Q_L and Q_H ($\phi_L(s) = det(sI - Q_L)$, $\phi_H(s) = det(sI - Q_H)$) do not have repeated roots.

Remark 5: Unlike $M/G/k$ and $M/G/1$ round-robin models where the heavy-traffic limits tend to be insensitive to the job size distribution beyond the first or second



(a) $i = 1$ (b) $i = 2$

Figure 3.3: Dependency of ϕ_i (Theorem 3.7) on $\mathbf{E}\left[\tau_H^{i+1}\right]$

moments, for the time-varying load model, we actually do have an interesting result in the "slow switching asymptote" ($\alpha \to \infty$): in the special case when $\tau_H \sim \text{Exp}(\gamma)$ and $\lambda_H > \mu_H$. Under transient overload during H states, as the mean durations of H and L states become long, the time-varying load system converges to a fluid system. For the special case mentioned, it is not hard to see that the mean response time of this fluid system can be derived from a $GI/M/1$ system with inter-arrival time distribution given by τ_L. As stated earlier, characterization of bounds for $GI/M/1$ via principal representations is known from the work of Eckberg [50]. We have proved that this characterization also holds under the fast switching asymptote, irrespective of the choice of arrival and service rates, and when both L and H state durations may be generally distributed.

We validate Theorem 3.7 numerically with Figure 3.3. Consider a time-varying load system with $\lambda_L = 4$, $\lambda_H = 8$, $\mu_L = \mu_H = 10$. We let τ_L have an exponential distribution with rate 10 and vary τ_H as is specified in the following. In Figure 3.3(a), τ_H has a two-phase hyperexponential ($H_2^\star$) distribution having a non-zero exponential phase and a phase of zero mean. The $H_2^\star$ allows us to hold the mean at 0.1 and vary the second moment $\mathbf{E}[\tau_H^2]$, which is indicated along the horizontal axis. The vertical axis shows $\Delta_i \equiv (\mathbf{E}[N_\alpha] - \mathbf{E}[N'_\alpha])/\alpha^i$ for $i = 1$, where N'_α indicates N_α with $\mathbf{E}[\tau_H^2] = 0.02$ (the lowest value studied in Figure 3.3(a)). Throughout we set $\alpha = 10^{-3}$, so that $o(\alpha^{i+1})$ terms are negligible relative to $\Theta(\alpha^i)$ term. Because $\mathbf{E}[\tau_H]$ is fixed, Δ_1 shows (approximately) how ϕ_1 depends on $\mathbf{E}[\tau_H^2]$. Indeed, we find that ϕ_1 grows linearly with $\mathbf{E}[\tau_H^2]$. In Figure 3.3(b), τ_H has a two-phase hyperexponential (H_2) distribution, which allows us to hold $\mathbf{E}[\tau_H] = 0.1$ and $\mathbf{E}[\tau_H^2] = 0.2$, and vary $\mathbf{E}[\tau_H^3]$, which is indicated along the horizontal axis. The vertical axis in Figure 3.3(b) shows Δ_2 (here, N'_α represents N_α with $\mathbf{E}[\tau_H^3] = 0.601$ (the lowest value studied in Figure 3.3(b))), which indicates (approximately) how ϕ_2 depends on $\mathbf{E}[\tau_H^3]$. Although

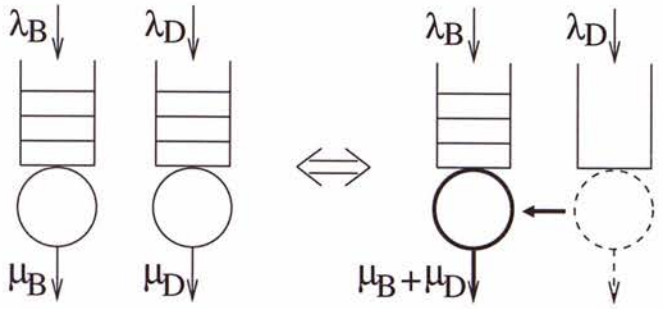


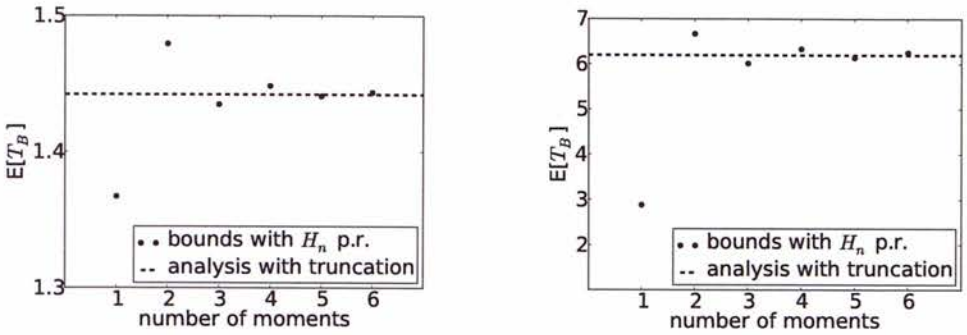
Figure 3.4: The N-sharing model – Service rate at the beneficiary queue is $\mu_B + \mu_D$ when the donor queue is empty and μ_B otherwise.

the line in Figure 3.3(b) is not as straight as Figure 3.3(a) due to numerical errors, we find that ϕ_2 does decrease linearly with $\mathbf{E}[\tau_H^3]$.

3.5.2 Application to analysis of N-sharing model

In this section, we apply the analysis of the time-varying load system to a work-stealing system with two $M/M/1$ queues, beneficiary and donor (see Figure 3.4). The two queues are independent except that the service rate at the beneficiary queue becomes larger when the donor queue is empty (thus the donor server can help the benefeciary queue, but not the other way round). Let λ_B (respectively, λ_D) be the arrival rate at the beneficiary (respectively, donor) queue. Let μ_B (respectively, μ_D) be the service rate at the beneficiary (respectively, donor) queue when the donor queue is nonempty. When the donor queue is empty, the service rate at the beneficiary queue becomes $\mu_B + \mu_D$. We assume that the jobs are preemptive, so that the service rate at the beneficiary queue changes from $\mu_B + \mu_D$ to μ_B immediately after a job arrives at the empty donor queue. The jobs arriving at the donor queue see a standard $M/M/1$ system with arrival rate λ_D and service rate μ_D.

Observe that the jobs arriving at the beneficiary queue, which we refer to as beneficiary jobs, see a time varying system, where $\lambda_H = \lambda_L = \lambda_B$, $\mu_H = \mu_B$, $\mu_L = \mu_B + \mu_D$, τ_L has an Exponential distribution with rate λ_D, and τ_H is the busy period of the $M/M/1$ system with arrival rate λ_D and service rate μ_D. To analyze the response time of beneficiary jobs, we need to consider a Markov chain that is infinite in two dimensions, where one dimension represents the number of beneficiary jobs and the other dimension represents the number of donor jobs. Such a Markov chain cannot be solved exactly, so that the prior work has investigated various approximations (e.g., truncation in [65] and approximating the donor busy period with moment matching



(a) $\rho = 0.6$ (b) $\rho = 0.9$

Figure 3.5: Bounding mean response time of beneficiary jobs in the N-sharing model. The system parameters are $\mu_B = \mu_D = 1$ and $\lambda_B = \lambda_D = \rho$.

in [122]). However, such approximations do not guarantee their accuracy and can be computationally expensive.

Now, because the busy period of an $M/M/1$ system has a hyperexponential distribution with a continuous spectrum [4], our results in Section 3.5.1 suggest that the stationary mean response time of beneficiary jobs, $\mathbf{E}[T_B]$, is likely to be extremized by lower and upper principal representations, given the first n moments of the busy period for $n = 1, 2, \ldots$. Figure 3.5 shows $\mathbf{E}[T_B]$ and the bounds obtained with principal representations. We fix $\mu_B = \mu_D = 1.0$ and vary $\rho = \lambda_D = \lambda_B$ as indicated below each figure. The dashed line shows the exact value of $\mathbf{E}[T_B]$, which is obtained by numerically analyzing a Markov chain. Here the state space of the Markov chain is truncated so that the number of jobs at the donor queue is at most a threshold, 200, and we verify that increasing the threshold does not change the results of the analysis. The resulting Markov chain is a quasi-birth-and-death (QBD) process that can be analyzed with a matrix analytic method.

A dot in Figure 3.5 shows a bound on $\mathbf{E}[T_B]$ obtained by replacing the busy period with a principal representation, where the number of moments used to determine the principal representation is shown on the horizontal axis. Here again, the bound is numerically computed by analyzing a QBD process via matrix analytic methods (but due to a small number of levels, the computational cost is much lower than truncation). Observe that the principal representations obtained by including an additional odd moment (1, 3, and 5 are shown in the figure) refine the lower bound on $\mathbf{E}[T_B]$, while the upper bound is refined by principal representation obtained via an additional even moment (2, 4, and 6 are shown in the figure). When five or six moments are used, the upper bound and the lower bound give nearly exact value (specifically, the two bounds differ by 0.62% in Figure 3.5(a) and 2.2% in

Figure 3.5(b)).

The results in Figure 3.5 justify the approximation in [122], where the donor busy period is approximated by matching its first three moments. The lower bound obtained with the first three moments gives a nearly perfect approximation, and using the fourth and higher moments do not significantly improve the bound. In determining the principal representations for the busy period, B, we have used the following expression obtained by manipulating the Laplace-Stieltjes transform of B (we omit the details): $\mathbf{E}\left[B^k\right] = \frac{k!}{\mu_D^k(1-\rho_D)^{2k-1}}\xi_k$, where $\rho_D = \lambda_D/\mu_D$, $\xi_1 = \xi_2 = 1$, $\xi_3 = 1+\rho_D$, $\xi_4 = 1+3\rho_D+\rho_D^2$, $\xi_5 = 1+6\rho_D+6\rho_D^2+\rho_D^3$, and $\xi_6 = 1+10\rho_D+20\rho_D^2+10\rho_D^3+\rho_D^4$.

3.6 Conjectures on tight bounds for general traffic

Let $\mathbf{m} = (m_0 = 1, m_1, m_2, \ldots, m_n) \in \mathbb{R}_+^n$ be such that there exists a positive random variable $\mathcal{X}$ with $\mathbf{E}[\mathcal{X}^i] = m_i,\ i = 0, \ldots, n$. For n odd, let $\mathcal{D}(\mathbf{m})$ denote the unique lower pr with moments $\mathbf{m}$ and support $[0, \infty)$ (and therefore has mass at ∞), and let $\mathcal{D}_B^*(\mathbf{m})$ denote the unique upper pr with moments $\mathbf{m}$ and support $[0, B]$. For n even, let $\mathcal{D}^*(\mathbf{m})$ denote the unique lower principal representation with moments $\mathbf{m}$ and support $(0, \infty]$, and let $\mathcal{D}_B(\mathbf{m})$ denote the unique upper pr with moments $\mathbf{m}$ and support $[0, B]$. (The star in the superscript is to emphasize a point mass at 0, and the B in the subscript emphasizes the point mass at the upper bound, B, of the support.)

Let

$$T_h(\mathbf{m}) = \sup_{\mu_X \in \mathcal{D}} \left\{ \mathbf{E}\left[T^{\mathcal{S}(X)}\right] \middle| \mathbf{E}\left[X^i\right] = m_i,\ i = 0, \ldots, n \right\},$$
$$T_l(\mathbf{m}) = \inf_{\mu_X \in \mathcal{D}} \left\{ \mathbf{E}\left[T^{\mathcal{S}(X)}\right] \middle| \mathbf{E}\left[X^i\right] = m_i,\ i = 0, \ldots, n \right\}.$$

where $\mathcal{S}(X)$ represents either the $M/G/k$ ($k \geq 2$) multi-server system, the $M/G/1$ round-robin system, or the time-varying load system, with X as the random variable for the service distribution for the $M/G/k$ and the $M/G/1$ round-robin models, or the duration of the L or H states for the time-varying load model, and T denotes the response time.

Conjecture 3.1 *Let* $\mathbf{m} = (m_0 = 1, m_1, \ldots, m_n)$, $n \geq 1$, *be a valid moment sequence for positive distributions. Let* $\mathbf{m}' = (m_0, m_1, \ldots, m_{n-1})$. *Then,*

Case 1: *n odd*

(i) $T_h(\mathbf{m}) = \lim_{B\to\infty} \mathbf{E}\left[T^{\mathcal{S}(\mathcal{D}_B^*(\mathbf{m}))}\right]$.

(ii) $T_l(\mathbf{m}) = \mathbf{E}\left[T^{\mathcal{S}(\mathcal{D}(\mathbf{m}))}\right]$.

(iii) $T_l(m_1, \ldots, m_{n-1}, x)$ *is strictly decreasing in* x.

Case 2: n ***even***

(i) $T_h(\mathbf{m}) = \mathbf{E}\left[T^{\mathcal{S}(\mathcal{D}^*(\mathbf{m}))}\right]$.

(ii) $T_l(\mathbf{m}) = \lim_{B\to\infty} \mathbf{E}\left[T^{\mathcal{S}(\mathcal{D}_B(\mathbf{m}))}\right]$.

(iii) $T_h(m_1, \ldots, m_{n-1}, x)$ *is strictly increasing in* x.

Further, for the $M/G/k$ *system: for* n *odd,* $T_h(\mathbf{m}) = T_h(\mathbf{m}')$*; and for* n *even (and additionally for* $\rho < (k-1)$ *if* $n = 2$*),* $T_l(\mathbf{m}) = T_l(\mathbf{m}')$. [1]

To state in simple language, the conjectures would imply the following for the $M/G/k$ multi-server model: If we are given only the mean of the service distribution, we only have enough information to fix a lower bound on $\mathbf{E}\left[W^{M/G/k}\right]$. This lower bound is given by $\mathbf{E}\left[W^{M/D/k}\right]$. If we are additionally given the second moment of the service distribution, we can fix an upper bound on $\mathbf{E}\left[W^{M/G/k}\right]$ (The conjectured upper bound is presented in Conjecture 2.1). By determining the third moment of service distribution, we can *refine* (tighten) our lower bound but this *lower bound is a decreasing function of the third moment.* The upper bound remains unchanged. Similarly, knowledge of the fourth moment will *refine the upper bound* on the mean waiting time (bring it down), and so forth for alternating higher even and odd moments. Further, these bounds are achieved by mixtures of point masses as dictated by the upper and lower pr's.

3.7 Towards a unified approach for moments-based bounds

While our results offer an intuitive justification for tight moments-based bounds via principal representations for the three queueing systems considered for general (i.e.,

[1] Intuitively, as we said before, this is true because the mass at ∞ is only present to satisfy the largest moment constraint. Karlin and Studden write ([90], page 152), "Whenever mass at ∞ is present, this mass may be ignored to obtain a measure representing only the moments $m_0, m_1, \cdots, m_{n-1}$." In the classical Markov-Krein framework, this treatment suffices under some conditions on the function $g(\cdot)$ whose expectation we are extremizing. However for queueing systems, whenever the sup/inf as defined above exist and involve the upper principal representation, we need to be slightly more careful. For example, for the case of $M/G/k$ with $n = 2$ and $\rho \geq (k-1)$ we can not ignore the mass at infinity and must define the sup/inf via the limit of a sequence of systems involving upper pr on finite support. This fact is highlighted via $M/G/1$ where given $n = 2$, the mean sojourn time is completely determined. However, if we ignore the mass at ∞ in the upper pr, we incorrectly obtain $\mathbf{E}[T^{M/D/1}]$!

non-asymptotic) traffic conditions, we are still quite far from proving the desired result. Further, we believe that similar results are likely to hold for other queueing systems as well. We now discuss some possible lines of attack for proving moments-based bounds for general queueing systems.

One line of approach to proving such results might be along what we have tried to do in the present chapter. One would first prove the desired result in an "appropriate" asymptotic regime, that is, where the effect of the entire distribution of the parameter of interest (e.g., the service distribution) is apparent. This is expected to be the easier step, and should offer insights into what distributions are extremal. The remaining open question would then be to prove that the extremality of the conjectured distributions is preserved when we are in non-asymptotic regime. This last step seems very challenging because there exist service distributions whose relative performance flip while going from light to heavy traffic.[2]

While the above approach sounds promising in that obtaining extremal distributions in asymptotic regimes would be tractable, proving such results for every new queueing system *ab initio* would be far from elegant.

A second line of approach could be that of Eckberg [50] for obtaining bounds on the mean response time of the $GI/M/k$ model. As we mentioned earlier, the mean response time of a $GI/M/k$ queue can be written as an increasing function of an implicit quantity σ that is itself an increasing function of the Laplace-Stieltjes transform $\tilde{A}(s) = \mathbf{E}\left[e^{-sA}\right]$ of the inter-arrival time duration A:

$$\sigma = \tilde{A}(\mu(1-\sigma)).$$

The functions e^{-sx} form a Tchebycheff system with moment functions x^i. Therefore from Theorem 3.1, the principal representations of the moment sequence would extremize the Laplace-Stieltjes transform point-wise, and hence the mean response time of the $GI/M/k$ queue. Employing a similar approach for the mean response time of queueing systems considered in this chapter by expressing these quantities as increasing functions of $\mathbf{E}[f(X)]$ for some function f which forms a Tchebycheff system with $f_i(x) = x^i$, and then directly applying Theorem 3.1, eludes us (and in light of the discussion in Section 3.3.4, seems not possible).

To overcome the above shortcomings, we propose a unified framework by posing the following *moment problem*: Observe that the solution to any queueing system can be

[2]Indeed, consider moment sequences $\mathbf{m} = (m_1, m_2)$ and $\mathbf{m}' = (m_1', m_2')$ with $m_1 = m_1'$ and $(m_1)^2 < m_2 < m_2'$. The lower pr of $\mathbf{m}$ yields a higher mean sojourn time than the upper pr of $\mathbf{m}'$ in light traffic. However, the mean sojourn time in heavy traffic is completely determined by the first two moments, and hence the lower pr of $\mathbf{m}$ yields a lower mean sojourn time than the upper pr of $\mathbf{m}'$ in heavy traffic. Also see the discussion in Section 3.3.4.

represented at some level by the fixed point of a stochastic recursive sequence (SRS). That is, there exists Φ such that

$$\mathbf{W} \stackrel{d}{=} \Phi(\mathbf{W}, S), \tag{3.6}$$

where $\mathbf{W}$ is the unknown random vector capturing the performance of the system, and $\stackrel{d}{=}$ denotes equality in distribution. For example, for the $GI/G/1$ FCFS model, the distribution of the customer average waiting time W is given by the Lindley recursion:

$$W \stackrel{d}{=} (W + S - A)^+$$

where S is the service distribution, and A is the inter-arrival time distribution. As another example, for the $GI/G/k/FCFS$ queueing system, let $\mathbf{W} = (W_1, W_2, \ldots, W_k)$ where $W_1 \leq W_2 \leq \ldots \leq W_k$ denote the Kiefer-Wolfowitz workload vector seen by arriving customer (equivalently, the ordered vector of times at which the k servers will idle, assuming the customer arriving at time $t = 0$ has size 0 and and there are no further arrivals). The distribution of $\mathbf{W}$ is then given by:

$$\mathbf{W} \stackrel{d}{=} \mathscr{R}\left((\mathbf{W} + X \cdot \mathbf{e_1} - A \cdot \mathbf{e})^+\right)$$

where $\mathbf{e_1}$ is a $k-$vector whose first element is 1 and the rest are 0, $\mathbf{e}$ is a k-vector all of whose elements are 1, and $\mathscr{R}$ is a function that reorders the elements of its argument in ascending order.

The final performance metric of interest would be $\mathbf{E}[g(\mathbf{W})]$ for some function g. Our goal is to seek bounds on $\mathbf{E}[g(\mathbf{W})]$, given the first n moments of X. *For what class of probability flows $\Phi(\cdot)$ and functions $g(\cdot)$ can these bounds be characterized along the Markov-Krein Theorem?*

Even partial progress on the above moment problem promises to yield bounds on many interesting queueing systems in a single shot – one only needs to check whether the SRS for the queueing system satisfies certain conditions. Further, an understanding of this problem should give insights into the common thread among queueing systems which share the Markov-Krein characterization property, but are otherwise seemingly very different. For example, what is the fundamental difference between the queueing systems described above and the following queueing system for which the principal representations achieve *identical* mean sojourn time (when n, the number of moment constraints, is even), yet the mean sojourn time is sensitive to the service distribution?

A queueing system where principal representations are non-extremal Consider a 2-server JSQ-PS system with Poisson arrival process : each server follows the

ideal Processor Sharing (PS) scheduling discipline, and new arrivals join the shorter queue (ties broken randomly, no jockeying between queues). It is easy to see that given any first n moments *with n even*, the service distributions corresponding to the upper and lower pr's yield identical mean sojourn time. Consider the case $n = 2$ – the mass at ∞ in the upper pr does not influence the mean sojourn time; jobs of size 0 in the lower pr depart the PS servers instantaneously on arrival. Thus both the upper and lower pr systems effectively behave as if the service distribution is deterministic (albeit, with different means; the arrival process is still Poisson but with different rates). This in turn implies that the distribution for the number of jobs in the upper and lower pr systems are identical, and thus by Little's law, so are the mean sojourn time. A formal proof is given after Theorem 5.1.

While the upper and lower pr yield the same mean sojourn time, this system is sensitive to the service distribution. Bonald and Proutiére [28] have proved that local balance is a necessary and sufficient condition for insensitivity, whereas shortest queue routing with static node capacities violates the local balance condition.

3.8 Summary and Open Questions

In this chapter we have taken a small but fundamental step towards solving three queueing systems which have not yielded exact analysis so far, one of them being the classical $M/G/k$ multi-server system whose analysis has remained open for more than 50 years. Our approach is different from prior attempts in the literature in that instead of trying to obtain an explicit expression for the mean response time as a function of the service distribution, we strive to identify the service distributions with given first n moments which minimize or maximize the mean response time, thus obtaining sharp lower and upper bounds on the mean response time given a partial characterization of the service distribution in terms of its moments.

We were initially motivated by experimental observations made in Chapter 2, and further emboldened by existence of results similar in spirit in the seemingly disconnected area of moment problems. To bridge this disconnect, our approach relied on looking at appropriate tractable asymptotic regimes where the effect of the entire service distribution is apparent (unlike heavy traffic regimes, for example), and extracting the extremal distributions. As our major contribution, we utilized the Markov-Krein theorem to prove that if the service distribution is restricted to lie in the completely monotone (CM) class of distributions, then given any first n moments, the extremal distributions are the principal representations within the hyperexponential class of distributions. For the $M/G/k$ multi-server system, we additionally proved that without the restriction of complete monotononicity, and given the first

$n = 2$ or $n = 3$ moments, these extremal distributions are given by the principal representations of the moment sequence. However, we found the Markov-Krein Theorem lacking for the latter purpose.

Finally, analogous to the classical Markov-Krein theorem for scalar functions, we propose exploration of Markov-Krein characterization of solutions of Stochastic recursive equations as a unified approach to identify and study queueing systems permitting moment-based characterization of extrema via principal representations of the moment sequence of the random variables driving them.

Impact: We have given strong analytical evidence for tight moments-based bounds for the mean sojourn time in three, as yet unsolved, queueing systems. However, the contributions of this chapter go beyond the queueing systems that have been the subject of analysis. We have proposed an analytical tool which would open a new area in the century old field of queueing theory. By viewing queueing systems as a special case of solutions of stochastic recursive sequences, we can utilize the rich set of tools from algebraic geometry, functional analysis and approximation theory and approach the problem of obtaining tight moments-bounds for more general queueing systems.

Open Problems: The most tractable question seems to be extending Theorem 3.4 (light-traffic extremality for $\mathbf{E}\left[W^{M/G/k}\right]$ without CM restriction) to higher moments. A deeper question of interest is, when can the ordering of mean sojourn time under light traffic extend to general arrival rates? Is ordering in both the light and heavy traffic asymptotes sufficient? Motivated by observations in Chapter 2, another question of practical relevance is to identify characteristics of the service distribution which would yield sharper bounds than achievable by moments. Can principal representations be identified when the constraints include these more representative characterization of the service distribution? What are the simplest non-trivial solutions to the moment problem proposed in Section 3.7?

3.A Proof of Theorem 3.7

As stated previously, to illustrate the main ideas behind the proof, we will instead consider an $M/M/1/1$ system in the 2-state environment process defined in Section 3.5. For this case, we only need to analyze the time average idle probability. Let p_L and p_H denote the idle probabilities at the *end* of L and H states, respectively, and let $\overline{p_L}$ and $\overline{p_H}$ be the time average idle probabilities during L and H states, respectively. Our focus is not on deriving the precise coefficients of α^i for all i because our goal is not to propose an approximation by extrapolating the fast-switching asymp-

tote (even though we can do so). Instead, we want to identify sufficient functional dependence of these coefficient on the moments of τ_L and τ_H to be able to conclude that principal representations extremize the performance metric of interest.

Let the distributions of τ_L be given by:

$$\tau_L \sim \begin{cases} \text{Exp}(\gamma_1) & \text{with probability } q_1 \\ \vdots & \\ \text{Exp}(\gamma_n) & \text{with probability } q_n \end{cases}$$

We begin with a simple lemma.

Lemma 3.2 *Consider an $M/M/1/1$ system with arrival rate λ and service rate μ. Let $\tau \sim \text{Exp}(\gamma)$, and let $p(t)$ denote the idle probability at time t. Then:*

$$p(\tau) = \frac{p(0) + \frac{\mu}{\gamma}}{1 + \frac{\mu+\lambda}{\gamma}} \tag{3.7}$$

Proof: The Chapman-Kolmogorov equation is given by:

$$\frac{dp(t)}{dt} = -\lambda p(t) + \mu(1 - p(t))$$

Integrating by parts:

$$p(\tau) = \int_0^\infty \gamma e^{-\gamma u} p(u) du = p(0) + \frac{1}{\gamma}\left(\mu - (\lambda + \mu)p(\tau)\right).$$

■

By conditioning on the which of the n phases of the L state duration occurs and using the above lemma, we can obtain p_L in terms of p_H for the αth system as:

$$p_L = \sum_{j=1}^{n} q_j \frac{p_H + \alpha\frac{\mu_L}{\gamma_j}}{1 + \alpha\frac{\mu_L+\lambda_L}{\gamma_j}} \tag{3.8}$$

$$= p_H \left(1 - \alpha(\mu_L + \lambda_L)\mathbf{E}[\tau_L] + \sum_{k=2}^{i+1} \alpha^k \mathbf{E}\left[\tau_L^k\right]\eta_k + \Theta(\alpha^{i+2})\right)$$

$$+ \alpha\mu_L \mathbf{E}[\tau_L] + \sum_{k=2}^{i+1} \alpha^k \mathbf{E}\left[\tau_L^k\right]\zeta_k + \Theta(\alpha^{i+2}) \tag{3.9}$$

where η_k and ζ_k are constants (functions of μ_L and λ_L only). Similarly,

$$p_H = p_L \left(1 - \alpha(\mu_H + \lambda_H)\mathbf{E}[\tau_H] + \sum_{k=2}^{i+1} \alpha^k \mathbf{E}\left[\tau_H^k\right]\theta_k + \Theta(\alpha^{i+2})\right) + \alpha\mu_H \mathbf{E}[\tau_H] + \sum_{k=2}^{i+1} \alpha^k \mathbf{E}\left[\tau_H^k\right]\kappa_k + \Theta(\alpha^{i+2}) \tag{3.10}$$

where, again, θ_k and κ_k are constants (functions of μ_H and λ_H only).

Eliminating p_H,

$$\begin{aligned} p_L = p_L \Big(1 - \alpha\left[(\mu_L + \lambda_L)\mathbf{E}[\tau_L] + (\mu_H + \lambda_H)\mathbf{E}[\tau_H]\right] \\ + \sum_{k=2}^{i} \alpha^k \sigma_k + \alpha^{i+1}\left[\mathbf{E}\left[\tau_L^k\right]\eta_k + \mathbf{E}\left[\tau_H^k\right]\theta_k\right] + \Theta(\alpha^{i+2})\Big) \\ + \alpha\left[\mu_L \mathbf{E}[\tau_L] + \mu_H \mathbf{E}[\tau_H]\right] + \sum_{k=2}^{i} \alpha^k \psi_k \\ + \alpha^{i+1}\left[\mathbf{E}\left[\tau_L^{i+1}\right]\zeta_k + \mathbf{E}\left[\tau_H^{i+1}\right]\kappa_k\right] + \Theta(\alpha^{i+2}) \end{aligned} \tag{3.11}$$

where σ_k and ψ_k for $2 \leq k \leq i$ involve $\mu_L, \mu_H, \lambda_L, \lambda_H$ and $\mathbf{E}[\tau_L^m]$ and τ_H^m for $1 \leq m \leq i$ (importantly, not $\mathbf{E}\left[\tau_L^{i+1}\right]$, $\mathbf{E}\left[\tau_H^{i+1}\right]$, or still higher moments). This gives

$$\begin{aligned} p_L = \frac{\mu_{avg}}{\mu_{avg} + \lambda_{avg}} \Bigg(1 + \frac{\alpha^i}{\mathbf{E}[\tau_L] + \mathbf{E}[\tau_H]} \Bigg[\frac{\mathbf{E}\left[\tau_L^{i+1}\right]\zeta_k + \mathbf{E}\left[\tau_H^{i+1}\right]\kappa_k}{\mu_{avg}} \\ + \frac{\mathbf{E}\left[\tau_L^{i+1}\right]\eta_k + \mathbf{E}\left[\tau_H^{i+1}\right]\theta_k}{\mu_{avg} + \lambda_{avg}}\Bigg] + \sum_{k=1}^{i} \alpha^k \phi_k\Bigg) + \Theta(\alpha^{i+1}) \end{aligned} \tag{3.12}$$

where again ϕ_k for $1 \leq k \leq i$ only involve μ, λ, and the first i moments of τ_L and τ_H. A similar expression holds for p_H. Note that as $\alpha \to 0$, the idle probability of the finite buffer system is indeed given by $\frac{\mu_{avg}}{\mu_{avg}+\lambda_{avg}}$.

Finally, the expression for the time avergae idle probability during L states is obtained as:

$$\overline{p_L} = \frac{1}{\mathbf{E}[\tau_L]} \sum_{j=1}^{n} \frac{\frac{q_j}{\gamma_j}\left(p_H + \alpha\frac{\mu_L}{\gamma_j}\right)}{1 + \alpha\frac{\mu_L + \lambda_L}{\gamma_j}} \tag{3.13}$$

The contributions to the α^i term in $\overline{p_L}$ are made by $O(\alpha^i)$ terms in p_H, and also from $\alpha\frac{q_j\mu_L}{\gamma_j^2}$ term in the numerator above. It is straightforward to see that the coefficient of the α^i term will again depend on only the first $(i+1)$ moments, and will be linear in the $(i+1)$st moments of τ_L and τ_H.

Chapter 4

Scheduling Policies for Database Concurrency Control: The $G/G/$PS-MPL Model

In this part of the thesis, we focus on designing scheduling policies for the smallest building block of a server farm: an individual back-end server. Two of the most common roles of these servers are to act as database servers, or as web/file servers. The architecture of such servers is based on multi-threaded time-sharing. A time-sharing server is commonly modeled using the ideal Processor Sharing service discipline where the server's aggregate service rate is invariant to the number of requests in service. However, database servers exhibit load-dependent service rate: As the number of requests at the server increases, initially the service rate increases due to more efficient use of the resources, but eventually drops due to context switching overheads and thrashing arising from resource contention. To avoid thrashing, servers maintain a constant population of threads thus imposing a limit (called the Multi-Programming-Limit (MPL)) on the maximum number of active threads. Whenever a request arrives and finds an idle thread, the thread is assigned to processing the new request. Requests arriving to find all threads active and busy wait in a buffer. In practice, the MPL is always chosen to maximize the server's capacity. However, there are no analytical results to understand the following questions: *Does choosing the peak efficiency point as the MPL minimize the mean response time? If not, what should the MPL be set to?*

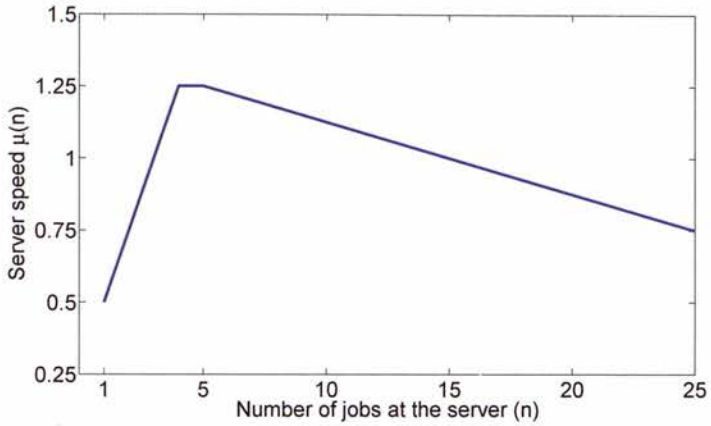


Figure 4.1: A prototypical service rate curve. The peak efficiency point for the curve shown is $K^* = 5$.

4.1 Introduction

The notion of time-sharing has been around since the earliest days of operating systems, as described in the first paper on Unix [130]. Time-sharing has several benefits. First, given that jobs often need different resources (CPU, I/O) at different times, time-sharing allows for increased throughput, typically allowing two jobs to complete in the same time as one, since they aren't likely to need the same resources at the same time. Another major benefit of time-sharing is that it allows small jobs to get out quickly; the small jobs are not stuck queueing behind big jobs as they would be in a first-come-first-served (FCFS) system, and therefore they don't have to suffer the delays of waiting for big jobs to complete.

However, time-sharing is most effective when there is a fixed Multi-Programming-Limit (MPL) imposed, so that not too many jobs time-share at once. Allowing too many jobs to time-share can lead to thrashing (due to the context-switching overhead), and reduce overall performance. This point has been observed repeatedly, starting with operating systems research in the 1970's [48] and 1980's [12, 26], and continuing to more recent research in Web server design [53, 88], and database implementation [77, 134]. Specifically, a system has a service rate curve which shows that the "speed" of the system increases when the number of jobs in the system increases from 1 to 2, and increases again as the number increases from 2 to 3, but the system speed starts to drop as the number of jobs in the system increases beyond some point. Figure 4.1 shows a typical service rate curve (see, e.g. [149, Figure 2]).

Model

To model a time-sharing system, we start with a G/G/1/PS queue where PS denotes "processor sharing," meaning that if there are n jobs in the system, they each receive $\frac{1}{n}$th of the system's processing capacity. Job sizes (or service requirements) are *i.i.d.*, with S denoting such a generic service requirement, and C_S^2 its squared coefficient of variation (SCV). Throughout, we assume that $\mathbf{E}[X] = 1$ without loss of generality.

In order to capture the fact that the speed of the system depends on the number of jobs at the server, we assume that our $G/G/1/PS$ server has state-dependent service rates $\mu(n)$. That is, when the number of jobs at the server is n, the speed of the server is $\mu(n)$, where $\mu(n)$ is chosen to match the system's service rate curve (Figure 4.1). As an example, a job of size x seconds which is sharing the server with n jobs (including itself) for its entire duration would require $\frac{x}{\mu(n)} \cdot n$ time to complete. We assume that the $\mu(n)$ curve is unimodal, that is, initially it is non-decreasing and then after some point the curve switches to being non-increasing. We define K' to be the smallest MPL which achieves the maximum speed, and K^* to be the largest MPL which achieves the maximum speed. For the $\mu(n)$ curve in Figure 4.1, $K' = 4$ and $K^* = 5$.

To complete our model, we now add an MPL parameter which limits the number of jobs that are allowed to concurrently share the server to some number MPL=K, and forces all remaining jobs to wait in a First-Come-First-Served (FCFS) buffer. We assume that the job sizes of the jobs in the system are not known, and size-based prioritization is not possible. We denote our model by the notation G/G/PS-MPL. Figure 4.2 depicts a G/G/PS-MPL system with MPL=4. When we additionally assume the arrival process to be Poisson, we will denote the system by M/G/PS-MPL. Throughout, we assume load-dependent service rates $\mu(n)$. So, for example, if there are $n = 10$ jobs in the G/G/PS-4 system, the server speed will be $\mu(4)$, since only 4 jobs time-share the server, while if there are $n<4$ jobs in the system, the speed will be $\mu(n)$. Thus the response time for a job of size x will be its queueing time plus its service time, where the service time will typically be $\frac{x}{\mu(4)} \cdot 4$, assuming that there are at least 4 jobs in the system during the job's time in system.

The goal of this chapter is, of course, to answer the question:

> *What is the optimal MPL for the G/G/PS-MPL model, so as to minimize mean response time?*

Obviously, the service rate curve plays a large role in the answer, and in fact, almost exclusively the MPL is chosen to maximize efficiency, e.g., [5, 26, 53]. For the curve shown in Figure 4.1, this would mean choosing the MPL to be $K' = 4$ or $K^* = 5$.

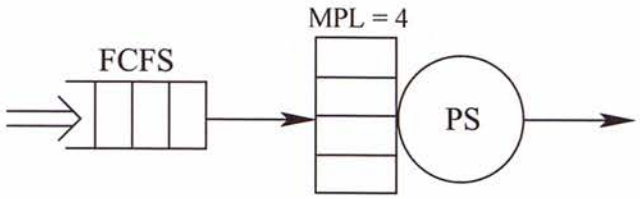


Figure 4.2: A G/G/PS-MPL queue with $MPL = 4$. Only 4 jobs can simultaneously share the server. The rest must wait outside in FCFS order.

Indeed, when the service distribution is exponential, maximizing efficiency is indeed the right answer regardless of the arrival process. The question of choosing the optimal MPL becomes non-trivial when the service distribution exhibits high variability, since with high variability service distributions, it is known that PS yields lower mean response time than FCFS by preventing small jobs from getting blocked behind big jobs. Thus, the optimal MPL must strike the right tradeoff between parallelism and efficiency. Towards this goal, we develop the first approximation for the mean response time of the GI/G/PS-MPL with general service rate curve. Additionally, we propose a novel heavy-traffic diffusion scaling for the GI/G/PS-MPL model, and more generally for non-work-conserving systems, and perform approximate analysis for the stationary distribution of the number of jobs under the proposed scaling. We will also answer the even harder question of how to dynamically vary the MPL when the arrival rate is not known and as load conditions change.

Prior Work

The non-triviality of choosing the optimal MPL for high variability service distributions arises because there is no known analysis even for the M/G/PS-MPL model. This is not surprising because the M/G/PS-MPL model is a generalization of the $M/G/k$ model the performance analysis of which, as we have seen in Chapters 2 and 3, is still largely an open problem. The $M/G/k$ system can be modeled by an M/G/PS-MPL system with MPL $= k$, and $\mu(n) = \mu \cdot n$, where μ is the speed of the individual servers. While the performance analysis of the Processor-Sharing queue has been well understood for years, and research on the M/G/1/PS queue has been abundant [32, 38, 101, 102, 104, 161, 163], very little is known about the M/G/PS-MPL queue. Most analyses of the M/G/PS-MPL queue do not allow for load-dependent service rates. For example, Itzhak and Halfin [16] derive a 2-moment approximation for the mean response time for the M/G/PS-MPL queue where the service rate is fixed, and Zhang and Zwart [162] have recently derived a heavy-traffic

diffusion approximation for $GI/G/$PS-MPL (referred to as the Limited Processor Sharing queue in [162]) with a constant service rate curve. There is one analysis of the $M/G/$PS-MPL that does involve state-dependent service rates, see Rege and Sengupta [127], which however assumes that job sizes are exponentially-distributed while our focus is on high-variability service distributions which are more representative of computer workloads. While Fredericks [61] warns that the exponential service distribution is not a good indicator of performance under high variability, he does not derive an approximation that allows for higher variability. Finally none of the above theoretical papers have tried to answer the question of how to set the MPL so as to minimize the mean response time.

While there is a large body of work on adaptive load control and admission control in resource-sharing systems, all of the existing work either ignores the crucial point of load-dependent service rates at the server, or the effect of job size variability. Elnikety et al. [53] propose monitoring the load of the server and admitting tasks as long as the resulting load does not exceed the peak efficiency point. Blake [26] also proposes operating at the peak efficiency point, but uses the fraction of jobs waiting in the virtual memory queue as an indicator of thrashing to control the MPL. Kamra et al. [88] model the server as an ideal $M/G/1/PS$ system thereby ignoring the state-dependence of the service rate. They monitor the response time of the departing jobs, and adjust the dropping probability of the arriving requests to achieve target response time for the admitted tasks. Our solutions differ from [88] in that we do not drop requests. Heiss and Wagner [77] propose a feedback mechanism to monitor the effect that changing the MPL has on the performance metric of interest. However, as the authors observe, this requires monitoring at least hundreds of departures before a control decision can be taken. Another drawback of the solution proposed in [77] is that the authors assume the system reaches stationarity after the control decision has been taken. This assumption is hardly justified, and can cause incorrect decisions due to a delay between the time the control action is taken, and the time its effect is observed. Schroeder et al. [134] consider the problem of setting a static MPL in the presence of variable job sizes, but the emphasis of [134] is to find a sufficiently small MPL so that jobs waiting in the FCFS buffer can be priority-ordered. Schroeder et al. also develop a feedback based controller based on measuring the throughput and response times, but ignore the state-dependence of service rate. Van der Weij, Bhulai and van der Mei [147] also look at admission control in a PS queue under the assumption that the service distribution is of phase type and the phases of all the jobs in the system are known. The authors assume a constant $\mu(n)$, and characterize the optimal admission control policy. In contrast, we assume that no information about the job sizes is available and hence size-based prioritization is not possible.

Summary of Results

The work presented in this chapter makes at least three contributions:

1. Optimal traffic-aware static policies
We derive the first 2-moment approximation for mean response time for the $M/G/$PS-MPL queue with state-dependent service rates, and extend this approximation to the $GI/G/$PS-MPL model. We had argued in Chapter 2 that no 2-moment approximation can be accurate for the $M/G/k$ model, and hence by extension for the $M/G/$PS-MPL model. Indeed, our approximation is not aimed at accurately predicting the exact mean response time. Rather, our goal is to well-characterize the *behavior* of mean response time as a function of the MPL to enable us to choose the MPL that achieves near optimal mean response time, and we do find this to be the case. Via extensive simulation experiments, we demonstrate that the optimal MPL setting can be much higher than the peak efficiency point under job size variability characteristic of computer workloads. In fact, we show examples where the optimal MPL operates the system at 85% of the peak efficiency, while reducing the mean response time by more than 65% compared to setting the MPL to maximize the service rate. Our results are verified across a variety of service distributions including Weibull, Pareto and Hyperexponential distributions. We refer to the static policy which uses the optimal static MPL as the OPT-STATIC policy.

2. Near-optimal traffic-oblivious dynamic policies
The above results assume jobs arrive according to a Poisson process with a known arrival rate and propose the best *static* MPL. However, we are interested in scenarios where the mean arrival rate may not be known, or the arrival process may not even be Poisson, exhibiting burstiness or temporal correlations. Our goal is to design *light-weight* MPL control policies that adapt to the traffic characteristics. By light-weight policies, we mean policies which take decisions based only on the instantaneous number of jobs in the buffer, $Q(t)$, and the instantaneous number of jobs at the server, $K(t)$.

We first consider the setting where the arrival process is known to be Poisson, but with an unknown mean arrival rate. We find that, unsurprisingly, static MPLs are very poor in handling uncertainty in the mean arrival rate. We then propose two light-weight MPL control policies, LIGHT-APPROX and POISSON-APPROX that robustly handle uncertainty in the mean arrival rate. The ***key idea in our approach*** is that by approximating the original service distribution via a 2-phase degenerate hyperexponential distribution, we are able to incorporate the effect of job size variability in our optimization problem, while $(Q(t), K(t))$ remains a Markov process. Thus, the control policies we obtain are a function only of $(Q(t), K(t))$. Via simulations we show that both LIGHT-APPROX and POISSON-APPROX are robust at

adapting to unknown mean arrival rate, resulting in near-optimal mean response time (under 19%) for a wide range of arrival rates when compared to the optimal static MPLs for each arrival rate. The computation of the POISSON-APPROX policy is enabled by a novel combination of Matrix geometric methods with the policy iteration algorithm which allows us to obtain exact solutions of Markov decision processes with infinite state space, and this technique is likely to be of independent interest.

Next, we consider the setting where not only is the mean arrival rate not known, but the arrival process is also bursty. We demonstrate that both LIGHT-APPROX and POISSON-APPROX are simultaneously robust to unknown mean arrival rate and burstiness of the arrival process, resulting in less than 25% higher mean response time than the mean response time for the optimal traffic-aware static MPL in the worst case. Surprisingly, we find that if the mean arrival rate is known, a static MPL optimized for a Poisson arrival process with the given mean arrival rate is also near-optimal when the arrival process is bursty with that mean arrival rate (that is, the interarrival times are i.i.d. but not exponentially distributed). However, burstiness can greatly worsen the performance of static policies when the mean arrival rate is unknown.

3. The first diffusion scaling and approximation for non-work conserving systems

Diffusion analysis is a powerful tool to approximate the behavior of a physical system as a stochastic process, and its application to queueing theory started with the work of Kingman [98] for single-server systems, and recently, relevant to our work, a diffusion scaling and analysis was proposed for the $GI/G/$PS-MPL model with $\mu(n)$ restricted to be a constant function [162]. Both of these systems are "work-conserving". The PS-MPL model we are considering is non-work-conserving in the sense that depending on the state of the system, the capacity or service rate available can be less than the maximum capacity. While diffusion scalings and approximation exist for specific examples of non-work-conserving systems (e.g., the $GI/G/k$ model, networks of queues), we propose the first approach to systematically obtain heavy traffic diffusion scaling for general non-work-conserving systems. Our **key idea** is to *reverse-engineer* the parameters (service rates) of the systems we aim to approximate so that the limiting distribution of the number of jobs in the system under the scaling converges to the distribution of the number of jobs in the original unscaled system under the tractable $M/M/$ arrival and service processes. Thus the approximation obtained via our proposed scaling is representative of the original system, and we believe that even for the $GI/G/k$ model it would yield a sharper approximation than existing scalings. As mentioned, the real strength of a diffusion scaling is developing a process level approximation for the queueing system of interest, but a rigorous treatment is beyond the scope of the present thesis. In this thesis, we will restrict

ourselves to an approximate analysis for the stationary distribution of the number of jobs under our proposed diffusion scaling.

Outline

In Section 4.2, we propose an approximation to the $M/G/$PS-MPL model and solve the problem of choosing the optimal static MPL for a general service distribution under the assumption that the arrival process is Poisson with a known arrival rate. In Section 4.3, we begin by demonstrating that the approach of choosing a single static MPL is fundamentally limited in its ability to handle variability in traffic arrival patterns. In Sections 4.3.2 and 4.3.3, we construct our dynamic MPL control policies LIGHT-APPROX and POISSON-APPROX, respectively. In Section 4.3.4, we evaluate these dynamic policies with respect to *(i)* robustness to unknown arrival rate, and *(ii)* robustness to burstiness of the arrival process against optimal traffic-aware static MPL policies. In Section 4.4, we present our heavy-traffic scaling for non-work-conserving systems and present preliminary approximations for the stationary behavior.

4.2 Choosing the best static MPL

Our first goal in this chapter is to address the question of *how to optimally set a multi-programming limit* in a resource-sharing system so as to minimize the mean response time (equivalently, minimize the mean number of jobs in the system). We assume that the arrival process is Poisson with a known mean arrival rate, and that the service distribution is known. In Section 4.2.1, we present some stochastic monotonicity results for the performance of PS-MPL systems under fairly general service distributions which motivate the need to appropriately choose the MPL based on the service distribution. In Section 4.2.2, we provide a simple approximation for the mean number of jobs in an M/G/PS-MPL system with state-dependent service rate involving only the first two moments of the service distribution, and demonstrate a service distribution for which the approximation is, in fact, exact. In Section 4.2.3, we present the OPT-STATIC policy, which uses our approximation to choose a static MPL based on the mean arrival rate and the first two moments of the service distribution. Even though our approximation involves only the first two moments of the service distribution, we show via experiments that it leads to optimal or near-optimal MPL selection for a range of distributions used to model computer workloads.

4.2.1 Stochastic monotonicity results

Let F be a distribution function for a non-negative random variable X, and f be the corresponding density function.

Definition 4.1 *Distribution F is said to belong to the class DFR (IFR) if the function $h(x) = \frac{f(x)}{1-F(x)}$ is decreasing (increasing).*

Definition 4.2 *Distribution F is said to belong to the class DMRL (IMRL) if the function $R(a) = \mathbf{E}[X - a | X \geq a]$ is decreasing (increasing).*

The classes IMRL (Increasing Mean Residual Life, also referred to as NWUE for New Worse than Used in Expectation) and DFR (Decreasing Failure Rate) both capture the notion that young jobs (those who have received less service) are more likely to finish earlier than old jobs. The condition DFR is equivalent to saying that the residual life of young jobs is stochastically smaller than the residual life of old jobs, while IMRL is equivalent to saying that the mean residual life of young jobs is smaller than the mean residual life of old jobs.

The following is a corollary of [121, Theorem 1].

Proposition 4.1 *In a $G/G/PS$-MPL system with a DFR service distribution, the number of jobs in the system at any time is a stochastically decreasing function of the MPL K, for $K \leq K^*$. For an IFR distribution, the number of jobs in the system is a stochastically increasing function of the MPL K, for $K \geq K'$.*

A similar proposition can be proved for the mean number of jobs (equivalently mean response time) by relaxing the assumptions on the arrival process and the service distribution.

Proposition 4.2 *In an $M/G/PS$-MPL system with an IMRL service distribution, the mean number of jobs in the system is a decreasing function of the MPL K, for $K \leq K^*$. For a DMRL distribution, the mean number of jobs in the system is an increasing function of the MPL K, for $K \geq K'$.*

Proof: From [128, Theorem 3.14], for IMRL distributions, it suffices to prove that for all x, the quantity $\overline{V}_x$, which denotes the mean workload in the system due to jobs with attained service less than x, is decreasing in the MPL K for $K \leq K^*$. From the proof of [121, Theorem 1], this is easily seen to hold. The proof for DMRL distributions is analogous. ∎

Intuitively, when the service distribution is DFR or IMRL, we prefer to serve young jobs as they are more likely to finish earlier. By choosing an MPL smaller than K^*, we do not gain serving capacity, since K^* achieves the maximum speed, and simultaneously limit the ability of new jobs (which are likely to be small) to enter service. Similarly, for IFR or DMRL service distributions, we prefer to serve old jobs as they are more likely to finish earlier. By choosing an MPL larger than K', we do not gain aggregate serving capacity, and we simultaneously reduce the capacity available to old jobs, as young jobs are allowed into service. Job size distributions belonging to class DFR and IMRL correspond to distributions which are more variable than the exponential distribution, and the above results show that there is no benefit in running at an MPL smaller than K^* in this case. However, there might be benefit in operating at an MPL higher than K^*, increasing the chance for small jobs to enter service and finish quickly even while losing aggregate service capacity in the process, as we show next.

4.2.2 2-moment approximation for *M/G/PS-MPL*

As mentioned earlier, there are no known analytical expressions or approximations for the mean number of jobs in an M/G/PS-MPL system with state-dependent service rate. We now propose a simple approximation for the mean number of jobs in an M/G/PS-MPL system involving only the first two moments of the service distribution.

Proposition 4.3 *Let* $\mathbf{E}[N]$ *denote the mean number of jobs in an M/G/PS-MPL system with arrival rate* λ*, state-dependent service rate* $\mu(n)$ *when there are* n *jobs at the PS server, with MPL=*K*, and a general service distribution with mean 1 and SCV* C_S^2*. Then,*

$$\mathbf{E}[N] \approx \mathbf{E}\left[N^S_{Exp}(K)\right] + \frac{C_S^2+1}{2}\mathbf{E}\left[N^Q_{Exp}(K)\right] \tag{4.1}$$

where $\mathbf{E}\left[N^Q_{Exp}(K)\right]$ *and* $\mathbf{E}\left[N^S_{Exp}(K)\right]$*, respectively, denote the mean number of jobs in the FCFS Queue and at the PS Server in an M/M/PS-MPL with the same state-dependent service rates as the original M/G/PS-MPL system, with MPL=*K *and exponential service distribution with mean 1. The expressions for* $\mathbf{E}\left[N^Q_{Exp}(K)\right]$ *and* $\mathbf{E}\left[N^S_{Exp}(K)\right]$ *are given by:*

$$\mathbf{E}\left[N^Q_{Exp}(K)\right] = \frac{\phi_{K+1}}{1+\sum_{i=1}^{\infty}\phi_i}\left(\frac{1}{1-\frac{\lambda}{\mu(K)}}\right)^2$$

$$\mathbf{E}\left[N^S_{Exp}(K)\right] = \frac{\sum_{i=1}^{K} i \cdot \phi_i + K \cdot \sum_{i=K+1}^{\infty} \phi_i}{1 + \sum_{i=1}^{\infty} \phi_i}$$

where ϕ_i's are the ratio of the stationary probabilities and the idle probability for an M/M/PS-MPL, and are given by:

$$\phi_i = \begin{cases} \Pi_{j=1}^{i} \frac{\lambda}{\mu(j)} & 1 \leq i \leq K, \\ \phi_K \cdot \left(\frac{\lambda}{\mu(K)}\right)^{i-K} & i > K. \end{cases}$$

Proposition 4.3 can be seen as a generalization of the Lee and Longton [108] approximation for the mean number of jobs in an M/G/K system, and agrees with the approximation given by Avi-Itzkah and Halfin when the service rate is independent of the state [16]. In Proposition 4.4, we show that approximation (4.1) is in fact exact for a degenerate hyperexponential distribution, H^*, with mean 1 and squared of coefficient of variation C_S^2.

Recall that a *degenerate hyperexponential distribution* with mean 1 and SCV C^2 is defined by:

$$H^*(C^2) \sim \begin{cases} 0 & \text{with probability } 1 - q = \frac{C^2-1}{C^2+1} \\ \text{Exp}\left(\frac{2}{C^2+1}\right) & \text{with probability } q = \frac{2}{C^2+1} \end{cases}$$

where $\text{Exp}(\nu)$ denotes an exponential random variable with mean $1/\nu$.

Proposition 4.4 *The mean number of jobs in an $M/H^*(C_S^2)$/PS-MPL system with arrival rate λ, state-dependent service rate $\mu(n)$ when there are n jobs at the PS server, and MPL=K is given by:*

$$\mathbf{E}\left[N_{H^*(C_S^2)}(K)\right] = \mathbf{E}\left[N^S_{Exp}(K)\right] + \frac{C_S^2 + 1}{2}\mathbf{E}\left[N^Q_{Exp}(K)\right]$$

where $\mathbf{E}\left[N^Q_{Exp}(K)\right]$ and $\mathbf{E}\left[N^S_{Exp}(K)\right]$ are as defined in Proposition 4.3.

Proof: We first observe that the $H^*(C_S^2)$ distribution consists of two classes of jobs, those of size 0 and those belonging to the exponential branch. The response time and hence the number of jobs belonging to the exponential class in the $M/H^*(C_S^2)$/PS-MPL system is not affected by the presence of zero-sized jobs. Therefore, the contribution to the mean number of jobs in the system consisting of jobs in the exponential class is precisely $\mathbf{E}\left[N^S_{Exp}\right] + \mathbf{E}\left[N^Q_{Exp}\right]$. The zero-sized jobs only contribute to the mean number in queue. However, since the scheduling policy is size-independent, the waiting time distribution of a zero-sized job is the same as the waiting time distribution

of a job belonging to the exponential class, but the arrival rate of zero-sized jobs is $\frac{C_S^2-1}{2}$ times the arrival rate of the exponential class. Therefore, the contribution of the zero-sized jobs to the mean number in system is $\frac{C_S^2-1}{2}\mathbf{E}\left[N_{Exp}^Q\right]$, proving the proposition. ∎

In Section 4.2.4 we extend Proposition 4.3 to obtain an approximation for a $GI/G/$PS-MPL system involving the first two moments of the interarrival time and service distributions.

4.2.3 The Opt-Static policy

We now introduce the OPT-STATIC policy to choose a near-optimal static MPL. The OPT-STATIC policy simply sets MPL $= \kappa$ where κ denotes the MPL that minimizes the right hand side of (4.1):

$$\kappa = \arg\min_K \left\{ \mathbf{E}\left[N_{Exp}^S(K)\right] + \frac{C_S^2+1}{2}\mathbf{E}\left[N_{Exp}^Q(K)\right] \right\} \tag{4.2}$$

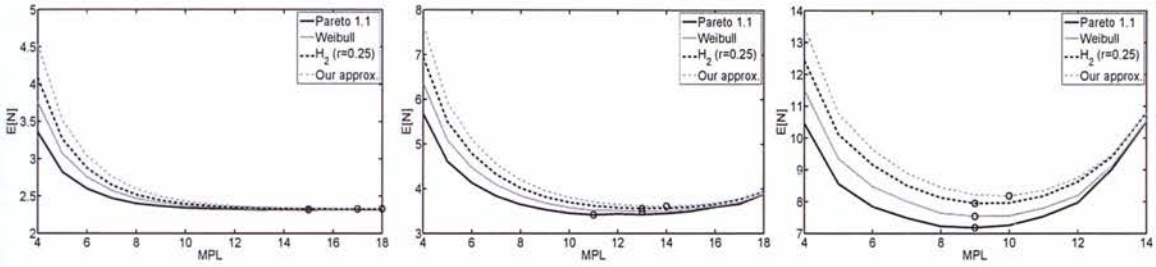
We now show that the OPT-STATIC policy is a good heuristic for minimizing the mean response time in an M/G/PS-MPL system with known mean arrival rate. In Figure 4.3, we present simulation results for the following three service distributions *all with mean 1 and* C_S^2*=19*:

- Weibull distribution with scale parameter $\frac{1}{6}$ and shape parameter $\frac{1}{3}$.
- Bounded Pareto distribution with shape parameter $\alpha = 1.1$ and support $[0.182, 178.759]$.
- A two-phase hyperexponential (H_2) distribution whose parameters are chosen so that, r, the fraction of the total load constituted by the phase with the smaller mean, is 0.25.

The results in Figure 4.3 assume that the state-dependent service rates of the PS server are given by the $\mu(n)$ curve shown in Figure 4.1. We will use the service rate curve shown in Figure 4.1 in all the numerical and simulation evaluations in this paper. Detailed simulation results for more scenarios appear in [68].

The main message of Figure 4.3 is that the optimal MPL can be much larger than the peak efficiency MPL of $K^* = 5$. For example, when $\lambda = 0.8$, the optimal static MPL for the bounded Pareto distribution is 11 with a resulting mean number of jobs around 3.4, while $K^* = 5$ results in 35% larger mean number of jobs at approximately 4.6. Second, as can be seen, even though approximation (4.1) is not

extremely accurate at predicting the mean number of jobs in the system for general distributions (and, as we have mentioned repeatedly, no approximation based on only the first two moments can be), it is robust in predicting the optimal or near-optimal MPL. Our approximation recommends MPL = 14 and the mean number of jobs in the system using our recommended MPL is around 3.45.



(a) $\lambda = 0.7$ (b) $\lambda = 0.8$ (c) $\lambda = 0.9$

Figure 4.3: The mean number of jobs in the system vs. MPL for the following distributions, all with mean 1 and SCV 19: (i) Bounded Pareto distribution with shape parameter 1.1 (ii) Weibull distribution (iii) Two-phase hyperexponential distribution with 25% of load constituted by the branch with the smaller mean. The arrival process considered is Poisson with the indicated mean arrival rate, λ. For reference, we have also shown our 2-moment approximation for the mean number of jobs in the system. The optimal MPL for each curve is shown with a circle.

Using approximation (4.1), it is easy to see why the mean number of jobs in the system is minimized at a larger MPL than the peak efficiency MPL of K^* when job sizes have high variability. To see this, start by considering the case of low variability: $C_S^2 = 1$. For this case, approximation (4.1) suggests that the optimal MPL is in fact K^*. As we increase the MPL beyond K^*, if the traffic intensity is not very high, $\mathbf{E}\left[N_{Exp}^Q\right]$ falls while $\mathbf{E}\left[N_{Exp}^S\right]$ increases. For a large enough C_S^2, the fall in $\frac{C_S^2+1}{2}\mathbf{E}\left[N_{Exp}^Q\right]$, and hence in the mean waiting time in the FCFS buffer, will be larger than the rise in $\mathbf{E}\left[N_{Exp}^S\right]$, which is the component representing the mean time to process a job at the PS server. Therefore, setting an MPL larger than K^*, and allowing small jobs to overtake the big jobs, leads to an overall reduction in the mean response time.

We would like to point out that the question of choosing the optimal multi-programming limit is closely related to the question of choosing the optimal number of servers in a multiserver system (that is, one fast vs. K slow servers), such as the $M/G/K$, but with a fundamentally different trade-off. In the presence of highly variable job

sizes, one wants to choose a large number of servers in a multiserver system to prevent small jobs from getting blocked behind large jobs. Similarly, in the PS-MPL system, we want to choose a high MPL to allow small jobs to overtake large jobs. In both cases, we are limited in our ability to increase the parallelism due to capacity wastage. While in a multiserver system, capacity is wasted when there are less than K jobs in the system, in the PS-MPL system, capacity is wasted when the multiprogramming limit K is set larger than the peak efficiency point K^*, and there are more than K^* jobs in the system. Therefore, in a multiserver system, high parallelism (large number of servers) is preferred when the traffic intensity is high, while in a PS-MPL system a high degree of parallelism (large MPL) is preferred when the traffic intensity is low.

4.2.4 Approximation for *GI/G/PS-MPL*

Proposition 4.5 *Let* $\mathbf{E}[N]$ *denote the mean number of jobs in a GI/G/PS-MPL system with state-dependent service rate* $\mu(n)$ *when there are n jobs at the PS server, MPL=K, a general service distribution with mean 1 and SCV* $C_S^2 \geq 1$*, and a general interarrival time distribution with mean* $\frac{1}{\lambda}$ *and SCV* $C_A^2 \geq 1$*. Then,*

$$\mathbf{E}[N] \approx \mathbf{E}\left[N_{S_{Exp}}\right] + \frac{C_S^2+1}{2}\mathbf{E}\left[N_{Q_{Exp}}\right]$$

where $\mathbf{E}\left[N_{S_{Exp}}\right]$ *and* $\mathbf{E}\left[N_{Q_{Exp}}\right]$ *denote, respectively, the mean number of jobs at the PS Server and in the FCFS Queue in a BPP/M/PS-MPL system with the same state-dependent service rates as the original GI/G/PS-MPL system, MPL=K, exponential service distribution with mean 1, mean arrival rate* λ *and i.i.d. geometric batch sizes with mean* $\frac{C_S^2+C_A^2}{C_S^2+1}$*. The expressions for* $\mathbf{E}\left[N_{S_{Exp}}\right]$ *and* $\mathbf{E}\left[N_{Q_{Exp}}\right]$ *are given by*

$$\mathbf{E}\left[N_{S_{Exp}}\right] = \frac{\sum_{i=1}^{K} i \cdot \phi_i + K \cdot \sum_{i=K+1}^{\infty} \phi_i}{1 + \sum_{i=1}^{\infty} \phi_i} \tag{4.3}$$

$$\mathbf{E}\left[N_{Q_{Exp}}\right] = \frac{\phi_{K+1}}{1 + \sum_{i=1}^{\infty} \phi_i} \left(\frac{C_S^2 + C_A^2}{(C_S^2+1)(1-\rho)}\right)^2 \tag{4.4}$$

where

$$\phi_i = \begin{cases} \prod_{j=1}^{i} \frac{\lambda \cdot (C_S^2+1) + \mu(j-1) \cdot (C_A^2-1)}{(C_S^2+C_A^2)\mu(j)} & 1 \leq i \leq K \\ \phi_K \cdot \left(\frac{\rho \cdot (C_S^2+1) + C_A^2 - 1}{C_S^2+C_A^2}\right)^{i-K} & i > K \end{cases}$$

and $\rho = \frac{\lambda}{\mu(K)}$.

In the special case of state-independent service rate, i.e. $\mu(n) = \mu$ for all n, our approximation for the mean number in system simplifies to:

$$\mathbf{E}[N] \approx (1 - p_b)\frac{C_S^2 + C_A^2}{C_S^2 + 1} \cdot \frac{\rho}{1-\rho} + p_b \frac{C_S^2 + C_A^2}{2} \cdot \frac{\rho}{1-\rho}$$

where $p_b = \left(1 - (1-\rho)\frac{C_S^2+1}{C_S^2+C_A^2}\right)^K$ denotes the probability (approximation thereof) that a job sees at least K jobs in the system on arrival. The above approximation is similar to the heavy-traffic approximation for GI/G/PS-MPL systems with state-independent service rates proposed by Zhang and Zwart [162], except that $p_b \approx \rho^{\frac{C_S^2+1}{C_S^2+C_A^2}K}$ in [162]. Indeed, in heavy-traffic ($\rho \to 1$, $K \to \infty$ as $\rho^K \to \theta$ for some constant θ), the two approximations converge.

Similar to Proposition 4.4, we can show the existence of a GI arrival process with an interarrival time SCV of C_A^2, and a service distribution with SCV C_S^2 (the $H^*(C_S^2)$ distribution) for which the approximation proposed in Proposition 4.5 is exact.

Proposition 4.6 *The mean number of jobs in an $BPP/H^*(C_S^2)$/PS-MPL system with mean arrival rate λ, i.i.d. batch sizes distributed according to a Geometric distribution with mean $\frac{C_A^2+1}{2}$, state-dependent service rate $\mu(n)$ when there are n jobs at the PS server, and MPL=K is given by:*

$$\mathbf{E}\left[N_{C_S^2,C_A^2}(K)\right] = \mathbf{E}\left[N_{S_{Exp}}(K)\right] + \frac{C_S^2+1}{2}\mathbf{E}\left[N_{Q_{Exp}}(K)\right]$$

where $\mathbf{E}\left[N_{Q_{Exp}}(K)\right]$ and $\mathbf{E}\left[N_{S_{Exp}}(K)\right]$ are as defined in Proposition 4.5.

4.3 Self-Adaptive MPL control policies

In the previous section, we considered the question of choosing the optimal static MPL under the assumption that the arrival process is Poisson, and that the arrival rate, λ, was known accurately. We begin this section by showing that the methodology of choosing a static MPL based on assuming a mean intensity for the Poisson arrival process is very fragile. In Table 4.1 we consider a Weibull service distribution with mean 1 and $C_S^2 = 19$, and show the mean number of jobs in the system for various settings of MPL and the mean arrival rate λ. We assume the service rate curve shown in Figure 4.1 with $K^* = 5$. The optimal MPL in Table 4.1 varies from 15, when $\lambda = 0.65$, to 5, when $\lambda = 1.15$. In fact, choosing the optimal static MPL assuming $\lambda \leq 0.85$ results in an unstable system when $\lambda = 1.15$.

MPL	4	5	6	7	8	9	10	11	12	13	14	15	95% c.i.
$\lambda = 0.65$	2.96	2.47	2.25	2.14	2.09	2.05	2.04	2.02	2.02	2.02	2.01	**2.01**	$\pm$ 0.007
$\lambda = 0.75$	4.84	3.88	3.44	3.17	3.00	2.89	2.84	2.79	2.77	2.76	**2.75**	2.76	$\pm$ 0.020
$\lambda = 0.85$	8.49	6.79	6.02	5.54	5.22	4.98	4.90	**4.85**	4.92	5.01	5.21	5.58	$\pm$ 0.294
$\lambda = 0.95$	15.96	13.19	12.52	**12.16**	12.17	12.63	13.49	15.03	18.05	23.13	32.88	60.79	$\pm$ 2.483
$\lambda = 1.05$	33.92	**29.51**	31.08	34.98	40.70	52.55	72.72	126.90					$\pm$ 4.273
$\lambda = 1.15$	92.84	**87.18**	114.99	183.61									$\pm$ 5.606

Table 4.1: Numerical results for mean number of jobs in system for different values of MPL and arrival rates. The arrival process was Poisson, and the service distribution was Weibull with mean 1, SCV 19. The optimal value for each setting of the mean arrival rate has been boldened.

There are at least two ways around this problem: The first is to robustly choose a single static MPL that works well for all λ. This necessarily implies operating the system at peak efficiency K^*, which we have already seen can be far from the optimal. The second approach is to learn the parameters of the arrival process and then choose the optimal static MPL for that particular arrival process. However, this approach will fail to adapt to variations in traffic on time scales smaller than needed for the learning algorithm to converge.

In this section, we are motivated by the question:

> *Are there light-weight, traffic-oblivious MPL control policies which perform as well as the traffic-aware optimal static MPL policies?*

By a traffic-oblivious control policy, we mean a policy that does not depend on knowing the arrival rate or the higher order characteristics of the arrival process.

In this section, we develop two dynamic MPL control policies - LIGHT-APPROX and POISSON-APPROX. Section 4.3.1 highlights the key ideas in our approach. Section 4.3.2 and Section 4.3.3, respectively, present the numerical algorithms involved in the construction of our traffic-oblivious dynamic MPL control policies LIGHT-APPROX and POISSON-APPROX. In Section 4.3.4 we evaluate our dynamic MPL control policies via simulations and demonstrate that our proposed MPL control policies exhibit robustness to both the traffic intensity and the burstiness of the arrival process.

4.3.1 Key Steps in Our Approach

Recall that, given a service distribution, our goal is to obtain MPL control policies which are (i) light-weight: adjust the MPL based only on the instantaneous queue

length, $Q(t)$, and the instantaneous MPL, $K(t)$, and (*ii*) traffic-oblivious: robust to variations in the arrival process.

To achieve our first goal, we consider a special class of service distributions, the degenerate hyperexponential distribution (H^*), which is a mixture of an exponential distribution, and a point mass at 0. Jobs of size 0 do not spend any time at the server. This coupled with the memoryless property of the exponential distribution, ensures that $(Q(t), K(t))$ is a Markov process. This ensures that we can obtain a light-weight dynamic MPL control policy, since any optimal MPL control policy for the H^* service distribution will only take decisions based on $(Q(t), K(t))$.

The next step in our approach is solving a stochastic dynamic programming problem to construct *families* of candidate dynamic MPL control policies. The LIGHT-APPROX and POISSON-APPROX policies correspond to two families of candidate policies. Under LIGHT-APPROX, the family of candidate policies is a set, $\{\pi_p\}$, where a particular policy π_p is constructed by solving an optimal MPL control problem for an H^* service distribution with parameter p (Eqn. (4.7)). Thus, while there is some unique $H^*(C_S^2)$ service distribution that matches the first two moments of the true service distribution, the family is constructed by looking at a range of H^* distributions. To solve the optimal control problem, we assume that we start in some initial state (Q_0, K_0), and find the policy that minimizes the sum of response time of jobs in the system given that there are no further arrivals. In the case of POISSON-APPROX, the family of candidate policies is the set, $\{\pi_{\lambda_p}\}$, where a particular policy π_{λ_p} is obtained by solving an optimal control problem for a Poisson arrival process with intensity λ_p and the $H^*(C_S^2)$ service distribution to minimize the time-average mean number of jobs in the system.

The final step in our approach is to choose one member of the family of candidate dynamic policies, so that the chosen policy is robust to the arrival process. To achieve this goal, we evaluate the candidate policies in the family for a Poisson arrival process with rates lying in an interval $[\underline{\lambda}, \overline{\lambda}]$ and $H^*(C_S^2)$ service distribution. Let $\mathbf{E}[N^*(\lambda)]$ denote the mean number of jobs in the system for a Poisson arrival process with intensity λ, and $H^*(C_S^2)$ service distribution, under the OPT-STATIC policy. The quantity $\mathbf{E}[N^*(\lambda)]$ is given by Proposition 4.4. Let $\mathbf{E}[N^\pi(\lambda)]$ denote the mean number of jobs in the system for the $H^*(C_S^2)$ service distribution and Poisson arrival process with intensity λ under a dynamic MPL control policy π. We define the worst-case relative error for a policy π as:

$$\epsilon(\pi) = \max_{\lambda \in [\underline{\lambda}, \overline{\lambda}]} \frac{\mathbf{E}[N^\pi(\lambda)] - \mathbf{E}[N^*(\lambda)]}{\mathbf{E}[N^*(\lambda)]} \tag{4.5}$$

Given a family of candidate policies $\{\pi_a\}$ with parameter a taking its values from

some set A, we choose the policy that minimizes the worst case relative error:

$$a^* = \arg\min_{a \in A} \epsilon(\pi_a) \tag{4.6}$$

Thus, in our case, π_{p^*} denotes the LIGHT-APPROX policy, and $\pi_{\lambda_p^*}$ denotes the POISSON-APPROX policy.

4.3.2 The Light-Approx policy

As a first step towards deriving the LIGHT-APPROX policy, we begin in Section 4.3.2 by formulating and solving a light-traffic optimal MPL control problem. We find that the solution to this problem exhibits both a fluid component, to guarantee stability, and a stochastic component, to handle variability in job sizes. In Section 4.3.2, we use the solution of the light-traffic optimal control problem to construct a family, $\{\pi_p\}$, of simple, light-weight MPL control policies, and in Section 4.3.2 we sketch the use of Matrix-Geometric methods to evaluate this family of candidate policies to enable selection of the appropriate policy, LIGHT-APPROX.

A light-traffic optimal control problem

In this section we solve an optimal light-traffic MPL control problem parametrized by p, by considering the following degenerate hyperexponential service distribution :

$$H^*(p) \sim \begin{cases} 0 & \text{with probability } p \\ \text{Exp}\,(1) & \text{with probability } 1-p \end{cases} \tag{4.7}$$

We assume that we start our PS-MPL system in some state (Q_0, K_0) at time $t = 0$, where a departure has taken place at time $t = 0^-$. The state variable Q_0 denotes the queue length at $t = 0^-$ and K_0 is one more than the number of jobs at the PS server left behind by the last departure. We assume that multiple zero-sized jobs admitted at the same time leave together. Thus K_0 does not necessarily denote the MPL at time $t = 0^-$. However, by our assumption of an $H^*(p)$ service distribution, each of the $(K_0 - 1)$ jobs at the server has remaining service requirement independent and identically distributed as Exp(1). Note that while the zero-sized jobs do not spend any time at the server, they still experience delays while waiting in the FCFS buffer. We *assume that there are no more arrivals* (hence the light-traffic). We can now take one of the following actions at time $t = 0$:

1. **Decrease MPL**: We do not admit another job from the queue into the PS server, decreasing the MPL to $K_0 - 1$.

2. **Keep MPL same**: We admit only one job from the queue into the PS server to replace the departing job, maintaining the MPL at K_0.

3. **Increase MPL by k**: We admit $k+1$ jobs from the queue into the PS server, increasing the MPL to $K_0 + k$.

Our aim is to take the optimal action in each state so as to achieve the following **goal**:

Minimize the expected sum of response times of jobs present in the system at time $t = 0$, given that there are no further arrivals.

If our goal was to minimize the time until the system empties, the optimal control would be to operate at MPL of K^*. However our performance metric is the mean response time. Note that we do not allow the preemption of an executing job to decrease the MPL. This is important because in a transaction processing system, for instance, killing an executing task involves unrolling the execution trace for the task and is significantly expensive. In our framework, we can only alter the MPL when a job departs, and hence we assume that there are no costs associated with changing the MPL.

The solution of the above optimal-control problem can be obtained in a straightforward fashion via stochastic dynamic programming. To do so, we associate a cost function $c(Q, K)$ with each state (Q, K), which represents the optimal expected sum of response times, given that we start in state (Q, K) at time $t = 0$, and an action function $\pi(Q, K)$, representing the optimal action in state (Q, K). The function $\pi(Q, K)$ takes values in the range $\{-1, 0, 1, 2, \ldots\}$ with -1 representing the action 'decrease MPL', 0 representing the action 'keep MPL same' and $k > 0$ representing the action 'increase MPL by k'.

The cost of states with zero queue length is simply:

$$c(0, K) = \sum_{i=1}^{K-1} \frac{i}{\mu(i)} \tag{4.8}$$

To see why the above is true, note that since the queue is empty and we do not allow preemption of executing jobs, the cost of state $(0, K)$ is the expected sum of response times of the $K - 1$ jobs executing at the server. The mean time until the departure of the first job is given by $\frac{1}{\mu(K-1)}$ since the server is processing at rate $\mu(K-1)$. The time until the first departure gets added to the response time of all the jobs in the system, and contributes $\frac{K-1}{\mu(K-1)}$ to $c(0, K)$, and so on for subsequent departures.

We represent by $c_{-1}(Q, K)$ the cost of state (Q, K) given that we take action 'decrease MPL' in state (Q, K). Similarly, $c_k(Q, K)$ $(k \in \{0, \ldots, Q-1\})$ denotes the cost of

state (Q, K) given that we take action 'increase MPL by k' in state (Q, K). Given $c_{-1}(Q, K)$ and $c_k(Q, K)$, the optimal action $\pi(Q, K)$ and the cost function $c(Q, K)$ are:

$$\pi(Q, K) = \arg\min_\delta c_\delta(Q, K) \qquad \delta \in \{-1, \ldots, Q-1\} \tag{4.9}$$

$$c(Q, K) = c_{\pi(Q,K)}(Q, K) \tag{4.10}$$

The function $c_{-1}(Q, K)$ is given by:

$$c_{-1}(Q, K) = \frac{Q + K - 1}{\mu(K-1)} + c(Q, K-1) \tag{4.11}$$

and $c_k(Q, K)$ is given by:

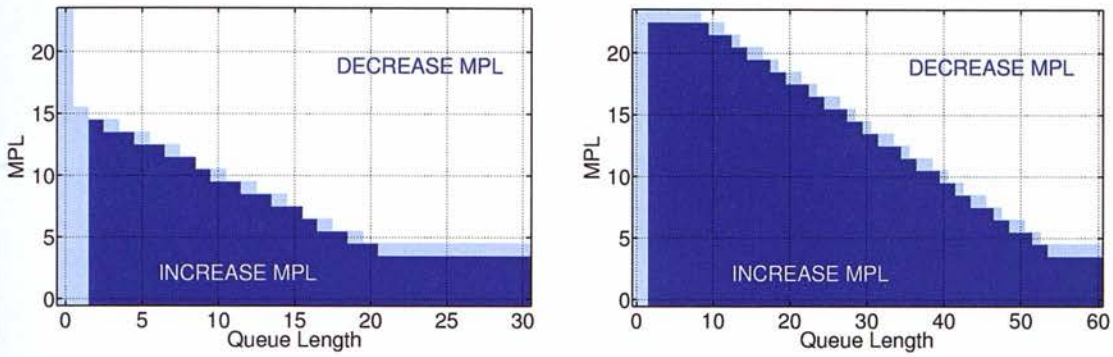
$$\begin{aligned} c_k(Q, K) = &\left[\frac{Q + K - 1}{\mu(K+k)} + c(Q-k-1, K+k)\right] \cdot (1-p)^{k+1} \\ &+ \sum_{i=1}^{k+1} c(Q-k-1, K+k+1-i) \cdot \binom{k+1}{i} (1-p)^{k+1-i} p^i \end{aligned} \tag{4.12}$$

In deriving the last equation, we have made use of the assumption that if multiple zero-sized jobs are admitted simultaneously, then they all leave together. This maintains the invariant that the K in state descriptor (Q, K) is one larger than the number of jobs at the server belonging to the exponential class, and we do not have to keep track or estimate the number of zero-sized jobs.

While in the problem formulation above, we have not imposed an upper bound on k, in practice we restrict $k \leq \Delta_{max}$ to prevent sudden jumps in MPL. For all the simulation results in this paper, we set $\Delta_{max} = 1$.

A family of traffic-oblivious MPL control policies

In Section 4.3.2 we formulated an optimal control problem parametrized by p, the fraction of zero-sized jobs in the $H^*(p)$ service distribution. By varying the parameter p, we obtain a family of MPL control policies. Let π_p denote the action function for the control problem with parameter p. Figure 4.4 shows the structure of π_p for $p = 0.3$ and $p = 0.5$ and the service rate curve shown in Figure 4.1. For example, if the current state is $(Q = 21, K = 10)$, under the $p = 0.3$ policy, the control is to decrease the MPL to 9 by not admitting a new job, while under $p = 0.5$ policy, the optimal control is to increase the MPL to 11 by admitting two jobs. The structure of the optimal solution has some interesting features:



(a) $p = 0.3$ (b) $p = 0.5$

Figure 4.4: The structure of the LIGHT-APPROX control policy for two values of the parameter p and $\Delta_{max} = 1$. The dark shaded area represents the region where the control decision is to 'increase MPL', and the light shaded region represents the decision 'keep MPL same'. Decision in the unshaded area is to decrease MPL.

1. For a given p, there is some minimum queue length $Q(p)$ such that the optimal action for $Q > Q(p)$ is to operate at the peak efficiency point. In Figure 4.4(a), $Q(p) = 20$ and the optimal control for $Q > Q(p)$ is to attain the peak efficiency MPL of $K^* = 5$. We call this the *fluid component* of the control policy. This fluid component provides robustness to the dynamic MPL policy against high arrival rates. Furthermore, as p increases, the threshold $Q(p)$ increases.

2. As the queue length decreases, the *stochastic component* of the control takes over, gradually increasing the MPL to a point with lower service rate than the most efficient point. This stochastic component gives our MPL control policy the ability to combat the job-size-variability when the traffic intensity is low.

The structure of the optimal control is quite intuitive. Whenever a decision to increase the MPL has to be taken, there are two scenarios: (i) with probability p the admitted job is of size zero in which case the decrease in server speed does not hurt any one, and (ii) with probability $1-p$, the admitted job belongs to the exponential class and in this case adds to the waiting time of everyone in the queue. If we define the 'threshold queue length' to be the point when we should increase the MPL and move to a less efficient service rate, then we see that this threshold queue length is an increasing function of p.

Given any action function π, we can translate it into a dynamic MPL control policy via the procedure in Figure 4.5.

Algorithm MPL_control(π)
Case: New arrival

- Let Q be the queue length and K be the MPL immediately after the arrival.
- Let $\pi(Q, K+1) = k$
  - if $k \geq 0$: admit $k+1$ jobs from the head of the FCFS buffer into the server and increase MPL to $K+k+1$
  - if $k < 0$: do nothing

Case: Departure

- Let Q be the queue length and K be the MPL immediately before the departure.
- Let $\pi(Q, K) = k$
  - if $k \geq 0$: admit $k+1$ jobs from the head of the FCFS buffer into the server and set MPL to $K+k$
  - if $k < 0$: reduce MPL to $K-1$ by not admitting any job from the FCFS buffer

Figure 4.5: The dynamic MPL control policy obtained from the action function π.

The LIGHT-APPROX control policy for a distribution with SCV C_S^2 is now chosen to be π_{p^*} such that:

$$p^* = \arg\min_p \epsilon(\pi_p) \tag{4.13}$$

where $\epsilon(\cdot)$ is given by (4.5). Experimentally, it suffices to carry out the optimization over a small set of parameters p (at a coarse granularity).

Evaluation of dynamic MPL control policies via Matrix-geometric analysis

In this section, we outline a method to numerically evaluate the mean number of jobs, $\mathbf{E}[N^\pi(\lambda)]$, for a dynamic MPL control policy π under the assumption of the $H^*(C_S^2)$ service distribution and a Poisson arrival process of intensity λ. Note that in Proposition 4.4 with static MPL, we were able to simplify the analysis of the $H^*(C_S^2)$ service distribution by ignoring the zero-sized jobs and focusing on the exponential class. This was because the admission control policy was independent of the queue-length. However, with a dynamic policy that looks at the queue-length, we need

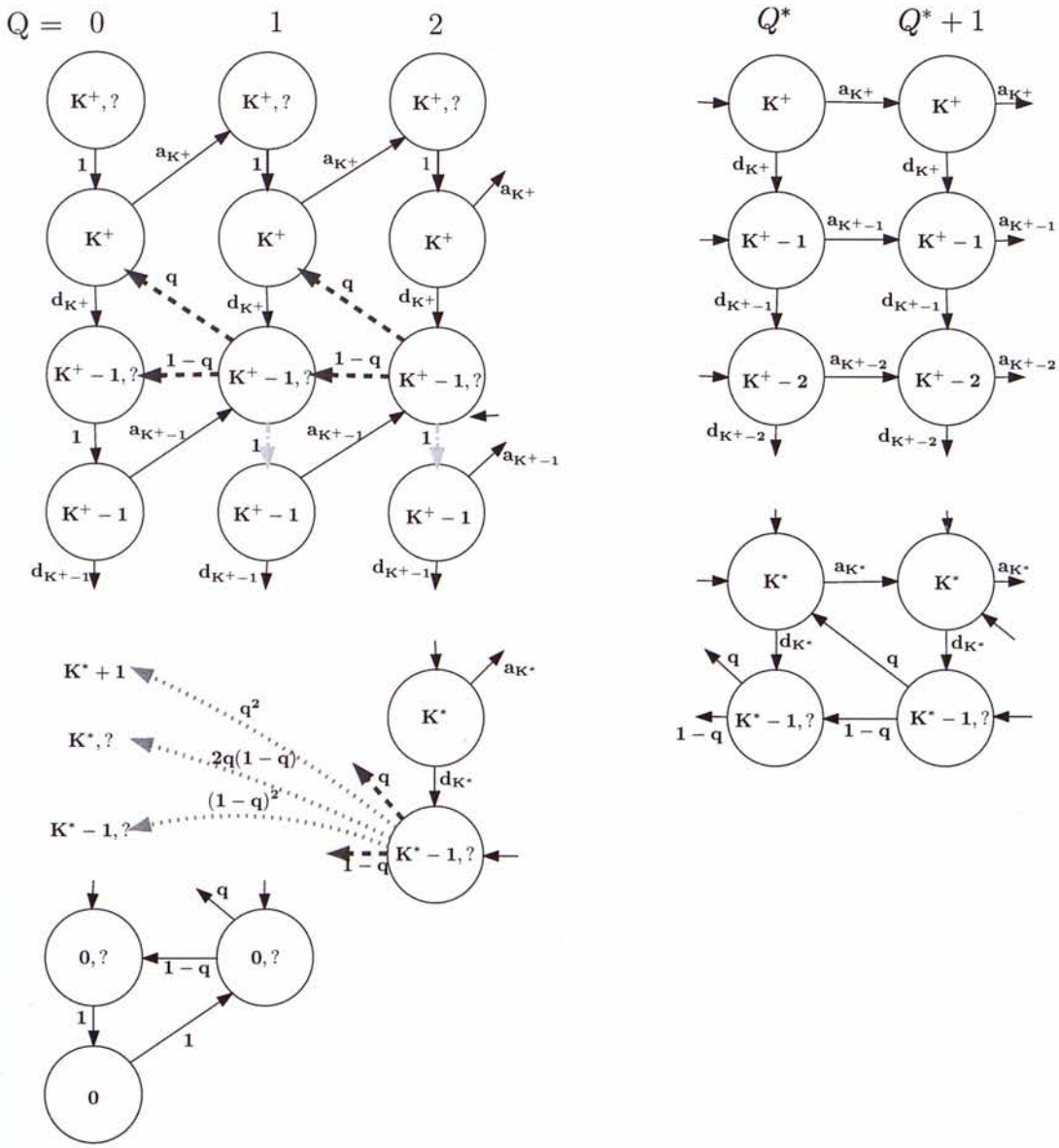


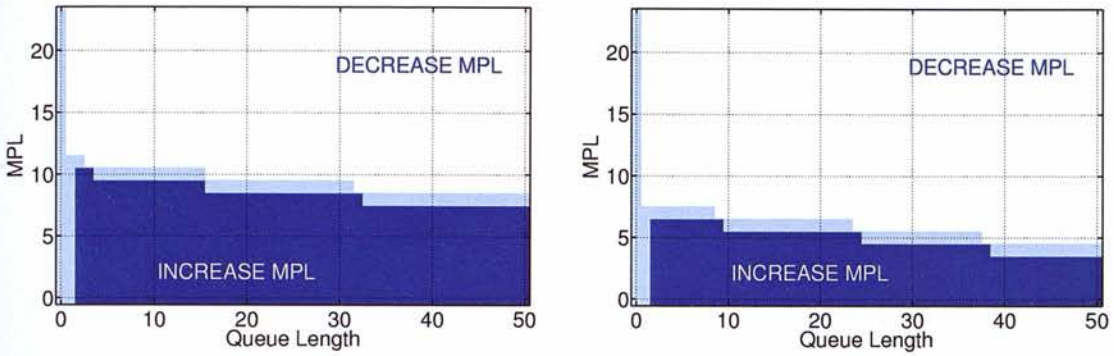
Figure 4.6: The embedded Markov chain for evaluation of dynamic MPL control policies. We use a_n to denote $\frac{\lambda}{\lambda+q\cdot\mu(n)}$ and $d_n = 1 - a_n$. For decision states with multiple alternatives (e.g., $(1, K^+ - 1, ?)$ and $(2, K^* - 1, ?)$), the dash-dotted arcs correspond to the decision to not admit any jobs, dashed arcs correspond to the decision to admit one job, and dotted arcs correspond to the decision to admit two jobs.

to keep track of how many zero-sized jobs are in the system. For succinctness, let $q = \frac{2}{C_S^2+1}$.

Assuming that under the dynamic policy π, there is some queue-length Q^* such that the optimal control for any queue length $Q \geq Q^*$ is to operate at the highest efficiency point K^*, we can model the system by a Markov chain with a repeating structure. The states of the Markov chain are pairs (Q, K) with Q denoting the queue length, and K denoting the number of jobs of the exponential class at the server. However, due to the zero-sized jobs, we can have arbitrarily large drops in Q. For example, if we are in state $(Q = 10, K = 5)$ and a departure takes place, and if all the jobs in the queue have size 0, which happens with non-zero probability, we jump to state $(Q = 0, K = 4)$. To take care of this problem, we introduce *decision* states represented as $(Q, K, ?)$. We transition to the decision state $(Q, K, ?)$ immediately after a departure takes place from the state $(Q, K + 1)$, or if an arrival takes place while in state $(Q - 1, K)$ and $Q < Q^*$. The state $(Q, K, ?)$ implements the admission control policy π, as well as handling zero-sized jobs, because now the jumps are bounded. For example, if the control in state $(Q, K, ?)$ is to admit 1 job, then with probability $(1 - q)$ the job is of size 0, and we transition to $(Q - 1, K, ?)$; otherwise, with probability q we transition to $(Q - 1, K + 1)$. However, the rate of transitioning from the decision states is infinite. Thus we will find it suitable instead to work in the framework of Semi-Markov processes. We will consider the embedded discrete time Markov chain where the transitions correspond to arrivals, departure and decisions taken in decision states in the original continuous time system. The embedded Markov chain is shown in Figure 4.6. We then solve for the stationary distribution of this embedded Markov chain via Matrix-Geometric method. We would like to point out that due to the special structure of the Markov chain in Figure 4.6 (the backward transition matrix is of rank 1), the rate matrix involved in the Matrix-geometric solution has an explicit solution in our case [125]. Finally, we obtain the stationary distribution of the number of jobs in the system by multiplying the probability of being in a state in the embedded chain with the mean residence time in that state, and normalizing.

4.3.3 The Poisson-Approx policy

The POISSON-APPROX policy is defined by constructing a family $\{\pi_{\lambda_p}\}$, where the candidate policy π_{λ_p} is obtained as follows: We consider a Poisson arrival process of intensity λ_p and the $H^*(C_S^2)$ service distribution, and solve the optimal dynamic MPL control problem to minimize the mean number of jobs. The policy π_{λ_p} is computed via the method of policy iteration, explained in Appendix 4.A. Figure 4.7 shows the structure of π_{λ_p} for $\lambda_p = 0.95$ and $\lambda_p = 1.05$. The POISSON-APPROX MPL control



(a) $\lambda_p = 0.95$ (b) $\lambda_p = 1.05$

Figure 4.7: The structure of the POISSON-APPROX control policy for two values of the parameter λ_p and $\Delta_{max} = 1$. The dark shaded area represents the region where the control decision is to 'increase MPL', and the light shaded region represents the decision 'keep MPL same'. Decision in the unshaded area is to decrease MPL.

policy is now chosen to be $\pi_{\lambda_p^*}$ where:

$$\lambda_p^* = \arg\min_{\lambda_p} \epsilon(\pi_{\lambda_p}) \tag{4.14}$$

where $\epsilon(\cdot)$ is defined in (4.5). As in the case of LIGHT-APPROX, it suffices to carry out the above optimization at a coarse granularity.

4.3.4 Performance Evaluation

In this section we show via simulations that our dynamic MPL control policies proposed in Sections 4.3.2 and 4.3.3 guarantee robustness against both misestimation of traffic intensity, and against higher order characteristics of the arrival process, such as the burstiness.

Robustness against traffic intensity estimation

We will now evaluate the LIGHT-APPROX and POISSON-APPROX policies for a Poisson arrival process with unknown arrival rate, λ, and compare them against the OPT-STATIC policy that is given the exact mean arrival rate. To do this, we show the mean number of jobs, $\mathbf{E}[N]$, under different arrival rates, obtained via simulations. Recall that Table 4.1 shows these results for the Weibull service distribution

p	0.2	**0.25**	0.3	0.35	0.4	0.45	0.5	95% c.int.
$\lambda = 0.65$	2.14	**2.06**	2.03	2.02	2.02	2.01	2.01	$\pm$ 0.004
$\lambda = 0.75$	3.26	**3.03**	2.89	2.84	2.80	2.80	2.80	$\pm$ 0.015
$\lambda = 0.85$	5.93	**5.50**	5.22	5.13	5.18	5.33	5.49	$\pm$ 0.049
$\lambda = 0.95$	12.47	**12.31**	12.31	12.83	13.83	15.48	17.03	$\pm$ 0.201
$\lambda = 1.05$	29.80	**30.73**	32.44	35.98	40.74	46.83	53.34	$\pm$ 0.381
$\lambda = 1.15$	89.07	**93.37**	99.87	108.30	120.12	132.35	143.67	$\pm$ 1.724

Table 4.2: Simulation results for mean number of jobs, $\mathbf{E}[N]$, for different parameters p of the LIGHT-APPROX policy and arrival rates, λ. The arrival process is Poisson(λ), and the service distribution is Weibull with mean 1, SCV 19.

λ_p	0.65	0.75	0.85	**0.95**	1.05	1.15	95% c.int.
$\lambda = 0.65$	2.01	2.01	2.02	**2.03**	2.13	2.47	$\pm$ 0.005
$\lambda = 0.75$	2.79	2.76	2.77	**2.84**	3.19	3.89	$\pm$ 0.015
$\lambda = 0.85$	5.82	5.24	4.89	**4.98**	5.65	6.77	$\pm$ 0.068
$\lambda = 0.95$	20.99	16.73	13.26	**11.87**	12.10	13.21	$\pm$ 0.183
$\lambda = 1.05$	66.04	53.76	40.95	**33.48**	29.67	29.46	$\pm$ 0.536
$\lambda = 1.15$	166.72	149.05	124.86	**104.40**	91.01	86.47	$\pm$ 1.967

Table 4.3: Simulation results for mean number of jobs, $\mathbf{E}[N]$, for different parameters λ_p of the POISSON-APPROX policy and arrival rates λ. The arrival process is Poisson(λ), and the service distribution is Weibull with mean 1, SCV 19.

and various values of static MPLs. In Table 4.2 we show the results for the mean number of jobs for the same Weibull service distribution under the LIGHT-APPROX policy, as a function of λ and the parameter p of the family $\{\pi_p\}$ of candidate policies. The optimization procedure (4.13) sets $p^* = 0.25$ from among the values shown in the table (column highlighted). Observe that the LIGHT-APPROX policy gives near optimal performance for each arrival rate as compared to Table 4.1 for λ up to 1.05 with approximately 13% larger mean number of jobs in the system than the optimal traffic-aware static policy when $\lambda = 0.85$. On the other hand, a single robustly chosen static MPL necessarily has to operate at the peak efficiency point and, as Table 4.1 shows, exhibits 41% larger mean response time than the optimal traffic-aware static policy when $\lambda = 0.75$.

Table 4.3 shows simulation results for the mean number of jobs with the POISSON-APPROX MPL control policy for various values of the parameter λ_p for the family $\{\pi_{\lambda_p}\}$ of candidate policies. The optimization procedure (4.14) sets $\lambda_p^* = 0.95$ from

among the values shown in the table (column highlighted). The POISSON-APPROX policy also achieves near-optimal performance for each arrival rate as compared to Table 4.1 with approximately 19.5% larger mean number of jobs in the system than the optimal traffic-aware static policy when $\lambda = 1.15$. Note that for these results, we have not completely optimized the λ_p parameter, and the performance of the POISSON-APPROX policy is likely to improve further.

While we have seen that both dynamic policies are far superior than any static policy when the mean arrival rate is not known, looking both at Tables 4.2 and 4.3, one can observe that neither dynamic policy significantly outperforms the OPT-STATIC policy if the mean arrival rate is known.

Robustness against burstiness in arrival process with unknown arrival rate

We now evaluate the robustness of our MPL control policies to unknown arrival rate for bursty arrival processes. To do so, we choose a batch Poisson arrival process (BPP). The batch sizes are i.i.d. geometric with mean 5. Table 4.4 shows the results for the mean number of jobs in the system with Weibull service distribution for various settings of static MPL and mean arrival rate λ of the arrival process. From Table 4.4, we see that when the arrival rate is not known, a robustly chosen static policy has to operate at $K^* = 5$, which results in 50% higher mean number of jobs than the optimal traffic-aware static policy when the mean arrival rate is $\lambda = 0.65$. Therefore a bursty arrival process can exacerbate the inadequacy of static MPL policies when the mean arrival rate is not known.

MPL	4	5	6	7	8	9	10	11	12	13	14	15	95% c.i.
$\lambda = 0.65$	5.80	4.85	4.35	3.99	3.75	3.58	3.46	3.37	3.31	3.26	3.25	**3.23**	$\pm$ 0.015
$\lambda = 0.75$	9.28	7.79	7.09	6.53	6.17	5.87	5.69	5.56	5.49	**5.47**	5.47	5.58	$\pm$ 0.083
$\lambda = 0.85$	15.43	13.20	12.29	11.67	11.38	**11.20**	11.22	11.43	11.86	12.75	13.90	15.85	$\pm$ 0.905
$\lambda = 0.95$	27.24	24.04	**23.86**	24.36	25.17	26.84	29.45	34.28	41.08	54.11	78.13	141.17	$\pm$ 3.495
$\lambda = 1.05$	53.044	**49.18**	53.36	60.67	71.38	90.52	130.34	210.05					$\pm$ 4.932
$\lambda = 1.15$	136.64	**131.35**	176.23	274.39									$\pm$ 7.308

Table 4.4: Simulation results for mean number of jobs, $\mathbf{E}[N]$, for different values of MPL and mean arrival rates λ. The arrival process is a batch Poisson process where the arriving batch sizes are geometrically distributed with mean 5, and the service distribution is Weibull with mean 1 and SCV 19. The optimal value for each setting of the mean arrival rate is boldened.

Table 4.5 shows the results for mean number of jobs in the system for the same setting as Table 4.4 for the LIGHT-APPROX MPL control policy as a function of the parameter p of the family $\{\pi_p\}$ of candidate policies for various values of the mean

arrival rate λ. The column for the parameter chosen by the LIGHT-APPROX policy has been highlighted. From Table 4.5, we find that LIGHT-APPROX policy is also robust to burstiness, while yielding at worst 25% higher mean response time than the optimal traffic-aware static MPL policy. Therefore, the LIGHT-APPROX policy is robust to the mean arrival rate, both for Poisson and for bursty arrival processes. The LIGHT-APPROX policy with parameter $p = 0.3$ outperforms the policy with LIGHT-APPROX policy $p = 0.25$ for the chosen setting, but as noted earlier, this is due to the fact that we have not optimized the parameter completely.

p	0.2	**0.25**	0.3	0.35	0.4	0.45	0.5	95% c.int.
$\lambda = 0.65$	4.39	**4.03**	3.75	3.53	3.41	3.34	3.32	± 0.017
$\lambda = 0.75$	7.30	**6.80**	6.39	6.10	5.91	5.90	6.00	± 0.040
$\lambda = 0.85$	12.66	**12.26**	11.88	11.77	12.02	12.67	13.45	± 0.104
$\lambda = 0.95$	23.83	**23.71**	24.05	24.97	26.61	29.42	32.59	± 0.312
$\lambda = 1.05$	49.54	**50.41**	52.11	55.58	60.66	67.07	74.57	± 0.569
$\lambda = 1.15$	133.77	**137.34**	141.70	149.05	158.22	171.73	182.81	± 2.458

Table 4.5: Simulation results for mean number of jobs, $\mathbf{E}[N]$, for different parameters p of the LIGHT-APPROX policy and mean arrival rates λ. The arrival process is a batch Poisson process where the arriving batch sizes are geometrically distributed with mean 5, and the service distribution is Weibull with mean 1 and SCV 19.

λ_p	0.65	0.75	0.85	**0.95**	1.05	1.15	95% c.int.
$\lambda = 0.65$	3.27	3.26	3.30	**3.48**	4.07	4.85	± 0.017
$\lambda = 0.75$	5.92	5.64	5.63	**5.84**	6.80	7.82	± 0.035
$\lambda = 0.85$	14.03	12.42	11.42	**11.17**	12.17	13.17	± 0.125
$\lambda = 0.95$	36.05	30.40	26.07	**23.89**	23.29	23.93	± 0.244
$\lambda = 1.05$	84.01	71.67	60.75	**54.10**	49.37	48.83	± 0.510
$\lambda = 1.15$	199.38	180.81	162.21	**148.86**	135.17	131.72	± 2.239

Table 4.6: Simulation results for mean number of jobs, $\mathbf{E}[N]$, for different parameters λ_p of the POISSON-APPROX policy and arrival rates λ. The arrival process is a batch Poisson process where the arriving batch sizes are geometrically distributed with mean 5, and the service distribution is Weibull with mean 1 and SCV 19.

Table 4.6 shows the results for mean number of jobs in the system under the same setting for the POISSON-APPROX MPL control policy as a function of the parameter λ_p of the family $\{\pi_{\lambda_p}\}$ of candidate policies. The column for the parameter chosen

by the POISSON-APPROX policy has been highlighted. From Table 4.6, we find that the POISSON-APPROX policy yields at worst 13% higher mean response time than the optimal traffic aware static MPL policy. Thus while both our policies are robust to bursty arrival processes, the POISSON-APPROX policy seems to marginally outperform the LIGHT-APPROX policy. These observations also hold true for other simulation experiments not shown here.

While we have demonstrated that our dynamic policies are much more robust than any static policy in handling burstiness when the mean arrival rate is not known, comparing Tables 4.1 and 4.4, we see that, surprisingly, if the mean arrival rate of the arrival process is known, the OPT-STATIC policy which optimizes for a Poisson arrival process with the given mean arrival rate remains near-optimal for a bursty arrival process.

4.4 A Heavy-Traffic Diffusion Scaling and Approximation for Non-Work-Conserving Systems

Diffusion approximations are very powerful tools for studying the dynamics of queueing systems and to observe the first order effects of system parameters on the performance of the queueing system. They are also an important first step in developing algorithms for stochastic control of queueing systems. While for certain special types of queueing systems, the diffusion approximations are very mature, there are no tools to study the very important class of queueing systems which are not work conserving, i.e., queueing systems where the service capacity varies as a function of the system state. In this section we propose a novel approach to derive heavy-traffic diffusion scaling for non-work conserving systems, and perform approximate analysis of the stationary distribution of the number of jobs in the system. We illustrate our approach via the $GI/G/$PS-MPL model with state-dependent service rates. The proposed scaling is quite general, and we hope that it will be applicable to other systems as well.

At a high level, the diffusion scaling is obtained by fixing a smooth distribution function $F(\cdot)$ on $[0, \infty)$ for the number of jobs in the system, and *engineering* a sequence of systems (the state-dependent service rates, to be precise) indexed by parameter r, so that under Poisson arrivals (the intensity of the arrival process is invariant in the scaling) and Exponential service distribution (the mean of the job size distribution is also fixed and we assume it to be 1 without loss of generality), the sequence of the distribution function of number of jobs scaled by r converges to

$F(\cdot)$ as :

$$\lim_{r\to\infty} F^{(r)}(\lceil xr \rceil) \to F(x) \quad \forall x \in [0,\infty), \tag{4.15}$$

where $F^{(r)}(\cdot)$ denotes the distribution function for the number of jobs in the rth system under a Poisson arrival process and Exponential job size distribution. Additionally, for the $GI/G/$PS-MPL system, we will impose the condition that the MPL for the rth system scales as $\theta \cdot r$, for a constant θ.

The motivation behind our 'reverse-engineered' heavy-traffic scaling should be obvious: Our goal is to approximate the behavior of a 'discrete' system with a bounded MPL, and hence we want our limiting continuous system to 'resemble' the original discrete system. A natural constraint to impose then is that under a Markovian workload, the limiting scaled system behave 'identically' to the original discrete system, and hence by design also guaranteeing us a non-degenerate limit.

In Section 4.4.1 we derive the heavy-traffic scaling for a $GI/G/$PS-MPL system. In Section 4.4.2 we present approximate analysis of the stationary distribution of the number of jobs in the system under our scaling by considering degenerate hyperexponential interarrival and service distributions. Since the degenerate hyperexponential distributions allow us to vary the first two moments, our approximate analysis provides strong indications as to what the true diffusion limit looks like. A rigorous diffusion analysis is a topic of ongoing research.

4.4.1 Heavy-traffic scaling for $GI/G/$PS-MPL

Consider a system with MPL K, and service rate curve $\mu(\cdot)$ where $\mu(n)$ is the service rate of the server when time sharing among n jobs. Let the arrival process have mean intensity λ, and the mean job size be 1 without loss of generality. Also, assume $\rho \equiv \frac{\lambda}{\mu(K)} < 1$. To arrive at the heavy-traffic scaling, we begin by considering the case where the arrival process is Poisson and the service distribution is Exponential, which we abbreviate as $M/M/$ in the remainder of the section.

Let $\pi^{M/M/}(i)$ denote the stationary probability that the system under $M/M/$ arrival process has i jobs in the system. Let $F(\cdot) : [0,\infty) \to (0,1]$ be a twice-differentiable non-decreasing cdf obtained from $\pi^{M/M/}$ as follows:

$$1 - F(j) = \sum_{i=0}^{j-1} \pi^{M/M/}(i) \qquad j = 0, 1, \ldots, K$$

$$1 - F(x) = (1 - F(K))\rho^{x-K} \qquad x > K$$

Thus, F is a twice-differentiable interpolation of the stationary distribution of the number of jobs in the $M/M/$PS-MPL system. Let $f(x) = \dot{F}(x)$. As briefly mentioned in the introduction, the diffusion scaling is obtained by creating a sequence of $M/M/$PS-MPL systems, indexed by a discrete parameter $r \in \mathbb{N}$, where the service rates of the rth system $\left\{\mu^{(r)}(n)\right\}$ are engineered to satisfy:

1. the MPL of the rth system is $r \cdot K$,
2. the arrival rate of the rth system is λ,
3. $\lim_{r\to\infty} F^{(r)}(\lceil r \cdot x \rceil) \to F(x)$ for $x \in [0, \infty)$,
   where $F^{(r)}(\cdot)$ denotes the stationary distribution function for the number of jobs in the rth system.

Obtaining the service rates: To achieve our goal, consider the rth system. Let $x = \frac{i}{r}$. We create the service rates so that the probability that the system is in state $i = r \cdot x$ is approximately $\frac{1}{r} f(x)$. If the service rate in state $i+1 = (x + 1/r) \cdot r$ is $\mu^{(r)}(x + \frac{1}{r})$, then we must have:

$$\left\{f\left(x + \frac{1}{N}\right)\right\} \frac{1}{N} = \frac{\lambda}{\mu^{(N)}((x + 1/N) \cdot N)} \left\{f\left(x\right)\right\} \frac{1}{N} \tag{4.16}$$

which gives:

$$\frac{\lambda}{\mu^{(r)}((x + 1/r) \cdot r)} = \frac{f(x + 1/r)}{f(x)} \approx 1 + \frac{1}{r} \cdot \frac{f'(x)}{f(x)}$$

or,

$$\frac{\lambda}{\mu^{(r)}(r \cdot x)} = 1 + \frac{1}{r} \frac{d \log f(x)}{dx} + o(1/r)$$

Therefore the state-dependent service rates of the rth system under the heavy traffic scaling are given by:

$$\mu^{(r)}(i) = \lambda \left(1 - \frac{1}{r} \left.\frac{d \log(f(x))}{dx}\right|_{x=\frac{i}{r}}\right) \tag{4.17}$$

Remark 1: Consider the case with a constant service rate μ and arrival rate λ. In this case $F(x) = 1 - \left(\frac{\lambda}{\mu}\right)^x = 1 - \rho^x$, so that $\mu^{(r)}(r \cdot x) \approx \lambda\left(1 - \frac{1}{r}\frac{d \log f(x)}{dx}\right) = \lambda\left(1 - \frac{\log \rho}{r}\right)$, which is exactly the diffusion scaling used in Zhang and Zwart [162].

Remark 2: As $r \to \infty$, the service rate curve uniformly converges to λ. Thus while the limiting system would be work-conserving on any compact interval of time, none of the pre-limit systems are work conserving. Work conservation was one of the critical tools used in [162] for diffusion analysis of $GI/G/$PS-MPL systems with constant $\mu(n)$ curve, and its absence makes the diffusion analysis non-trivial. We will however be able to provide an approximation for the stationary distribution via special arrival and service processes.

4.4.2 Analysis of the Heavy-traffic Diffusion scaling

A simple approximation for a $GI/GI/$ arrival process, but involving only the first two moments of the interarrival time and service distributions, can be obtained by considering H_2^* interarrival times and H_2^* job sizes. Let C_A^2 denote the squared coefficient of variation of the interarrival time distribution and C_S^2 denote the SCV of the service distribution. The main result of this section is that the mean number of jobs in the stationary system is approximately given by:

$$\mathbf{E}[N] \approx \frac{\int_{u=0}^{\infty} \min\{u, K\} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du}{\int_{u=0}^{\infty} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du} + \frac{C_S^2+1}{2} \cdot \frac{\int_{u=0}^{\infty} (u-K)^+ f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du}{\int_{u=0}^{\infty} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du}$$

where $f(\cdot)$ is the pdf corresponding to the distribution function $F(\cdot)$ – the smooth interpolation of the distribution of number of jobs in the $M/M/$PS-MPL system.

We will begin with the analysis of a discrete $H_2^*/H_2^*/$PS-MPL system, and then specialize the results to the heavy-traffic diffusion limit.

Analysis of $H_2^*/H_2^*/$PS-MPL model: Let $\pi_{Exp}^{H_2^*/H_2^*/}(i)$ denote the stationary probability distribution for the number of jobs belonging to the Exponential branch in the system. That is, the probability that under the $H_2^*/H_2^*/$ arrival process described above, the system has i Exponential jobs in the system. Then as we established in Propositions 4.5 and 4.6:

$$\pi^{H_2^*/H_2^*/}(i) = \frac{\phi^{H_2^*/H_2^*/}(i)}{\sum_{i=0}^{\infty} \phi^{H_2^*/H_2^*/}(i)}$$

where,

$$\phi^{H_2^*/H_2^*/}(i) = \begin{cases} 1 & i = 0, \\ \phi^{H_2^*/H_2^*/}(i-1) \cdot \frac{\lambda \cdot (C_S^2+1) + \mu(i-1) \cdot (C_A^2-1)}{(C_A^2+C_S^2) \cdot \mu(i)} & 1 \le i \le K, \\ \phi^{H_2^*/H_2^*/}(K) \cdot \left(\frac{\rho \cdot (C_S^2+1) + C_A^2 - 1}{C_S^2+C_A^2} \right)^{i-K} & i > K. \end{cases}$$

Given the above, we can obtain the mean number of jobs in the system as:

$$\mathbf{E}\left[N^{H_2^*/H_2^*/}\right] = \mathbf{E}\left[N_{S_{Exp}}^{H_2^*/H_2^*/}\right] + \frac{C_S^2+1}{2}\mathbf{E}\left[N_{Q_{Exp}}^{H_2^*/H_2^*/}\right] \tag{4.18}$$

where $\mathbf{E}\left[N_{S_{Exp}}^{H_2^*/H_2^*/}\right]$ and $\mathbf{E}\left[N_{Q_{Exp}}^{H_2^*/H_2^*}\right]$, denote, respectively, the mean number of jobs from the Exponential branch at the PS Server and in the FCFS Queue, and are given by

$$\mathbf{E}\left[N_{S_{Exp}}^{H_2^*/H_2^*/}\right] = \frac{\sum_{i=1}^{K} i \cdot \phi^{H_2^*/H_2^*/}(i) + K \cdot \sum_{i=K+1}^{\infty} \phi^{H_2^*/H_2^*/}(i)}{1 + \sum_{i=1}^{\infty} \phi_i} \tag{4.19}$$

$$\mathbf{E}\left[N_{Q_{Exp}}^{H_2^*/H_2^*}\right] = \frac{\phi^{H_2^*/H_2^*/}(K+1)}{1 + \sum_{i=1}^{\infty} \phi^{H_2^*/H_2^*/}(i)} \left(\frac{C_S^2 + C_A^2}{(C_S^2+1)(1-\rho)} \right)^2. \tag{4.20}$$

Mean number of jobs under the diffusion scaling and $H_2^*/H_2^*/$ arrival process: We now consider the sequence of PS-MPL systems described by the service rate curves $\mu^{(r)}(i)$ defined in (4.17) and use the expressions in (4.18)-(4.20) to find the mean number of jobs under stationarity for an H_2^*/H_2^* arrival process.

Let $\phi_{Exp}^{(r)}(i)$ denote the probability that there are i jobs of the Exponential branch in the rth system. From our exact equations:

$$\frac{\phi_{Exp}^{(r)}(r \cdot x + 1)}{\phi_{Exp}^{(r)}(r \cdot x)} = \frac{\lambda \cdot (C_S^2+1) + \mu^{(r)}(r \cdot x) \cdot (C_A^2-1)}{(C_A^2+C_S^2) \cdot \mu^{(r)}(r \cdot x)} \quad \ldots x > 0 \tag{4.21}$$

$$= \frac{(C_S^2+1) + \frac{\mu^{(r)}(r \cdot x)}{\lambda} \cdot (C_A^2-1)}{(C_A^2+C_S^2) \cdot \frac{\mu^{(r)}(r \cdot x)}{\lambda}} \tag{4.22}$$

$$= \frac{(C_S^2+1) + \left(1 - \frac{1}{r}\frac{f'(x)}{f(x)}\right) \cdot (C_A^2-1)}{(C_A^2+C_S^2) \cdot \frac{1}{1+\frac{1}{r} \cdot \frac{f'(x)}{f(x)}}} + o(1/r) \tag{4.23}$$

$$= 1 + \frac{1}{r} \cdot \frac{C_S^2+1}{C_A^2+C_S^2} \cdot \frac{f'(x)}{f(x)} + o(1/r) \tag{4.24}$$

Which gives:

$$\log \phi_{Exp}^{(r)}((x+1/r)\cdot r) - \log \phi_{Exp}^{(r)}(r\cdot x) \sim \frac{C_S^2+1}{C_S^2+C_A^2}\cdot[\log f(x+1/r) - \log f(x)] \tag{4.25}$$

Therefore:

$$\phi_{Exp}^{(r)}(r\cdot x) \sim \phi_{Exp}^{(r)}(0)\cdot\left(\frac{f(x)}{f(0)}\right)^{\frac{C_S^2+1}{C_S^2+C_A^2}}$$

Normalizing to obtain $\phi_{Exp}^{(r)}(0)$:

$$1 = \phi_{Exp}^{(r)}(0)\left[\sum_{i=0}^{\infty}\left(\frac{f\left(\frac{i}{r}\right)}{f(0)}\right)^{\frac{C_S^2+1}{C_S^2+C_A^2}}\right] \tag{4.26}$$

$$\sim \phi_{Exp}^{(r)}(0)\cdot r\cdot\int_{x=0}^{\infty}\left(\frac{f(x)}{f(0)}\right)^{\frac{C_S^2+1}{C_S^2+C_A^2}}dx \tag{4.27}$$

which yields:

$$\phi_{Exp}^{(r)}(r\cdot x) = \frac{\left(\frac{f(x)}{f(0)}\right)^{\frac{C_S^2+1}{C_S^2+C_A^2}}}{\int_{x=0}^{\infty}\left(\frac{f(x)}{f(0)}\right)^{\frac{C_S^2+1}{C_S^2+C_A^2}}dx}\cdot\frac{1}{r} + o(1/r) \tag{4.28}$$

$$= \frac{f(x)^{\frac{C_S^2+1}{C_S^2+C_A^2}}}{\int_{x=0}^{\infty} f(x)^{\frac{C_S^2+1}{C_S^2+C_A^2}}dx}\cdot\frac{1}{r} + o(1/r) \tag{4.29}$$

Denote by $N^{(r)}, N_{Exp}^{(r)}, N_{S_{Exp}}^{(r)}, N_{Q_{Exp}}^{(r)}$ the stationary number of jobs in the system, and the stationary number of jobs belonging to the Exponential branch in the system, at the server, and in the queue, respectively, for the rth system. Let $N^* = \lim_{r\to\infty}\frac{N^{(r)}}{r}$, and similarly define N_{Exp}^*, $N_{S_{Exp}}^*$ and $N_{Q_{Exp}}^*$. Let F_{Exp}^* be the distribution function of N_{Exp}^*. We then have:

$$F_{Exp}^*(x) = \frac{\int_{u=0}^{x} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}}du}{\int_{u=0}^{\infty} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}}du} \tag{4.30}$$

Therefore the approximation for N^* (including all jobs) is given by:

$$\mathbf{E}[N^*] = \mathbf{E}\left[N^*_{S_{Exp}}\right] + \left(\frac{C_S^2+1}{2}\right)\mathbf{E}\left[N^*_{Q_{Exp}}\right] \tag{4.31}$$

$$= \frac{\int_{u=0}^{\infty} \min\{u, K\} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du}{\int_{u=0}^{\infty} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du} + \left(\frac{C_S^2+1}{2}\right)\frac{\int_{u=0}^{\infty} (u-K)^+ f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du}{\int_{u=0}^{\infty} f(u)^{\frac{C_S^2+1}{C_S^2+C_A^2}} du} \tag{4.32}$$

Remark 3: Our goal is to approximate a discrete system by its corresponding diffusion approximation. However, for the original system, we have a discontinuous distribution function, and hence the choice of the interpolating F (equivalently f) is somewhat arbitrary and in our control. Further, while we needed f to be differentiable to define the scaling, the approximation (4.32) is defined even when f is continuous.

4.5 Summary and Open Questions

In this chapter we addressed the problem of concurrency control for resource-sharing system such as database servers by limiting the maximum number of active threads to avoid thrashing. We modeled such systems as Processor Sharing servers with load-dependent service rates. We proved that, contrary to common practice, imposing a static multi-programming limit (MPL) to maximize the system efficiency (service rate) is not always optimal for minimizing the mean response time when the job sizes exhibit high variance, and used analysis to propose a simple heuristic rule to choose the optimal static MPL under the assumption that traffic is Poisson with a known arrival rate.

Next, we showed that a static MPL policy cannot be robust to varying traffic patterns, such as variability in the mean arrival rate. We proposed two simple MPL control policies, LIGHT-APPROX and POISSON-APPROX, that adjust the MPL based on knowledge of only the instantaneous queue length. We showed that our dynamic MPL control policies exhibit robustness to both an unknown mean arrival rate and to burstiness in the arrival process.

As a third contribution, we proposed a novel heavy-traffic diffusion scaling to study non-work-conserving systems, such as the one which was the focus of study of this chapter. Our scaling was arrived at via reverse engineering the system parameters

so as to be more representative of the original system that is being approximated. We presented an approximate analysis of the stationary distribution of the number of jobs under the proposed scaling.

Impact: The work in this chapter highlights an important system design principle: maximizing efficiency is not always optimal for minimizing the mean response time, and the optimal operating point critically depends on the workload. The existing work on concurrency control has ignored the effects of the workload. We also demonstrated existence of simple concurrency control policies which adapt to fluctuations in the demand without needing to learn the demand. We believe that the techniques presented to develop these traffic-oblivious control policies will be applicable to more general stochastic settings. While the majority of literature on robust dynamic control focuses on solving the corresponding optimal control problem in the fluid regime (i.e., when the backlog is large), our techniques allow one to obtain optimal control policies which exhibit components of both stochastic and fluid control (i.e., from close to empty to when the backlog is large). Finally, we hope that our heavy-traffic scaling for non-work-conserving systems will allow researchers to revisit the existing work on diffusion analysis of multi-server queueing systems and queueing networks and obtain refined approximations.

Open Problems: Are the traffic-oblivious dynamic MPL control schemes proposed in this chapter robust to the traffic demand in a formal sense, and are there provably better traffic-oblivious concurrency control mechanisms? Performing a rigorous diffusion analysis of the proposed heavy-traffic scaling is subject of ongoing research by the author, and we believe that the approximate results presented for the stationary behavior under the proposed scaling would match the diffusion analysis.

4.A Policy Iteration to Construct Candidate Poisson-Approx Policies

The goal of this section is to explain the policy iteration algorithm to find the optimal MPL control policy π_{λ_p}, for a Poisson arrival process with intensity λ_p and the $H^*(C_S^2)$ service distribution matching the true service distribution .

Let us first recall how policy iteration works [23]. We begin with some MPL control policy π^0 (in our case, a good initial policy is the threshold MPL policy which operates at the peak efficiency point K^*). Let γ^0 be the average cost (in our case the mean number of jobs in the system) of this policy. We then define the differential cost function $h^0(\cdot)$ associated with each state, where $h^0(s_i)$ denotes the differential cost to reach some state s_0 starting in state s_i under π^0. That is, $h^0(s_i)$ denotes

the difference between the mean total cost to reach state s_0, and the product of γ^0 and the mean total time to reach state s_0, given that we start in state s_i. The vector of differential costs, $h^0(\cdot)$, and the average cost, γ^0, are obtained by solving the following linear system of equations:

$$h^0(s_0) = 0$$
$$h^0(s_i) = c(s_i)\tau(s_i) - \gamma^0\tau(s_i) + \sum_j p_{ij}(\pi^0(s_i))h(s_j)$$

where $\tau(s_i)$ is the mean residence time in state s_i, $c(s_i)$ is the cost per unit of time in state s_i, and $p_{ij}(\pi^0(s_i))$ represents the probability that we transition from state s_i to s_j when control $\pi^0(s_i)$ is applied in state s_i. This is called the policy evaluation step. We then perform the policy improvement step to obtain the policy π^1. To do this, for each state s_i, we choose $\pi^1(s_i)$ as the control which satisfies:

$$c(s_i)\tau(s_i) - \gamma^0\tau(s_i) + \sum_j p_{ij}(\pi^1(s_i))h(s_j) = \min_{a\in A_i}\left[c(s_i)\tau(s_i) - \gamma^0\tau(s_i) + \sum_j p_{ij}(a)h(s_j)\right]$$

where A_i is the set of possible actions in state i. We then keep performing policy evaluation and improvement until two consecutive policies are the same, or have the same average cost.

The policy iteration step can be easily performed once the policy evaluation step is performed. The policy evaluation step is clearly tractable when the state space is finite. We now show it is also tractable when the state space is infinite but repeating, obeying the conditions for Matrix-Geometric analysis. In the remaining section, we focus on the procedure for performing the policy evaluation step for such infinite state space systems, and specializing it to the problem of solving the optimal dynamic MPL control problem.

Consider a fixed policy π, and let P^π denote the probability transition matrix:

$$P^\pi = \begin{bmatrix} L_0 & F_0 & 0 & 0 & 0 & \cdots \\ B_0 & L & F & 0 & 0 & \cdots \\ 0 & B & L & F & 0 & \cdots \\ & \vdots & & & \vdots & \end{bmatrix}$$

Let $\mathbf{h}_0$ be the vector of differential costs for the 0th (non-repeating) level, $\mathbf{h}_i$ $(i \geq 1)$ be the differential cost vector for the ith (repeating) level of the state space, and γ be the average cost under policy π. Denote by R the rate matrix (for the embedded chain) which is the least non-negative solution to:

$$R = F + RL + R^2B$$

Let G be the solution to the following equation:

$$G = B + LG + FG^2$$

We now note that G and R have the following probabilistic interpretations [119]: by conditioning on the first transition, it is easy to see that $G(j,k)$ denotes the conditional probability that the chain eventually reaches level $i-1$ and the state it enters is $(i-1,k)$, given the chain starts in state (i,j). Similarly, conditioning on the last transition before visiting $(i+1,k)$, one can see that the entry $R(j,k)$ represents the mean number of visits to state $(i+1,k)$ until it first enters level i again, given that the chain starts in state (i,j). Let J be given by:

$$J = L + FG$$

Then $J(j,k)$ represents the conditional probability that the chain enters level i again before entering level $i-1$, and that the state it enters is (i,k), given the chain starts in state (i,j). We can now write the differential cost of some state (i,j) for $i \geq 2$ as

$$h(i,j) = c(i,j)\tau(i,j) - \gamma\tau(i,j) + \sum_k B(j,k)h(i-1,k) + \sum_k J(j,k)h(i,k)$$
$$+ \sum_{m=1}^{\infty} \sum_k R^m(j,k)\left[c(i+m,k)\tau(i+m,k) - \gamma\tau(i+m,k)\right]$$

or,

$$\mathbf{h}_i = diag(\boldsymbol{\tau}_i)\mathbf{c}_i - \gamma.\boldsymbol{\tau}_i + B\mathbf{h}_{i-1} + J\mathbf{h}_i + \sum_{m=1}^{\infty} R^m(diag(\boldsymbol{\tau}_{i+m})\mathbf{c}_{i+m} - \gamma\boldsymbol{\tau}_{i+m}) \quad \ldots i \geq 2 \tag{4.33}$$

where $\mathbf{c}_i$ is the column vector of cost per unit of time for states in level i, and $\boldsymbol{\tau}_i$ is the column vector of mean residence time for states in level i. Thus,

$$\mathbf{h}_i = (I-J)^{-1}\left(diag(\boldsymbol{\tau}_i)\mathbf{c}_i - \gamma\boldsymbol{\tau}_i + B\mathbf{h}_{i-1} + \sum_{m=1}^{\infty} R^m(diag(\boldsymbol{\tau}_{i+m})\mathbf{c}_{i+m} - \gamma\boldsymbol{\tau}_{i+m})\right)$$

Thus, we can express $\mathbf{h}_2$ in terms of $\mathbf{h}_1$ and solve for $\mathbf{h}_0$ and $\mathbf{h}_1$. We can then obtain subsequent cost vectors as needed while performing the policy improvement step.

We now address the problem of evaluating a dynamic MPL control policy, π. Let K^+ denote the maximum MPL used by policy π and let Q^* denote the queue length beyond which policy π uses the MPL K^* (see Figure 4.6). For computational reasons we restrict Q^* to be at most 50. As stated earlier, in our case, the matrix B is of rank 1. Specifically, we can write $B = \boldsymbol{\nu} \cdot \boldsymbol{\alpha}$, where $\boldsymbol{\nu} = \boldsymbol{e}_1$ (the column vector with

first entry 1, and rest 0), and $\boldsymbol{\alpha} = [(1-q)\ q\ 0\ \ldots\ 0]$. Therefore, in our case the matrices G and R have an explicit solution [125]:

$$G = \boldsymbol{e} \cdot \boldsymbol{\alpha}$$
$$R = F(I - L - F\boldsymbol{e}\boldsymbol{\alpha})^{-1}$$

where $\boldsymbol{e}$ is the column vector of all 1s.

Denote by $\mathbf{s_i}$ the state vector for level i, $i \geq 1$:

$$\mathbf{s}_i = \begin{bmatrix} (Q^* + i - 1, K^* - 1, ?) \\ (Q^* + i - 1, K^*) \\ \vdots \\ (Q^* + i - 1, K^+) \end{bmatrix}$$

with the cost and mean residence time vectors given by $\mathbf{c}_i = (Q^* + i - 1) \cdot \boldsymbol{e} + \mathbf{K}$ and

$$\boldsymbol{\tau}_i = \begin{bmatrix} 0 \\ \frac{1}{\lambda + q \cdot \mu(K^*)} \\ \vdots \\ \frac{1}{\lambda + q \cdot \mu(K^+)} \end{bmatrix} = \boldsymbol{\tau}, \qquad \mathbf{K} = \begin{bmatrix} K^* - 1 \\ K^* \\ \vdots \\ K^+ \end{bmatrix}$$

We can thus simplify (4.33) to:

$$\mathbf{h}_i = diag(\boldsymbol{\tau})\mathbf{c}_i - \gamma\boldsymbol{\tau} + B\mathbf{h}_{i-1} + J\mathbf{h}_i + \sum_{m=1}^{\infty} R^m(diag(\boldsymbol{\tau})\mathbf{c}_{i+m} - \gamma\boldsymbol{\tau})$$
$$= B\mathbf{h}_{i-1} + J\mathbf{h}_i + \left[(I-R)^{-1}\left((Q^* + i - 2 - \gamma).\boldsymbol{\tau} + diag(\boldsymbol{\tau}) \cdot \mathbf{K}\right) + \left((I-R)^{-1}\right)^2 \boldsymbol{\tau}\right]$$

or,

$$\mathbf{h}_i = (I-J)^{-1}\left\{B\mathbf{h}_{i-1} + \left[(I-R)^{-1}\left((Q^* + i - 2 - \gamma).\boldsymbol{\tau} + diag(\boldsymbol{\tau})\mathbf{K}\right) + \left((I-R)^{-1}\right)^2 \boldsymbol{\tau}\right]\right\} \tag{4.34}$$

Thus, the solution of our system is given by the following system of linear equations for $\mathbf{h}_0, \mathbf{h}_1, \gamma$:

$$\begin{aligned} \mathbf{h}_0 &= diag(\boldsymbol{\tau}_0)\mathbf{c}_0 - \gamma\boldsymbol{\tau}_0 + L_0\mathbf{h}_0 + F_0\mathbf{h_1} \\ \mathbf{h}_1 &= diag(\boldsymbol{\tau})\mathbf{c}_1 - \gamma\boldsymbol{\tau} + B_0\mathbf{h}_0 + L\mathbf{h}_1 + F\mathbf{h_2} \\ &= diag(\boldsymbol{\tau})\mathbf{c}_1 - \gamma(I + F(I-J)^{-1}(I-R)^{-1})\boldsymbol{\tau} + B_0\mathbf{h}_0 \end{aligned} \tag{4.35}$$

$$+ (L + F(I - J)^{-1}B)\mathbf{h}_1 + F(I - J)^{-1}\left\{\left[(I - R)^{-1}(Q^*.\boldsymbol{\tau}\right.\right.$$
$$\left.\left.+diag(\boldsymbol{\tau})\mathbf{K}) + \left((I - R)^{-1}\right)^2\boldsymbol{\tau}\right]\right\} \tag{4.36}$$

and the additional constraint $h(0,0) = 0$.

In the method of policy iteration, the policy evaluation and policy improvement steps are repeated until two policies with the same cost are obtained. In the experiments presented in this paper, we stopped when the relative improvement between consecutive policies was below 0.01%, which took at most 7 iterations in each case (less than 30 seconds on a 3.2 GHz Pentium 4 CPU with 1 GB of memory).

Chapter 5

Load Balancing for Webserver Farms: Analysis of Join-the-Shortest-Queue Policy for PS servers

Load balancers are the most critical component of multi-server systems: *Which server from among the hundreds in a data center should be assigned a particular task? How much information about the state of each server needs to be collected before such decisions can be taken with confidence?* Join-the-Shortest-Queue (JSQ) is one of the most popular load balancing heuristics, but until now all analysis and optimality results of JSQ have been limited to First-Come-First-Serve (FCFS) server farms, whereas it is known that web servers are better modeled by the Processor Sharing (PS) scheduling discipline. We provide the first approximate analysis of JSQ in the PS server farm model for general job size distributions, obtaining the distribution of queue length at each queue. We also discover interesting insensitivity properties for PS server farms with JSQ load balancing, and discuss the near-optimality of JSQ. Finally, we propose a novel many-server heavy-traffic regime to study load balancing policies. We use the proposed many-servers scaling to present a new closed-form approximation for the joint distribution of queue lengths under Exponential service distribution. The analysis of the many-servers scaling leads to many useful insights into the behavior of JSQ, including the first approximation for the distribution of response time. Finally, we use the proposed scaling to analytically study load balancing policies for the case where server speeds are heterogeneous.

5.1 Introduction

In this chapter, we are motivated by web server farm architectures serving static requests. Requests for files (or HTTP pages) arrive at a front-end dispatcher, which *immediately* routes the request to one of the servers in the farm for processing using a load balancing or task assignment policy. It is important that the dispatcher not hold back the arriving connection request, or the client will time out and possibly submit more requests. The bottleneck resource at a web server is often the uplink bandwidth. This bandwidth is shared by all files requested in a round-robin manner with a small granularity, which is well-modeled by the idealized processor sharing (PS) scheduling policy [75]. We are thus interested in a *PS server farm with immediate dispatch.* Time sharing servers are beneficial in that they allow "short jobs" to get processed quickly without being stuck waiting behind long jobs, and are thus 'fair'. This is particularly important, since measurements have shown that requested files sizes, and the associated service requirements, are highly variable, (e.g., heavy-tailed [21, 42])
.

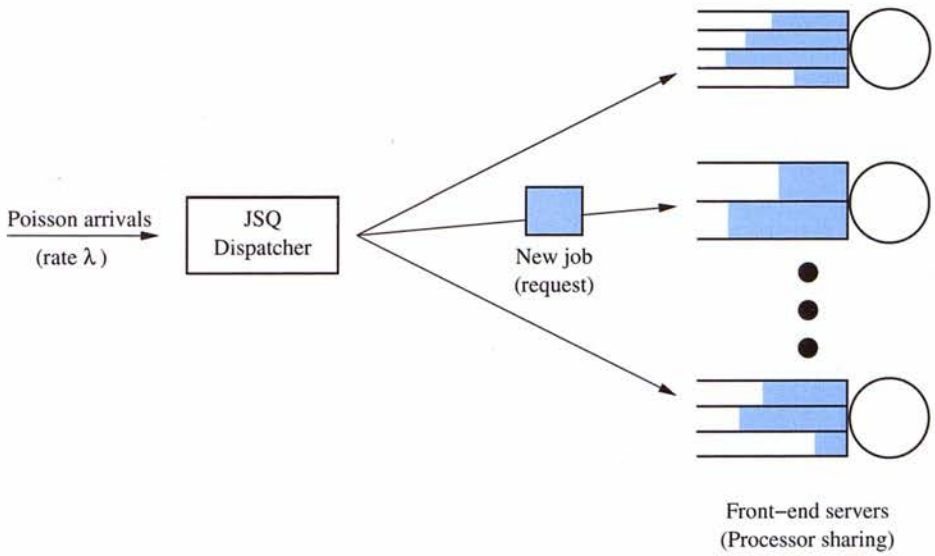


Figure 5.1: Server farm with front-end dispatcher and K identical processor sharing back-end servers.

Join-the-Shortest-Queue (JSQ) policy is the most popular load balancing heuristic used in PS server farms today; e.g., it is used in Cisco Local Director, IBM Network Dispatcher, Microsoft Sharepoint and F5 Labs BIG/IP. Under JSQ, an incoming

request is routed to the server with the least number of unfinished requests. Thus, JSQ strives to balance load across the servers, reducing the probability of one server having several jobs while another server sits idle. From the point of view of a new arrival, it is a *greedy policy* for the case of PS servers, because the arrival would prefer sharing a server with as few jobs as possible. We refer to a PS server farm with JSQ routing as a *JSQ/PS server farm.*

Model and Notation

We model the arrival process of jobs as a stationary Poisson process. We assume that there is a single dispatcher (router) and K identical PS servers, each with unlimited waiting space, as depicted in Figure 5.1. We assume that dispatching is immediate using the JSQ policy. Ties are broken by randomly choosing (with equal probabilities) among the servers with the fewest jobs. No jockeying is allowed between the servers (once a job is dispatched to a server, it stays there until completion).

Consequently, the JSQ/PS server farm acts as an $M/G/K$/JSQ/PS queueing model, with JSQ denoting the policy used to assign arrivals to the servers and PS denoting the scheduling rule (service discipline) used by each server. Jobs arrive as a Poisson stream with rate λ and are dispatched immediately to one of the K servers with the fewest jobs. For most of the chapter, we will assume that the servers are identical with speed μ. The service requirements are drawn independently from a general distribution with mean 1 (the G) and service is performed at each server according to PS. *We define the load of this system, ρ, as the per-server load* $\rho = \lambda/(K\mu)$ (unlike previous chapters). We sometimes use the extra notation $M(\lambda)/G(\mu)/K$/JSQ/PS to denote that the arrival rate is λ and the server's speed is μ. We will use N to denote the random variable for the number of jobs at a *single* PS queue in the server farm.

Summary of Results

Despite the ubiquity of JSQ/PS server farms, analytical results on the performance of JSQ in this setting are very limited. The existing analysis on JSQ involves *First-Come-First-Serve (FCFS) server farms*, where the servers employ FCFS scheduling. Within the JSQ/FCFS setting, almost all analysis is restricted to 2 servers, often with exponentially-distributed job sizes. For more than 2 servers, while some very appealing approximations exist, the accuracy of those approximations decreases as the number of servers is increased or as the job-size distribution becomes more variable. Prior work is detailed in Section 5.2.

In this chapter we provide the first analysis of the JSQ/PS model. In particular, we provide a way to approximate the steady-state distribution of queue-length (number of jobs in the system) at a server, which also yields the mean response time via Little's Law. While our analysis is approximate, the accuracy of our approximation is extremely good: $< 3\%$ error for mean response time and only slightly more for the second moment of queue length. More importantly, the error does *not* seem to increase beyond 3% with increased numbers of servers, or with an increase in job-size variability. We compliment this approximation with asymptotic analysis of the stationary joint distribution of queue lengths at the servers, and the stationary distribution of response time for Exponential service distribution under a novel many-servers heavy-traffic limit.

1. Bounded-Sensitivity

We begin by investigating the sensitivity of the $M/G/K$/JSQ/PS model to the variability of the service distribution in Section 5.3. In Theorem 5.1 we prove that for the *degenerate hyperexponential* (H_2^*) service distribution, the joint distribution of the number of jobs at the servers (and hence the mean response time) depends on the service distribution only through its mean. We then perform numerical experiments in the light-traffic regime with the more broad class of H_2 service distribution, and observe an interesting bounded-sensitivity phenomenon. To examine other job-size distributions, we resort to extensive simulations of a wide class of distributions, including hyperexponential distributions, Erlang distributions, Weibull distributions, the deterministic distribution and bimodal distributions (mixture of two point masses), which further lend support to the *bounded-sensitivity* hypothesis. Thus we have:

$$M/G/K/JSQ/PS \approx M/M/K/JSQ/PS$$

where the approximation is quite close for at least the first two moments of queue length.

2. Single Queue Approximation for $M/M/K$/JSQ/PS

Based on the results and observations of Section 5.3, we turn focus on analyzing the $M/M/K$/JSQ/PS model. We accomplish this goal in what we believe is an interesting innovative way. In Section 5.4 we introduce a new approximation technique for server farms, which we call the *single-queue approximation* (SQA). The key idea behind SQA is the following: Instead of analyzing the entire multi-server model, we just concentrate on a single queue in the server farm, say queue Q, and model its behavior *independently* of all the other queues. To capture the effect of the other queues, without directly considering them, we model the arrival process into queue Q by a stochastic point process with state dependent rates. In particular, we assume that the arrival process into queue Q has stochastic intensity $\lambda(N_Q(t))$, where $N_Q(t)$ is the queue length of Q at time t and $\lambda(n)$ is the long-run arrival rate when Q has

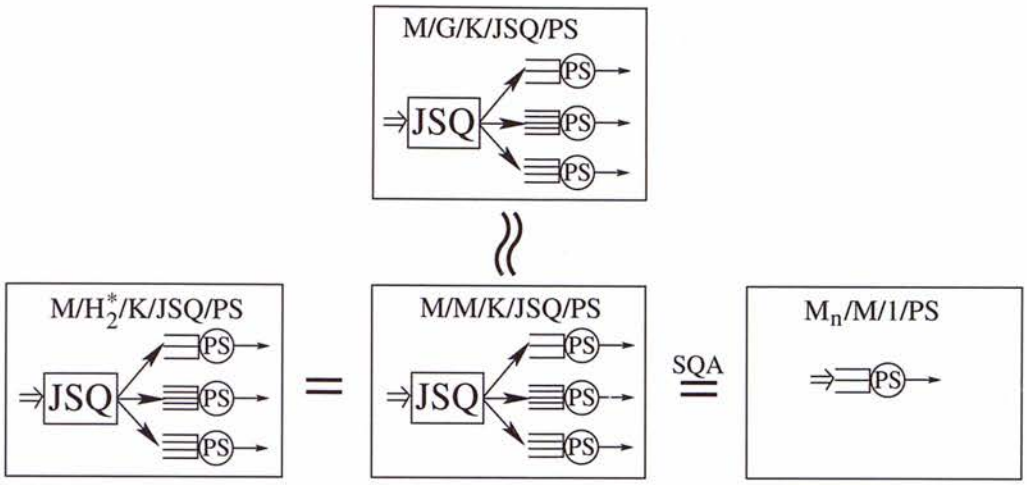


Figure 5.2: A pictorial view of some results in the chapter.

n customers in the original multi-server model.

To determine the $\lambda(n)$'s, we begin with extensive simulation experiments, and find stunning regularity in the results: $\lambda(n) \approx \mu\rho^K$ for all $n \geq 3$. We support the observation by Theorem 5.3 which proves that

$$\frac{\lambda(n)}{\mu} \to \rho^K, \text{ as } n \to \infty$$

in the case where $K = 2$, and provide further support in Section 5.6 where we prove that under a suitable (in fact the only non-degenerate) many-servers scaling, $\lambda(n) = \mu\rho^K$ for $n \geq 3$. It is this critical observation that lends SQA its tractability for the JSQ-PS model, leaving only three parameters to be determined: $\lambda(0)$, $\lambda(1)$ and $\lambda(2)$, which we determine using a combination of analysis and simulation, obtaining closed-form expressions for all the conditional arrival rates as functions of λ, μ and K.

Figure 5.2 pictorially summarizes some of the results in this chapter. It is important to note that once we know that: $M/G/K/JSQ/PS \approx M/M/K/JSQ/PS \equiv M/M/K/JSQ/FCFS$, we can apply other methods in the literature to solve the $M/M/K/JSQ/FCFS$ as well, e.g. Blanc [27], Nelson and Philips [117], Lin and Raghavendra [110].

3. Near-optimality of JSQ for $M/G/K/\cdot/$PS

In Section 5.5, we address the question:

Are there smart load balancing policies that substantially outperform JSQ?

There certainly exist sample paths where JSQ can yield mean response time twice as much as the optimal *even for deterministic job sizes.* However, for the $M/G/K/\cdot/\text{PS}$ model, we find via simulations that JSQ is impressively close to optimal, despite using far less information about system state than the other routing policies against which it is compared.

4. Many-server asymptotics, and load balancing with heterogeneous servers In Section 5.6, we fill the gap left by lack of exact analysis for the $M/M/K/\text{JSQ/PS}$ model by introducing a many-server "heavy-traffic" scaling, and presenting the stationary analysis under the proposed scaling. The scaling is obtained by letting the number of servers $K \to \infty$, while simultaneously increasing the arrival rate so that ρ^K converges to a positive constant $0 < \theta < 1$. Equivalently $K\mu - \lambda$ converges to a constant. The intuition is that under the proposed scaling, the *marginal queue length distribution* at a single server converges to a limit, and is the only scaling where the response time converges to a non-degenerate limit in distribution.

The many-server analysis can be seen as a complement to the approximation of Nelson and Philips [117] and Blanc [27] which are tight when traffic is light, and also to the exact analysis of $K = 2$ case by Adan, Wessels and Zijm [8, 9]. In addition, the analysis provides useful insights into the behavior of the JSQ load balancing policy. We use our scaling to propose the **first approximation for the distribution of response time for $M/M/K/\text{JSQ/PS}$ model**. The author is currently working on an analysis for general service distribution under the proposed scaling to obtain closed-form bounded-sensitivity results.

Finally, we utilize our many-server heavy-traffic scaling to analyze optimal routing policies for the $M/M/K/\cdot/\text{PS}$ model, and prove that, counter to intuition, JSQ remains optimal, while the greedy policy which sends to the server where the arrival gets served at the maximum rate is far from optimal.

5.2 Prior Work

There has been no previous mathematical analysis of the $M/G/K/\text{JSQ/PS}$ model. However, Bonomi [29] conducted a simulation study for the special case of two servers. He showed that, among all policies that base their decisions only on the queue lengths at the servers, JSQ minimizes the mean response time for the PS scheduling rule and exponential service requirements. Bonomi also proposed policies that improve slightly upon JSQ (5% improvement), for some general job-size distributions, by exploiting the remaining service times of jobs. He showed via simulation that common load-balancing schemes that perform well for JSQ/FCFS do not per-

form well for JSQ/PS. Bonomi observed that, while Least-Work-Left (LWL) is good for FCFS, it is not good for PS. However, we find that LWL is not always bad; see Figure 5.8.

By contrast, there is a lot of work on the JSQ/FCFS model (recall that under exponential workloads, JSQ/FCFS is equivalent to JSQ/PS with respect to the stationary queue length distribution). However, even the $M/M/K$/JSQ/FCFS model remains quite intractable. Several authors, including Koole, Sparaggis and Towsley [107], Winston [157], and Ephremides et al. [54], consider the optimality of JSQ for FCFS servers in certain constrained settings involving a job-size distribution with non-decreasing likelihood ratio and various assumptions on not knowing job sizes a priori. Note, however, that JSQ is far from optimal for FCFS servers with highly-variable job sizes [41, 73].

Almost all papers analyzing JSQ/FCFS performance are limited to 2 servers, an exponential job-size distribution and the mean response time metric. Among the classic papers are Kingman [97] and Flatto and McKean [58]. They use generating functions to derive the joint probability distribution of queue lengths and express the mean response time as an infinite sum, which in practice requires truncation to compute. Wessels, Adan, and Zijm [8] show that Kingman's result can be derived more intuitively via the compensation approach. Approximations for the mean response time have been obtained by state space truncation of the Markov chain [39, 64, 126], and Lui, Muntz and Towsley obtain bounds for JSQ (and its heterogeneous counterpart, the Minimum Expected Response time policy) by constructing models which upper or lower bound the mean response time under JSQ and can be analyzed numerically [112]. Heavy traffic approximations for JSQ/FCFS also exist and are evaluated in [59, 105]. Lastly, Boxma and Cohen [33] obtain a functional representation for the mean response time using boundary value approach. These methods are exact. However they are not always computationally efficient and do not generalize to higher values of K.

For analyzing the mean response time for $M/M/K$/JSQ/FCFS with $K > 2$ servers, only approximations exist. Nelson and Philips [117] use the following idea: They look at the steady-state probability of the $M/M/K$/FCFS queue (with a central queue) as an estimate for the total number of jobs in the JSQ/FCFS system, and then assume that the jobs in the system are divided equally (within 1) among each of the queues. Lin and Raghavendra [110] follow the approach of approximating the number of busy servers by a binomial distribution and then also assume that the jobs are equally divided among each of the queues (within 1). The Nelson and Philips demonstrates error less than 8% for K up to 16 with exponentially distributed job sizes for the cases presented in the paper. They also provide an empirically obtained correction factor which drops the error to 2%. However, we show that the Nelson-

Philips approximation can overestimate the mean response time by as much as a factor of 2. Lin and Raghavendra method yields less than 3.5% error for K up to 64 for the cases presented. Blanc [27] presents a numerical approximation based on a power-series expansion of the state probabilities as functions of the load of the system. Blanc's approximation performs well when the load or number of servers is small. There are also some numerical methods papers that don't lead to a closed-form solution, but are accurate and computationally efficient for not-too-large K, see for example [6, 10, 112].

Recently Bramson et al. [35] have presented asymptotic analysis of 'JSQ-type' dispatching schemes. They consider dispatching policies of the following kind: an arrival picks d random servers out of K, and joins the most favorable (shortest queue, least work) among them. The authors consider the limit where $K \to \infty$, and the arrival rate increases as $\lambda = \theta \cdot K$ $(0 < \theta < 1)$. For PS servers with shortest queue criterion, the authors are able to show an *insensitivity result*: the mean response time depends only on the mean of the service distribution and not on the higher order characteristics. Intuitively one expects such a result to hold because, as $K \to \infty$, the queues become asymptotically independent of each other. Thus the arrival process into a particular queue is a *state-dependent Poisson process*, and insensitivity of mean response time to higher moments of the service distribution under such an arrival process is a well-known result. Previously, Mitzenmacher [114] had characterized the steady-state joint queue length distribution under the same asymptotic scaling and dispatching policy for exponential service distributions. At the outset, it is not clear if and when such an insensitivity result holds for the JSQ-PS model we consider.

There has also been a lot of work investigating the optimality of routing policies for heterogeneous servers under Exponential service distribution. Nelson and Towsley [118] propose a policy, Greedy-Throughput, which routes jobs to servers so as to maximize the number of departures before the next arrival, and thus depend on knowing the arrival rate. Shenker and Weinrib [135, 136] propose policies that estimate the value functions of the solution of a certain stochastic dynamic programming problem by observing the state, and thus are able to adapt to changes in the arrival rate.

5.3 Bounded-sensitivity of JSQ/PS Model

The first question that must be raised when performing analysis of a queueing model is: Does the service distribution influence the performance at all? This question acquires further prominence for the JSQ/PS model because it is well established that for a Poisson arrival process, the mean response time of a single PS queue depends on the service distribution only through the mean [91]. Further, Bramson et al. [35]

have recently proved that a similar insensitivity also holds asymptotically when jobs are routed to shortest of a small set of randomly chosen servers, and the total number of servers grows to infinity. In this Section, we show that while the JSQ/PS model does not exhibit complete insensitivity, there is evidence of *bounded-sensitivity* – the effect of higher order characteristics of the service distribution beyond the mean is bounded.

Insensitivity with the Degenerate Hyperexponential Distribution

We begin by proving that for the special degenerate hyperexponential class of service distribution, the $M/G/K/$JSQ/PS model exhibits perfect insensitivity. Recall the definition of the degenerate hyperexponential distribution (denote by H_2^*):[1]

A random variable X distributed according to the H_2^* distribution with mean $1/\mu$ and SCV C^2, is given by

$$X \sim \begin{cases} 0 & w.p.\ p \\ \text{Exp}(\mu^*) & w.p.\ 1-p\ , \end{cases}$$

where $p = (C^2-1)/(C^2+1)$ and $\mu^* = \mu(1-p)$. We will use the shorthand $H_2^*(\mu^*, p)$ to denote the above.

As mentioned before, the degenerate hyperexponential distribution is a relatively minor modification of the Exponential distribution, but provides an additional parameter to represent the full range of SCV C^2 from 1 to ∞. The next result shows that if the job sizes are drawn from an H_2^* distribution, then the steady-state queue-length distribution and the mean response time in the resulting $M/H_2^*/K/$JSQ/PS model depend only on the mean job size, and not on the remaining free parameter; i.e., we have insensitivity within this H_2^* class.

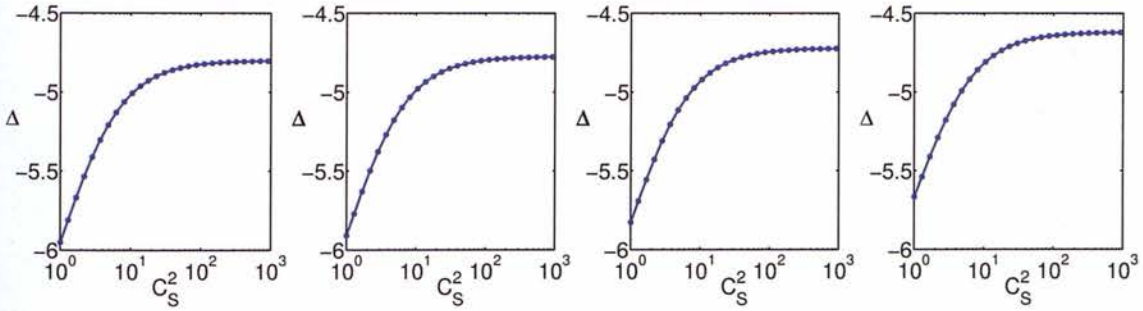
Theorem 5.1 *The queueing systems $M(\lambda)/H_2^*(1-p,p)/K/JSQ/PS$ and $M(\lambda)/M(1)/K/JSQ/PS$ have identical steady-state queue-length distributions and mean steady-state response times. Moreover, the response-time distribution of the $M(\lambda)/H_2^*(1-p,p)/K/JSQ/PS$ system is a mixture of a unit point mass at 0, with probability p, and the response-time distribution of the $M(\lambda)/M(1)/K/JSQ/PS$ system multiplied by $1/(1-p)$, with probability $1-p$.*

[1] We have already seen in the previous chapters that the H_2^* distribution is extremely useful approximately capture the variability of job sizes in multi-server systems. See also, [151, 155].

Proof: The jobs with size 0 do not have to wait, since the servers are doing processor sharing. Therefore, only accounting for the non-zero-sized jobs, the joint distribution of the original $M(\lambda)/H_2^*(1-p,p)/K/\text{JSQ/PS}$ system is identical to an $M(\lambda(1-p))/M(1-p)/K/\text{JSQ/PS}$. However, the latter system can be thought of as an $M(\lambda)/M(1)/K/\text{JSQ/PS}$ system seen on a slower time scale, and thus have the same stationary joint queue length distribution, proving the first part of the theorem. From the perspective of response time, the response time of the p-proportion of zero-sized jobs is the deterministic distribution with mean 0, while the remaining $(1-p)$-proportion of non-zero-sized jobs experience an $M(\lambda(1-p))/M(1-p)/K/\text{JSQ/PS}$ system. By employing the time scaling argument again, the $(1-p)$-proportion of non-zero-sized jobs experience a response time $1/(1-p)$ times higher than that in an $M(\lambda)/M(1)/K/\text{JSQ/PS}$ system. ∎

Bounded-Sensitivity for H_2 service distribution in light traffic

The provable insensitivity of Theorem 5.1 is for a very special class of service distributions. We will show that this insensitivity property does not extend exactly to other job-size distributions, but an approximate form of it does; i.e., we have *near-insensitivity* or *bounded sensitivity.*



(a) $\lambda = 0.005$ (b) $\lambda = 0.01$ (c) $\lambda = 0.02$ (d) $\lambda = 0.04$

Figure 5.3: Light-traffic numerical results illustrating bounded-sensitivity under H_2 service distribution with mean 1. For each value of SCV C_S^2 shown, the largest values of mean number of jobs within the H_2 class of service distribution was evaluated numerically for a 2-server JSQ/PS system with a finite buffer of 5 at each server. The Y-axis shows $\Delta = \frac{2\mathbf{E}[N]-(2\rho+4\rho^3)}{\rho^4}$.

In Figure 5.3, we show results from experiments with the two-phase hyperexponential service distribution in light traffic. We considered a 2-server JSQ/PS system where

each server has a finite buffer space of 5 jobs, and numerically solve for the stationary distribution for job size distributions with SCV ranging from 1 to 1000. Since the buffer is finite, to remove the effect of lost jobs, we chose arrival rates small enough so as to make the loss probability negligible. For each value of SCV, we find that H_2 distribution that results in the largest mean number of jobs in the system and show that result in Figure 5.3. We observe that in light traffic, the mean number of jobs for a 2-server system has the expansion: $\mathbf{E}[N] \sim \rho + 2\rho^3 + \frac{\Delta}{2} \cdot \rho^4 + o(\rho^4)$, where the effect of the job size variability shows up in the coefficient Δ. It is this quantity that is plotted in Figure 5.3.

There are a couple of important observations: As $\rho \to 0$, the coefficient of the ρ^4 term, Δ, does converge to a non-degenerate function of the service distribution. Further, as the job size variability $C_S^2 \to \infty$, Δ remains bounded (reminiscent with (3.4), it appears to grow as $a + \frac{1}{C_S^2 + b}$). Therefore, there is strong evidence of bounded sensitivity to the service distribution in the JSQ/PS model.

Near-Insensitivity for General Job-Size Distributions

As further evidence of near-insensitivity, we simulate an $M/G/K$/JSQ/PS system with the following job-size distributions (all with mean 2, in increasing order of C_S^2):

1. `Deterministic`: point mass at 2 (variance = 0)
2. `Erlang2`: sum of two exponential random variables with mean 1 (variance = 2)
3. `Exponential`: exponential distribution with mean 2 (variance = 4)
4. `Bimodal-1`: (mean = 2, variance = 9)

$$X = \begin{cases} 1 & w.p.\ 0.9 \\ 11 & w.p.\ 0.1 \end{cases}$$

5. `Weibull-1`: Weibull with shape parameter = 0.5 and scale parameter = 1 (heavy-tailed, mean = 2, variance = 20)
6. `Weibull-2`: Weibull with shape parameter = $\frac{1}{3}$ and scale parameter = $\frac{1}{3}$ (heavy-tailed, mean = 2, variance = 76)
7. `Bimodal-2`: (mean = 2, variance = 99)

$$X = \begin{cases} 1 & w.p.\ 0.99 \\ 101 & w.p.\ 0.01 \end{cases}$$

The load was set at $\rho = 0.9$ and simulations were run for $K = 2, 4, 8$ and 16 servers. For each value of K and each distribution, the simulation was run 50 times, each run consisting of $K \times 10^7$ departures. Statistics for completed requests were considered. Figure 5.4 shows the 95% confidence intervals for the mean response time and second moment of queue length, for each service distribution and $K = 2, 8$ (complete simulation results appear in [70]). The mean response time in Figure 5.4 never deviates by more than 2% from the exponential case, regardless of the job-size distribution, and the deviation for the second moment of queue length is barely over 3%.

This section has aimed to provide ample justification for approximating the mean response time of an $M/G/K$/JSQ/PS system by an $M/M/K$/JSQ/PS system. We address this latter goal in the next section.

5.4 Single-Queue-Approximation for $M/M/K$/JSQ/PS

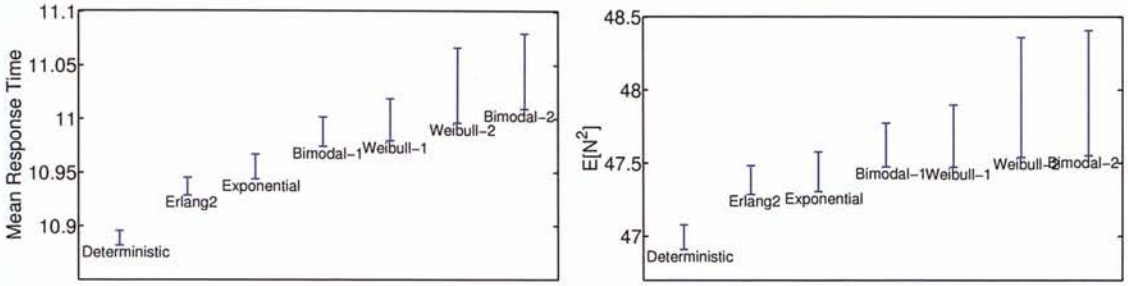
To understand SQA, it helps to recall that the main obstacle in analyzing routing policies such as JSQ is that the states of all the queues are correlated, necessitating a multidimensional state space for the system. Thus exact analysis requires that we work with the vector of queue lengths and possibly also the remaining service requirements of all jobs at each server. The SQA method allows one to approximate the marginal queue length distribution of each queue in the server farm by modeling each queue independently of the other, thereby avoiding the above difficulties.

Consider a queue Q in the server farm. Under SQA, we model Q by a queue Q', where the arrival rate of jobs into Q' depends only on the queue length of Q', and not on the state of any other queues. Thus SQA approximates each queue of the $M/G/K$/JSQ/PS model by an associated $M_n/G/1/PS$ model, where M_n denotes a state-dependent Markovian arrival process. Specifically, at time t, the arrival process acts as a Poisson process with rate $\lambda(N_{Q'}(t))$, where $N_{Q'}(t)$ is the queue length of Q' at time t and $\{\lambda(n) : n \geq 0\}$ is a deterministic sequence with $\lambda(n)$ being the actual long-run arrival rate into queue Q (of the original server farm) conditioned on the queue length of Q being n. We define $\lambda(n)$ in Definition 5.1.

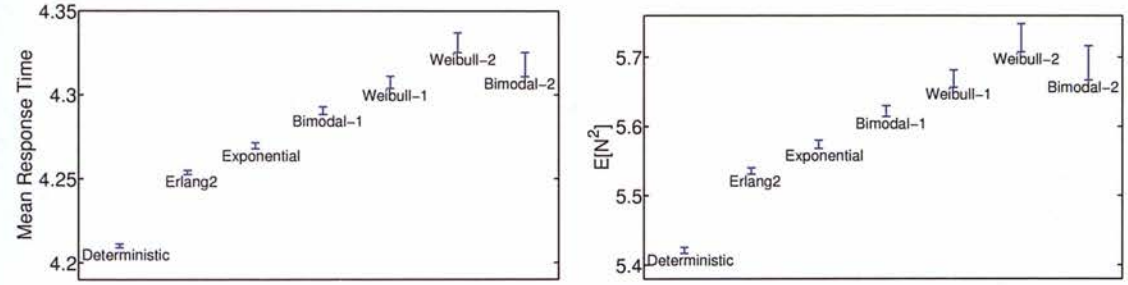
Definition 5.1 *Given a general $M/G/K/\mathcal{R}/\mathcal{S}$ model, the conditional arrival rate into one designated queue Q given that it has n jobs, $\lambda(n)$, is defined as*

$$\lambda(n) = \lim_{t \to \infty} \frac{A_n(t)}{T_n(t)} , \tag{5.1}$$

where $A_n(t)$ is the number of arrivals into Q during the time interval $[0, t]$ that see



(a) $K = 2$



(b) $K = 8$

Figure 5.4: 95% Confidence intervals for mean response time (left column) and second moment of queue length (right column) in the $M/G/K$/JSQ/PS model with $\rho = 0.9$ and mean job size 2 for different job-size distributions based on simulations. The service distributions are arranged on the x-axis in order of increasing C_S^2 (the C_S^2 values are $\{0, 1, 2.25, 5, 19, 24.75\}$, respectively).

n jobs at Q on arrival (excluding themselves), while $T_n(t)$ is the total time spent by Q with n jobs during the time interval $[0, t]$.

Formally, the arrivals form a stochastic point process with stochastic intensity $\lambda(N_{Q'}(t))$, as defined in §II.3,5 in Brémaud [36].

The state-dependence in the arrival rate $\lambda(n)$ is intended to capture some of the dependence inherent in the full $M/G/K$/JSQ/PS model. Consider an $M/G/K$/JSQ/PS model with outside arrival rate λ. The average arrival rate into each queue is λ/K. However, if we condition on the fact that some designated queue has n jobs, then the arrival rate into that designated queue is no longer λ/K. In fact, with JSQ routing,

we expect that the long-term arrival rates into that designated queue, $\lambda(n)$, should *decrease* as n increases, because it is likely that at least one other queue is shorter than the designated queue. This is precisely what happens: $\lambda(0)$ is larger than λ/K, but $\lambda(n)$ decreases as n increases. In this way, having state-dependent arrival rates captures some of the influence of the other queues on the designated queue.

The SQA method is not limited to the $M/G/K$/JSQ/PS model. We can consider other routing policies $\mathcal{R}$ (see e.g., Definition 5.2) for the K-server model and other scheduling rules $\mathcal{S}$ at this single queue. We can also accommodate heterogeneous servers. We now specify a class of routing policies for which SQA works well.

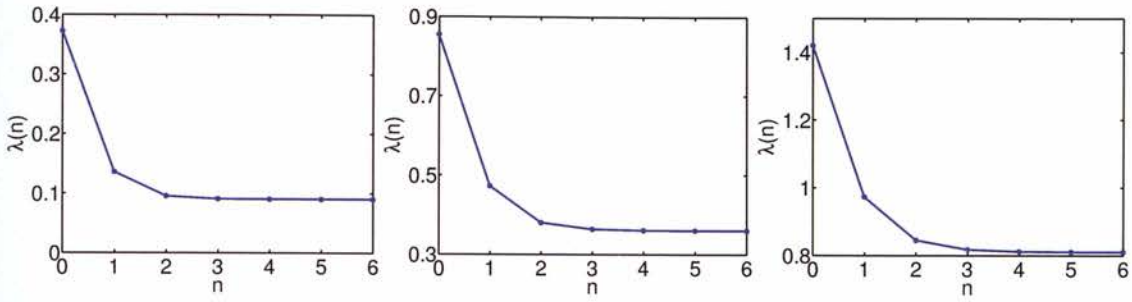
Definition 5.2 *A* stationary queue-length-dependent routing policy *is a time-stationary routing policy that uses only information about queue lengths at the servers at the instant of an arrival. The decisions may be made probabilistically, and may be biased in favor of certain servers (allowing the modeling of heterogeneous servers).*

In fact, SQA in some senses is a misnomer. If in addition to accurately computing the conditional arrival rates, we could also compute the conditional departure rates $\mu(n)$, then SQA yields the exact marginal distribution for the number of jobs at a designated queue and not just an approximation (intuitively, since the number of arrivals while in state n, and departures while in state $n+1$ are within ± 1, $\frac{\lambda(n)}{\mu(n+1)} \rightarrow \lim_{t\rightarrow\infty} \frac{T_{n+1}(t)}{T_n(t)}$). However, when the service distribution is Markovian, finding the arrival rates is sufficient to produce the exact stationary queue-length distribution:

Theorem 5.2 *Consider an $M/M/K/\mathcal{R}/\mathcal{S}$ model, where $\mathcal{R}$ is any stationary queue-length-dependent routing policy, e.g., JSQ, and $\mathcal{S}$ is any stationary, size-independent, work-conserving scheduling policy, e.g., PS. Assume that this model has a unique proper steady-state distribution. Let Q be any particular server in the $M/M/K/\mathcal{R}/\mathcal{S}$ model. Then SQA with the exact conditional arrival rates $\lambda(n)$ yields the same steady-state queue-length distribution as in the original $M/M/K/\mathcal{R}/\mathcal{S}$ model.*

Proof: The result easily follows via sample path arguments employing conditional arrival and departure rates, see [52] (Theorem 1.9, Section 1.4.2, page 21). ■

What makes SQA especially appealing for approximating the $M/M/K$/JSQ/PS model is that due to a surprising regularity in the conditional arrival rates, the number of parameters to be approximated essentially reduces to two. We elaborate on this next.



(a) $\rho = 0.3$ (b) $\rho = 0.6$ (c) $\rho = 0.9$

Figure 5.5: Illustrating the convergence of conditional arrival rates, $\lambda(n)$, for a given queue of an $M/M/K$/JSQ/PS, with mean job size 1, where $K = 2$.

5.4.1 The Conditional Arrival Rates

The feasibility of the SQA method hinges on obtaining the conditional arrival rates $\lambda(n)$, $n \geq 0$, defined in (5.1). In this section we will derive closed-form approximations for these conditional arrival rates. Our results here draw on extensive simulation experiments in which we estimated these conditional arrival rates for a range of job-size distributions and other model parameters.

First, we observed that the conditional arrival rates rapidly converge to a limiting value as n (the number of jobs at the queue) increases. Indeed, we found that

$$\frac{\lambda(n)}{\mu} \approx \rho^K \quad \text{for all} \quad n \geq 3 \,, \tag{5.2}$$

for $\rho \leq 0.95$. Simulations of the $M/M/K/JSQ/FCFS$ model showed this approximation to be consistently within 2% of the actual values (provided that ρ is not too extreme, i.e., for $0.3 \leq \rho \leq 0.95$). We will provide further analytical arguments justifying this observation in Section 5.6.1. This fact is illustrated in Figure 5.5 for the case of $K = 2$. We also prove this convergence in the limit for the case $K = 2$ in Theorem 5.3 below, but it is easy to see the intuition behind the result. Since JSQ tries to equalize queue lengths, if a particular designated queues has many jobs, then there is a high probability that all servers are busy. In this case, the total number of jobs in the systems starts behaving as a central queue $M/M/K$. For the number of jobs to increase by 1 at a queue, the total number of jobs in the system should increase by K, which gives the tail decay rate as ρ^K.

Theorem 5.3 *For the $M(\lambda)/M(\mu)/2/JSQ/PS$ system,*

$$\lim_{n\to\infty} \frac{\lambda(n)}{\mu} = \rho^2 \tag{5.3}$$

The proof follows directly from the work of Adan et al. [8] on using the compensation approach to analyze the $M/M/2/JSQ/FCFS$ queue and will use Lemmas 5.1 and 5.2 mentioned below. We begin by reviewing the notation. Let $\pi_{m,n}$ be the stationary probability that length of queue 1 is m and length of queue 2 is n. For $m \geq 0$ and $r \geq 0$, define $q_{m,r}$ as:

$$q_{m,r} = \pi_{m,m+r} \tag{5.4}$$

That is, $q_{m,r}$ is the probability that queue 1 is the shorter queue and has m jobs and queue 2 has $m + r$ jobs.

Lemma 5.1 *[Adan et al. [8]] The stationary probabilities $q_{m,r}$ for $m \geq 0$ and $r \geq 1$ are given by:*

$$q_{m,r} = Cx_{m,r}$$

The normalization constant C is given by

$$C = \frac{2(1-\rho^2)(2-\rho)}{\rho(2+\rho)}$$

and

$$x_{m,r} = \sum_{i=0}^{\infty} d_i(\alpha_i^m + c_i\alpha_{i+1}^m)\beta_i^r \tag{5.5}$$

where α_i, β_i, c_i and d_i's are given by the following recursion scheme:

$$\begin{aligned}
d_0 &= 1 \\
\alpha_0 &= \rho^2 \\
\beta_0 &= \frac{\rho^2}{2+\rho} \\
\alpha_i\alpha_{i+1} &= 2\rho\beta_i^2 \\
\beta_i\beta_{i+1} &= \alpha_{i+1}^2/(2\rho + \alpha_{i+1}) \\
c_i &= -\frac{\alpha_{i+1} - \beta_i}{\alpha_i - \beta_i} \\
d_{i+1} &= -\frac{(\alpha_{i+1}+\rho)/\beta_{i+1} - (\rho+1)}{(\alpha_{i+1}+\rho)/\beta_i - (\rho+1)} c_i d_i
\end{aligned}$$

We will use the following lemma to bound the infinite sum of (5.5) by a finite sum.

Lemma 5.2 *The infinite sum for $x_{m,r}$ $(m \geq 0, r \geq 1)$ in (5.5) can be bounded by the following finite sums:*

$$(\alpha_0^m + c_0\alpha_1^m)\beta_0^r + d_1(\alpha_1^m + c_1\alpha_2^m)\beta_1^r = \underline{x_{m,r}} < x_{m,r} < \overline{x_{m,r}} = (\alpha_0^m + c_0\alpha_1^m)\beta_0^r \quad (5.6)$$

Proof: Let $s_i = |d_i(\alpha_i^m + c_i\alpha_{i+1}^m)\beta_i^r|$. In [8] (Lemma 8), authors prove that:

$$s_{i+1} < Rs_i$$

where $R = 4/(4+2\rho+\rho^2) < 1$. Also as a consequence of Lemma 1 of [8], $d_{i+1}/d_i < 0$. That is d_i alternate signs, d_0 being defined to equal 1. Hence,

$$\begin{aligned} x_{m,r} &= s_0 - s_1 + s_2 - s_3 + s_4 - \ldots \\ &< s_0 - s_1 + Rs_1 - s_3 + Rs_3 - \ldots \\ &= s_0 - (1-R)(s_1 + s_3 + \ldots) \\ &< s_0 \\ &\stackrel{\text{def}}{=} \overline{x_{m,r}} \end{aligned}$$

and,

$$\begin{aligned} x_{m,r} &= s_0 - s_1 + s_2 - s_3 + s_4 - s_5 + \ldots \\ &> s_0 - s_1 + s_2 - Rs_2 + s_4 - Rs_4 + \ldots \\ &= s_0 - s_1 + (1-R)(s_2 + s_4 + \ldots) \\ &> s_0 - s_1 \\ &\stackrel{\text{def}}{=} \underline{x_{m,r}} \end{aligned}$$

■

Proof of Theorem 5.3: Let Π_n be the stationary probability that there are n jobs in queue 1. Since we know SQA is exact, we can express the conditional arrival rates, $\lambda(n)$, as

$$\lambda(n) = \mu\frac{\Pi_{n+1}}{\Pi_n} = \mu\frac{\sum_{i=0}^{\infty}\pi_{n+1,i}}{\sum_{i=0}^{\infty}\pi_{n,i}}$$

Let $x_{m,0} = C^{-1}q_{m,0}$. Since for $m > 0$,

$$q_{m,0} = \frac{1}{1+\rho}(2\rho q_{m-1,1} + q_{m,1}) \quad (5.7)$$

we also have the following bounds on $x_{m,0}$:

$$\frac{1}{1+\rho}(2\rho\underline{x_{m-1,1}} + \underline{x_{m,1}}) = \underline{x_{m,0}} < x_{m,0} < \overline{x_{m,0}} = \frac{1}{1+\rho}(2\rho\overline{x_{m-1,1}} + \overline{x_{m,1}})$$

Expressing π's in terms of the x's gives us the following bounds on $\lambda(n)$:

$$\underline{\lambda(n)} < \lambda(n) < \overline{\lambda(n)} \tag{5.8}$$

where,

$$\underline{\lambda(n)} = \mu\frac{\underline{x_{n+1,0}} + \sum_{i=1}^{\infty}\underline{x_{n+1,i}} + \sum_{j=0}^{n}\underline{x_{j,n+1-j}}}{\overline{x_{n,0}} + \sum_{i=1}^{\infty}\overline{x_{n,i}} + \sum_{j=0}^{n-1}\overline{x_{j,n-j}}} \tag{5.9}$$

$$\overline{\lambda(n)} = \mu\frac{\overline{x_{n+1,0}} + \sum_{i=1}^{\infty}\overline{x_{n+1,i}} + \sum_{j=0}^{n}\overline{x_{j,n+1-j}}}{\underline{x_{n,0}} + \sum_{i=1}^{\infty}\underline{x_{n,i}} + \sum_{j=0}^{n-1}\underline{x_{j,n-j}}} \tag{5.10}$$

The expression for $\overline{\lambda(n)}$ in (5.10) is obtained by upper bounding the numerator, Π_{n+1}, and lower bounding the denominator, Π_n. Doing the opposite gives $\underline{\lambda(n)}$ (5.9).

To prove the convergence of $\lambda(n)$, we will prove

$$\lim_{n\to\infty}\underline{\lambda(n)} = \lim_{n\to\infty}\overline{\lambda(n)} = \mu\rho^2$$

We will first show the convergence of $\overline{\lambda(n)}$. Proof for $\underline{\lambda(n)}$ is similar. Now,

$$\overline{\lambda(n)} = \mu\frac{\overline{x_{n+1,0}} + \sum_{i=1}^{\infty}\overline{x_{n+1,i}} + \sum_{j=0}^{n}\overline{x_{j,n+1-j}}}{\underline{x_{n,0}} + \sum_{i=1}^{\infty}\underline{x_{n,i}} + \sum_{j=0}^{n-1}\underline{x_{j,n-j}}} = \mu\frac{S_{n+1}}{S_n + T_n} \tag{5.11}$$

where,

$$\begin{aligned}
S_i =& \frac{\beta_0}{1+\rho}[2\rho(\alpha_0^{i-1} + c_0\alpha_1^{i-1}) + (\alpha_0^i + c_0\alpha_1^i)] + (\alpha_0^i + c_0\alpha_1^i)\frac{\beta_0}{1-\beta_0} \\
&+ \beta_0\left(\frac{\alpha_0^i - \beta_0^i}{\alpha_0 - \beta_0} + c_0\frac{\beta_0^i - \alpha_1^i}{\beta_0 - \alpha_1}\right) \\
T_i =& d_1\left[\frac{\beta_1}{1+\rho}[2\rho(\alpha_1^{i-1} + c_1\alpha_2^{i-1}) + (\alpha_1^i + c_1\alpha_2^i)] + (\alpha_1^i + c_1\alpha_2^i)\frac{\beta_1}{1-\beta_1}\right. \\
&\left.+\beta_1\left(\frac{\alpha_1^i - \beta_1^i}{\alpha_1 - \beta_1} + c_1\frac{\beta_1^i - \alpha_2^i}{\beta_1 - \alpha_2}\right)\right]
\end{aligned}$$

Dividing the numerator and denominator of (5.11) by α_0^{n-1}, taking $\lim_{n\to\infty}$ and noting that $\frac{\alpha_1}{\alpha_0} < 1$, $\frac{\alpha_2}{\alpha_0} < 1$, $\frac{\beta_0}{\alpha_0} < 1$ and $\frac{\beta_1}{\alpha_0} < 1$:

$$\lim_{n\to\infty}\overline{\lambda(n)} = \mu\alpha_0\frac{\frac{\beta_0}{1+\rho}[2\rho + \alpha_0] + \alpha_0\frac{\beta_0}{1-\beta_0} + \beta_0\left(\frac{\alpha_0}{\alpha_0-\beta_0}\right)}{\frac{\beta_0}{1+\rho}[2\rho + \alpha_0] + \alpha_0\frac{\beta_0}{1-\beta_0} + \beta_0\left(\frac{\alpha_0}{\alpha_0-\beta_0}\right)} \tag{5.12}$$

$$= \mu\alpha_0$$
$$= \mu\rho^2 \tag{5.13}$$

Similarly,

$$\lim_{n\to\infty} \underline{\lambda(n)} = \lim_{n\to\infty} \mu \frac{S_{n+1} + T_{n+1}}{S_n} = \mu\rho^2$$

and hence convergence of $\lambda(n)$ follows by convergence of its upper and lower bounds.

We believe that we can generalize the proof to any finite K, however we state it only for $K = 2$. ∎

Observe that it makes intuitive sense that $\lambda(n)$, the average arrival rate into a designated queue conditioned on that queue having n jobs, should decrease as n is increased, because, if the designated queue has many jobs then it is likely that other queues have fewer jobs than itself.

Next, consistent with the other near-insensitivity results, we have observed that these conditional arrival rates also exhibit near-insensitivity; there is almost no dependence on the variability of the job-size distribution. This fact is illustrated in Table 5.1 for the case of $K = 4$, with hyperexponential job-size distributions having squared coefficient of variation ranging from 1 to 64, where r denotes the fraction of load made up by one branch of the hyperexponential (hence $r = 0.5$ denotes a hyperexponential with balanced load on its branches). The near-insensitivity of the $\lambda(n)$'s provides further justification for focusing on the special case of an exponential job-size distribution.

Based on the key observation in (5.2), our task has been reduced to obtaining approximations for the first 3 conditional arrival rates: $\lambda(0), \lambda(1)$ and $\lambda(2)$. The following lemma, allows us to reduce our task further to just deriving two conditional arrival rates, $\lambda(0)$ and $\lambda(2)$, since $\lambda(1)$ can be estimated from these, assuming the relation in (5.2).

Lemma 5.3 *Under the approximating approximation of* (5.2) *for the* $M/M/K/JSQ/PS$ *model, we obtain*

$$\lambda(1) = \mu \frac{\left[\frac{\mu}{\lambda(0)} \frac{\rho - \rho^{K+1}}{(1-\rho)} + \rho^K - 1\right]}{1 + \lambda(2)/\mu - \rho^K} \ . \tag{5.14}$$

Proof: Since all the servers are homogeneous, the time-average arrival rate into any one queue is $\lambda/K = \mu\rho$. By Theorem 5.2, SQA is exact given the conditional arrival rates. Therefore, we can write the time average arrival rate into any server as

$$\mu\rho = \sum_{n=0}^{\infty} \Pi_n \lambda(n) \ .$$

		$\lambda(0)$	$\lambda(1)$	$\lambda(2)$	$\lambda(3)$	$\lambda(4)$	$\lambda(5)$	$\lambda(6)$
$C^2 = 0$		2.2379	0.9865	0.6931	0.6575	0.6605	0.6645	0.6678
$C_S^2 = 1$		2.2125	0.9962	0.7098	0.6631	0.6573	0.6550	0.6543
$C_S^2 = 2$	$r = 0.1$	2.2080	1.0000	0.7123	0.6629	0.6541	0.6516	0.6542
	$r = 0.5$	2.2074	0.9975	0.7119	0.6609	0.6520	0.6522	0.6525
	$r = 0.9$	2.2077	0.9947	0.7114	0.6649	0.6560	0.6554	0.6557
$C_S^2 = 4$	$r = 0.1$	2.2068	1.0041	0.7144	0.6611	0.6513	0.6531	0.6522
	$r = 0.5$	2.2018	0.9992	0.7150	0.6653	0.6585	0.6553	0.6520
	$r = 0.9$	2.2075	0.9971	0.7110	0.6630	0.6572	0.6560	0.6549
$C_S^2 = 16$	$r = 0.1$	2.2032	1.0092	0.7201	0.6641	0.6544	0.6521	0.6536
	$r = 0.5$	2.1957	0.9982	0.7181	0.6649	0.6534	0.6510	0.6559
	$r = 0.9$	2.2091	0.9965	0.7146	0.6672	0.6598	0.6567	0.6572
$C_S^2 = 64$	$r = 0.1$	2.2061	1.0104	0.7157	0.6572	0.6515	0.6497	0.6597
	$r = 0.5$	2.1893	0.9959	0.7233	0.6702	0.6569	0.6526	0.6529
	$r = 0.9$	2.2072	0.9964	0.7136	0.6668	0.6583	0.6573	0.6554

Table 5.1: Conditional arrival rates for $M/H_2/K/JSQ/PS$ with $K = 4$ and $\rho = 0.9$, where the hyperexponential (H_2) distribution has parameters C_S^2 and r with mean 1, and the variability of H_2 ranges from $C_S^2 = 1$ to $C_S^2 = 64$. Results from simulation. (Conditional arrival rates for $M/D/K/JSQ/PS$ are also shown for reference in the top line.)

By Little's law (focusing on the servers), $1 - \Pi_0 = \rho$. Using that with (5.2), we obtain

$$\begin{aligned} \mu\rho &= (1-\rho)\lambda(0) + (1-\rho)\frac{\lambda(0)}{\mu}\lambda(1) + (1-\rho)\frac{\lambda(0)\lambda(1)}{\mu^2}\lambda(2) \\ &\quad + \left(\rho - (1-\rho)\frac{\lambda(0)}{\mu} - (1-\rho)\frac{\lambda(0)\lambda(1)}{\mu^2}\right)\rho^K \end{aligned} \tag{5.15}$$

This gives the desired approximation for $\lambda(1)$. ■

The approximations for $\lambda(2)$ and $\lambda(0)$ were obtained empirically using MATLAB's curve fitting toolbox (version 1.1.5), which uses a trust-region method for a nonlinear least-squares fit. For each value of load, ρ, we approximate $\lambda(2)$ as a function of K by a simple exponential function of the form

$$\lambda(2) \approx \mu\left(u_\rho v_\rho^K\right) \tag{5.16}$$

Empirical fit yields the following functions of ρ:

$$u_\rho = c_3\rho^3 + c_2\rho^2 + c_1\rho + c_0 \quad \text{and} \quad v_\rho = c_2'\rho^2 + c_1'\rho + c_0' \,,$$

where $c_3 = -0.29$, $c_2 = 0.8822$, $c_1 = -0.5349$, and $c_0 = 1.0112$, while $c_2' = -0.1864$, $c_1' = 1.195$, and $c_0' = -0.016$.

For $\lambda(0)$, we used a function with two exponential terms, namely,

$$\lambda(0) \approx \mu \left(a_\rho - b_\rho c_\rho^K - d_\rho e_\rho^K \right) \tag{5.17}$$

where $c_\rho, e_\rho < 1$. The constant a_ρ in (5.17) is clearly the limit as $K \to \infty$. The following lemma gives the value of this limit.

Lemma 5.4

$$\lim_{K \to \infty} \frac{\lambda(0)}{\mu} = \frac{\rho}{1-\rho} \tag{5.18}$$

Proof: For any value of $\rho < 1$, as the number of servers becomes large enough, any arrival will find at least one server idle with high probability. Therefore, $\lambda(i) \approx 0$ for $i \geq 1$. Equating the expressions for time average arrival rates into any queue,

$$(1-\rho)\lambda(0) = \mu\rho \quad \text{or} \quad \frac{\lambda(0)}{\mu} = \frac{\rho}{1-\rho} .$$

■

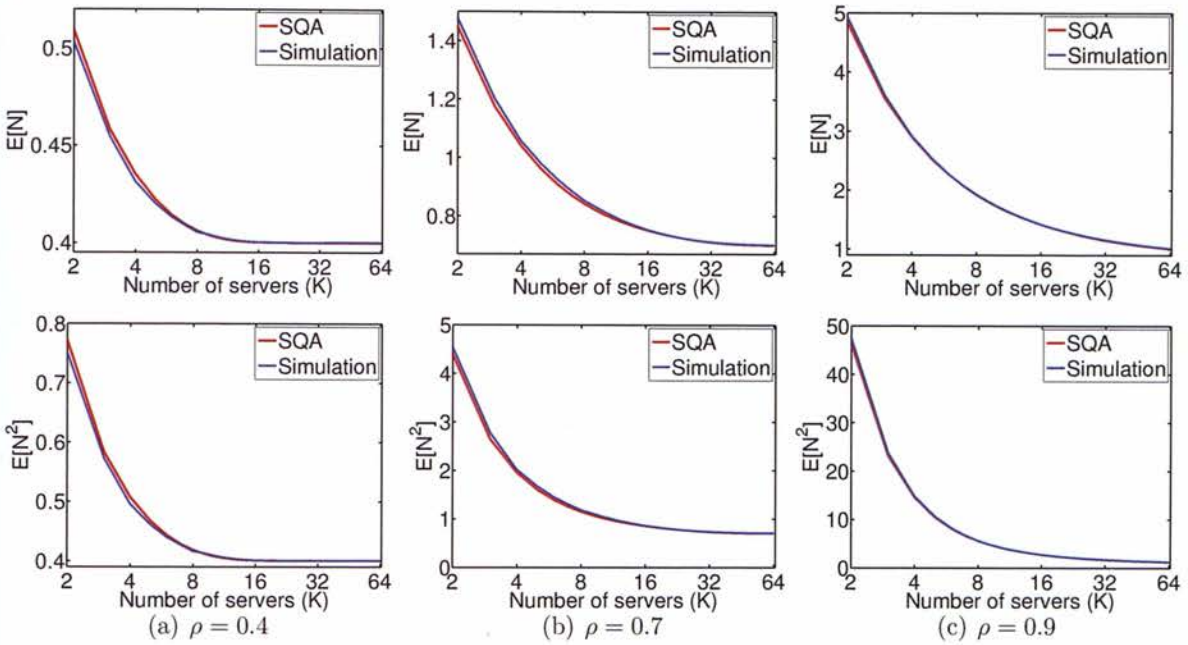
The remaining functions b_ρ, c_ρ, d_ρ, and e_ρ were determined empirically for $0.3 \leq \rho \leq 0.95$; we did not have accurate enough simulations outside this range. The final functions are

$$\begin{aligned}
b_\rho &= \frac{-0.0263\rho^2 + 0.0054\rho + 0.1155}{\rho^2 - 1.939\rho + 0.9534} \\
c_\rho &= -6.2973\rho^4 + 14.3382\rho^3 - 12.3532\rho^2 + 6.2557\rho - 1.005 \\
d_\rho &= \frac{-226.1839\rho^2 + 342.3814\rho + 10.2851}{\rho^3 - 146.2751\rho^2 - 481.1256\rho + 599.9166} \\
e_\rho &= 0.4462\rho^3 - 1.8317\rho^2 + 2.4376\rho - 0.0512
\end{aligned}$$

5.4.2 Evaluating the Approximation

In this section we evaluate our SQA approximation for the $M/G/K$/JSQ/PS model, where the conditional arrival rates used in the SQA are the approximate ones derived in Section 5.4.1. Our approach is not exact even for the case of an exponential service distribution, because the conditional arrival rates are approximate. Therefore, we first evaluate our method for exponential distributions, and afterwards, we consider general service distributions.

Exponential Job Sizes



(a) $\rho = 0.4$ (b) $\rho = 0.7$ (c) $\rho = 0.9$

Figure 5.6: The top row shows the effectiveness of SQA in predicting mean queue length, and the bottom row shows the effectiveness of SQA in predicting the second moment of queue length. Results are shown for three values of load: $\rho = \{0.4,\ 0.7,\ 0.9\}$, K up to 64 servers.

Theorem 5.2 implies that SQA is exact if the conditional arrival rates are correct. In this section, we apply SQA with our approximate conditional arrival rates to determine the first two moments of queue lengths for exponential service requirements. The results are shown in Figure 5.6, where N represents the queue length of a single queue in the server farm.

From Figure 5.6, it is difficult to see that the SQA method with our derived approximate conditional arrival rates exhibits any error at all, when compared with simulations. However, the error is actually $< 2\%$ for mean queue length and $< 2.4\%$ for the second moment of queue length, when the number of servers is up to $K = 64$ and $\rho = 0.9$. Given that we have exponential job sizes, this error is solely due to error in the approximation of the conditional arrival rates.

Looking at Figure 5.6, we see

$$\lim_{K \to \infty} \mathbf{E}[N] = \rho.$$

This is expected because, when $\rho < 1$ and the number of servers increases, arrivals find idle servers with probability 1. Thus the system resembles an infinite server system.

General Job Sizes

We now move on to the case of general job-size distributions. Figure 5.7 shows the 95% confidence intervals for the first and second moment of queue length obtained from simulations of the original M/G/K/JSQ/PS server farm for the distributions mentioned in Section 5.3. Each plot also shows the results of the SQA approximation: the analysis of the $M_n/G/1/PS$ system with the conditional arrival rates derived in Section 5.4.1. The results are also summarized in Tables 5.2 and 5.3.

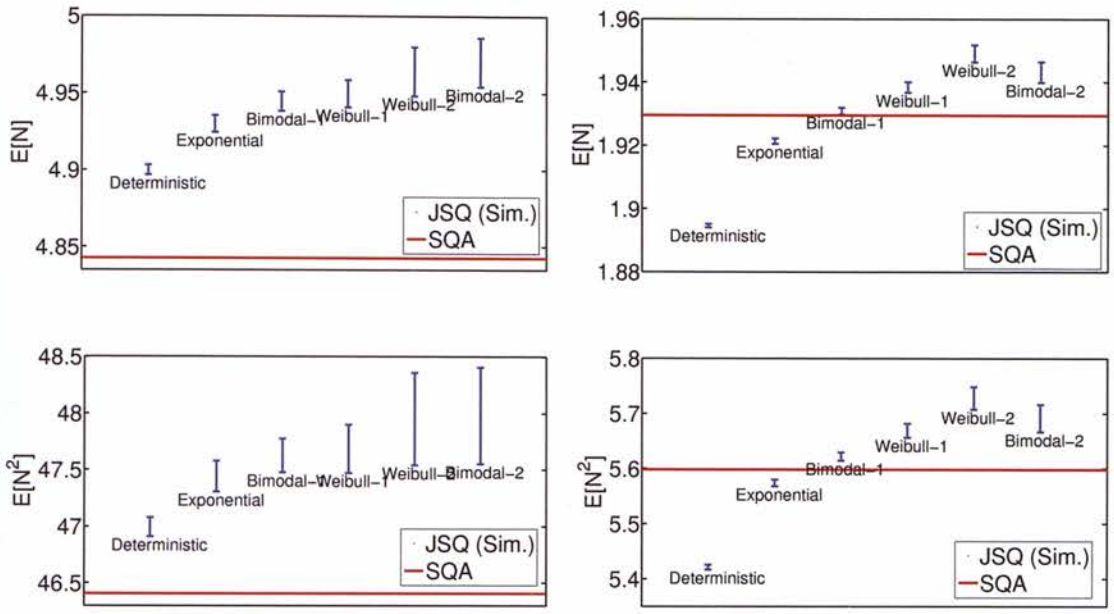
	$K = 2$			$K = 8$		
	$E[N]^{JSQ}$	$E[N]^{SQA}$	% error	$E[N]^{JSQ}$	$E[N]^{SQA}$	% error
Deterministic	4.8999	4.8426	1.1676	1.8946	1.9295	1.8449
Erlang2	4.9216	4.8426	1.6055	1.9142	1.9295	0.8015
Exponential	4.9298	4.8426	1.7678	1.9213	1.9295	0.4260
Bimodal-1	4.9445	4.8426	2.0592	1.9308	1.9295	0.0668
Weibull-1	4.9495	4.8426	2.1589	1.9384	1.9295	0.4573
Weibull-2	4.9640	4.8426	2.4456	1.9490	1.9295	1.0010
Bimodal-2	4.9700	4.8426	2.5618	1.9431	1.9295	0.7004

Table 5.2: Evaluation of SQA: First moment of queue length, obtained via simulation versus SQA, evaluated on distributions mentioned in Section 5.3.

The error is at most 2.6% for mean queue length, and at most 3.3% for the second moment of queue length when $\rho = 0.9$.

5.5 Optimal Load Balancing for PS Servers

So far, we have only considered the commonly used JSQ routing policy. However, it is natural to wonder how good a routing policy JSQ is for PS server farms. In this section we show, via simulation, that it is unlikely that there is a routing policy



(a) $K = 2$ (b) $K = 8$

Figure 5.7: Comparison of the first and second moments of queue length at a single queue in the JSQ/PS server farm with those obtained using SQA for various service distributions with load $\rho = 0.9$ and number of servers $K = 2$ and 8. The top row shows $\mathbf{E}[N]$ and the bottom row shows $\mathbf{E}[N^2]$.

which outperforms JSQ by more than about 10%. We also pose many interesting open problems regarding the optimality of JSQ.

Figure 5.8 compares the performance of JSQ for a PS server farm with that of several other policies, via simulation, on a range of job size distributions, defined in Section 5.3. The policies shown are:

Random – We flip a fair coin in deciding to which queue an incoming job should be assigned. Note that in this case, each queue looks like an $M/G/1/PS$ queue with arrival rate λ/K.

Round-Robin (RR) – Assign jobs in Round-Robin order, where if the previous job was assigned to queue i mod K, then the next job will be assigned to queue $(i + 1)$ mod K.

Least-Work-Left (LWL) – Each job is assigned to the queue with the least total

	$K = 2$			$K = 8$		
	$E[N^2]^{JSQ}$	$E[N^2]^{SQA}$	% error	$E[N^2]^{JSQ}$	$E[N^2]^{SQA}$	% error
Deterministic	46.9934	46.4050	1.2523	5.4210	5.5982	3.2690
Erlang2	47.3844	46.4050	2.0669	5.5354	5.5982	1.1352
Exponential	47.4411	46.4050	2.1840	5.5738	5.5982	0.4375
Bimodal-1	47.6244	46.4050	2.5606	5.6217	5.5982	0.4187
Weibull-1	47.6847	46.4050	2.6837	5.6688	5.5982	1.2464
Weibull-2	47.9491	46.4050	3.2203	5.7277	5.5982	2.2616
Bimodal-2	47.9787	46.4050	3.2801	5.6912	5.5982	1.6343

Table 5.3: Evaluation of SQA: Second moment of queue length, obtained via simulation versus SQA, evaluated for distributions mentioned in Section 5.3.

remaining work.

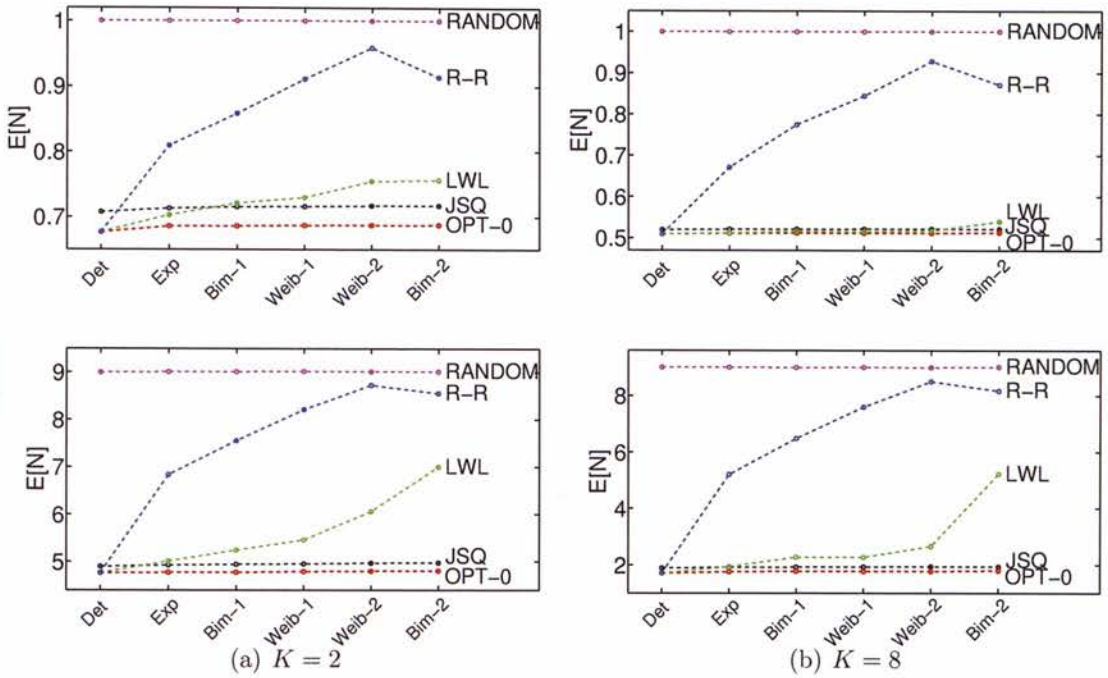
Join-Shortest-Queue (JSQ) – Each job is assigned to the queue with the fewest number of jobs. Ties are broken by flipping a fair coin.

OPT-0 – Each incoming job is assigned so as to minimize the mean response time for all jobs currently in the system, *assuming that there are* 0 *future arrivals.* Note that we are not being greedy from the perspective of the incoming job, but rather trying to minimize across all the jobs in the system. This policy is followed for each successive incoming arrival. The OPT-0 policy was introduced Bonomi [29].

Observe that policies OPT-0 and Least-Work-Left are both less practical than the other policies because they require knowledge of the job sizes.

There are many interesting things to see in Figure 5.8. First, we note that OPT-0 is in fact the best routing policy across all job-size distributions of those policies shown. Also JSQ is very close to OPT-0, within no more than 10%. This is surprising because JSQ utilizes only the *number* of jobs at each queue, whereas OPT-0 uses the remaining sizes of all jobs and the size of the incoming job.

From an insensitivity perspective, we see that that there are some policies, e.g., OPT-0 and JSQ, that are nearly insensitive to the job-size distribution, whereas other policies, e.g., LWL and RR, are highly sensitive to the job-size distribution. It is an interesting question whether there is some detectable common characteristic among those routing policies that are nearly insensitive to the job-size distribution under PS server farms. This is an important question in light of the fact that empirical workloads in Web server farms are very variable.



(a) $K = 2$ (b) $K = 8$

Figure 5.8: Comparison of the first moment of queue length for JSQ, Least Work Left (LWL), Round Robin (R-R) and Random routing policies for $K = 2$ and $K = 8$ servers for a PS server farm with a range of job-size distributions. The service distributions are arranged on the x-axis in order of increasing C_S^2 (the C_S^2 values are $\{0, 1, 2.25, 5, 19, 24.75\}$, respectively).
blah blah

Turning to the question of optimality, note that the case of deterministic job sizes yields the lowest mean response times, as compared with other job-size distributions, and that all three policies: RR, LWL, and OPT-0, yield the *same* performance for the case of deterministic job sizes – in fact, they behave identically on every sample path when the job-size distribution is deterministic, and Theorem 5.7 (Appendix 5.A) shows that they are optimal in a strong sense. Conjecture 5.1 below hypothesizes that this value is the minimum response time possible across all policies and job-size distributions for PS farms.

Conjecture 5.1 *For an $M/G/K/\mathcal{R}/PS$ system, where the job-size distribution has mean 1, we conjecture that setting $G \equiv$ `Deterministic`(1) and $\mathcal{R} \equiv RR$ results in the lowest possible mean response time, over all other pairs $(G, \mathcal{R})$.*

Conjecture 5.1 gives us a handle on evaluating the optimality of JSQ. Making use of the fact that JSQ is one of the policies that is nearly insensitive to the job-size distribution, by the above conjecture, it would suffice to compare the performance of JSQ under deterministic job sizes with RR under deterministic job sizes. Even under the narrowed scope of deterministic job sizes, the comparison between JSQ and RR is not obvious, however, because JSQ can differ from RR both in tie-breaks and non-tie-break situations. If we do not impose the restriction of Poisson arrival process, then the following theorem shows that there exist arrival sequences where JSQ can yield response times up to twice as much as RR for deterministic service distribution, *even without any tie breaks.* We conjecture this factor of two to be tight.

Theorem 5.4 *For the JSQ/PS task assignment policy, with any arbitrary tie-breaking rule,*

$$\sup_{\sigma} \frac{\mathbf{E}\left[T^{\sigma/D/K/JSQ/PS}\right]}{\mathbf{E}\left[T^{\sigma/D/K/RR/PS}\right]} \geq 2$$

The supremum is taken over all finite arrival sequences σ.

Proof: Let the number of servers K be even, and consider the following arrival sequence σ^*: At time $t = 0$, a batch of $\frac{3K}{2}$ jobs arrive. Thus servers 1 to $\frac{K}{2}$ get 2 jobs each and will serve these until time $t = 2$. Servers $\frac{K}{2} + 1$ to K get one job each and idle at time $t = 1$. At time $2 - \epsilon$, a batch of K jobs arrives, and since servers 1 to $\frac{K}{2}$ are still busy with 2 jobs (each of which has remaining size $\frac{\epsilon}{2}$), JSQ will send 2 jobs each to servers $\frac{K}{2} + 1$ to K. However RR (LWL) will send a job each to servers 1 to K. The arrival sequence continues as follows: at times $t = 2i - \epsilon$, $i = 2, 3, \ldots$, batches of K jobs arrive. Under RR (LWL), the subsequent batches of K jobs are distributed evenly among the K servers. Under JSQ, the $\frac{K}{2}$ servers which are idle at the arrival instant of the batch get two jobs each. As the number of arrivals increases, and for ϵ small enough:

$$\frac{\mathbf{E}\left[T^{\sigma^*/D/K/JSQ/PS}\right]}{\mathbf{E}\left[T^{\sigma^*/D/K/RR/PS}\right]} = 2$$

■

5.6 Many-Servers Heavy-Traffic Analysis of Load Balancing Policies

As we have mentioned before, there are very few results on the exact analysis of JSQ/PS, and even the proposed approximations (including the one in Section 5.4) only give results for the distribution of number of jobs in the system. There are no results on the sojourn time distribution. In this section, we propose a novel heavy-traffic scaling to enable study of load balancing policies, and as a first step, we perform the analysis of the stationary joint distribution of queue lengths, and the sojourn time distribution. As the name suggests, our scaling is obtained by letting the number of servers K grow to ∞, while the arrival rate also grows to match the capacity. However unlike the popular square root rule where $K\mu - \lambda = \Theta(\sqrt{K})$, in our scaling the arrival rate grows so as to ensure $K\mu - \lambda = \Theta(1)$. While we call this scaling heavy traffic, under our scaling the marginal queue length of each server converges to a limit, and thus is very useful as a tool to obtain approximations, analyze routing policies for heterogeneous servers, and to study the effect of service distribution. Our scaling is similar to the recent work on Non-Degenerate Slowdown scaling for the central queue $M/M/K$ model by Atar [15].

We employ this scaling to present a new approximation for the $M/M/K$/JSQ/PS model in Section 5.6.1. Our closed-form approximation is accurate when the number of servers is large and the average load per server is close to one, and thus should be seen as a complement to the existing approximations [27, 117]. In addition, we present the first approximation for the *distribution* of response time for the $M/M/K$/JSQ/PS model. In Section 5.6.2 we discuss some further insights gained via our many-server analysis. In Section 5.6.3, we analyze load balancing policies for heterogeneous servers under the many-servers regime and prove that, rather counterintuitively, joining the shorter queue remains optimal.

5.6.1 A new approximation for the $M/M/K$/JSQ/PS model

We begin with a formal definition of the many-server heavy-traffic limit, followed by the analysis of the stationary joint distribution of the number of jobs at the servers in Theorem 5.5, and the stationary distribution of response time in Theorem 5.6.

Definition 5.3 *The many-servers heavy-traffic limiting system with parameter θ ($0 < \theta < 1$) is obtained via a sequence of $M/M/r$/JSQ/PS systems indexed by a discrete parameter r, such that:*

1. *The speed of the servers in the rth system is μ,*

2. *The number of servers in the rth system is r,*

3. *The arrival rate of the rth system, $\lambda^{(r)}$ is chosen to satisfy $\left(\rho^{(r)}\right)^r = \left(\frac{\lambda^{(r)}}{r\mu}\right)^r = \theta$. Equivalently, $\rho^{(r)} = \frac{\lambda^{(r)}}{r\mu} = 1 + \frac{\log\theta}{r}$.*

Stationary joint distribution of number of jobs

Theorem 5.5 *Let $N^{(r)}$ denote the number of job in the rth system. Then:*

$$\lim_{r\to\infty} \mathbf{Pr}\left[N^{(r)} > \alpha r\right] = \begin{cases} 1 & \alpha \le 1, \\ \frac{1}{C}\cdot\frac{1}{|\log\theta|} + \frac{1}{C}\frac{\left[(\log\theta-1)(1-(\alpha-1)\left(\frac{\theta}{e}\right)^{\alpha-2})-(1-\left(\frac{\theta}{e}\right)^{\alpha-2})\right]}{(\log\theta-1)^2} & 1 < \alpha \le 2, \\ \frac{1}{C}\cdot\frac{\theta^{\alpha-2}}{|\log\theta|} & 2 < \alpha. \end{cases}$$

where C, the normalizing constant, is given by

$$\begin{aligned} C &= \frac{e}{\theta}\int_0^1 ue^{u(\log\theta-1)}du + \int_0^\infty e^{u\log\theta}du \\ &= \frac{1}{(\log\theta-1)} - \frac{1}{(\log\theta-1)^2} + \frac{e}{\theta(\log\theta-1)^2} - \frac{1}{\log\theta} \end{aligned}$$

Further, the system exhibits the following state-space collapse: Conditioning on $N^{(r)} = \alpha r$ $(i < \alpha < i+1,\ i \in \mathbb{N})$, the number of servers with $(i+1)$ jobs is $(\alpha - i)r + \Theta(1)$, the number of servers with i jobs is $((i+1)-\alpha)r + \Theta(1)$, and the distribution of the number of servers with $(i-1)$ jobs is Geom $(\alpha - i) - 1$, where Geom(p) is the geometric distribution with success probability p (equivalently, as the number of jobs in an $M/M/1$ with load $(i+1-\alpha)$). The number of servers with less than $(i-1)$ or more than $(i+1)$ jobs is $o(1)$.

Proof: We begin by showing how the first part of the theorem follows from the second. The proof involves an idea similar to SQA. We look at the whole system as a single queue and consider the Markov chain for the total number of jobs in the system. The problem of finding conditional arrival rates now becomes trivial – they are simply $\lambda^{(r)}$. The interesting question is finding the conditional departure rates, and since our servers are homogeneous and job size distribution is Exponential, this would be given by $M^{(r)}(j) = \left(r - I^{(r)}(j)\right)\mu$, where $I^{(r)}(j)$ is the expected number of idle queues in the rth system conditioning on the total number of jobs being j.

Consider $\alpha < 1$: That is, the number of jobs is less than the number of servers. In this case, the system is unstable on a transient scale and hence there is no probability mass for $\alpha < 1$.

Now consider $\alpha > 2$: then according to the second statement in the theorem, the number of idle queues is $o(1)$. Thus the departure rate is $r\mu - o(1)$, while under our scaling, the arrival rate is $r\mu - \Theta(1)$. Therefore, for $\alpha > \beta \geq 2$, for the rth system:

$$\frac{\mathbf{Pr}\left[N^{(r)} = [\alpha r]\right]}{\mathbf{Pr}\left[N^{(r)} = [\beta r]\right]} = \Pi_{j=[\beta r]+1}^{[\alpha r]} \left(\frac{\lambda^{(r)}}{\mu\left(r - I^{(r)}(j)\right)}\right)$$
$$= \Pi_{j=[\beta r]+1}^{[\alpha r]} \left(\frac{r + \log\theta}{r - I^{(r)}(j)}\right)$$

In the above, we use $[x]$ to denote the integer part of x. Taking logarithm of both sides:

$$\log \mathbf{Pr}\left[N^{(r)} = [\alpha r]\right] - \log \mathbf{Pr}\left[N^{(r)} = [\beta r]\right] = \sum_{j=[\beta r]+1}^{[\alpha r]} \left(1 + \frac{\log\theta}{r} + o(1)\right)$$
$$= (\alpha - \beta)\log\theta + o(1)$$

Therefore, for $\alpha > \beta \geq 2$,

$$\frac{\mathbf{Pr}\left[N^{(r)} = [\alpha r]\right]}{\mathbf{Pr}\left[N^{(r)} = [\beta r]\right]} \sim \theta^{\alpha-\beta}$$

Finally, consider $1 < \alpha < 2$: This is the regime where difference between JSQ and central queue $M/M/K$ really emerges. While there would be no idleness in $M/M/K$, under JSQ, the expected number of idle queues $I^{(r)}([\alpha r]) = \frac{2-\alpha}{\alpha-1}$. Therefore, for $1 < \alpha < \beta \leq 2$:

$$\log \mathbf{Pr}\left[N^{(r)} = [\beta r]\right] - \log \mathbf{Pr}\left[N^{(r)} = [\alpha r]\right]$$
$$= \sum_{j=[\alpha r]+1}^{[\beta r]} \left(\log\frac{\lambda^{(r)}}{\mu} - \log\left(r - I^{(r)}(j) + o(1)\right)\right)$$
$$= (\beta - \alpha)r\log r(1 + \frac{\log\theta}{r}) - r\int_{\alpha}^{\beta} \log r\left(1 - \frac{1}{r}\cdot\frac{2-u}{u-1}\right)du + o(1)$$
$$= (\beta - \alpha)r\log r + (\beta - \alpha)\log\theta - (\beta - \alpha)r\log r + \int_{\alpha}^{\beta}\frac{2-u}{u-1}du + o(1)$$
$$= (\beta - \alpha)r\log r + (\beta - \alpha)\log\theta - (\beta - \alpha)r\log r - \int_{\alpha}^{\beta} du + \int_{\alpha}^{\beta}\frac{du}{u-1} + o(1)$$
$$= (\beta - \alpha)(\log\theta - 1) + \log\frac{\beta - 1}{\alpha - 1} + o(1)$$

or,

$$\frac{\mathbf{Pr}\left[N^{(r)} = [\alpha r]\right]}{\mathbf{Pr}\left[N^{(r)} = [\beta r]\right]} \sim \frac{\beta - 1}{\alpha - 1}\left(\frac{\theta}{e}\right)^{\beta - \alpha} \tag{5.19}$$

By integrating, the expression for the distribution of $\lim_{r\to\infty} \frac{N^{(r)}}{r}$ in the Theorem statement follows.

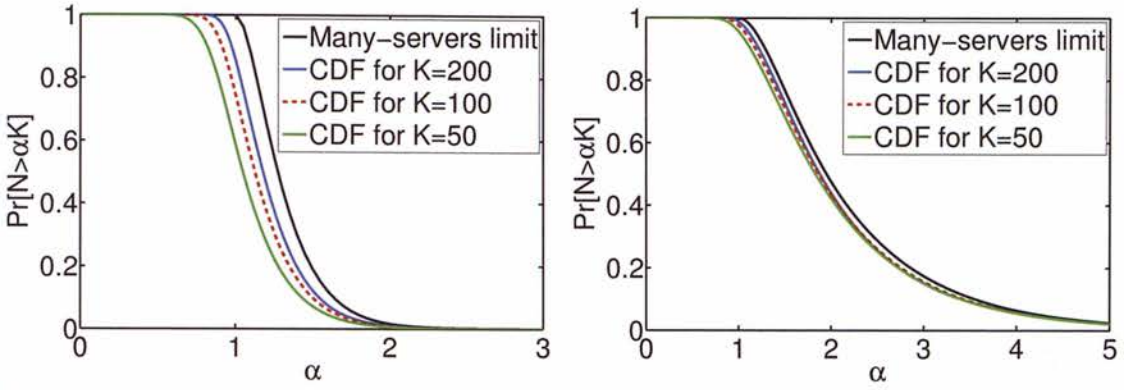
We do not provide a rigorous proof of the second part of the theorem as it can be verified by setting up the balance equations and plugging in the stationary distribution from the theorem statement. Instead, we provide an intuitive argument for why the expression for the distribution is so. Suppose there are more than αr jobs in the system with $2 \leq i < \alpha < i + 1$. Since JSQ tries to equalize queue lengths, we expect servers to have only have either i or $i + 1$ jobs. However, servers with i jobs can have departures creating servers with $i - 1$ jobs. Since $i \geq 2$, these servers are not idle, and thus do not cause loss in departure rate. Further, these servers with $i - 1$ jobs are being created at an average rate of approximately $((i + 1) - \alpha)\mu r$, and destroyed (that is, getting an ith job) at rate of $\mu r + \Theta(1)$. Thus the number of these servers 'behaves' similar to the number of jobs in an $M/M/1$ with arrival rate $((i + 1) - \alpha)\mu r$ and service rate μr, and we expect the number of these servers to be $\Theta(1)$. Since conditional departure rate is μr, and the arrival rate is $\mu r - \Theta(1)$, it would take $\Theta(r)$ time for the number of jobs to change by $\Theta(r)$ (If we view the system as a random walk where the up/down probabilities are $\frac{1}{2} \pm \Theta\left(r^{-1}\right)$; it takes $\Theta(r^2)$ steps for a displacement of $\Theta(r)$). However, if we look at the number of servers with $i - 1$ jobs, it achieves stationarity in $\varnothing(1)$ time (the $M/M/1$ queue mimicing the evolution of these servers has mean busy period of $\Theta(\frac{1}{r})$) during which period the rate of their creation does not change.

However, if the number of jobs is αr with $1 = i < \alpha < i + 1 = 2$, then the the servers with $i - 1$ jobs are, in fact, idle causing a drop in the departure rate. However, this drop is $\Theta(1)$, and we still have that the total number of jobs changes by $\Theta(r)$ at a time scale of $\Theta(1)$, while the number of idle servers achieves the stationary distribution at a time scale of $\Theta\left(\frac{1}{r}\right)$. ∎

Corollary 5.1 *Under the many-servers heavy-traffic limit with parameter θ, the mean number of jobs per server is given by:*

$$\mathbf{E}\left[\lim_{r\to\infty} \frac{N^{(r)}}{r}\right] = 1 + \frac{1}{C(\log\theta)^2} + \frac{1}{C|\log\theta|} + \frac{1}{C(\log\theta - 1)} - 2\frac{1}{C(\log\theta - 1)^2} + 2\frac{1 - \left(\frac{\theta}{e}\right)^{-1}}{C(\log\theta - 1)^3} \tag{5.20}$$

where C is the normalization constant from Theorem 5.5.



(a) $\theta = 0.0059$ (b) $\theta = 0.3660$

Figure 5.9: Simulation results showing convergence of the stationary distribution of the number of jobs in the system to the many-servers heavy-traffic limit for two values of θ. $\theta = 0.0059$ corresponds to $\rho = 0.95$ for $K = 100$, and $\theta = 0.3660$ corresponds to $\rho = 0.99$ for $K = 100$.

Figure 5.9 shows the speed of convergence of the stationary distribution of the number of jobs in the system to the many-servers limit as the number of servers is increased while holding $\rho^K = \theta$ fixed. The approximation is tight under very heavy traffic, but we believe that the ideas in the proof of Theorem 5.5 can be combined with existing approximations which perform well in light-traffic to obtain sharper approximations for all parameter settings.

Stationary distribution of response time

Theorem 5.6 *Let $T^{(r)}(\alpha)$ denote the response time of an arrival that sees $[\alpha \cdot r]$ jobs in the rth system on arrival with $1 \leq i < \alpha < i+1$. Let $T(\alpha) = \lim_{r\to\infty} T^{(r)}(\alpha)$. Then, $T(\alpha)$ has a phase-type (PH) distribution with parameters $(\boldsymbol{\beta}, \mathbf{B})$*

$$\boldsymbol{\beta} = \begin{pmatrix} i+1-\alpha \\ \alpha - i \end{pmatrix}; \qquad \mathbf{B} = \begin{pmatrix} -\left(\frac{\mu}{i} + \gamma_\alpha\right) & \left(\frac{\mu}{i} + \gamma_\alpha\right)\frac{i\gamma_\alpha}{\mu + i\gamma_\alpha} \\ \mu\frac{i}{i+1} & -\mu \end{pmatrix} \tag{5.21}$$

where $\gamma_\alpha = \frac{\alpha - i}{i+1-\alpha}$.
In other words, $T(\alpha)$ is given by the time until absorption into state 0 in a Markov

chain with states $\{0, i, i+1\}$, *initial state vector* $\mathbf{q}$ *and generator* $\mathbf{Q}$, *where:*

$$\mathbf{q} = \begin{pmatrix} 0 \\ i+1-\alpha \\ \alpha - i \end{pmatrix}; \qquad \mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 \\ \left(\frac{\mu}{i}+\gamma_\alpha\right)\frac{\mu}{\mu+i\gamma_\alpha} & -\left(\frac{\mu}{i}+\gamma_\alpha\right) & \left(\frac{\mu}{i}+\gamma_\alpha\right)\frac{i\gamma_\alpha}{\mu+i\gamma_\alpha} \\ \mu\frac{1}{i+1} & \mu\frac{i}{i+1} & -\mu \end{pmatrix}. \tag{5.22}$$

Proof: According to Theorem 5.5, the arrival will either join a server with queue length $i-1$ or i with overwhelmig probability. Let $T_i(\alpha)$ be the response time given the arrival joins a server with queue size $(i-1)$ (and thus begins its sojourn in a server with i jobs in total), and let $T_{i+1}(\alpha)$ be the response time given the arrival joins a server with queue size i (and thus begins its sojourn in a server with $i+1$ jobs in total). The distribution of $T_i(\alpha)$ is the time until absorption in state 0 starting from state i in the Markov chain with generator $\mathbf{Q}$ above, and similarly, $T_{i+1}(\alpha)$ is the time until absorption in state 0 starting from state $i+1$ in the Markov chain with generator $\mathbf{Q}$.

According to Theorem 5.5, the arrival joins a server with queue length $i-1$ with probability $(i+1-\alpha)$, and a server with queue length i with probability $(\alpha - i)$, and these correspond to the initial state vector $\mathbf{q}$ (or, $\boldsymbol{\beta}$) in the theorem statement.

Case 1: arrival starts sojourn in a queue with $i+1$ jobs in total : Note that it takes $\Theta(r)$ time for the number jobs in the system to change by $\Theta(r)$, while the sojourn time of the arrival is $\Theta(1)$, and thus we can assume that the state of the system is tightly concentrated around the stationary distribution conditioned on $\alpha \cdot r + o(r)$ jobs. Given this knowledge, the only event that can affect the tagged arrival is a departure from its queue, as no arrivals will happen into a queue with $i+1$ jobs. Thus, with rate μ, the server with the tagged job loses a job to become a server with i jobs, and with probability $\frac{i}{i+1}$, the departing job is not the tagged job. The ensuing sojourn time of the tagged job in this case is equal in distribution to $T_i(\alpha)$. This explains the last rwo of $\mathbf{Q}$.

Case 2: arrival starts sojourn in a queue with i jobs in total : The events that can affect the server with the tagged job in this case are a departure as well as an arrival. Note that as we mentioned before, the number of queues of size $(i-1)$ behaves as the number of jobs in an $M/M/1$ with departure rate r and load $(i+1-\alpha)$, and hence busy period of $\Theta(\frac{1}{r})$. On the other hand, the events at the tagged server happen at rate $\Theta(1)$, and hence we can assume that the probability that an external arrival finds no queue of size $i-1$ and hence is assigned to a queue of size i is $(\alpha - i)$. External arrivals happen at a rate of r, and conditioning on the arrival joining a queue of size i, it joins the tagged queue with probability $\frac{1}{(i+1-\alpha)r} + o\left(\frac{1}{r}\right)$. Thus, the arrival process into the tagged queue is (asymptotically, as $r \to \infty$) a Poisson

process with rate $r(\alpha - i) \cdot \frac{1}{(i+1-\alpha)r} = \frac{\alpha - i}{i+1-\alpha} = \gamma_\alpha$. We are now ready to explain the middle row of $\mathbf{Q}$. An arrival or a departure happens at aggregate rate of $\mu + \gamma_\alpha$. However, a departure that is not the tagged departure causes the queue length to drop to $i-1$, where it only spends $\theta\left(\frac{1}{r}\right)$ time before being assigned a job and pushed back to queue length of i. Thus, we only need to worry about the departure of the tagged job, which happens at rate of $\frac{\mu}{i}$, or an arrival from outside, which happens at rate γ_α. The arrival causes the Markov chain to transition to state $(i+1)$, from where the ensuing sojourn time of the tagged job is equal in distribution to $T_{i+1}(\alpha)$.
■

5.6.2 Further Consequences

A couple of interesting observations immediately follow from Theorem 5.5.

Accuracy of existing approximations: The popularity of JSQ stems from the fact that it is believed to be a good approximation to the $M/M/K$ system. Switching the statement around, $M/M/K$ is a good approximation for the performance of JSQ. How far off can the performance of $M/M/K$ be from JSQ? Under the many-servers heavy-traffic limit, the probability distribution for the total number of jobs in an $M/M/K$ is

$$\lim_{r\to\infty} \mathbf{Pr}\left[N^{(r)}_{M/M/r} \geq \alpha r\right] = \begin{cases} 1 & \alpha \leq 1 \\ \theta^{\alpha-1} & \alpha > 1 \end{cases}$$

which gives an approximation for the mean number of jobs in the system as:

$$\mathbf{E}\left[\lim_{r\to\infty} \frac{N^{(r)}_{M/M/r}}{r}\right] = 1 + \frac{1}{|\log\theta|} \tag{5.23}$$

Numerically comparing (5.20) against (5.23) in the many-servers limit, the performance of JSQ is at most 14% off ($\theta = 0.124$). We have found that the figure of 14% is approximately the maximum performance gap between central queue $M/M/K$ and JSQ even under non-asymptotic regime.

We now turn to the question: How well do popular approximations in the literature for JSQ compare in the many-servers heavy-traffic limit? We choose the Nelson Philips [117] approximation, according to which the mean number of jobs under JSQ is approximated by:

$$\mathbf{E}\left[N_{JSQ(NP)}\right] \approx \frac{\lambda}{\mu}\left(1 + \left\lfloor \frac{N_{M/M/K}}{K} \right\rfloor\right)$$

which under the many-servers heavy-traffic scaling becomes

$$\mathbf{E}\left[\lim_{r\to\infty}\frac{N^{(r)}_{JSQ(NP)}}{r}\right] = 1 + \frac{1}{1-\theta} \tag{5.24}$$

By letting $\theta \to 0$, the Nelson Philips approximation can be made arbitrarily close to a factor of 2 times the JSQ limit. Thus, in the worst case, Nelson Philips approximation for JSQ can be off by 100%. This error stems from the approximation assumption employed in [117] that an arrival seeing n jobs in the system on arrival will join a server with $\left\lfloor \frac{n}{K} \right\rfloor$ jobs. However, as our analysis reveals, if $K \leq n \leq 2K$, then it is very likely that at least one server is idle.

Analytic justification for (5.2)**:** The derivation of conditional arrival rates in Section 5.4 relied on a crucial approximation assumption based on empirical evidence: $\frac{\lambda(n)}{\mu} \approx \rho^K$ for $n \geq 3$. We now formally justify the assumption, using Theorem 5.5. Let us fix a designated queue Q. For $i \geq 3$, the probability that Q has i jobs is given by:

$$\pi(i) = \sum_j \mathbf{Pr}[Q = i | N = j] \cdot \mathbf{Pr}[N = j]$$

which in the many servers limit is given by,

$$\approx \int_{\alpha=i-1}^{\alpha=i+1} \theta^{\alpha-2} \left[\mathbf{1}_{\alpha<i}\{\alpha - (i-1)\} + \mathbf{1}_{\alpha>i}\{i+1-\alpha\}\right] d\alpha$$

Therefore, the conditional arrival rate $\lambda(i)$ for $i \geq 3$ in the many server limit is:

$$\begin{aligned}
&\frac{\lambda(i)}{\mu} \\
&= \frac{\begin{array}{c}\int_{\alpha=i-1}^{i} \theta^{\alpha-2}\{\alpha-(i-1)\}\cdot 0 d\alpha + \int_{\alpha=i}^{i+1} \theta^{\alpha-2}(i+1-\alpha)\frac{\lambda}{K(i+1-\alpha)}\cdot(1-(i+1-\alpha))d\alpha \\ + \int_{\alpha=i+1}^{i+2} \theta^{\alpha-2}(i+2-\alpha)\frac{\lambda}{K} d\alpha\end{array}}{\int_{\alpha=i-1}^{\alpha=i+1} \theta^{\alpha-2}\left[\mathbf{1}_{\alpha<i}\{\alpha-(i-1)\} + \mathbf{1}_{\alpha>i}\{i+1-\alpha\}\right] d\alpha} \\
&= \frac{\lambda}{K} \frac{\int_{\alpha=i}^{i+1} \theta^{\alpha-2}(\alpha-i)d\alpha + \int_{\alpha=i+1}^{i+2} \theta^{\alpha-2}(i+2-\alpha)}{\int_{\alpha=i-1}^{\alpha=i} \theta^{\alpha-2}(\alpha-(i-1))\,d\alpha + \int_{\alpha=i}^{i+1} \theta^{\alpha-2}(i+1-\alpha)\,d\alpha} \\
&= \frac{\lambda}{K}\theta \\
&\approx \theta = \rho^K
\end{aligned}$$

5.6.3 Optimal load balancing for heterogeneous servers

The optimality of JSQ for homogeneous servers for Exponential service distribution (and when job sizes can not be observed) was established long ago [29]. However, if the servers speeds are heterogeneous, finding the optimal policy quickly becomes intractable due to the explosion of state space. Heuristic policies have been proposed (see, e.g., [136]) motivated by policies for central queue $M/M/K$ models with heterogeneous servers. What makes the problem non-trivial is that there is no single load-balancing policy that is optimal across all arrival intensities, and thus thus the optimal policy must learn the arrival rate. One popular traffic-oblivious heuristic is to send a new arrival to the server where the arrival gets served at the largest rate. That is, if the ith server has speed μ_i and n_i jobs in the buffer, then the job is sent to the server with the largest value of $\frac{\mu_i}{n_i+1}$. Alternately, this policy can be seen as sending the job to the server where its expected response time is minimized *if the servers were FCFS*, and hence we will refer to it as the Minimum Expected Response time (MER) policy.

In this section we prove that, rather counterintuitively, the MER policy is suboptimal in the many-servers regime. Instead, sending the job to the server with fewer number of jobs (irrespective of the speeds) and only using the server speeds for tie breaking is optimal while being traffic-oblivious.

We will compare the following load balancing heuristics:

1. **Minimum-Expected-Response time (MER):** The job is sent to server with the smallest value of $\frac{n+1}{\mu}$. Equivalently, where the job's expected response time is minimized under FCFS scheduling.

2. **Join-Shortest-Queue with smart tie-breaking (JSQ)**: The job is sent to the server with the fewest number of jobs, if this server is unique. Otherwise ties are broken in favor of faster servers.

3. **HYBRID**: A combination of MER and JSQ – JSQ is followed if some server is idle, otherwise if all servers are busy then the job is sent to the server with smallest value of $\frac{n+1}{\mu}$.

We first generalize the many-servers limit to heterogeneous servers, and for ease of exposition, consider the case where servers can have two possible speeds μ_1 or μ_2.

Definition 5.4 *In the many-servers limit with parameters $\mu_1, \mu_2, \beta_1, \beta_2$ with $\beta_1 + \beta_2 = 1$, the total number of servers K grows to ∞, while the number of servers of speed μ_1 grows as $K_1 = \beta_1 K$, and that of speed μ_2 grows as $K_2 = \beta_2 K$. Without loss of generality we will assume $\mu_1 > \mu_2$.*

The many-servers regime defined above is very natural in large data centers where servers are bought in volume once or twice a year, and are phased out over a period of 3-5 years, leading to heterogeneous equipment.

In the next section, we will compare the load balancing schemes in light traffic regime, and prove that while MER is suboptimal, both JSQ and HYBRID are optimal in many-servers light-traffic. To resolve between JSQ and HYBRID, we will look at many-servers heavy-traffic. Finally we will present simulation results comparing the policies.

Comparison of Policies in Many-servers Light-Traffic

As mentioned previously, in the many-servers limit, the system capacity grows to infinity, and thus the arrival rate λ must also grow to infinity for a non-degenerate limit. In the light traffic regime, the arrival rate grows so that $\frac{\lambda}{K} = \gamma$ (a constant), where $\gamma < (\beta_1\mu_1 + \beta_2\mu_2)$. Therefore the offered load is a constant fraction of the capacity. We now proceed to analyze our load balancing policies in the light-traffic limit.

Analysis of MER: The analysis splits in two cases:
Case 1: $(\gamma < \beta_1\mu_1)$ In this case, the capacity of the the fast servers is sufficient to handle the offered load. Thus, arrivals find idle fast servers with overwhelming probability and hence the expected response time converges to $\frac{1}{\mu_1}$.
Case 2: $(\gamma > \beta_1\mu_1)$ In this case, the slower servers must be used to handle load. However, the MER policy does not route any jobs to a slower server until all fast servers have at least $\left\lceil \frac{\mu_1}{\mu_2} \right\rceil - 1$ jobs (depending on tie breaking rule when $\frac{\mu_1}{\mu_2}$ is an integer). Thus, under stationarity, all fast servers have at least $\left\lceil \frac{\mu_1}{\mu_2} \right\rceil - 1$ jobs, while $\frac{(\gamma-\beta_1\mu_1)K}{\mu_2} + o(K)$ slow servers have 1 job. Moreover, all jobs find at least some server idle with overwhelming probability.

Analysis of JSQ: The analysis again splits in two cases:
Case 1: $(\gamma < \beta_1\mu_1)$ This case is identical to MER since ties are broken in favor of faster servers, and hence the expected response time converges to $\frac{1}{\mu_1}$.
Case 2: $(\gamma > \beta_1\mu_1)$ In this case again, the slower servers must be used to handle load. Unlike MER, JSQ starts routing jobs to slower servers as soon as all fast servers have 1 job. Moreover, all jobs find at least some server idle with overwhelming probability. Thus, JSQ gives strictly smaller mean response time than MER. Moreover, as can be seen, JSQ minimizes the mean response time in light traffic. This is because the number of jobs in the system is lower bounded by the number of servers needed for stability ($\beta_1 K_1 + \frac{(\gamma-\beta_1\mu_1)K}{\mu_2}$), and JSQ keeps these many servers occupied with

exactly one job.

Analysis of HYBRID: The HYBRID policy mimics JSQ as long as there are idle servers, which in the light-traffic regime happens with overwhelming probability. Thus JSQ and HYBRID yield identical mean response time.

We thus already see that MER, while greedy, is not a smart policy in the many-servers regime. Instead, JSQ and HYBRID yield better performance. Intuitively, by trying to keep all servers busy, JSQ and HYBRID use the full system capacity. To compare JSQ and HYBRID, we next look at heavy-traffic regime.

Comparison of Policies in Many-servers Heavy-Traffic

Similar to Definition 5.3, the many-servers heavy-traffic limit is achieved by scaling the arrival rate so that $(\beta_1\mu_1 + \beta_2\mu_2)K - \lambda = \Theta(1)$. We will only compare JSQ and HYBRID under this regime. As illustrated in the proof of Theorem 5.5, to compare JSQ and HYBRID, we have to compare the inefficiencies induced by idle servers. The greater the number of idle servers, and the greater their speed, the worse the load balancing policy is. We will therefore look at the distribution of idle queues as a function of N, the number of jobs in the system.

Analysis of JSQ: The analysis splits into three cases:
Case $N > 2K$**:** There are no idle queues in this case, because almost all servers have at least 2 jobs, and thus even after having a departure and before getting an arrival, these servers are busy.
Case $(1+\beta_1)K < N < 2K$**:** In this case almost all fast servers have 2 jobs, and even after departures, do not become idle. However, a constant fraction of slow servers have one job, and they will give rise to $\Theta(1)$ idle queues.
Case $K < N < (1+\beta_1)K$**:** Both fast and slow servers will give rise to idle queues. The total number of idle queues is distributed according to an $M/M/1$ with arrival rate $((1+\beta_1)K - N)\mu_1 + (\beta_2 K)\mu_2$ (the total service rate of servers with 1 job), and departure rate $\lambda = \gamma K$. However, since JSQ gives priority to fast idle servers over slow idle servers, the idle servers evolve according to a 2-class preemptive priority queueing system – the number of fast idle servers is distributed according to an $M/M/1$ with arrival rate $((1+\beta_1)K - N)\mu_1$ and departure rate $\lambda = \gamma K$, and the remaining idle servers are slow.

Analysis of HYBRID: To compare the HYBRID policy with JSQ, it will now suffice to look at whether under HYBRID, there are more or less queues of size 1, as these are instrumental in giving birth to idle queues and hence loss in efficiency. For N up to $(1+\beta_1)K$, HYBRID again behaves exactly like JSQ because fast servers will get the second job before slow servers get the second job. However, for $N > (1+\beta)K$,

under HYBRID, the fast servers will get a third job before slow servers get a second job. Therefore, the number of idle slow server under HYBRID will be larger than the number of idle slow servers under JSQ. Only after $N > \left(\left\lfloor 2\frac{\mu_1}{\mu_2} \right\rfloor \beta_1 + 2\beta_2\right) K$ will there be no idleness in the system. Therefore, for every value of $N > K$, the conditional departure rate under JSQ will be at least the conditional departure rate under HYBRID, and hence JSQ stochastically minimizes the number of jobs in the system in the many-servers limit. However, note that until $N = (1+\beta_1)K$, HYBRID mimics JSQ, which is optimal. Hence, unless traffic is very high, HYBRID and JSQ will have similar performance.

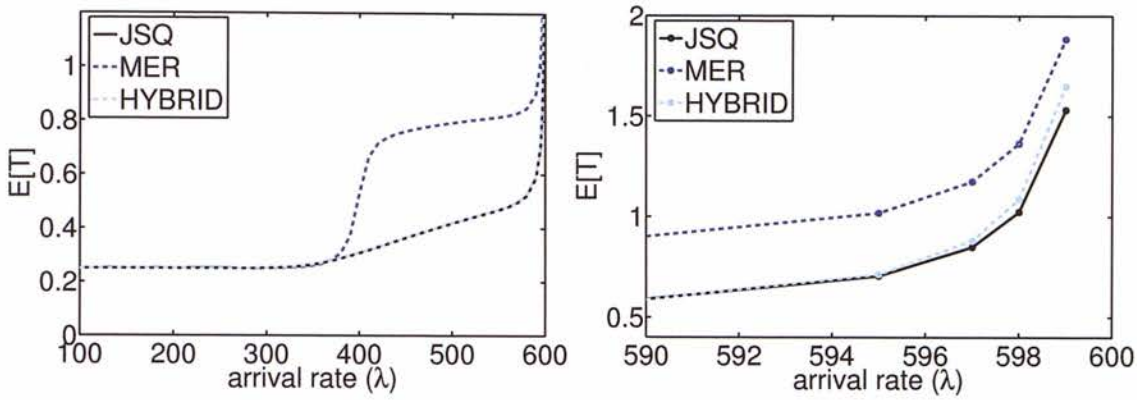
In the next section we will compare MER, JSQ, and HYBRID in the many-servers regime, and also compare them to an elegant traffic-aware policy Greedy-Throughput [118] when not in the many-servers regime.

Simulation Results

Figure 5.10 presents the simulation results for the many-server regime. We simulate a queueing system with $K_1 = 100$ servers of speed $\mu_1 = 4$, and $K_2 = 400$ servers of speed $\mu_2 = 1$. The total capacity is 600, and the capacity of the fast servers alone is 400. We vary the arrival rate and plot the mean response time as a function of the arrival rate. In Figure 5.6.3, we present the simulation results for the light-traffic case, that is where the system is far from critical load. We see that simulations verify our analysis. While the arrival rate is less than the capacity of fast servers, arrivals find an idle fast server with high probability and hence the mean response time is $\frac{1}{\mu_1} = 0.25$. As λ increases beyond 400, under MER, all the fast servers are occupied by 3 jobs before slow servers are utilized. Thus the mean response time jumps to 0.75, and then increases linearly as more and more slow servers are used. However, under JSQ and HYBRID, the slow servers are utilized immediately, and the response time increases linearly without the jump. We further see that JSQ and HYBRID yield identical mean response time in light traffic.

Figure 5.6.3, we show the same results, but when arrival rate is close to the capacity of the server farm. We now see that JSQ starts outperforming HYBRID, as analysis predicts. However, this behavior emerges at a very high arrival rate.

In Figure 5.11, we compare the policies analyzed in this section when the number of servers is not very large. All the policies we have analyzed are very simple traffic-oblivious policies in that they do not require knowing the arrival rate λ (and indeed are optimal in the many-servers across all values of λ). However, when the numbers of servers is not large, the optimal load balancing critically depends on the arrival rate. We will compare the proposed traffic-oblivious policies against an



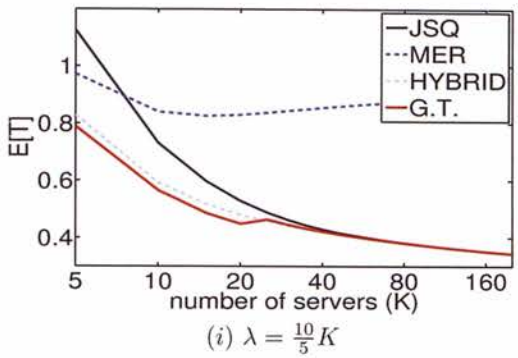
(a) Light Traffic

(b) Heavy Traffic

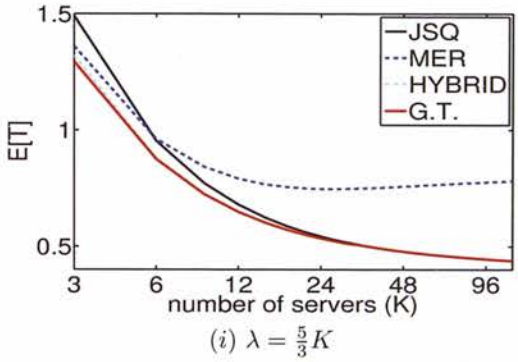
Figure 5.10: Simulation results comparing mean response time under JSQ, MER, and HYBRID load balancing policies in the many servers regime. The speeds of fast and slow servers were 4 and 1, respectively, and their numbers were 100 and 200 for a total service rate of 600.

elegant *traffic-aware* policy proposed in the literature – Greedy Throughput [118]. The Greedy Throughput heuristic sends an arriving job to the queue so as to maximize the number of departures before the next arrival. This amounts to maximizing $\left(\frac{\mu_i}{\mu_i+\lambda}\right)^{n_i+1}$. We note that this policy takes the right decision at extrema of arrival rate. When $\lambda \to 0$, Greedy Throughput routes the job to the queue maximizing $1 - \frac{\lambda(n_i+1)}{\mu_i}$, and thus is the same as MER. When $\lambda \to \infty$, Greedy Throughput routes the job to the queue maximizing $\left(\frac{\mu_i}{\lambda}\right)^{n_i+1}$ – and thus aims to equalize queue lengths, and among queues of equal size, routes to the faster server – identical to JSQ. Shenker and Weinrib [136] have performed extensive comparisons of central queue variants of JSQ (called Never Queue (NQ) in [136]), MER (called to Shortest Expected Delay (SED) in [136]), and Greedy Throughput policies.

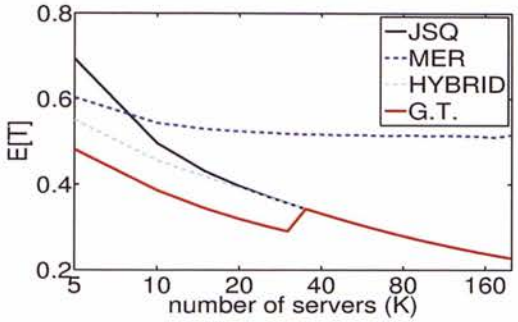
The experiments in Figure 5.11 are split in two groups: the figures on the left are for a server farm where the heterogeneity in server speeds is high ($\frac{\mu_1}{\mu_2} = 8$), and on the right are when the heterogeneity is lower ($\frac{\mu_1}{\mu_2} = 4$). For each server farm scenario, we fix different values of 'average load', i.e., the ratio of the arrival rate to the server farm capacity, and plot the mean response time as a function of the number of servers. We have chosen three values of 'average load' for each scenario: the arrival rate is larger than the capacity of the fast servers in the top figure, in the middle figure they are equal, and the arrival rate is smaller than the capacity of the



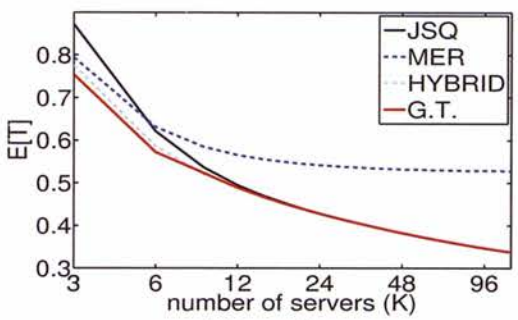
(*i*) $\lambda = \frac{10}{5}K$



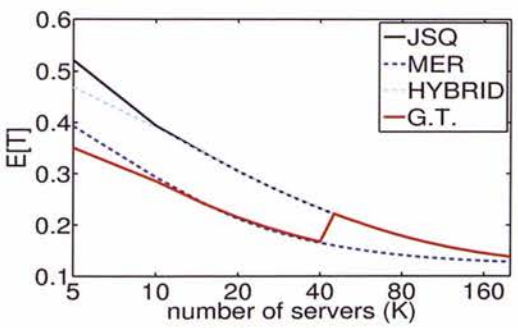
(*i*) $\lambda = \frac{5}{3}K$



(*ii*) $\lambda = \frac{8}{5}K$

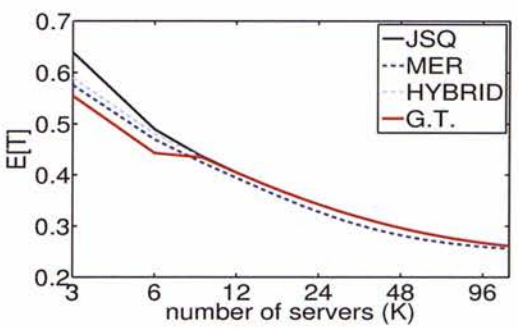


(*ii*) $\lambda = \frac{4}{3}K$



(*iii*) $\lambda = \frac{6}{5}K$

(a) $\mu_1 = 8, \mu_2 = 1;\ \beta_1 = \frac{1}{5}, \beta_2 = \frac{4}{5}$



(*iii*) $\lambda = K$

(b) $\mu_1 = 4, \mu_2 = 1;\ \beta_1 = \frac{1}{3}, \beta_2 = \frac{2}{3}$

Figure 5.11: Simulation results comparing mean response time under JSQ, MER, HYBRID, and Greedy-Throughput policies in the non-many-servers regime. The x-axis shows the total number of servers and the y-axis shows the mean response time. The figures on the left represent high server speed heterogeneity (8 : 1), and the figures on the right represent low heterogeneity (4 : 1). In each figure, the ratio of the arrival rate to server farm capacity is constant, and this ratio decreases from top to bottom as indicated.

fast servers in the bottom figure. The following observations are immediate:

- Among the traffic-oblivious policies, HYBRID provides the best compromise. It consistently outperforms JSQ in the non-many-servers regime, and this performance gap increases when the degree of heterogeneity is higher. HYBRID is only outperformed by MER when the server heterogeneity is very high and arrival rate is smaller than the capacity of fast servers.

- When the arrival rate is smaller than the capacity of fast servers, MER performs the best among traffic oblivious policies, and even outperforms traffic aware Greedy Throughput. However, when the arrival rate is close or above the capacity of fast servers, MER becomes suboptimal even at relatively small number of servers.

5.7 Summary and Open Questions

We present the first analysis of JSQ load balancing for server farms with Processor Sharing servers, which are more representative of computing applications than First-Come-First-Served servers. We have introduced several new ideas which we believe will be applicable in much more general settings. The first is the idea of Single Queue Approximation (SQA), whereby one designated queue in the farm can be analyzed *in isolation* of all the other queues, where a state-dependent arrival rate is used to, in some sense, capture the effect of the other queues. Understanding what these state-dependent arrival rates look like is also a very interesting topic that we introduce and study via analysis and simulation.

Second, and perhaps most interesting, is the notion of *bounded-sensitivity*, and the discovery that the $M/G/K$/JSQ/PS farm exhibits bounded-sensitivity to the service distribution, apart from the mean job size. This is particularly intriguing in light of the fact that so many other routing policies for PS server farms, like Least-Work-Left or Round-Robin, do not exhibit this insensitivity property. Via simulations, we conjecture that despite being simple and oblivious to the sizes of jobs, the performance of JSQ is comparable to load balancing policies which know the sizes of all the jobs in the system.

Finally, we proposed a novel many-servers heavy-traffic scaling to study load balancing polices for server farms. Based on this scaling, we proposed a new closed-form approximation for the stationary joint distribution of queue length in the $M/M/K$/JSQ/PS model, which we believe can be combined with existing light traffic approximations to provide a uniformly sharp approximation. We also presented

the first approximation for the distribution of response time. Finally, we utilized our many server scaling to analyze load balancing policies for heterogeneous server farms, and proposed the traffic-oblivious HYBRID policy – a cross between Join-Shortest-Queue (JSQ) and Minimum-Expected-Response Time (MER) heuristics, which performs favorably compared to traffic-aware policies.

Impact: Join-the-Shortest-Queue has been the subject of numerous papers – both from the point of view of analyzing its performance, and for finding optimal load balancing policies for heterogeneous server farms. However, most of the existing work has been limited to numerical studies. Our many-servers scaling for the first time allows us to analytically address these questions. For example, we proved that popular approximations for the mean response time of the $M/M/K$/JSQ/PS queueing system can be off by 100% in the many-servers regime. We also proved that, counter to intuition, JSQ load balancing while ignoring the server speeds is optimal for heterogeneous server farms in the many-servers regime. Finally, we propose investigating the bounded-sensitivity phenomenon of the $M/G/K$/JSQ/PS queueing system in the many-servers regime, which should lead to useful intuition into this behavior. Our results are not meant to invalidate the existing work, but instead to complement them to develop better approximations for the performance of JSQ, and to develop better load balancing algorithms (for example, the HYBRID policy).

Open Problems: The present chapter perhaps opens many more areas of exploration than it closes. The first question of interest is to combine the many-server heavy-traffic approximation (Theorem 5.5) with existing light-traffic approximations in a principled manner to arrive at an approximation that is sharp across all parameter settings (number of servers and arrival rate).

The second question is to further investigate the notion of bounded-insensitivity: Are there closed-form bounds on the effect of job-size distribution on the mean response time of $M/G/K$/JSQ/PS model? We believe that our proposed many-server regime is the right tool to approach this problem, and the author has made preliminary progress as of writing of this thesis. However, a deeper question is to investigate the cause of this phenomenon (we have already observed a similar behavior in Section 3.4 for the quantum-based Round-Robin problem). When can insensitivity under tractable service distributions, such as the degenerate hyperexponential (H_2^*) be employed to conclude near- or bounded-insensitivity? Is size-independent scheduling a sufficient condition? E.g., $M/G/K$/LWL/PS is insensitive under H_2^*, yet its performance is very sensitive to the service distribution.

Finally, algorithmically, is it possible to find policies that can significantly outperform JSQ for PS servers? We have seen evidence that even size-aware policies do not perform significantly better. However, for heterogeneous servers, JSQ is likely to be

quite inferior to size-aware load balancing policies.

5.A Optimality of Least-Work-Left Routing for Deterministic Job Sizes

Theorem 5.7 *In the $G/D/K/\cdot/PS$ system, Least-Work-Left routing policy minimizes the mean sojourn time.*

Proof: We will use backward induction technique as outlined in [111] to prove that Least Work Left minimizes the sum of response time in the $G/D/K/PS$ model.

Consider an arrival sequence of n jobs. Let π be any routing policy and let $\pi^{(k)}$ be the routing policy that routes the first k arrivals according to π, and the remaining jobs using Least-Work-Left. Let T_P be the sum of response times of the jobs under the routing policy P. We will show,

$$T_{\pi^{(k-1)}} \leq T_{\pi^{(k)}} \tag{5.25}$$

We will use backward induction to prove the above. First we make the following observation: With deterministic job sizes (of size 1), a job arriving into a queue with workload w increases the sum of response times of jobs in system (including itself) by $1 + 2w$.

Basis step: $\ell = n$ Straight forward using the above observation.

Inductive step: Assume that for a given k, $1 \leq k \leq n$, (5.25) holds for all $k \leq \ell \leq n$. We will prove that (5.25) holds for $\ell = k - 1$.

We will prove this by creating another policy, $\gamma^{(k)}$, that behaves like π for the first $k-1$ arrivals and does LWL at the kth arrival. Further, we will show that

$$T_{\gamma^{(k)}} \leq T_{\pi^{(k)}} \tag{5.26}$$

Now, $\pi^{(k-1)}$ is a policy that behaves like $\gamma^{(k)}$ for the first k arrivals and then does LWL (since $\gamma^{(k)}$ does LWL at the kth arrival). Applying induction step to $\gamma^{(k)}$,

$$T_{\pi^{(k-1)}} \leq T_{\gamma^{(k)}} \leq T_{\pi^{(k)}}$$

The policy $\gamma^{(k)}$ is constructed as follows: Let $\pi^{(k)}$ route the kth job to queue r, whereas the queue with the least work left is s. The policy $\gamma^{(k)}$ routes the kth job to s. For subsequent arrivals, if $\pi^{(k)}$ routes the job to s, $\gamma^{(k)}$ routes to r, and if $\pi^{(k)}$

routes the job to r, $\gamma^{(k)}$ routes it to s. The routing of other jobs under $\gamma^{(k)}$ and $\pi^{(k)}$ is identical.

It is easy to see that (5.26) holds under the above construction, thus completing the proof of optimality of LWL with deterministic job sizes. ■

In fact, as a consequence of Theorem 5.1 of Liu, Nain and Towsley [111], the following strong theorem holds.

Theorem 5.8 *In the $G/D/K/\cdot/PS$ system, Least-Work-Left routing policy minimizes the vector of workloads at the queues in the increasing Schur convex ordering.*

Chapter 6

Energy-Efficient Dynamic Capacity Provisioning in Server Farms

In this chapter we turn to an important algorithmic problem: efficiently provisioning the number of servers in a server farm so as to optimize energy/response-time tradeoffs. Traffic demand experienced by data centers and cloud computing infrastructures is highly non-stationary, exhibiting not only seasonal and diurnal variations, but also unpredictable surges. Therefore, server farms which are provisioned to handle the peak demand usually have many servers idle. While one would like to turn servers off when they become idle to save energy, the large setup cost (both, in terms of setup time and energy penalty) needed to switch the server back on can adversely affect performance. The problem is made more complex by the fact that today's servers provide multiple sleep or standby states which trade off the setup cost with the power consumed while the server is 'sleeping'.

We employ the metric of Energy-Response time Product (ERP) to capture the energy-performance tradeoff, and present theoretical results on the optimality of server farm management policies. For a stationary demand pattern, we prove that there exists a very small, natural class of policies that always contains the optimal policy for a single server, and conjecture it to contain a near-optimal policy for multi-server systems. For time-varying demand patterns, we propose a simple, traffic-oblivious policy and provide empirical evidence for its near-optimality.

6.1 Introduction

Motivation

Server farm power consumption accounts for more than 1.5% of the total electricity usage in the U.S., at a cost of nearly $4.5 billion [146]. The rising cost of energy and the tremendous growth of data centers will result in even more expenditures on power consumption. Unfortunately, due to over-provisioning, only 20-30% of the total server capacity is used on average [22]. This over-provisioning results in idle servers which can consume as much as 60% of their peak power.

While a lot of energy can be saved by turning *idle* servers *off*, turning on an *off* server incurs a significant cost. The *setup cost* takes the form of both a time delay, which we refer to as the *setup time*, and an *energy penalty*. Another option is to put *idle* servers into some *sleep* state. While a server in *sleep* mode consumes more power than an *off* server, the setup cost for a sleeping server is lower than that for an *off* server. Today's state-of-the-art servers come with an array of *sleep* states, leaving it up to the server farm manager to determine which of these is best.

Goal and metric

There is a clear tradeoff between leaving *idle* servers on, and thus minimizing mean response time, versus turning *idle* servers *off* (or putting them to *sleep*), which hurts response time but may save power. Optimizing this tradeoff is a difficult problem, since there are an infinite number of possible server farm management policies. Our goal in this chapter is to find a simple class of server farm management policies, which optimize (or nearly optimize) the above tradeoff. We also seek simple rules of thumb that allow designers to choose from this class of near-optimal policies. In doing so, we greatly simplify the job of the server farm manager by reducing the search space of policies that he/she needs to choose from.

To capture the tradeoff involved in energy and performance, and to compare different policies, we use the Energy-Response time Product (ERP) metric, also known as the Energy-Delay Product (EDP) [63, 87, 89, 96, 137]. For a control policy π, the ERP is given by:

$$ERP^{\pi} = \mathbf{E}[P^{\pi}] \cdot \mathbf{E}[T^{\pi}]$$

where $\mathbf{E}[P^{\pi}]$ is the long-run average power consumed under the control policy π, and $\mathbf{E}[T^{\pi}]$ is mean customer response time under policy π. Minimizing ERP can be seen as maximizing the "performance-per-watt", with performance being defined as the

inverse of mean response time. While ERP is widely accepted as a suitable metric to capture energy-performance tradeoffs, we believe we are the first to analytically address optimizing the metric of ERP in server farms.

Note that there are other performance metrics that also capture the tradeoff between response time and energy, for example, a weighted sum of the mean response time and mean power (ERWS) [13, 17, 156]. However, the ERWS metric implies that a reduction in mean response time from 1001 sec to 1000 sec is of the same value as a reduction from 2 sec to 1 sec. By contrast, the ERP metric implies that a reduction in mean response time from 2 sec to 1 sec is better than a reduction from 1001 sec to 1000 sec. One reason for the popularity of ERWS is that it is a nicer metric to handle analytically, being a single expectation, and hence additive over time. Therefore, one can optimize the ERWS metric via Markov Decision Processes, for example. From the point of view of worst case sample path based analysis, this metric allows one to compare arbitrary policies to the optimal policy via potential function arguments [84]. However, ERP, being a product of two expectations, does not allow a similar analysis. Other realistic metrics of interest include minimizing total energy given bounds on, say, the 95%tile of response times.

Summary of Contributions

We consider a specific set of server farm management policies (defined in Table 6.1) and prove that it contains the optimal policy for the case of a single server, and also contains a near-optimal policy for the case of multi-server systems, assuming a stationary demand pattern. For the case of time-varying demand patterns, we develop a traffic-oblivious policy that can auto-scale the server farm capacity to adapt to the incoming load. Via simulations, we show that our traffic-oblivious policy performs well when the server farm is offered a general time-varying arrival process. Throughout this chapter, for analytical tractability, we make the assumption of exponentially distributed job sizes and a Poisson arrival process. Setup times are assumed to be Deterministic. We formally define the traffic model and the model for servers' *sleep* state dynamics in Section 6.3.

- We begin with the question of designing the optimal power management policy for an $M/M/1$ queue in Section 6.4. While the range of possible policies is large, for example, immediately put a server to *sleep* when it goes *idle* and then delay turning on the server until a certain number of jobs have accumulated in the queue (to amortize setup cost) transitioning to shallower sleep states on arrivals, we prove that one of the policies, NEVEROFF, INSTANTOFF or SLEEP, is always optimal. Refer to Table 6.1 for the exact definitions of these

Policy	Single-Server	Multi-Server
NEVEROFF	Whenever the server goes *idle*, it remains *idle* until a job arrives.	A fixed optimally chosen number n^* (with respect to ERP) of servers are maintained in the *on* or *idle* states. If an arrival finds a server *idle*, it starts serving on the *idle* server. Arrivals that find all n^* servers *on* (busy) join a central queue from which servers pick jobs when they become *idle*.
INSTANTOFF	Whenever the server goes *idle*, it turns *off*. The server then remains *off* until there is no work to process, and begins to turn *on* as soon as work arrives.	Whenever a server goes *idle*, and there are no jobs in the queue, the server turns *off*. Otherwise it picks a job from the queue to serve. At any moment in time, there are some number of servers that are *on* (busy), and some number of servers that are in *setup* (transitioning from *off* to *on*). Every arrival puts a server into *setup* mode, unless the number of servers in *setup* already exceeds the number of jobs in the queue. A job does not necessarily wait for the full setup time since it can be run on a different server that becomes free before the setup time is complete, leaving its initially designated server in *setup*.
SLEEP(S)	Whenever a server goes *idle*, it goes into the *sleep* state S. It remains in *sleep* state S until there is no work to process, and begins to wake up as soon as work arrives.	A fixed optimally chosen number n^* of servers are maintained in the *on*, *off* or *sleep* states. Whenever a server goes *idle*, and there are no jobs in the queue, it goes into the *sleep* state S. Otherwise it picks a job from the queue to serve. Every arrival wakes a sleeping server and puts it into *setup*, unless the number of servers in *setup* already exceeds the number of jobs in the queue.

Table 6.1: A summary of the different policies considered in this chapter, and their description in the single-server and multi-server cases.

policies.

- In Section 6.5, we consider the case of managing servers in an $M/M/\infty$ multi-server systems. The arrival process is Poisson with a known mean arrival rate. We assume that there are enough servers so that we are not constrained by the available capacity. Again, the range of policies to choose from is vast. For example, some servers could be turned *off* when *idle*, some could be moved to a specific *sleep* state, and the rest may be kept *idle*. One could also delay turning on an *off* server until a certain number of jobs have accumulated in the queue, or delay turning *off* an *idle* server until some time has elapsed. Via a combination of analysis and numerical experiments, we conjecture that one

of NEVEROFF, INSTANTOFF or SLEEP (defined in Table 6.1 for a multi-server system) is near-optimal.

- In Section 6.6 we consider a time-varying arrival pattern with the aim of finding policies which can auto-scale the capacity while being oblivious to the traffic intensity. This situation is even more complicated than in Section 6.5, since a server farm management policy might now also take into account the history of arrivals or some predictions about the future arrivals. For the time-varying case, we introduce a new policy DELAYEDOFF. Under the DELAYEDOFF policy, a server is only turned *off* if it does not receive any jobs to serve within an interval of length t_{wait}. If an arrival finds more than one server *idle* on arrival, it is routed to the server which was *most recently busy* (MRB) (alternately, the server which is the farthest from turning *off*). Otherwise, the arriving job turns *on* an *off* server.
  The MRB routing policy proposed above turns out to be crucial for the near-optimality of DELAYEDOFF. Intuitively, MRB routing increases the variance of the idle periods of the servers when compared to random or round-robin routing, and yields the property that the longer a server has been idle, the longer it is likely to stay idle.Policies similar to DELAYEDOFF have been proposed in the literature but applied to individual devices [51, 84, 129], whereas in our case we propose to apply it to a pool of homogeneous interchangeable servers under MRB routing. We provide simulation evidence in favor of the auto-scaling capabilities of DELAYEDOFF and show that it compares favorably to an offline, traffic-aware capacity provisioning policy.

6.2 Prior work

Prior analytical work in server farm management to optimize energy-performance tradeoff can be divided into *stochastic analysis*, which deals with minimizing average power/delay or the tail of power/delay under some probabilistic assumptions on the arrival sequence, and *worst-case analysis*, which deals with minimizing the cost of worst-case arrival sequences.

Stochastic Analysis

The problem of server farm management is very similar in flavor to two well studied problems in the stochastic analysis community: operator staffing in call centers and inventory management. In call center staffing, the servers are operators, who require

a salary (power) when they are working. Similarly to our problem, these operators require a setup cost to bring an employee into work, however, importantly, all analysis in call center staffing has ignored this setup cost.

The operator staffing problem involves finding the number of operators (servers) which minimize a weighted sum of delay costs experienced by users and the monetary cost of staffing operators. While this problem has received significant attention under the assumption of stationary (non-time-varying) demand (see [31] for recent results), there is significantly less work for the time-varying case, one exception being [85]. In [85], the authors consider the problem of dynamic staffing based on knowing the demand pattern so as to maintain a target probability of a user finding all servers busy on arrival.

Within inventory management, the problem of capacity provisioning takes the form: how much inventory should one maintain so as to minimize the total cost of unused inventory (holding cost, in our case *idle* power) and waiting cost experienced by orders when there is no inventory in stock (queueing delay of users). Conceptually this problem is remarkably similar to the problem we consider, and the two common solution strategies employed, known as Make to Order and Make to Stock, are similar in flavor to what we call INSTANTOFF and NEVEROFF, respectively (see [7], for example). However, in our case servers can be turned *on* in parallel, while in inventory management it is assumed that inventory is produced *sequentially* (this is similar to allowing at most one server to be in *setup* at any time).

Worst-case Analysis

There has been significant amount of work done in power management from the point of view of minimizing worst case sample path cost, for example ERWS (See [83] for a recent survey). Again, none of the prior work encompasses a setup time and is more applicable to a single device than a server farm. The performance metrics used are also very different from ERP.

The work can primarily be split in terms of results on speed scaling algorithms, and results on algorithms for powering down devices. In the realm of speed scaling, the problem flavors considered have been minimizing energy or maximum temperature while meeting job deadlines [18, 19, 160], minimizing mean response time subject to a bound on total energy [124], and minimizing the ERWS [17, 156]. However, again all these papers assume that the speed level can be switched *without any setup costs*, and hence are mainly applicable to single stand-alone devices, since in multi-server systems setup costs are required to increase capacity.

The work on powering down devices is more relevant to the problem we consider,

and due to sample path guarantees, these results naturally lead to traffic-oblivious powering down schemes. In [84] the authors consider the problem of minimizing total energy consumed under the constraint that a device must instantly turn on when a job arrives. Further, [84] assumes that there is *no setup time* while turning on a device, only an energy penalty.

6.3 Model

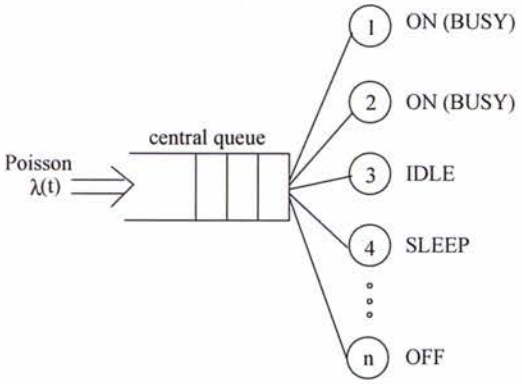


Figure 6.1: Illustration of server farm model for studying power management algorithms

Figure 6.1 illustrates our server farm model. We assume n homogeneous servers of capacity one, where each server can process any job, and thus the servers are interchangeable. Jobs arrive from outside the system, to a central queue, according to a Poisson process. In Sections 6.4 and 6.5, we consider a fixed arrival rate, λ. However, in Section 6.6, we consider a time-varying arrival rate, $\lambda(t)$. We assume the job sizes are *i.i.d.* Exponentially distributed random variable with mean $\mathbf{E}[S]$. The quantity $\rho(t) = \lambda(t)\mathbf{E}[S]$ is used to denote the instantaneous load, or the rate at which work is entering the system at time t.Therefore, ρ represents the minimum number of servers needed to maintain a stable system (same as in Chapter 2).

Each server can be in one of the following states: *on* (busy), *idle*, *off*, or any one of $N-1$ *sleep* states: S_1, S_2, ..., S_{N-1}. For convenience, we sometimes refer to the *idle* state as S_0 and the *off* state as S_N. The associated power values are P_{ON}, $P_{IDLE} = P_{S_0}$, P_{S_1}, ..., $P_{S_N} = P_{OFF}$. We shall assume the ordering $P_{ON} > P_{IDLE} > P_{S_1} > \ldots > P_{S_{N-1}} > P_{OFF} = 0$. The server can only serve jobs in the *on* state. P_{ON} need not necessarily denote the peak power at which a job is served, but is

used as a proxy for the average power consumed during the service of a job. Indeed, while applying our model, we would first profile the workload to measure the average power consumed during a job's execution, and use it as P_{ON}. The time to transition from initial state, S_i, to final state, S_f, is denoted by $T_{S_i \to S_f}$ and is deterministic. Rather obviously, we assume $T_{ON \to IDLE} = T_{IDLE \to ON} = 0$. Further, the average power consumed while transitioning from state S_i to S_f is given by $P_{S_i \to S_f}$.

Model Assumptions: For analytical tractability, we will introduce some additional assumptions. We will assume that the time to transition from a state to any state with lower power is zero. Therefore, $T_{ON \to OFF} = T_{S_i \to OFF} = 0$, for all i. This assumption is justified because the time to transition back to a higher power state is generally considerably larger than the time to transition to the lower power state, and hence dominates the performance penalties. Further, we will assume that the time to transition from a state S_i to any higher power state is only dependent on the low power state, and we will denote this simply as T_{S_i}. Therefore, $T_{OFF \to IDLE} = T_{OFF \to S_i} = T_{OFF}$, for all i. Note that $0 = T_{IDLE} < T_{S_1} < \ldots < T_{S_{N-1}} < T_{OFF}$. This assumption is justified because in current implementations there is no way to go between two *sleep* states without first transitioning through the $IDLE$ state. Regarding power usage, we assume that when transitioning from a lower power state, S_i, to a higher power state S_f, we consume power $P_{S_i \to S_f} = P_{ON}$.

The results of this chapter are derived under the *Model Assumptions* which were validated experimentally.

Simulation Parameters: All the simulation results presented in this chapter are based on the following server characteristics: $T_{OFF} = 200s$, $T_{Sleep} = 60s$, $P_{OFF} = 0W$, $P_{Sleep} = 10W$, $P_{IDLE} = 150W$ and $P_{ON} = 240W$. These parameter values are based on measurements for the Intel Xeon E5320 server, running the CPU-bound LINPACK [82] workload.

6.4 Optimal Single Server policies

As the first step towards our goal of finding policies for efficiently managing server pools, we analyze the case of a single server system. Recall that our aim is to find the policy that minimizes ERP under a Poisson arrival process of known intensity. Theorem 6.1 below states that for a single server, remarkably, the optimal policy is included in the set {NEVEROFF, INSTANTOFF, SLEEP} (defined in Table 6.1).

Theorem 6.1 *For the single server model with a Poisson(λ) arrival process and i.i.d. Exponentially distributed job sizes, the optimal policy for minimizing ERP is*

either NEVEROFF, INSTANTOFF *or* SLEEP(S), *where S is the optimally chosen* sleep *state among the existing* sleep *states.*

Remark 1: Theorem 6.1 is quite non-intuitive, and in general we do not expect such a result to hold for other metrics such as ERWS. The theorem rules out a large class of policies, for example those which may randomize between transitioning to different *sleep* states, or policies which move from one *sleep* state to another, or those which may wait for a few jobs to accumulate before transitioning to the *on* state. While *ERP*, being a product of expectations, is a difficult metric to address analytically, for the single-server case we are able to obtain tight optimality results by deriving explicit expressions for ERP.

Proof of Theorem 6.1: We give a high-level sketch of the proof in terms of four lemmas, whose proofs are deferred to Appendix 6.A. These lemmas successively narrow down the class of optimal policies, until we are left with only NEVEROFF, INSTANTOFF and SLEEP.

Definition 6.1 *Let Π_{mixed} denote the class of randomized policies whereby a server immediately transitions to power state S_i ($i \in \{0, \ldots, N\}$) with probability p_i on becoming* idle. *Given that the server went into power state S_i, with probability q_{ij} it stays in S_i and waits until j jobs accumulate in the queue, where $\sum_{j=1}^{\infty} q_{ij} = 1$. Once the target number of jobs have accumulated, the server immediately begins transitioning to the* on *state, and stays there until going* idle.

Lemma 6.1 *Under a Poisson arrival process and general i.i.d. job sizes, the optimal policy lies in the set Π_{mixed}.*

Lemma 6.2 *Consider a policy $\pi \in \Pi_{mixed}$ with parameters as in Definition 6.1. The mean response time for policy π under a Poisson(λ) arrival process with i.i.d.* Exp(μ) *job sizes is given by:*

$$\mathbf{E}[T] = \frac{\sum_{i=0}^{N} p_i \sum_{j=1}^{\infty} q_{ij} r_{ij}}{\sum_{i=0}^{N} p_i \sum_{j=1}^{\infty} q_{ij} \left(j + \lambda T_{S_i}\right)} \tag{6.1}$$

where,

$$r_{ij} = \frac{j + \lambda T_{S_i}}{\mu - \lambda} + \left[j T_{S_i} + \frac{j(j-1)}{2\lambda} + \frac{\lambda T_{S_i}^2}{2} \right] \tag{6.2}$$

and the average power for policy π is given by:

$$\mathbf{E}[P] = \frac{\sum_{i=0}^{N} p_i \sum_{j=1}^{\infty} q_{ij} \left(j(\rho P_{ON} + (1-\rho) P_{S_i}) + \lambda T_{S_i} P_{ON}\right)}{\sum_{i=0}^{N} p_i \sum_{j=1}^{\infty} q_{ij} \left(j + \lambda T_{S_i}\right)}. \tag{6.3}$$

Lemma 6.3 *The optimal strategy for a single server must be pure. That is,* $p_i = 1$ *for some* $i \in \{0, \ldots, N\}$, *and* $q_{in_i} = 1$ *for some integer* $n_i \geq 1$.

Lemma 6.4 *The optimal pure strategy dictates that* $n_i = 1$, *if the optimal* sleep *state is* S_i.

Lemma 6.1 is proved using a sample path argument and crucially depends on the assumption of a Poisson arrival process and the *Model Assumptions* for the *sleep* states of the server, and in fact holds for any metric that is increasing in mean response time and mean power. Lemma 6.3 relies on the structure of ERP metric. While Lemma 6.3 also holds for the ERWS metric (with a much simpler proof), it does not necessarily hold for general metrics such as the product of the mean power and the square of the mean response time. Lemma 6.4 also relies on the structure of the ERP metric and does not hold for other metrics such as ERWS. ■

Corollary 6.1

$$\mathbf{E}[T] = \frac{1}{\mu - \lambda} + \frac{T_{S_i}(1 + \lambda T_{S_i}/2)}{1 + \lambda T_{S_i}} \tag{6.4}$$

$$\mathbf{E}[P] = \frac{\rho P_{ON} + (1 - \rho) P_{S_i} + \lambda T_{S_i} P_{ON}}{1 + \lambda T_{S_i}} \tag{6.5}$$

where $S_i = IDLE$ *for* NEVEROFF, $S_i = OFF$ *for* INSTANTOFF, *and* S_i *is the* sleep *state that we transition to in* SLEEP.

Proof: Follows by substituting $p_i = 1$ and $q_{i1} = 1$ in Lemma 6.2. ■

The expressions in Corollary 6.1 allow us to determine regimes of load and mean job sizes for which each of NEVEROFF, INSTANTOFF and SLEEP policy is best with respect to ERP. Numerically, we have found that NEVEROFF is typically superior to the other policies, unless the load is low *and* the mean job size is high, resulting in very long idle periods. In the latter case, INSTANTOFF or one of the SLEEP policies is superior, depending on the parameters of the *sleep* and *off* states. Eqs. (6.4) and (6.5) are also helpful for guiding a server architect towards designing useful *sleep* states by enabling the evaluation of ERP for each candidate *sleep* state.

6.5 Near-Optimal Multi-server policies

In this section, we extend our results for single server systems to the multi-server systems with a fixed known arrival rate, with the goal of minimizing ERP. Inspired

by the results in Section 6.4, where we found the best of NEVEROFF, INSTANTOFF and SLEEP to be the optimal policy, we intuit that one of NEVEROFF, INSTANTOFF and SLEEP will be close to optimal in the multi-server case as well. We make this intuition precise in Section 6.5.1, and provide simple guidelines for choosing the right policy from among this set in Section 6.5.2.

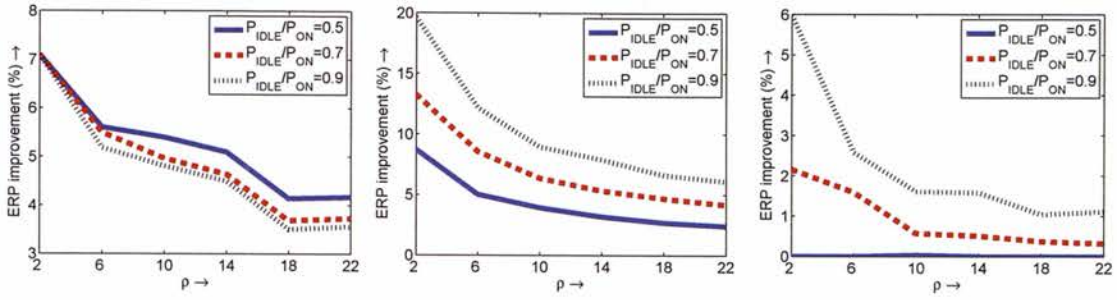
6.5.1 A Near-optimality conjecture

Conjecture 6.1 *Let Π_{S_i} denote the class of policies which only involve the states* on, idle *and the S_i* sleep *state. For arbitrary S_i (that is P_{S_i} and T_{S_i}), the ERP of the best of* NEVEROFF *and* SLEEP *with* sleep *state S_i is within 30% of the ERP of the optimal policy in Π_{S_i} when $\rho \geq 10$. When $\rho \geq 20$, the performance gap is smaller than 20%.*

The main idea behind Conjecture 6.1 is obtaining reasonably good lower bounds on the ERP for the optimal policy, and then numerically optimizing the performance gap with respect to the lower bound. We justify Conjecture 6.1 in Appendix 6.B.

We believe that in reality, the simple NEVEROFF, INSTANTOFF, and SLEEP policies are better than our Conjecture suggests. To justify this claim, we perform the following simulation experiment. We focus on the class of policies involving *on*, *idle* and *off* states. Note that as we mentioned earlier, due to the metric of ERP, we cannot utilize the framework of Markov Decision Processes/Stochastic Dynamic Programming to numerically obtain the optimal policy. Instead we limit ourselves to the following class of threshold policies:
THRESHOLD(n_1, n_2): At least n_1 servers are always maintained in either the *on* or *idle* states, but no more than n_2 servers are ever turned on. If an arrival finds a server *idle*, it begins service. If the arrival finds all servers *on* (busy) or turning on, but this number is less than $n_2 \geq n_1$, then the arrival turns on an *off* server. Otherwise the arrival waits in a queue. If a server becomes *idle* and the queue is empty, the server turns *off* if there are at least n_1 other servers which are *on*.

The THRESHOLD policy can be seen as a mixture of NEVEROFF with n_1 servers, and INSTANTOFF with $(n_2 - n_1)$ servers. Thus, THRESHOLD represents a broad class of policies (since n_1 and n_2 can be set arbitrarily), which includes NEVEROFF and INSTANTOFF. In Figure 6.2, we show the gain in ERP afforded by the optimal THRESHOLD policy over the best among optimal NEVEROFF and INSTANTOFF for various values of ρ, T_{OFF} and $\frac{P_{IDLE}}{P_{ON}}$. We see that if T_{OFF} is small (Figure 6.2 (a)), the ERP gain of the THRESHOLD policy over the best of NEVEROFF and INSTANTOFF is marginal ($< 7\%$). This is because in this case, INSTANTOFF is



(a) $T_{OFF} = 0.1\mathbf{E}[S]$ (b) $T_{OFF} = 0.5\mathbf{E}[S]$ (c) $T_{OFF} = 2\mathbf{E}[S]$

Figure 6.2: Comparison of the performance of THRESHOLD policy against the best of optimal NEVEROFF and INSTANTOFF policies. The y-axis shows the percentage improvement in ERP afforded by the THRESHOLD policy.

close to optimal. At the other end, when T_{OFF} is large (Figure 6.2 (c)), the ERP gain of the THRESHOLD policy over the best of NEVEROFF and INSTANTOFF are again marginal ($< 6\%$), because now NEVEROFF is close to optimal. We expect the optimal THRESHOLD policy to outperform the best of NEVEROFF and INSTANTOFF when T_{OFF} is moderate (comparable to $\frac{P_{IDLE} \cdot \mathbf{E}[S]}{P_{ON}}$). In Figure 6.2 (b), we see that this is indeed the case. However, the gains are still moderate (an improvement of 10% when $\rho \geq 10$ and at most 7% when $\rho \geq 20$ when P_{IDLE} is high).

6.5.2 Choosing the right policy

Based on the results of Section 6.5.1, to provision a multi-server system with a fixed known arrival rate, it suffices to only consider the policies NEVEROFF, INSTANTOFF and SLEEP. The goal of this section is to develop a series of simple rules of thumb that will help a practitioner choose between these policies. The specific questions we answer in this section are:

Question 1: What is the optimal number of servers, n^*, for the NEVEROFF policy?
Question 2: What is the optimal number of servers, n^*, for the SLEEP policy?
Question 3: Which of INSTANTOFF, NEVEROFF, and the various SLEEP policies should be chosen?

Before presenting the rules of thumb to answer the above questions, we present a well-known result regarding the $M/M/K$ queueing system which forms the basis of further analysis.

Lemma 6.5 (Halfin and Whitt [72]) *Consider a sequence of $M/M/s_n$ systems*

with load ρ_n in the nth system. Let α_n denote the probability that an average customer finds all servers busy in the nth system. Then,

$$\lim_{\rho_n \to \infty} \alpha_n = \alpha(\beta) \text{ if and only if } \lim_{\rho_n \to \infty} \frac{s_n - \rho_n}{\sqrt{\rho_n}} = \beta. \tag{6.6}$$

The function $\alpha(\beta)$ is given by

$$\alpha(\beta) = \left[1 + \sqrt{2\pi}\beta\Phi(\beta)e^{\frac{\beta^2}{2}}\right]^{-1} \tag{6.7}$$

where $\Phi(\cdot)$ is the c.d.f. of a standard Normal variate. Under the above conditions, the mean number of jobs in the nth system, $\mathbf{E}\left[N^{M/M/s_n}\right]$, satisfies:

$$\lim_{\rho_n \to \infty} \frac{\mathbf{E}\left[N^{M/M/s_n}\right] - \rho_n}{\sqrt{\rho_n}} = \frac{\alpha(\beta)}{\beta}. \tag{6.8}$$

Rule of Thumb #1: Choosing n^* for NeverOff
For the parameter regime where NEVEROFF is the chosen policy,

$$n^* = \rho + \beta^*(P_{IDLE}/P_{ON})\sqrt{\rho} + o(\sqrt{\rho}) \tag{6.9}$$

where $\beta^*(\cdot)$ is the following function:

$$\beta^*(x) = \arg\min_{\beta>0}\left(\frac{\alpha(\beta)}{\beta} + \beta \cdot x\right). \tag{6.10}$$

A very good approximation $\beta^*(x) \approx \frac{0.4105x^2+0.8606x+0.0395}{x^2+0.5376x+0.01413}$ is obtained via the MATLAB curve fitting toolbox, with a maximum absolute relative error of $< 0.75\%$.

Justification: Consider a sequence of $M/M/s_n$ systems with load ρ_n in the nth system. Let $s_n \sim \rho + g(\rho_n) + o(g(\rho_n))$. From [72], we have that $\mathbf{E}\left[N^{M/M/s_n}\right] \sim \rho_n + \frac{\rho_n}{g(\rho_n)}\alpha_n$ where α_n denotes the stationary probability that all s_n servers are busy in the nth system. Also, $\mathbf{E}\left[P^{M/M/s_n}\right] \sim \rho P_{ON} + g(\rho_n)P_{IDLE}$, which gives

$$\mathbf{E}\left[N^{M/M/s_n}\right] \cdot \mathbf{E}\left[P^{M/M/s_n}\right] = \rho_n^2 P_{ON}\left(1 + \frac{\alpha_n}{g(\rho_n)} + \frac{g(\rho_n)}{\rho_n}\frac{P_{IDLE}}{P_{ON}} + o() \text{ terms}\right).$$

When $g(\rho_n) = \omega(\sqrt{\rho_n})$, $\alpha_n \to 0$, and the expression in the parenthesis is $1 + \omega\left(1/\sqrt{\rho_n}\right)$. When $g(\rho_n) = o(\sqrt{\rho_n})$, $\alpha_n \to 1$, and the expression in the parenthesis is again $1 + \omega\left(1/\sqrt{\rho_n}\right)$. Thus, the optimal choice is $g(\rho_n) = \beta\sqrt{\rho_n} + o(\sqrt{\rho_n})$ for some constant β. This yields:

$$ERP^{NEVEROFF} \sim \rho_n \mathbf{E}[S] P_{ON}\left(1 + \frac{\frac{\alpha(\beta)}{\beta} + \beta\frac{P_{IDLE}}{P_{ON}}}{\sqrt{\rho_n}}\right) \tag{6.11}$$

Optimizing the above yields the expression for β^*. ∎

For the ERWS metric, the rule $n^* = \rho + \beta\sqrt{\rho}$ is known to be near-optimal in practice. It is popularly known as the "square-root staffing rule", or the Quality and Efficiency Driven regime because it balances the sub-optimality in the performance (Quality) and resource utilization (Efficiency), both being $\Theta\left(\frac{1}{\sqrt{\rho}}\right)$, and hence optimizing the ERWS metric. Here we have shown that the square-root staffing rule also optimizes the ERP metric, albeit with a different β.

Rule of Thumb #2: Choosing n^* for Sleep

For the parameter regime where SLEEP with *sleep* state S_i is the chosen policy,

$$n^* = \rho' + \beta^*(P_{S_i}/P_{ON})\sqrt{\rho'} + o(\sqrt{\rho'}) \tag{6.12}$$

where $\rho' = \rho\left(1 + \frac{T_{S_i}}{\mathbf{E}[S]}\right)$ and $\beta^*(\cdot)$ is given by (6.10).

Justification: The justification for Rule of Thumb #2 is along the same lines. We expect the SLEEP(S_i) policy to outperform NEVEROFF when T_{S_i} is small enough so that almost all jobs turn on a *sleeping* server and get served there. This is equivalent to an $M/G/\infty$ system with $G \sim S + T_{S_i}$. However, since $P_{S_i} > 0$, we optimize the number of servers by following Rule of Thumb #1, but with mean job size replaced by $\mathbf{E}[S] + T_{S_i}$, or equivalently $\rho' \leftarrow \rho\left(1 + \frac{T_{S_i}}{\mathbf{E}[S]}\right)$, and $P_{IDLE} \leftarrow P_{S_i}$. This gives us:

$$ERP^{Sleep(S_i)} \sim \rho\mathbf{E}[S]\left(1 + \frac{T_{S_i}}{\mathbf{E}[S]}\right)^2 P_{ON}\left(1 + \frac{\frac{\alpha(\beta)}{\beta} + \beta\frac{P_{S_i}}{P_{ON}}}{\sqrt{\rho\left(1 + \frac{T_{S_i}}{\mathbf{E}[S]}\right)}}\right) \tag{6.13}$$

∎

Rule of Thumb #3: Which policy to use?

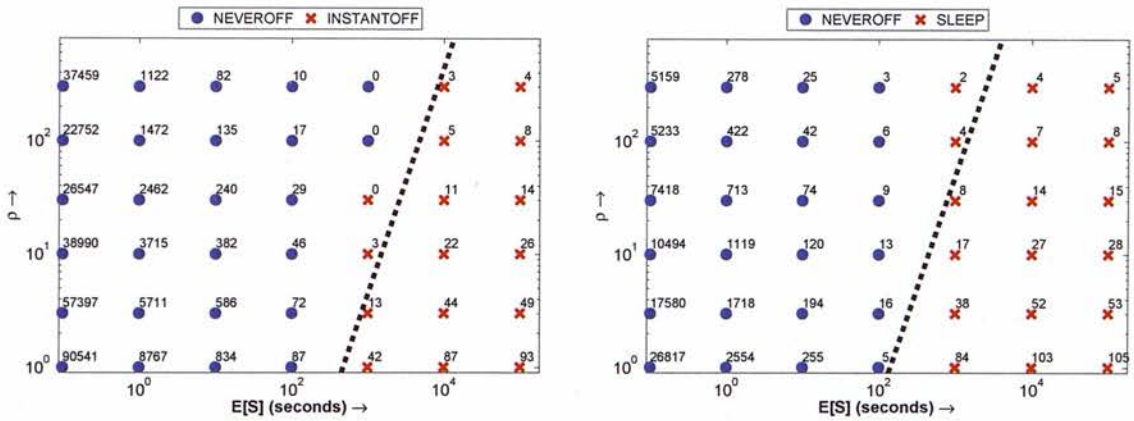
We associate each policy with an index, and choose the policy with the smallest index. The index for INSTANTOFF is given by $\left(1 + \frac{T_{OFF}}{\mathbf{E}[S]}\right)^2$. The index for NEVEROFF is given by $\left(1 + \frac{\gamma(P_{IDLE}/P_{ON})}{\sqrt{\rho}}\right)$, and for SLEEP with state S_i by $\left(1 + \frac{T_{S_i}}{\mathbf{E}[S]}\right)^2\left(1 + \frac{\gamma(P_{S_i}/P_{ON})}{\sqrt{\rho\left(1+\frac{T_{S_i}}{\mathbf{E}[S]}\right)}}\right)$.

The function $\gamma(\cdot)$ is given by

$$\gamma(x) = \min_{\beta>0}\left(\frac{\alpha(\beta)}{\beta} + \beta \cdot x\right) \tag{6.14}$$

with $\alpha(\beta)$ given by (6.7). A very good approximation $\gamma(x) \approx \frac{5.444x^2+2.136x+0.006325}{x^2+4.473x+0.9012}$ is obtained via the MATLAB curve fitting toolbox, with a maximum relative error of $< 0.6\%$ for $x \geq 0.025$.

Justification: We justify the heuristic rule of thumb by proposing approximations for the ERP metric under INSTANTOFF, NEVEROFF, and the SLEEP policies. We expect the INSTANTOFF policy to outperform NEVEROFF and SLEEP when T_{OFF} is small enough compared to $\mathbf{E}[S]$, so that the penalty to turn on an *off* server is negligible compared to the necessary cost of serving the job. In this regime, we can approximate the ERP of INSTANTOFF by $ERP^{INSTANTOFF} \approx \lambda P_{ON} (\mathbf{E}[S] + T_{OFF})^2$, which is an upper bound obtained by forcing every job to run on the server that it chooses to turn on on arrival. The ERP of NEVEROFF with optimal number of servers is approximated by Eq. (6.11), with $\rho_n = \rho$ and $\beta = \beta^*(P_{IDLE}/P_{ON})$. For SLEEP, we again expect SLEEP(S_i) policy to outperform NEVEROFF when T_{S_i} is small enough so that almost all jobs turn on a *sleeping* server and get served there. In this regime, we can approximate the ERP of SLEEP by Eq. (6.13), with $\beta = \beta^*(P_{S_i}/P_{ON})$. Using the above approximations for ERP, we can choose between the INSTANTOFF, NEVEROFF and SLEEP policies. ■



(a) (b)

Figure 6.3: Verifying the accuracy of Rule of Thumb #3. The relative performance of NEVEROFF, INSTANTOFF and SLEEP policies for a multi-server system are shown as functions of load (ρ) and mean job size ($\mathbf{E}[S]$) based on simulations. Figure (a) shows NEVEROFF vs. INSTANTOFF. The crosses indicate the region of superiority of INSTANTOFF over NEVEROFF. Figure (b) shows NEVEROFF vs. SLEEP. The crosses indicate the region of superiority of SLEEP over NEVEROFF. The numbers associated with each point denote the % improvement of the superior algorithm over the inferior. The dashed lines indicate the theoretically predicted split based on Rule of Thumb #3.

If we compare INSTANTOFF and NEVEROFF, Rule of Thumb #3 says that if T_{OFF} is sufficiently small compared to $\mathbf{E}[S]$ and $\frac{1}{\sqrt{\rho}}$, then one should choose INSTANTOFF.

Figure 6.3(a) verifies the accuracy of the above rule of thumb. Observe that in the region where our rule of thumb mispredicts the better policy, the gains of choosing either policy over the other are minimal. Similarly, the dashed line in Figure 6.3(b) indicates that the theoretically predicted split between the NEVEROFF and SLEEP policies is in excellent agreement with simulations.

6.6 Traffic-oblivious dynamic capacity provisioning and Applications

Thus far we have considered a stationary demand pattern. In this section we propose a policy, DELAYEDOFF, and provide empirical evidence towards favorable performance of our proposed policy when the arrival process is Poisson with an unknown non-stationary arrival rate $\lambda(t)$, with $\rho(t) = \lambda(t)\mathbf{E}[S]$. In Section 6.6.2, we propose a slight modification of the DELAYEDOFF policy, called the DELAYEDOFF-INDEX policy, which is easier to implement and is flexible enough to adapt to diverse application scenarios.

6.6.1 The DelayedOff policy

The previous policies that we have considered, NEVEROFF, SLEEP and INSTANTOFF, do not satisfy our goal. NEVEROFF and SLEEP are based on a fixed number of servers n^*, and thus do not auto-scale to time-varying demand patterns. INSTANTOFF is actually able to scale capacity in the time-varying case, since it can turn on servers when the load increases, and it can turn *off* servers when there isn't much work in the system. However, when T_{OFF} is high, we will see that INSTANTOFF performs poorly with respect to ERP.

We now define our proposed traffic-oblivious auto-scaling policy, DELAYEDOFF.
DELAYEDOFF: DELAYEDOFF is a capacity provisioning policy similar to INSTANTOFF, but with two major changes. First, under DELAYEDOFF, we wait for a server to *idle* for some predetermined amount of time, t_{wait}, before turning it *off*. If the server gets a job to service in this period, its idle time is reset to 0. The parameter t_{wait} is a constant chosen independent of load, and thus DELAYEDOFF is a truly traffic-oblivious policy. Second, if an arrival finds more than one servers *idle* on arrival, instead of joining a random *idle* server, it joins the server that was most recently busy (MRB). Equivalently, and perhaps more precisely, the arrival is sent to the server which will turn *off* farthest in the future. We will later see that MRB routing is *crucial* to the near-optimality of DELAYEDOFF.

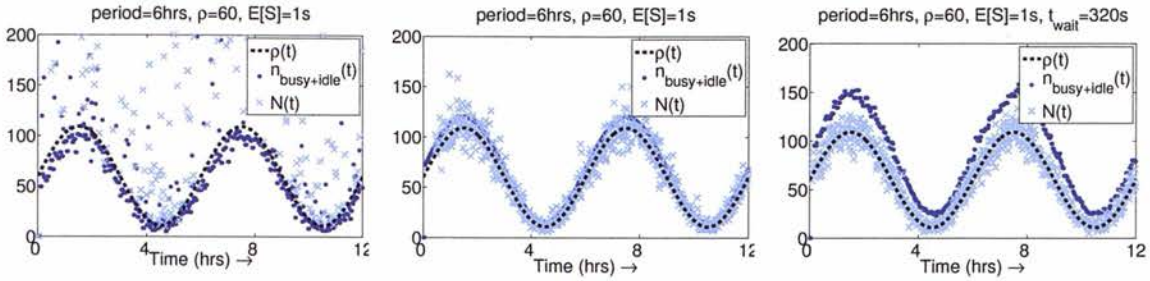
We will demonstrate the superiority of DELAYEDOFF by comparing it against two other policies, the first being INSTANTOFF, and the second being an offline, traffic-aware hypothetical policy, LOOKAHEAD. LOOKAHEAD runs the NEVEROFF policy, with n^* changing as a function of time. LOOKAHEAD smartly calculates $n^*(t)$ for each time t, given the $\rho(t)$ forecast. To do this, we use an idea proposed in [85], which is to compute what we will call the "effective load" at time t (referred to as the "offered load" in [85]), $\rho_{\text{eff}}(t)$, as:

$$\rho_{\text{eff}}(t) = \int_{-\infty}^{t} e^{-\mu(t-u)}\lambda(u)du.$$

The quantity $\rho_{\text{eff}}(t)$ denotes the mean number of jobs in the system at time t under the assumption that every job in the system can have its own server. The number of servers to have *on* at time t, $n^*(t)$, is then chosen to be $n^*(t) = \rho_{\text{eff}}(t) + \beta^*\sqrt{\rho_{\text{eff}}(t)}$, where β^* is given by (6.10).

Figure 6.4 illustrates the performance of INSTANTOFF, LOOKAHEAD and DELAYEDOFF in the case of a time-varying arrival pattern that resembles a sine curve with a period of 6 hours. In all the simulations, we set $\mathbf{E}[S] = 1sec$, and $T_{OFF} = 200secs$ (hence T_{OFF} is high). Figure 6.4(a) shows that INSTANTOFF auto-scales poorly as compared to the other policies, in particular $ERP^{InstantOff} \approx 6.8 \times 10^5 Watts \cdot sec$, with $\mathbf{E}[T] \approx 13.17sec$ and $\mathbf{E}[P] \approx 5.19 \times 10^4 Watts$. By contrast, LOOKAHEAD, shown in Figure 6.4(b), scales very well with the demand pattern. The ERP of LOOKAHEAD is $ERP^{LookAhead} \approx 1.64 \times 10^4 Watts \cdot sec$, with $\mathbf{E}[T] \approx 1.036sec$ and $\mathbf{E}[P] \approx 1.58 \times 10^4 Watts$. Unfortunately, as pointed out above, LOOKAHEAD requires knowledge of the future arrival pattern to be able to have $n^*(t)$ servers on at time t (in particular, it needs knowledge of the demand curve T_{OFF} units in advance). Thus, while LOOKAHEAD performs very well in a time-varying situation, it is not an online strategy, and is thus, not practical. Figure 6.4(c) illustrates the excellent auto-scaling capability of DELAYEDOFF for the sinusoidal arrival pattern. Here, $t_{wait} = 320s$ is chosen according to Rule of Thumb #4 presented later in this section. For the case in Figure 6.4(c), $ERP^{DelayedOff} \approx 1.89 \times 10^4 Watts \cdot sec$ with $\mathbf{E}[T] \approx 1.002sec$ and $\mathbf{E}[P] \approx 1.89 \times 10^4 Watts$. The ERP for DELAYEDOFF is only slightly higher than that of LOOKAHEAD, and far lower than that of INSTANTOFF. DELAYEDOFF slightly over-provisions capacity compared to LOOKAHEAD due to its traffic-oblivious nature. We verify this last observation analytically.

While analyzing DELAYEDOFF even under stationary traffic is a formidable challenge, we justify its excellent auto-capacity-scaling capabilities via the following modest proposition which suggests that under a Poisson arrival process with unknown intensity, DELAYEDOFF achieves near-optimal ERP. Thus, if the rate of change of the arrival rate is less than T_{OFF} (as was the case in Figure 6.4(c)), we expect DE-



(a) INSTANTOFF (b) LOOKAHEAD (c) DELAYEDOFF

Figure 6.4: Dynamic capacity provisioning capabilities of INSTANTOFF, LOOKAHEAD and DELAYEDOFF. The dashed line denotes the load at time t, $\rho(t)$, the dots denotes the number of servers that are busy or idle at time t, $n_{busy+idle}(t)$, and the crosses represent the number of jobs in the system at time t, $N(t)$.

LAYEDOFF to still achieve near-optimal ERP. This is because we are able to turn servers on before the queue builds up.

Proposition 6.1 *Consider a server farm with Poisson arrival process and Exponential service distribution. Let ρ denote the average load. Under* DELAYEDOFF *with MRB routing, the number of servers on is given by $\rho + \Theta(\sqrt{\rho})$, as $\rho \to \infty$.*

Proof: We first provide an alternate way of viewing the MRB routing. Consider a server farm with infinitely many servers, where we assign a unique rank to each server. Whenever there are n jobs in the server farm, they instantaneously move to servers ranked 1 to n. We now claim that there are m servers on at time t under MRB routing and DELAYEDOFF if and only if there are m servers on at time t in the alternate model under DELAYEDOFF. To see this, let the rank of servers at time t under MRB be defined by the last time they were *idle* (rank 1 server has been idle the shortest and so on). Once a server goes *idle* and gets rank n (thus the number of jobs in the system drops to $n-1$), its rank remains n until the number of jobs in the system increases to n.

Define the idle period for server $n+1$, $I(n)$, to be the time that elapses between the instant that the number of jobs in the system transitions from $n+1$ to n until it next reaches $n+1$. It is easy to see that the setup delay, T_{OFF} does not affect the distribution of $I(n)$. A rank $n+1$ server turns *off* when $I(n) > t_{wait}$. Analyzing DELAYEDOFF now reduces to analyzing the idle periods of servers in an $M/M/\infty$. Let $N(t)$ denote the number of jobs in an $M/M/\infty$ at time t. Iglehart [81] proved

that as $\rho \to \infty$, the process $X(t) = \frac{N(t)-\rho}{\sqrt{\rho}}$ converges to the solution $Y(t)$ of the following mean reverting Ornstein-Uhlenbeck process:

$$dY(t) = -mY(t)dt + \sigma dW(t); \quad m = \frac{1}{\mathbf{E}[S]}, \ \sigma^2 = \frac{2}{(\mathbf{E}[S])^2} \tag{6.15}$$

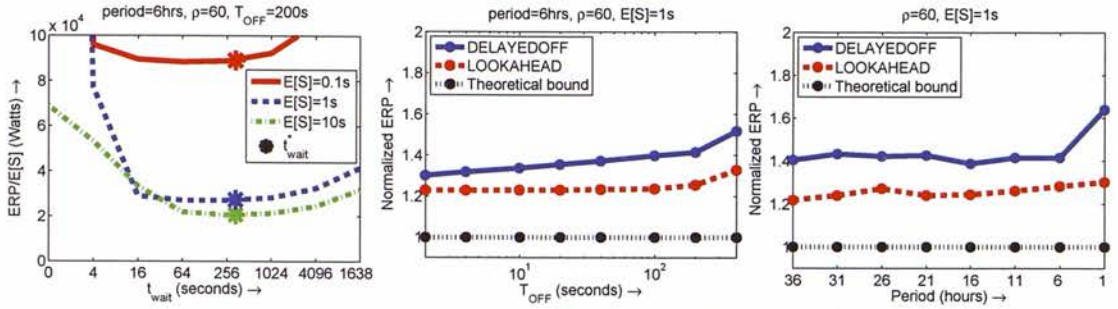
where $W(t)$ is the standard Brownian motion. The process $Y(t)$ mixes on a $\Theta(1)$ time scale, and the first passage time distributions for $Y(t)$, defined as $T_{u,v} = \inf\{t \geq 0 : Y(t) = v | Y(0) = u\}$ are only known via their Laplace transforms [57]. Thus while any exact analysis of DELAYEDOFF seems intractable, we can make the following conclusion: if we observe the system at a random point in time T, and find $N(T) = \rho + u\sqrt{\rho}$, the probability that the server ranked $\rho + v\sqrt{\rho}$ will be *on*, is exactly the probability that $t^* = \sup\{t \geq 0 : N(T-t) \geq \rho + v\sqrt{\rho}\} \leq t_{wait}$. However, t^* has the same distribution as $T_{u,v}$, and thus for any v, there is a constant probability that server with $\rho + v\sqrt{\rho}$ is *on*. [1] ■

We now address the question of choosing the optimal value of t_{wait}, which we denote as t^*_{wait}.

Rule of Thumb #4: Choosing t^*_{wait}.

As mentioned above, an exact analysis of DELAYEDOFF seems intractable since it involves first passage time distributions of the Ornstein-Uhlenbeck process, and thus analytically obtaining the optimal value of t_{wait} is equally intractable. However, empirically we have found that a good choice for the t_{wait} parameter is $t^*_{wait} \approx T_{OFF} \cdot \frac{P_{ON}}{P_{IDLE}}$. The rule of thumb is along similar lines as the power down strategy proposed in [84] and is based on an amortization argument. Once the server has wasted $P_{IDLE} \cdot t^*_{wait}$ units of power in *idle*, it amortizes the cost of turning the server on later and paying the penalty of $P_{ON} \cdot T_{OFF}$. While a reader familiar with work on powering down scheme might find our DELAYEDOFF policy not novel, we would like to point out a conceptual difference between the use of DELAYEDOFF in our work and in the prior literature. The prior literature uses DELAYEDOFF type schemes for stand-alone devices, obtaining constant factor sub-optimality. However, we are applying DELAYEDOFF to each device in a server farm, and are artificially creating an arrival process via MRB so as to make the idle periods of the servers highly variable. It is not surprising that the optimal value t^*_{wait} should be independent of ρ. As we mentioned above, by mapping the DELAYEDOFF policy to an Ornstein-Uhlenbeck process, for a fixed $\mathbf{E}[S]$, as we increase the arrival rate and hence ρ, the limit $Y(t)$ does not change, and thus the behavior of a server with rank $\rho + c\sqrt{\rho}$ would remain invariant to ρ for a fixed t_{wait}. However, what is interesting is that even as $\mathbf{E}[S]$ varies, our rule of thumb for t^*_{wait} holds as shown in Figure 6.5(a).

[1] In the published version of this work [67], there is an error in Theorem 2 and Corollary 1. While Lemma 7 on mean idle periods is indeed correct, it does not imply Theorem 2.



(a) Effect of $\mathbf{E}[S]$ on optimal t_{wait} (b) Effect of T_{OFF} on ERP (c) Effect of time period on ERP

Figure 6.5: (a) Verifying the accuracy of Rule of Thumb #4. The graph shows the effect of t_{wait} on ERP for the DELAYEDOFF policy, in the case of a sinusoidal demand curve, with average $\rho = 60$ and $\mathbf{E}[S] = 0.1, 1, 10s$. Different values of t_{wait} result in different ERP values. However, $t^*_{wait} = T_{OFF} \cdot \frac{P_{ON}}{P_{IDLE}} = 320s$ does well for all values of $\mathbf{E}[S]$. (b) The graph shows the difference in ERP of the DELAYEDOFF and LOOKAHEAD policies. The ERP values are normalized by the theoretical lower bound. (c) The graph shows the effect of decreasing the period of the sinusoidal demand curve on the ERP. Results suggest that decreasing the period of the demand curve does not effect the ERP significantly.

Figure 6.5(b) compares the ERP of DELAYEDOFF against the ERP of LOOKAHEAD for different T_{OFF} values. We normalize the ERP values with the theoretical lower bound of $\rho P_{ON} \cdot \mathbf{E}[S]$. Throughout the range of T_{OFF} values, we see that DELAYEDOFF, with t_{wait} chosen based on Rule of Thumb #4, performs within 10% of LOOKAHEAD, based on the ERP. The ERP of both, DELAYEDOFF and LOOKAHEAD are within 70-80% of the ERP values of the theoretical lower bound. Figure 6.5(c) shows the effect of decreasing the period of the sinusoidal demand curve on the ERP. We see that the ERP of DelayedOff increases as the period decreases, but this change is not very significant. Thus, we can expect DELAYEDOFF to perform well for time-varying demand patterns, as long as the rate of change of demand is not too high.

Trace-based simulation results: Thus far we have only looked at simulation results for arrival patterns that look like a sinusoidal curve. However, not all demand patterns are sinusoidal. We now consider a real-life demand pattern based on traces from the 1998 World Cup Soccer website, obtained from the Internet Traffic Archives [3]. The trace contains approximately 90 days worth of arrival data, with more than 1.3 billion arrivals. The data contains very bursty arrivals, with the arrival rate varying by almost a factor of 10, between periods of peak demand and low demand. In particular, the rate of change of arrival rate is sometimes much higher

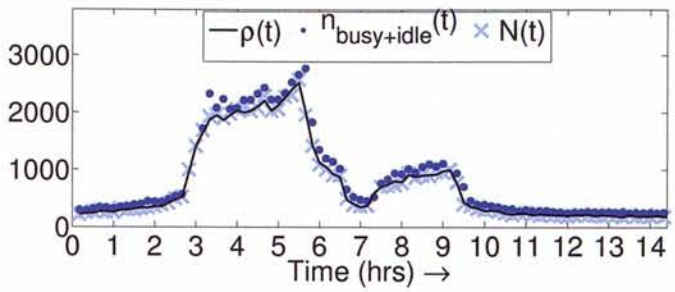


Figure 6.6: DELAYEDOFF simulation results based on a subset of arrival traces collected from the Internet Traffic Archives, representing 15 hours of bursty traffic during the 1998 Soccer world cup finals. Observe that DELAYEDOFF scales very well even in the case of bursty traffic.

than $T_{OFF} = 200s$. We run DELAYEDOFF on this trace, and compare our results against LOOKAHEAD. Throughout, we assume Exponentially distributed job sizes, with mean 1 second.

Figure 6.6 shows our simulation results for a subset of the arrival traces, corresponding to the most bursty traffic. We see that DELAYEDOFF (with optimally chosen $t_{wait} = 320s$) adapts extremely well to the time-varying traffic. In fact, over the entire duration of 90 days, the ERP of DELAYEDOFF was within 15% of the ERP of LOOKAHEAD. Thus, we conclude that DELAYEDOFF performs very well even in the case of unpredictable and bursty traffic.

6.6.2 An Index-based proxy for DelayedOff and applications

In this section we discuss a proxy policy for DELAYEDOFF that is easier is to implement, and can be easily modified for several application scenarios.

DELAYEDOFF-INDEX: In DELAYEDOFF-INDEX, we apriori assign static distinct ranks to all the servers in the server farm. Like DELAYEDOFF, under DELAYEDOFF-INDEX as well, we wait for a server to *idle* for some predetermined amount of time, t_{wait}, before turning it *off*. If the server gets a job to service in this period, its idle time is reset to 0. However, unlike MRB routing, if an arrival finds more than one servers *idle* on arrival, then it joins the highest ranked idle server (this server may not be the one that is farthest from turning *off*).

The intuition behind DELAYEDOFF-INDEX is the same as behind DELAYEDOFF – by

repeatedly sending the jobs to a preferred set of servers, we are creating variability in the idle periods. Empirically we have found that the performance of DELAYEDOFF and DELAYEDOFF-INDEX to be indistinguishable. However, the DELAYEDOFF-INDEX policy is practically appealing because it can be easily modified to adapt to many application scenarios, a couple of which we mention below:

Example 1: Dynamic capacity provisioning for heterogeneous server farms Our description and analysis of the DELAYEDOFF policy depended on the fact that the servers were homogeneous, while this is almost never true for data centers. However, by assigning static ranks to the servers, giving preference to faster and more energy efficient servers, and employing the DELAYEDOFF-INDEX policy, we can perform dynamic capacity scaling in heterogeneous compute environments.

Example 2: Dynamic capacity provisioning in cloud infrastructures An assumption that we have made throughout this chapter is that servers do not timeshare their capacity, and that further, a single job can exhaust the server's resources. As virtualization and cloud computing becomes ubiquitous, this assumption is broken. A typical virtual machine (VM) request might ask for 1GB of RAM and 2 GHz of processing capacity, while the physical servers are provisioned with several GBs of memory, and several cores. While it is hard to imagine how the DELAYEDOFF policy would extend to this application scenario, DELAYEDOFF-INDEX does the trick. A public cloud service like Amazon EC2 [2] allows users to choose from a small set of instance types. Based on the popularity of these instance types, physical servers could be partitioned into virtual servers, and these virtual servers can be assigned static ranks (ensuring that virtual servers in the same physical server get contiguous ranks). Now DELAYEDOFF-INDEX policy can be used to schedule VMs on these virtual servers. A physical server will be turned off once all its virtual servers have idled for t_{wait} units of time.

6.7 Summary and Open Questions

In this chapter we address the algorithmic question of energy-performance tradeoff in server farms, and utilized the metric of Energy-Response Time Product (ERP) to capture the aforementioned tradeoff. Via the first analysis of the ERP metric, we proves that a very small natural class of server farm management policies suffices to find the optimal or near-optimal policy. We furthermore develop rules of thumb for choosing the best among these policies given the workload and server farm specifications. The impact of our results is two-fold: (i) Our results eliminate the complexity of finding the optimal server farm management policy in a high-dimensional search

space, and (ii) Our analytical evaluation of the policies advocated in this chapter with respect to ERP can guide server designers towards developing a smaller set of *sleep* states with the most impact.

We first proved that for a single server under a Poisson arrival process, the optimal policy with respect to ERP is either (*a*) to always keep the server *on* or *idle* (NEVEROFF), or (*b*) to always turn a server *off* when *idle* and to turn it back *on* when work arrives (INSTANTOFF), or (*c*) to always put the server in some *sleep* state when *idle* (SLEEP). Next, based on analysis and numerical experiments, we conjecture that for a multi-server system under a Poisson arrival process, the multi-server generalizations of NEVEROFF, INSTANTOFF and SLEEP suffice to find a near-optimal policy. Finally we consider the case of a time-varying demand pattern and propose a simple traffic oblivious policy, DELAYEDOFF, which turns servers on when jobs arrive, but waits for a specific amount of time, t_{wait}, before turning them *off*. Through a clever routing policy, DELAYEDOFF achieves asymptotically near-optimal performance in simulations for a stationary Poisson arrival process with an unknown arrival rate, as the load becomes large. We also proposed a variant, DELAYEDOFF-INDEX, which allows extending DELAYEDOFF to dynamic capacity provisioning in heterogeneous server farms and in cloud computing infrastructure.

Open Problems: In order to prove the optimality results in this chapter, we have made some assumptions: (i) The servers are interchangeable (any job can serve on any server), (ii) The server farm is homogeneous, (iii) The job-sizes are Exponentially distributed. If some or all of these assumptions were to be relaxed, then our optimality results might look different. Proving optimality results without the above assumptions constitutes ongoing work. Perhaps most important extension would be energy-management policies under I/O bound workloads, which severely limit the load balancing flexibility. Proving guarantees on performance of traffic-oblivious policies under some smoothness conditions on the demand pattern is also a theoretically challenging goal.

6.A Proof of Theorem 6.1

Proof of Lemma 6.1: We first note that if the server is in the *on* state and there is work in the system, then the optimal policy never transitions into a *sleep* state. Suppose, by contradiction, an optimal policy π transitioned into a *sleep* state at time t_0 with work in the queue and then later transitioned through some *sleep* state until finally transitioning to the *on* state at time t_1. We could transform this into a policy π' with equivalent power consumption, but lower mean response time by deferring the powering down until all the work present in the system at t_0 has finished (say at

t_2), and then transitioning through the same *sleep* states as π, finally transitioning to the *on* (or *idle*) state at time $t_2 + (t_1 - t_0)$.

Next, we prove that the only instants at which an optimal policy takes actions will be job completions, job arrivals, or when the server finishes transition from a low power state to a higher power state. Here we assume that once a transition to a *sleep*, *idle* or *on* state has been initiated from a lower power state, it can not be interrupted. We have already argued that no actions happen during a busy period when the server is in the *on* state. Therefore to prove that control actions only happen at the claimed events, it remains to show that actions do not occur while the server is in *idle* or *sleep* states (and not in transition or *on*) and an arrival has not occurred. To achieve this, it suffices to show that there exists a Markovian optimal control for the ERP metric. Note that $\mathbf{E}[T] = \lim_{T\to\infty} \frac{1}{\lambda T}\mathbf{E}\Big[\int_{t=0}^{T} N(t)dt\Big]$ and $\mathbf{E}[P] = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\Big[\int_{t=0}^{T} P(t)\Big]$, where $N(t)$ and $P(t)$ denote the number of jobs and power consumption, respectively, at time t. Thus the optimal decision at time t depends only on the future evolution of the system, and not on the finite history in $[0, t]$. (Note that these statements are not true if we replace $\mathbf{E}[T]$ and $\mathbf{E}[P]$ by their discounted versions, e.g. $\mathbf{E}[P_\gamma] = \int_{t=0}^{\infty} \gamma^t P(t)dt$ for some $0 < \gamma < 1$.) By the memoryless property of the Poisson arrival process, the claim follows.

Finally, we will show that once a policy goes into a *sleep* state when the server goes *idle*, the only other state it will transition to next is *on*. To see this, suppose the server went into *sleep* state S_i. Now, the server will not go into *sleep* state S_j for $j > i$ (and hence to a state with lower power) on a job arrival, otherwise it would have been better to transition to S_j when the server first went *idle*. If the server transitions to a *sleep* state S_k for $k < i$ (thus a state with higher power) but not the *on* state, and later transitions to the *on* state, it would instead have been better to transition directly to the *on* (since the transition times are the same by the *Model Assumptions*), finish processing the work and then transition to state S_k instantaneously.

So far, we have argued that the optimal policy must (i) immediately transition to *idle* or a *sleep* state when the work empties (recall that we have assumed these transitions to be instantaneous), (ii) immediately transition to the *on* state on some subsequent arrival, and (iii) is Markovian. However, the optimal control need not necessarily be a deterministic function of the current state. We therefore use p_i and q_{ij} to denote the class of possible optimal control policies Π_{mixed}. ∎

Proof of Lemma 6.2: The proof proceeds via renewal reward theory. We define a renewal cycle for the server as the time from when a server goes *idle* (has zero work),

until it next goes *idle* again. Thus we can express:

$$\mathbf{E}[T] = \frac{\mathbf{E}[\text{total response time per cycle}]}{\mathbf{E}[\text{number of jobs per cycle}]} \quad ; \quad \mathbf{E}[P] = \frac{\mathbf{E}[\text{total energy per cycle}]}{\mathbf{E}[\text{duration per cycle}]}.$$

Now consider a specific case, where the server goes into *sleep* state S_i on becoming *idle*, and starts transitioning to the *on* state when n_i jobs accumulate. There can be more arrivals while the server is turning on. We denote the number of arrivals during transition from S_i by X_i, and note that X_i is distributed as a Poisson random variable with mean λT_{S_i}. Thus, after the server turns on, it has $n_i + X_i$ jobs in the queue, and thus the time until the server goes *idle* is distributed as a sum of $n_i + X_i$ busy periods of an $M/M/1$ system. The sum of the response times of jobs that are server during this renewal cycle has two components:
1. Sum of waiting times of all jobs before the server turns on (term 1 below): The waiting time of the jth of the first n_i jobs is $\sum_{k=j+1}^{n_i} T_\lambda(k) + T_{S_i}$, where $\{T_\lambda(\cdot)\}$ are *i.i.d.* Exp(λ) random variables, and $T_\lambda(k)$ denotes the time between the $(k-1)$st and kth arrival of the cycle. By the properties of the Poisson arrival process, the (unordered) waiting time of each of the X_i jobs is an independent $U([0, T_{S_i}])$ random variable. Adding an taking expectation, we get the term 1 as shown below in (6.16).
2. Sum of the response times from when the server turns on until it goes idle (term 2 below): Since the sum of response time of the jobs that are served during the renewal cycle is the same for any non-preemptive size-independent scheduling policy, we will find it convenient to schedule the jobs as follows: We first schedule the first of $n_i + X_i$ arrivals and do not schedule any of the $n_i + X_i - 1$ remaining jobs until the busy period started by the first job completes. Then we schedule the second of the $n_i + X_i$ jobs, holding the remaining jobs until the busy period started by this job ends, and so on. The sum of the response times is thus given by the sum of response times in $n_i + X_i$ i.i.d. $M/M/1$ busy periods, and the additional waiting time experienced by the initial $n_i + X_i$ arrivals. By renewal theory, the expectation of the sum of response times of the jobs served in an $M/M/1$ busy period with arrival rate λ and service rate μ is given by the product of the mean number of jobs served in a busy period $\left(\frac{1}{1-\frac{\lambda}{\mu}}\right)$ and the mean response time per job $\left(\frac{1}{\mu-\lambda}\right)$. This gives the first component of term 2. The additional waiting time of the jth of the $n_i + X_i$ initial arrivals due to our scheduling policy is given by the sum of durations of $j-1$ $M/M/1$ busy periods, each of expected length $\frac{1}{\mu-\lambda}$. Adding this up for all the $n_i + X_i$ jobs and taking expectation, we get the second component of term 2.

$$\underbrace{n_i\left(\frac{n_i-1}{2\lambda}+T_{S_i}\right)+\mathbf{E}[X_i]\frac{T_{S_i}}{2}}_{\text{term 1}}+\underbrace{\frac{1}{1-\rho}\cdot\frac{n_i+\mathbf{E}[X_i]}{\mu-\lambda}+\mathbf{E}\left[\frac{(n_i+X_i)(n_i+X_i-1)}{2(\mu-\lambda)}\right]}_{\text{term 2}} \tag{6.16}$$

$$=\frac{1}{1-\rho}\left(\frac{n_i+\mathbf{E}[X_i]}{\mu-\lambda}+\left[n_iT_{S_i}+\frac{n_i(n_i-1)}{2\lambda}+\frac{\lambda T_{S_i}^2}{2}\right]\right)\;=\;\frac{r_{in_i}}{1-\rho}$$

The final expression in (6.1) is obtained by combining the above with the renewal reward equation, and noting that the mean number of jobs served in this renewal cycle is given by $\frac{n_i+\mathbf{E}[X_i]}{1-\rho}$.

$$\mathbf{E}[T]=\frac{\mathbf{E}[\text{total response time per cycle}]}{\mathbf{E}[\text{number of jobs per cycle}]}=\frac{\sum_{i=0}^{N}p_i\sum_{n_i=1}^{\infty}q_{in_i}\frac{r_{in_i}}{1-\rho}}{\sum_{i=0}^{N}p_i\sum_{n_i=1}^{\infty}q_{in_i}\frac{n_i+\lambda T_{S_i}}{1-\rho}}$$
$$=\frac{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}r_{ij}}{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}(j+\lambda T_{S_i})}$$

The proof for $\mathbf{E}[P]$ is analogous. The duration of a cycle is composed of three different times:
1. Time spent waiting for n_i jobs to queue up: The expected duration is $\frac{n_i}{\lambda}$, with expected total energy consumed given by $\frac{n_i}{\lambda}P_{S_i}$.
2. Time to wake up the server: This is T_{S_i}, with total energy consumed by the server during this time as $T_{S_i}P_{ON}$.
3. (n_i+X_i) busy periods: The expected time it takes for the server to go idle again is the expected duration of n_i+X_i busy periods, given by $\frac{n_i+\lambda T_{S_i}}{\mu-\lambda}$ with total energy consumed being $\frac{n_i+\lambda T_{S_i}}{\mu-\lambda}P_{ON}$.

Thus, we have:

$$\mathbf{E}[P]=\frac{\mathbf{E}[\text{total energy per cycle}]}{\mathbf{E}[\text{duration per cycle}]}=\frac{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}\left[\frac{j}{\lambda}\cdot P_{S_i}+T_{S_i}\cdot P_{ON}+\frac{j+\lambda T_{S_i}}{\mu-\lambda}\cdot P_{ON}\right]}{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}\left[\frac{j}{\lambda}+T_{S_i}+\frac{j+\lambda T_{S_i}}{\mu-\lambda}\right]}$$
$$=\frac{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}\left(j(\rho P_{ON}+(1-\rho)P_{S_i})+\lambda T_{S_i}P_{ON}\right)}{\sum_{i=0}^{N}p_i\sum_{j=1}^{\infty}q_{ij}\left(j+\lambda T_{S_i}\right)}.$$

■

Proof of Lemma 6.3: To prove that the optimal strategy is pure, we only need to note that the expressions for both the mean response time and average power are

of the form

$$\mathbf{E}[T] = \frac{q_1 t_1 + \ldots + q_n t_n}{q_1 m_1 + \ldots + q_n m_n}; \quad \mathbf{E}[P] = \frac{q_1 u_1 + \ldots + q_n u_n}{q_1 m_1 + \ldots + q_n m_n},$$

where n is the number of pure strategies that the optimal strategy is randomizing over. for some discrete probability distribution $\{q_1 \ldots, q_n\}$. We will show that when $n = 2$, the optimal strategy is pure, and the proof will follow by induction on n. For $n = 2$, we consider $\mathbf{E}[T]$ and $\mathbf{E}[P]$ as a function of q_1 over the extended domain $q_1 \in (-\infty, +\infty)$, and show that there is no local minima of $\mathbf{E}[T] \cdot \mathbf{E}[P]$ in $q_1 \in (0, 1)$. Further, note that both $\mathbf{E}[T]$ and $\mathbf{E}[P]$ are of the form $a + \frac{b}{c+dq_1}$ for some constants a, b, c, d. While the lemma would trivially follow if the product of $\mathbf{E}[T]$ and $\mathbf{E}[P]$ were a concave function of q, this is not true in our case because one/both of $\mathbf{E}[T]$ and $\mathbf{E}[P]$ may be convex, and hence we proceed through a case analysis:

Case 1: Both $\mathbf{E}[T]$ and $\mathbf{E}[P]$ are increasing or decreasing in q_1, except for a shared discontinuity at $q_1 = \frac{m_2}{m_2 - m_1}$. In this case, trivially, $\mathbf{E}[T]\mathbf{E}[P]$ is also increasing/decreasing in the interval $q_1 \in [0, 1]$ as both the functions are positive in this interval, and thus the minimum of $\mathbf{E}[T] \cdot \mathbf{E}[P]$ is either at $q_1 = 0$ or at $q_1 = 1$.

Case 2: One of $\mathbf{E}[T]$ and $\mathbf{E}[P]$ is an increasing function and the other is a decreasing function of q_1 (except for the shared discontinuity at $q_1 = \frac{m_2}{m_2 - m_1}$). In this case, as $q_1 \to \frac{m_2}{m_2 - m_1}$, $\mathbf{E}[T] \cdot \mathbf{E}[P] \to -\infty$. Second, due to the form of $\mathbf{E}[T]$ and $\mathbf{E}[P]$, it is easy to see that their product has at most one local optimum. Finally, we can see that as $q_1 \to \pm\infty$, $\mathbf{E}[T]\mathbf{E}[P] \to \frac{(t_1 - t_2)(m_1 - m_2)}{(u_1 - u_2)^2}$, which is finite. Combining the previous three observations, we conclude that there is no local minima in the interval $q_1 \in (0, 1)$. In other words, in the interval $q_1 \in [0, 1]$, the minimum is achieved at either $q_1 = 0$, or $q_1 = 1$. The inductive case for n follows by considering only two variables, q_n and q', where q' is a linear combination of $q_1, q_2, \ldots, q_{n-1}$, and applying the inductive assumption. ∎

Proof of Lemma 6.4: We now know that the optimal power down strategy is of the following form: the server goes into a fixed *sleep* state, S_i, on becoming *idle*. It then waits for some deterministic n_i arrivals before transitioning into the *on* state. We will show that under optimality, $n_i = 1$. The basic idea is to minimize the product of Eqs. (6.1) and (6.3). We first show that if $m = \lambda T_{S_i} > 1$, then the policy where the server goes to *idle* state (recall $T_{IDLE} = 0$) has a lower $\mathbf{E}[T]\mathbf{E}[P]$ than going into *sleep* state S_i with *any* n_i. Thus $\lambda T_{S_i} < 1$ is a necessary condition for optimality of *sleep* state S_i:

Lemma 6.6 *When $\lambda T_{S_i} \geq 1$,* NEVEROFF *has a lower ERP than a policy involving sleep state S_i with any $n_i > 0$.*

Proof: We will prove the above fact by upper bounding P_{IDLE} by P_{ON}, which only makes the ERP of NEVEROFF worse. Under the above assumption, the ERP values

for NEVEROFF and $n_i = n$ are given by:

$$\mathbf{E}[T] \cdot \mathbf{E}[P]|_{NeverOff} = \frac{P_{ON}}{\mu - \lambda}$$

$$\mathbf{E}[T] \cdot \mathbf{E}[P]|_{n_i=n} = \left[\frac{\frac{n+m}{\mu-\lambda} + \frac{1}{\lambda}\left(nm + \frac{n^2-n}{2} + \frac{m^2}{2}\right)}{(n+m)^2}\right] \cdot (\rho n + m)P_{ON}, \quad \text{where } m = \lambda T_{S_i}$$

Cross-multiplying the terms, we can say that

$$\begin{aligned}
& \mathbf{E}[T] \cdot \mathbf{E}[P]|_{NeverOff} < \mathbf{E}[T] \cdot \mathbf{E}[P]|_{n_i=n} \\
\iff \quad & \rho(n+m)^2 - \left[\rho(n+m) + (1-\rho)\left(\frac{(m+n)^2}{2} - \frac{n}{2}\right)\right](\rho n + m) < 0 \\
\iff \quad & \rho^2\left[-n(m+n) + n\left(\frac{(m+n)^2}{2} - \frac{n}{2}\right)\right] + \rho\left[n(m+n) + (m-n)\left(\frac{(m+n)^2}{2} - \frac{n}{2}\right)\right] \\
& -m\left[\frac{(m+n)^2}{2} - \frac{n}{2}\right] < 0 \qquad (6.17)
\end{aligned}$$

It is easy to check that the LHS of Eq. (6.17) is negative at $\rho = 0$, and is zero at $\rho = 1$. Since this expression is quadratic, it suffices to show that the derivative of the above at $\rho = 1$ is positive. This would imply that the curve lies below X-axis in the interval $\rho \in [0, 1)$ for $m, n > 1$. The derivative at $\rho = 1$ is given by:

$$\begin{aligned}
& -n(m+n) + (m+n)\left[\frac{(m+n)^2}{2} - \frac{n}{2}\right] \\
& = (m+n)\left[\frac{(m+n)^2}{2} - \frac{3n}{2}\right]
\end{aligned}$$

For $m, n > 1$, it is easy to see that $(m+n)^2 > 3n$, and hence the derivative at $\rho = 1$ is indeed positive. ■

Next, we show that when $\lambda T_{S_i} < 1$, the optimal value of n_i is in fact $n_i = 1$. We already know that λT_{S_i} is a necessary condition for the optimality of the pure policy involving S_i, and we thus show that in this case the optimal value of $n_i = 1$. Thus, the optimal policy involving S_i must be SLEEP(S_i).

Lemma 6.7 *When $\lambda T_{S_i} < 1$, $n_i = 1$ is the optimal policy involving sleep state S_i.*

Proof: Since we know from Lemma 6.3 that the optimal n_i will be at positive integral values, we can create alternate functions for $\mathbf{E}[P]$ and $\mathbf{E}[T]$ that agree at integral points and have continuous derivatives. If optimal value obtained from these

continuous functions is indeed $n_{ON} = 1$ then we are done. Let $m = \lambda T_{S_i}$. Further, we assume $P_{S_i} = 0$ as a higher P_{S_i} only favors a lower n_i. These smooth functions are given by:

$$\begin{aligned}
\mathbf{E}[T] &= \frac{\frac{x+m}{\mu-\lambda} + \frac{1}{\lambda}\left[x \cdot m + \frac{x^2-x}{2} + \frac{m^2}{2}\right]}{x+m} \\
&= \frac{1}{\mu-\lambda} + \frac{x+m}{2\lambda} - \frac{x}{2\lambda(x+m)} \\
&= \frac{1}{\mu-\lambda} - \frac{1}{2\lambda} + \frac{x+m}{2\lambda} + \frac{m}{2\lambda(x+m)} \\
\mathbf{E}[P] &= \rho \cdot P_{ON} + (1-\rho)\frac{\lambda T_{S_i} \cdot P_{ON}}{x + \lambda T_{S_i}} \\
&= \frac{(\rho x + m)P_{ON}}{x+m} \\
&= \rho \cdot P_{ON} + \frac{m(1-\rho)P_{ON}}{x+m}
\end{aligned}$$

The product $\mathbf{E}[T] \cdot \mathbf{E}[P]$ can be written as $ax + b + \frac{c}{x+m} + \frac{d}{(x+m)^2}$. Therefore, there are 3 local optima, and the second derivative changes sign only once. Further, the curve approaches $-\infty$ when $x \to -\infty$, $+\infty$ when $x \to +\infty$, and again $+\infty$ when $x \to -m$. Further, as $x \to -\infty$, the sign of the second derivative is $-sgn(c)$, and as $x \to +\infty$, the sign of the derivative is $sgn(c)$. In either case, since the curve is convex for some interval in $(-\infty, 0]$, $+\infty$ at $x = -\frac{1}{m}$, and the second derivative changes sign only once, proving that the derivative of $\mathbf{E}[T] \cdot \mathbf{E}[P]$ is positive at $x = 1$ suffices to show that there is no local minima for $x > 1$. (This is because in $[0, +\infty)$, the curve is either convex decreasing at $x = 0$ and then switches to concave, or is convex in the entire interval.)

Taking derivative of the log of the product we get:

$$\frac{\partial}{\partial x}\log\left(\mathbf{E}[T]\mathbf{E}[P]\right) = \frac{1 + \frac{(1-\rho)}{\rho}\left[m + \frac{2x-1}{2}\right]}{x + m + \frac{(1-\rho)}{\rho}\left[mx + \frac{x^2-x}{2} + \frac{m^2}{2}\right]} + \frac{\rho}{\rho x + m} - 2\frac{1}{x+m}$$

$$\implies \frac{\partial}{\partial x}\log\left(\mathbf{E}[T]\mathbf{E}[P]\right)\Big|_{x=1} = \frac{1 + \frac{(1-\rho)}{\rho}\left[m + \frac{1}{2}\right]}{1 + m + \frac{(1-\rho)}{\rho}\left[m + \frac{m^2}{2}\right]} + \frac{\rho}{\rho + m} - 2\frac{1}{1+m}$$

Now,

$$\frac{\partial}{\partial x} \log\left(\mathbf{E}[T]\mathbf{E}[P]\right)\Big|_{x=1} > 0 \iff$$
$$\left[\rho + (1-\rho)(m+\frac{1}{2})\right] \cdot [(\rho+m)(1+m)]$$
$$- \left[\rho(1+m) + (1-\rho)(m+\frac{m^2}{m})\right] \cdot [\rho(1-m)+2m] > 0$$

The last inequality involves a quadratic in ρ on LHS. It is easy to check that when $\rho = 0$ and $m < 1$, the quadratic is positive. Further, when $\rho = 1$, the value of the quadratic polynomial is 0. Thus it suffices to show that the slope of the above quadratic at $\rho = 1$ is negative (when $m < 1$). This would imply that the above inequality is satisfied in the interval $\rho \in [0, 1)$. Indeed, it can be checked that the derivative at $\rho = 1$ is given by $-\frac{m^3}{2} - \frac{1}{2} < 0$. Thus, we have proved that $n_i > 1$ is not optimal for $m < 1$. Thus, $n_i = 1$ is optimal. ■ ■

6.B Justification for Conjecture 6.1

The core problem is in coming up with a tight lower bound for $\mathbf{E}[T]\mathbf{E}[P]$ for the optimal policy. We have a trivial lower bound of $\mathbf{E}[T] \geq \mathbf{E}[S]$, and $\mathbf{E}[P] \geq \rho P_{ON}$. However, this is very loose when ρ is small and T_{OFF} is large.

To illustrate the **key ideas** in our approach to obtaining the lower bound, we begin by considering the case where there are no sleep states. The first idea we use is to give the optimal policy additional capability. We do so by allowing the optimal policy to turn a server on from *off* instantaneously (zero setup time). Consequently, each server is either *on* (busy), *idle*, or *off*. However there is still an energy penalty of $P_{ON}T_{OFF}$. Secondly, we use an accounting method where we charge the energy costs to the jobs, rather than to the server. Thus, each job contributes towards the total response time cost and to the total energy cost. Thirdly, we obtain a lower bound by allowing the optimal policy to choose the state it wants an arrival to see independently for each arrival. This allows us to decouple the decisions taken by the optimal policy in different states. We make this last point clearer next.

An arrival that finds the n jobs in the system (excluding itself) could find the system in one of the following states:

1. At least one server is *idle*: Here, the optimal policy would schedule the arrival on the *idle* server. In this case, we charge the job $\mathbf{E}[S]$ units for mean response

time. Further, the server would have been *idle* for some period before the arrival, and we charge the energy spent during this idle period, as well as the energy to serve the arrival, to the energy cost for the job. However, if under the optimal policy, there is an *idle* server when the number of jobs increases from n to $n+1$, there must have been a server *idle* when the number of servers last went down from $n+1$ to n. Furthermore, some server must have remained *idle* from then until the new arrival which caused the number of jobs to go to $n+1$ (and hence there were no jobs in the queue during this period). Thus, this idle period is exactly the idle period of an $M/M/n+1$ with load ρ, denoted by $I(n)$, where the idle period is defined as the time for the number of jobs to increase from n to $n+1$.

2. No server is *idle*, arrival turns on an *off* server: Here, we charge the arrival $\mathbf{E}[S]$ units for mean response time, and $P_{ON}\mathbf{E}[S] + T_{OFF}P_{ON}$ for energy.

3. No server is *idle*, arrival waits for a server to become idle: This case is slightly non-trivial to handle. However, we will lower bound the response time of the job by assuming that the arrival found n servers busy with the n jobs. Further, until a departure, every arrival turns on a new server and thus increases the capacity of the system. Thus, this lower bound on queueing time can be expressed as the mean time until first departure in an $M/M/\infty$ system starting with n jobs. We denote this by $D(n)$. The energy cost for the job will simply be $P_{ON}\mathbf{E}[S]$.

We will give the optimal strategy the capability to choose which of the above 3 scenarios it wants for an arrival that occurs with n jobs in the system. Since the response time cost of scenario 1 and 2 are the same, only one of them is used, depending on whether $P_{IDLE}\mathbf{E}[I(n)] > P_{ON}T_{OFF}$ or not. Let $P_{waste}(n) = \min\{P_{IDLE}\mathbf{E}[I(n)], P_{ON}T_{OFF}\}$. Let q_n denote the probability that the optimal policy chooses the best of scenarios 1 and 2 for an arrival finding n jobs in the system, and with probability $1-q_n$ it chooses scenario 3. Since we are interested in obtaining a lower bound, we will further assume that the probability of an arrival finding n jobs in the system, p_n, is given by the pdf of a Poisson random variable with mean ρ, which is indeed a stochastic lower bound on the stationary number of jobs in the system. We thus obtain the following optimization problem:

$$\begin{aligned}\mathbf{E}\left[T^{OPT}\right]\mathbf{E}\left[P^{OPT}\right] &\geq \lambda \min_{\{q_n\}} \left(\mathbf{E}[S] + \sum_n p_n(1-q_n)\mathbf{E}[D(n)]\right)\left(P_{ON}\mathbf{E}[S] + \sum_n p_n q_n P_{waste}(n)\right) \\ &\geq \lambda \min_{\{q_n\}} \left(\sum_n p_n \sqrt{(\mathbf{E}[S] + (1-q_n)\mathbf{E}[D(n)])(P_{ON}\mathbf{E}[S] + q_n P_{waste}(n))}\right)^2\end{aligned}$$

(By Cauchy-Schwarz inequality)

$$= \lambda \left(\sum_n p_n \sqrt{\min \{P_{ON}\mathbf{E}[S] + P_{waste}(n), P_{ON}(\mathbf{E}[S] + D(n))\}} \right)^2$$

The last equality was obtained by observing that the minimum occurs at $q_n = 0$ or $q_n = 1$. The rest of the argument is numerical. We have written a program that computes the above lower bound for a given ρ, T_{OFF}, P_{IDLE} and P_{ON} values. We then compare it against the cost of the NEVEROFF with optimal n^*, and against the following upper bound on the cost of INSTANTOFF: $\lambda P_{ON} (\mathbf{E}[S] + T_{OFF})^2$. This upper bound is obtained by forcing every job to run on the server that it chooses to *setup* on arrival. For each value of ρ, we then search for the T_{OFF} value that maximizes the ratio of the cost of the best of NEVEROFF and INSTANTOFF to the above lower bound, and bound the relative performance of the best of NEVEROFF and INSTANTOFF against the theoretical optimal as a function of ρ and the ratio $\frac{P_{IDLE}}{P_{ON}}$.

The proof for Theorem 6.1 with sleep states now proceeds along the same lines as we have described above. For Theorem 6.1, we have $P_{S_i} > 0$, so the optimal policy does not have infinite servers to work with. Let us say the optimal policy works with N servers. We first add a cost of $\frac{NP_{S_i}}{\lambda}$ to the energy cost of all jobs, and get back a system with $P_{ON} \leftarrow P_{ON} - P_{S_i}$ and $P_{IDLE} \leftarrow P_{IDLE} - P_{S_i}$. We now have the following three scenarios an arrival that sees n jobs in the system could encounter:

1. At least one server is *idle*: In this case we must have $n < N$, and the response time is $\mathbf{E}[S]$ and the energy penalty is $(P_{ON} - P_{S_i})\mathbf{E}[S] + (P_{IDLE} - P_{S_i})I(n)$.

2. Arrival finds no *idle* servers and there is a sleeping server: In this case we may turn on a sleeping server and the energy penalty is $(P_{ON} - P_{S_i})\mathbf{E}[S] + P_{ON}T_{S_i}$. However, the new arrival may be jumping ahead of jobs in the queue. There are at least $(n - (N-1))^+$ of them.

3. Arrival finds no *idle* server and the job waits: In this case the response time is given by $\mathbf{E}[S] + D(n)$ where $D(n)$ denotes the time until first departure in an $M/M/N$ starting with n jobs. The energy cost is just $(P_{ON} - P_{S_i})\mathbf{E}[S]$.

As before, only one of scenarios 1 or 2 is used, and we define

$$P_{waste} = \min\{P_{ON}T_{S_i}, (P_{IDLE} - P_{S_i})I(n)\mathbf{1}_{n<N}\}.$$

Our optimization problem then is:

$$\min_{\substack{\{p_n\},\{q_n\} \\ \{p_n\} \geq_{st} Poisson(\rho)}} \lambda \left(\mathbf{E}[S] + \sum_{i=0}^{\infty} p_i (1 - q_i) \mathbf{E}[D(n)] \right) \left(\frac{NP_{S_i}}{\lambda} + (P_{ON} - P_{S_i}) \mathbf{E}[S] + \sum_{i=0}^{\infty} p_i q_i P_{waste}(i) \right)$$

The problem with using the above approach is the following: consider a *sleep* state with P_{S_i} very close to P_{IDLE} and $T_{S_i} \ll 1$. In this case, the above problem is optimized for $N = \rho + 1$ (that too because we have a lower bound on N) as follows: for every job, we assume there is a sleeping server which we can wake up for negligible power penalty and negligible response time penalty. Thus we have the following gap in the current accounting method: once there are at least N jobs in the system, a new arrival is allowed to jump ahead of someone in the queue - so either we have jobs in queue, or we have an *idle* server which we are not taking into account. We may try to get around this by not charging jobs for response time when they queue up, but instead charge them for the number of jobs they see. However, we need to argue that the job either pays the penalty of turning on a server, or of waiting. However, we can't charge the job for waiting if we are also charging jobs for queue lengths they see.

To get around this problem, we will charge every job $\mathbf{E}[S]$ units for their service time, $\alpha < 1$ times the cost of the queue lengths they see, and $1 - \alpha$ times the cost of their waiting time. We can then optimize over α to get a good lower bound. We now show the steps in detail:

1. At least one server is *idle*: In this case we must have $n < N$, and the response time is $\mathbf{E}[S]$ and the energy penalty is $(P_{ON} - P_{S_i})\mathbf{E}[S] + (P_{IDLE} - P_{S_i})I(n)$.

2. Arrival finds no *idle* servers and there is a sleeping server: In this case we may turn on a sleeping server and the energy penalty is $(P_{ON} - P_{S_i})\mathbf{E}[S] + P_{ON}T_{S_i}$. The response time penalty is $\mathbf{E}[S] + \frac{1}{\lambda}\alpha(\max\{0, n - N + 1\})$.

3. Arrival finds no *idle* server and the job waits: In this case the response time is given by $\mathbf{E}[S] + (1 - \alpha)D(n)$. The energy cost is just $(P_{ON} - P_{S_i})\mathbf{E}[S]$.

Let $q_{n,1}$ be the probability that scenario 1 is used when there are n jobs, and so on. Our optimization problem then is:

$$\max_{\alpha} \min_{\{q_{n,1},q_{n,2},q_{n,3}\}|\{p_n\} \geq_{st} Poisson(\rho)} \lambda \left(\mathbf{E}[S] + \sum_{i=0}^{\infty} p_i (q_{i,2} \alpha \frac{(i - N + 1)^+}{\lambda} + q_{i,3}(1 - \alpha)\mathbf{E}[D(i)] \right)$$
$$\cdot \left(\frac{NP_{S_i}}{\lambda} + (P_{ON} - P_{S_i})\mathbf{E}[S] + \sum_{i=0}^{\infty} p_i (q_{i,1}(P_{IDLE} - P_{S_i})\mathbf{E}[I(i)] + q_{i,2}P_{ON}T_{S_i}) \right)$$

We note that the optimal values for the $q_{i,k} \in \{0, 1\}$. Applying Cauchy-Schwarz, we reduce this to a term-by-term minimization, and then we maximize over α.

Chapter 7

Summary

Queueing theory has traditionally been used for performance evaluation and optimization in application areas such as telecommunications systems, bandwidth sharing systems, inventory and production managament, and call centers. In this thesis, we have argued that queueing theory can also provide answers to the questions faced by designers of today's computing server farms. However, since the workloads and architectures of modern computing server farms are very different from telecommunications and manufacturing systems, new analytical tools and models must be developed for queueing theory to be relevant to computing applications.

The work presented in this thesis should be appealing to theoreticians, as well as system designers. From the theoretical perspective, we have addressed many challenging open problems, and raised questions which would lead to a deeper understanding of queueing systems. From the practical perspective, we have proposed new algorithms for resource management, and provided insights into the behavior of the queueing models considered. While not complete solutions in themselves, we hope that the policies proposed in this thesis will be combined with profiling and control theory techniques to solve the problems faced by systems designers. We briefly recapitulate the major contributions and open problems from the thesis below.

7.1 Theoretical Contributions

Moments-based bounds for solutions of Stochastic recursive sequences

In Chapter 2, we demonstrated the insufficiency of existing analyses of the classical $M/G/k$ queue by proving that no approximation for the mean waiting time that

only uses the first two moments of the service distribution can be accurate when the variance of the service distribution is large. Thus, in Chapter 3, we began with the goal of obtaining bounds on the mean sojourn time of $M/G/k$ via higher moments of the service distribution. We presented two more examples of queueing systems for which we could prove, in appropriate light-traffic asymptotic regimes, that the mean sojourn time was extremized by certain principal representations of these moment constraints, and achieved this via a link to the Markov-Krein Theorem and theory of Tchebycheff systems. These links have been established for the $GI/M/k$ model, but are not transparent for queueing systems such as the $M/G/k$. As a new research area, we propose to formally investigate these connections by considering a more general problem: Given a stochastic fixed point equation

$$W \stackrel{d}{=} \Phi(W, S)$$

when can sharp bounds on $\mathbf{E}[W]$ be achieved by principal representations of moment constraints on S?

We have provided evidence that we expect more conditions than the classical Markov-Krein theorem to be needed, and hence new theory may indeed be required. However, we believe this is a promising approach to tackle the classical unsolved $M/G/k$ model, and to provide a template for solving new queueing systems.

Queueing Models with Bounded-Sensitivity

The $M/G/1/PS$ model is one of the foremost examples of queueing systems where the mean sojourn time and the distribution of number of jobs in system exhibits perfect insensitivity to higher order characteristics of the service distribution beyond the mean [91]. In Chapter 5 and Section 3.4, we saw examples of queueing systems which, unlike the $M/G/1/PS$, do not exhibit perfect insensitivity but show evidence of bounded-sensitivity – as the variance of the service distribution increases, the mean sojourn time initially increases and then asymptotes to an upper bound. For the $M/G/1$/Round-Robin model, we were able to prove this phenomenon in light traffic under restrictions of completely monotone service distribution and Exponentially distributed quantum sizes. For the $M/G/k$/JSQ/PS, we provided numerical and simulation evidence. While both systems are in some senses a modification of the $M/G/1/PS$ queue, we saw that the bounded-sensitivity does not carry over to other load balancing policies for the $M/G/k/\cdot$/PS model.

It is a very fascinating question to explore under what conditions would a queueing system exhibit such bounded-sensitivity? We were able to intuit this phenomenon for the $M/G/1$/Round-Robin and $M/G/k$/JSQ/PS models via the example of the

degenerate hyperexponential H_2^* service distribution (while in the former it provides an upper bound, in the latter it yields identical performance as $M/M/k$/JSQ/PS). When can this be a sufficient criterion, at least for size-independent scheduling disciplines?

New heavy-traffic scalings

In this thesis we have presented two new heavy-traffic scalings to study questions which have not yet been addressed via this tool.

1. **Heavy-traffic scaling for non-work-conserving systems:** A non-work-conserving system is defined as one where the system capacity can be less than the maximum depending on the system state. Apart from the $G/G/k$ queueing model and Jackson type queueing networks, non-work-conserving queueing systems have not been subjected to study via the powerful tool of diffusion analysis which can provide approximations for the behavior not just in stationarity, but also as a stochastic process. We were motivated by the application of Database servers (Section 4.4) where depending on the number of jobs at the server, the aggregate server capacity can vary. None of the known scalings in the literature were sufficient for our purposes, and we thus proposed a general and principled approach to deriving heavy-traffic diffusion scaling for non-work-conserving systems – start with the original discrete system that is to be approximated, and then reverse engineer the system parameters such that the stationary distribution of the limiting system under Poisson arrivals and Exponential service approaches that of the original system under the same workload. We presented a preliminary approximation for the stationary distribution under our scaling, and a complete rigorous analysis is left as future work. We believe that our scaling will yield sharper approximations even for the $G/G/k$ model as the limiting system is more representative of the original finite-server system.

2. **Many-servers heavy-traffic scaling for load balancing policies:** In Section 5.6, we proposed a many-servers heavy-traffic scaling to study Joint-the-Shortest-Queue (JSQ) load balancer. Under the proposed scaling, the server farm capacity (K) and arrival rate (λ) grow simultaneously while maintaining constant slack capacity ($K - \lambda = \Theta(1)$). Here we were walking on a razor's edge: a higher slack capacity causes the mean sojourn time to converge to the mean job size, and any smaller slack capacity causes the multi-server system to collapse to a single server system. We presented a very simple analysis of JSQ

policy under the proposed scaling leading to new insights into its behavior, including an approximation for the sojourn time distribution. Similar scaling was recently independently proposed for analyzing central queue systems [15], and we believe this to be a very exciting regime for discovering qualitative behavior of load balancing policies.

7.2 System Design Insights

Maximizing efficiency is not always optimal

System designers are often faced with the problem of choosing an operating point for their system. For example, in Chapter 4, we encountered the problem of concurrency control via imposing a Multi-Programming Limit (MPL) – too little concurrency can lead to inefficient resource utilization, while too much concurrency can again lead to loss of throughput due to context switch overhead. Another design question of similar flavor is choosing the quantum size for CPU scheduling: too small a quantum size can lead to wasted capacity, while a large quantum size can hurt the performance of interactive jobs (jobs with short CPU bursts). A popular rule of thumb is to choose the MPL or quantum size that maximizes the efficiency of the system because a system close to instability is undesirable. As we show in this thesis, the metrics of maximizing efficiency and minimizing response time are not equivalent, and the right decision depends on the workload and the demand. We also presented a traffic-oblivious concurrency mechanism that can adapt to the changes in demand – when the demand is low, the concurrency level is increased driven by the variance in job sizes; when the demand is high, the concurrency level is adjusted to attain maximum efficiency.

Simple load balancing heuristics can be good

Modern day load balancers use numerous parameters to measure the 'health' of the servers in the system before taking load balancing or task assignment decisions. In Chapter 5 we saw that the very simple load balancing rule of just sending a new job to the server with the fewest jobs can be near optimal while being oblivious to remaining work at the servers. We found this rule to be asymptotically optimal even when server speeds are heterogeneous.

See the forest for the trees

Driven by the goal of energy-efficiency, a lot of effort is going into designing state-of-the-art servers with low power sleep states. While innovations in hardware are indeed the future of low-power computing, in Chapter 6 we saw that by looking at the entire data center as a single entity, significant energy savings can be achieved just via smart software. For example, our proposed policy DELAYEDOFF turns off servers after they have idled for some threshold time. Such time-out policies are commonplace, but applied at device level alone they are insufficient. By adding a smart dispatcher (MRB or DELAYEDOFF-INDEX) and keeping the same set of servers busy, we can induce the necessary variance in idle periods of the servers for such time-out based rules to work. Similarly, we suggest that the development of algorithms for server side speed-scaling, data layout in data centers, geographic load balancing, and incentive mechanisms for traffic shaping should be developed in a holistic rather than piecemeal manner.

Bibliography

[1] http://www.research.ibm.com/bluegene/.

[2] http://aws.amazon.com/ec2/instance-types/.

[3] The internet traffic archives: WorldCup98. Available at http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[4] J. Abate and W. Whitt. Simple spectral representations for the M/M/1 queue. *Queueing Systems*, 3(4):321–345, 1988.

[5] M. Abouzour. Automatically tuning database server multiprogramming level. Master's thesis, University of Waterloo, 2007.

[6] I. Adan, G. van Houtum, and J. van der Wal. Upper and lower bounds for the waiting time in the symmetric shortest queue system. *Annals of Operations Research*, 48:197–217, 1994.

[7] I. Adan and J. v. d. Wal. Combining make to order and make to stock. *OR Spektrum*, 20:73–81, 1998.

[8] I. Adan, J. Wessels, and W. Zijm. Analysis of the symmetric shortest queue problem. *Stochastic Models*, 6:691–713, 1990.

[9] I. Adan, J. Wessels, and W. Zijm. Analysis of the asymmetric shortest queue problem. *Queueing Systems*, 8:1–58, 1991.

[10] I. Adan, J. Wessels, and W. Zijm. Matrix-geometric analysis of the shortest queue problem with threshold jockeying. *Operations Research Letters*, 13:107–112, 1993.

[11] I. J. B. F. Adan and J. Resing. *Queueing theory.* Eindhoven University of Technology, 2002.

[12] R. Agrawal, M. J. Carey, and M. Livny. Models for studying concurrency control performance: alternatives and implications. *SIGMOD Rec.*, 14(4):108–121, 1985.

[13] S. Albers and H. Fujiwara. Energy-efficient algorithms for flow time minimization. *ACM Trans. Algorithms*, 3(4):49, 2007.

[14] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: A Fast Array of Wimpy Nodes. In *SOSP '09: Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pages 1–14, New York, NY, USA, 2009. ACM.

[15] R. Atar. A diffusion regime with nondegenerate slowdown. Preprint `http://webee.technion.ac.il/people/atar/NDS-rev.pdf`, Accessed: 24 April, 2011.

[16] B. Avi-Itzhak and S. Halfin. Expected response times in a non-symmetric time sharing queue with a limited number of service positions. *In Proceedings of ITC*, 12:5.4B.2.1–7, 1988.

[17] N. Bansal, H.-L. Chan, and K. Pruhs. Speed scaling with an arbitrary power function. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 693–701, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[18] N. Bansal, T. Kimbrel, and K. Pruhs. Dynamic speed scaling to manage energy and temperature. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 520–529, Washington, DC, USA, 2004. IEEE Computer Society.

[19] N. Bansal, T. Kimbrel, and K. Pruhs. Speed scaling to manage energy and temperature. *J. ACM*, 54(1):1–39, 2007.

[20] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. *Proceeding of ACM SIGMETRICS/Performance'98*, pages 151–160, 1998.

[21] P. Barford and M. E. Crovella. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of Performance '98/SIGMETRICS '98*, pages 151–160, July 1998. Software for Surge is available from Mark Crovella's home page.

[22] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.

[23] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1-2. Athena Scientific, 3rd edition, 2007.

[24] D. Bertsimas and K. Natarajan. A semidefinite optimization approach to the steady-state analysis of queueing systems. *Queueing Syst.*, 56(1):27–39, 2007.

[25] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15:780–804, 2005.

[26] R. Blake. Optimal control of thrashing. In *Proceedings of ACM SIGMET-*

RICS'82, 1982.

[27] J. P. C. Blanc. The power-series algorithm applied to the shortest-queue model. *Operations Research*, 40(1):157–167, 1992.

[28] T. Bonald and A. Proutière. Insensitive bandwidth sharing in data networks. *Queueing Syst. Theory Appl.*, 44:69–100, May 2003.

[29] F. Bonomi. On job assignment for a parallel system of processor sharing queues. *IEEE Transactions on Computers*, 39(7):858–869, 1990.

[30] A. Borovkov. *Stochastic Processes in Queueing Theory.* Nauka, Moscow, 1972.

[31] S. Borst, A. Mandelbaum, M. I. Reiman, and M. Centrum. Dimensioning large call centers. *Operations Research*, 52:17–34, 2000.

[32] S. Borst and R. Núñez-Queija. Introduction to special issue on queueing models for fair resource sharing. *Queueing Syst.*, 53(1-2):5–6, 2006.

[33] O. Boxma and J. Cohen. *Boundary value problems in queueing system analysis.* North Holland, 1983.

[34] O. Boxma, J. Cohen, and N. Huffels. Approximations in the mean waiting time in an $M/G/s$ queueing system. *Operations Research*, 27:1115–1127, 1979.

[35] M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. In *Proceedings of ACM SIGMETRICS'10*, pages 275–286, New York, NY, USA, 2010.

[36] P. Brémaud. *Point Processes and Queues.* Springer, New York, 1981.

[37] D. Burman and D. Smith. A light-traffic theorem for multi-server queues. *Math. Oper. Res.*, 8:15–25, 1983.

[38] E. G. Coffman, Jr., R. R. Muntz, and H. Trotter. Waiting time distributions for processor-sharing systems. *J. Assoc. Comput. Mach.*, 17:123–130, 1970.

[39] B. Conolly. The autostrada queueing problem. *J. Appl. Prob.*, 21:394–403.

[40] G. Cosmetatos. Some approximate equilibrium results for the multiserver queue $(M/G/r)$. *Operational Research Quarterly*, 27:615–620, 1976.

[41] M. Crovella, M. Harchol-Balter, and C. Murta. On choosing a task assignment policy for a distributed server system. *J. Parallel and Distributed Computing*, 59(2):204–228, 1999.

[42] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceeding of ACM SIGMETRICS'96*, pages 160–169, May 1996.

[43] D. Daley and T. Rolski. Some comparibility results for waiting times in single- and many-server queues. *J. Appl. Prob.*, 21:887–900, 1984.

[44] D. J. Daley. Some results for the mean waiting-time and workload in $GI/GI/k$

queues. In J. H. Dshalalow, editor, *Frontiers in queueing: models and applications in science and engineering*, pages 35–59. Boca Raton, FL, USA, 1997.

[45] J. H. A. de Smit. A numerical solution for the multiserver queue with hyperexponential service times. *Oper. Res. Lett.*, 2(5):217–224, 1983.

[46] J. H. A. de Smit. The queue $GI/M/s$ with customers of different types or the queue $GI/H_m/s$. *Adv. in Appl. Probab.*, 15(2):392–419, 1983.

[47] J. H. A. de Smit. The queue $GI/H_m/s$ in continuous time. *J. Appl. Probab.*, 22(1):214–222, 1985.

[48] P. J. Denning, K. C. Kahn, J. Leroudier, D. Potier, and R. Suri. Optimal multiprogramming. *Acta Informatrica*, 7:197–216, 1976.

[49] A. Downy and M. Harchol-Balter. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, August 1997.

[50] A. Eckberg Jr. Sharp bounds on Laplace-Stieltjes transforms, with applications to various queueing problems. *Math. Oper. Res.*, 2(2):132–142, 1977.

[51] L. Eggert and J. D. Touch. Idletime scheduling with preemption intervals. *SIGOPS Oper. Syst. Rev.*, 39(5):249–262, 2005.

[52] M. El-Taha and S. Stidham. *Sample-Path Analysis of Queueing System.* Kluwer, Boston, 1999.

[53] S. Elnikety, E. Nahum, J. Tracy, and W. Zwaenepoel. A method for trasparent admission control and request schedulin in e-commerce web sites. In *World-Wide-Web Conference*, 2004.

[54] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transac. on Auto. Cont.*, AC-25(4):690–693, 1980.

[55] A. K. Erlang. Sandsynlighetsregning og telefonsamtaler (in Danish). *Nytt tidsskrift for Matematik B 20*, 1909. Later in French: Calcul des probabilitès et conversations téléphoniques. *Revue général De l'Electricité, 18, 1925.*

[56] A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31:245–279, 1998.

[57] S. Finch. Ornstein-Uhlenbeck process. http://algo.inria.fr/csolve/ou.pdf, Accessed: 26 April, 2011.

[58] L. Flatto and H. McKean. Two queues in parallel. *Communication on Pure and Applied Mathematics*, 30:255–263, 1977.

[59] G. Foschini and J. Salz. A basic dynamic routing problem and diffusion. *IEEE Trans. Comm.*, 26(3):320–328, 1978.

[60] S. Foss and D. Korshunov. Heavy tails in multi-server queue. *Queueing Syst.*, 52(1):31–48, 2006.

[61] A. Fredericks. Approximations for customer viewed delays in multiprogrammed, transaction oriented computer systems. *Bell System Technical Journal*, 59(9):1559–1574, 1980.

[62] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: tutorial, review, and research prospects. *Manufacturing and Service operations Management*, 5:79–141, 2003.

[63] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9):1277–1284, 1996.

[64] W. Grassmann. Transient and steady state results for two parallel queues. *Omega*, 8:105–112, 1980.

[65] L. Green. A queueing system with general use and limited use servers. *Operations Research*, 33(1):168–182, 1985.

[66] V. Gupta. Finding the optimal quantum size: Sensitivity analysis of the $M/G/1$ round-robin queue. *SIGMETRICS Perform. Eval. Rev.*, 36(2):104–106, 2008.

[67] V. Gupta, A. Gandhi, M. Harchol-Balter, and M. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. In *PERFORMANCE 2010*, Namur, Belgium, Nov. 2010.

[68] V. Gupta and M. Harchol-Balter. Self-adaptive admission control policies for resource-sharing systems. Technical Report CMU-CS-09-115, School of Computer Science, Carnegie Mellon University, 2009.

[69] V. Gupta, M. Harchol-Balter, A. Scheller-Wolf, and U. Yechiali. Fundamental characteristics of queues with fluctuating load. In *Proceedings of ACM SIGMETRICS '06*, pages 203–215, 2006.

[70] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt. Simulation results for JSQ server farms with processor sharing servers. Technical Report CMU-CS-07-151, School of Computer Science, Carnegie Mellon University, 2007.

[71] B. Halachmi and W. Franta. A diffusion approximation to the multi-server queue. *Management Science*, 24(5):522–529, 1978.

[72] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29(3):567–588, 1981.

[73] M. Harchol-Balter. Task assignment with unknown duration. *JACM*, 49(2):260–288, 2002.

[74] M. Harchol-Balter and B. Schroeder. Evaluation of task assignment policies for supercomputing servers. In *Proceedings of 9th IEEE Symposium on High*

Performance Distributed Computing (HPDC '00), August 2001.

[75] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems*, 21(2):207–233, 2003.

[76] M. v. H. H.C. Tijms and A. Federgruen. Approximations for the steady-state probabilities in the $M/G/c$ queue. *Adv. Appl. Prob.*, 13:186–206, 1981.

[77] H.-U. Heiss and R. Wagner. Adaptive load control in transaction processing systems. In *Proceedings of the 17th International Conference on Large Data Bases (VLDB)*, 1991.

[78] J. L. Hellerstein, V. Morrison, and E. Eilebrecht. Applying control theory in the real world: experience with building a controller for the .net thread pool. *SIGMETRICS Perform. Eval. Rev.*, 37(3):38–42, 2009.

[79] P. Hokstad. Approximations for the $M/G/m$ queue. *Operations Research*, 26(3):510–523, 1978.

[80] P. Hokstad. The steady state solution of the $M/K_2/m$ queue. *Adv. Appl. Prob.*, 12(3):799–823, 1980.

[81] D. L. Iglehart. Limiting diffusion approximations for the many server queue and the repairman problem. *J. Appl. Probab.*, 2(2):429–441, 1965.

[82] Intel Corp. Intel Math Kernel Library 10.0 - LINPACK. http://www.intel.com/cd/software/products/asmo-na/eng/266857.htm, 2007.

[83] S. Irani and K. R. Pruhs. Algorithmic problems in power management. *SIGACT News*, 36(2):63–76, 2005.

[84] S. Irani, S. Shukla, and R. Gupta. Algorithms for power savings. *ACM Trans. Algorithms*, 3(4):41, 2007.

[85] O. B. Jennings, A. M, W. A. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Science*, 42:1383–1394, 1996.

[86] M. A. Johnson and M. T. Taaffe. Tchebycheff systems for probabilistic analysis. *American Journal of Mathematical and Management Sciences*, 13(1-2):83–111, 1993.

[87] P. Juang, Q. Wu, L.-S. Peh, M. Martonosi, and D. W. Clark. Coordinated, distributed, formal energy management of chip multiprocessors. In *ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design*, pages 127–130, New York, NY, USA, 2005. ACM.

[88] A. Kamra, V. Misra, and E. M. Nahum. Yaksha: A self-tuning controller for managing the performance of 3-tiered web sites. In *Twelfth IEEE International Workshop on Quality of Service (IWQOS)*, 2004.

[89] C. W. Kang, S. Abbaspour, and M. Pedram. Buffer sizing for minimum energy-delay product by using an approximating polynomial. In *GLSVLSI '03: Proceedings of the 13th ACM Great Lakes symposium on VLSI*, pages 112–115, New York, NY, USA, 2003. ACM.

[90] S. Karlin and W. J. Studden. *Tchebycheff systems: With applications in analysis and statistics.* John Wiley & Sons Interscience Publishers, New York, 1966.

[91] F. P. Kelly. *Reversibility and Stochastic Networks.* Chichester, 1979.

[92] J. Kiefer and J. Wolfowitz. On the theory of queues with many servers. *Trans. Amer. Math. Soc.*, 78:1–18, 1955.

[93] J. Kiefer and J. Wolfowitz. On the characteristics of the general queueing process with applications to random walk. *Ann. Math. Statist.*, 27:147–161, 1956.

[94] T. Kimura. Diffusion approximation for an $M/G/m$ queue. *Operations Research*, 31:304–321, 1983.

[95] T. Kimura. Approximations for multi-server queues: system interpolations. *Queueing Systems*, 17(3-4):347–382, 1994.

[96] J. Kin, M. Gupta, and W. Mangione-Smith. The filter cache: an energy efficient memory structure. *Microarchitecture, IEEE/ACM International Symposium on*, 0:184, 1997.

[97] J. Kingman. Two similar queues in parallel. *Biometrika*, 48:1316–1323, 1961.

[98] J. Kingman. On queues in heavy traffic. *J. R. Statist. Soc.*, 24(2):383–392, 1962.

[99] J. Kingman. Inequalities in the theory of queues. *J. R. Statist. Soc.*, 32(1):102–110, 1970.

[100] J. F. Kingman. The first Erlang century–and the next. *Queueing Syst. Theory Appl.*, 63:3–12, December 2009.

[101] L. Kleinrock. Analysis of a time-shared processor. *Naval research logistics quarterly*, pages 59–73, 1964.

[102] L. Kleinrock. Time-shared systems: A theoretical treatment. *J. Assoc. Comput. Mach.*, 14:242–261, 1967.

[103] L. Kleinrock. *Queueing Systems, Volume I: Theory.* Wiley-Interscience, 1975.

[104] L. Kleinrock. *Queueing Systems; Volume 2: Computer Applications.* Wiley, New York, 1976.

[105] C. Knessl, B. Matkowsky, Z. Schuss, and C. Tier. Two parallel $M/G/1$ queues where arrivals join the system with the smaller buffer content. *IEEE Trans.*

Comm., 35(11):1153–1158, 1987.

[106] J. Köllerström. Heavy traffic theory for queues with several servers. I. *J. Appl. Prob.*, 11:544–552, 1974.

[107] G. Koole, P. D. Sparaggis, and D. Towsley. Minimizing response times and queue lengths in systems of parallel queues. *J. Appl. Prob.*, 36:1185–1193, 1999.

[108] A. Lee and P. Longton. Queueing process associated with airline passenger check-in. *Operations Research Quarterly*, 10:56–71, 1959.

[109] H. L. Lee and M. A. Cohen. A note on the convexity of performance measures of $M/M/c$ queueing systems. *J. Appl. Probab.*, 20(4):920–923, 1983.

[110] H. Lin and C. Raghavendra. An analysis of the join the shortest queue (JSQ) policy. In *Proc. 12th Int'l Conf. Distributed Computing Systems*, pages 362–366, 1992.

[111] Z. Liu, P. Nain, and D. Towsley. Sample path methods in the control of queues. *Queueing Systems*, 21(3-4):293–335, Sept. 1995.

[112] J. Lui, R. Muntz, and D. Towsley. Bounding the mean response time of the minimum expected delay routing policy: an algorithmic approach. *IEEE Trans. Comp.*, 44(12):1371–1382, 1995.

[113] B. Ma and J. Mark. Approximation of the mean queue length of an $M/G/c$ queueing system. *Operations Research*, 43(1):158–165, 1995.

[114] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, 2001.

[115] M. Miyazawa. Approximation of the queue-length distribution of an $M/GI/s$ queue by the basic equations. *J. Appl. Prob.*, 23:443–458, 1986.

[116] A. Müller and D. Stoyan. *Comparison methods for stochastic models and risks.* Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., Chichester, 2002.

[117] R. Nelson and T. Philips. An approximation to the response time for shortest queue routing. *ACM Perf. Eval. Review*, 17:181–189, 1989.

[118] R. Nelson and D. Towsley. On maximizing the number of departures before a deadline on multiple processors. Technical report, Amherst, MA, USA, 1987.

[119] M. Neuts. *Matrix-geometric solutions – An algorithmic approach.* The Johns Hopkins University Press, Baltimore, MD, 1981.

[120] S. Nozaki and S. Ross. Approximations in finite-capacity multi-server queues with Poisson arrivals. *J. Appl. Prob.*, 15(4):826–834, 1978.

[121] M. Nuyens and W. van der Weij. Monotonicity in the limited processor sharing

queue. Technical Report PNA-E0802, CWI, 2008.

[122] T. Osogami, M. Harchol-Balter, and A. Scheller-Wolf. Analysis of cycle stealing with switching costs and thresholds. *Performance Evaluation*, 61(4):347–369, 2005.

[123] T. Osogami and R. Raymond. Semidefinite optimization for transient analysis of queues. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):363–364, 2010.

[124] K. Pruhs, P. Uthaisombut, and G. Woeginger. Getting the best response for your erg. *ACM Trans. Algorithms*, 4(3):1–17, 2008.

[125] V. Ramaswami and G. Latouche. A general class of Markov processes with explicit matrix-geometric solutions. *OR Spektrum*, 8:209–218, 1986.

[126] B. Rao and M. Posner. Algorithmic and approximation analyses of the shorter queue model. *Naval Research Logistics*, 34:381–398, 1987.

[127] K. Rege and M. Sengupta. Sojourn time distribution in a multiprogrammed computer system. *AT&T Tech. J.*, 64:1077–1090, 1985.

[128] R. Righter, J. G. Shanthikumar, and G. Yamazaki. On extremal service disciplines in single-stage queueing systems. *J. Appl. Probab.*, 27(2):409–416, 1990.

[129] A. Riska, N. Mi, E. Smirni, and G. Casale. Feasibility regions: exploiting tradeoffs between power and performance in disk drives. *SIGMETRICS Perform. Eval. Rev.*, 37(3):43–48, 2009.

[130] D. M. Ritchie and K. Thompson. The Unix time-sharing system. *C. ACM*, 17(7):365–375, 1974.

[131] S. M. Ross. *Stochastic Processes, 2nd Edition.* Wiley, 1996.

[132] A. Scheller-Wolf and K. Sigman. New bounds for expected delay in FIFO $GI/GI/c$ queues. *Queueing Systems*, 26(1-2):169–186, 1997.

[133] A. Scheller-Wolf and R. Vesilo. Structural interpretation and derivation of necessary and sufficient conditions for delay moments in FIFO multiserver queues. *Queueing Syst.*, 54(3):221–232, 2006.

[134] B. Schroeder, M. Harchol-Balter, A. Iyengar, E. Nahum, and A. Wierman. How to determine a good multi-programming level for external scheduling. In *Proceedings of the 22nd International Conference on Data Engineering*, Atlanta, GA, April 2006.

[135] S. Shenker and A. Weinrib. A symptotic analysis of large heterogeneous queueing systems. In *Proceedings of ACM SIGMETRICS'88*, pages 56–62, New York, NY, USA, 1988. ACM.

[136] S. Shenker and A. Weinrib. The optimal control of heterogeneous queueing systems: A paradigm for load-sharing and routing. *IEEE Trans. Comput.*,

38:1724–1735, December 1989.

[137] M. R. Stan and K. Skadron. Power-aware computing: Guest editorial. *IEEE Computer*, 36(12):35–38, December 2003.

[138] K. Stordahl. The history behind the probability theory and the queuing theory. *Telektronikk*, pages 123–140, 2007.

[139] D. Stoyan. A continuity theorem for queue size. *Bull. Acad. Sci. Pollon.*, 21:1143–1146, 1973.

[140] D. Stoyan. Approximations for $M/G/s$ queues. *Math. Operationsforsch. Statist. Ser. Optimization*, 7:587–594, 1976.

[141] D. Stoyan. *Comparison methods for queues and other stochastic models.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Ltd., Chichester, 1983. Translation from the German edited by Daryl J. Daley.

[142] B. Sun and S. Li. Improving effectiveness of customer service in a cost-efficient way - empirical investigation of service allocation decisions with out-sourced centers. Working Paper, Tepper School of Business, Carnegie Mellon University, 2006.

[143] H. Takagi. *Queueing Analysis, Vol. 1: Vacation and Priority Systems.* North-Holland, 1991.

[144] Y. Takahashi. An approximation formula for the mean waiting time of an $M/G/c$ queue. *J. Opns. Res. Soc. Japan*, 20:147–157, 1977.

[145] H. Thorisson. The queue $GI/GI/k$: finite moments of the cycle variables and uniform rates of convergence. *Comm. Statist. Stochastic Models*, 1(2):221–238, 1985.

[146] U.S. Environmental Protection Agency. EPA Report on server and data center energy efficiency. 2007.

[147] W. van der Weij, S. Bhulai, and R. van der Mei. Optimal scheduling policies for the limited processor sharing queue. Technical Report WS2008-5, Department of Mathematics, Vrije University, 2008.

[148] E. van Doorn and J. Regterschot. Conditional PASTA. *Oper. Res. Lett.*, 7:229–232, 1988.

[149] M. Welsh, D. Culler, and E. Brewer. Seda: an architecture for well-conditioned, scalable internet services. *SIGOPS Oper. Syst. Rev.*, 35(5):230–243, 2001.

[150] W. Whitt. The effect of variability in the $GI/G/s$ queue. *J. Appl. Prob.*, 17:1062–1071, 1980.

[151] W. Whitt. Comparison conjectures about the $M/G/s$ queue. *OR Letters*, 2(5):203–209, 1983.

[152] W. Whitt. On approximations for queues, I: Extremal distributions. *AT&T Bell Labs Technical Journal*, 63:115–138, 1984.

[153] W. Whitt. Approximations for the $GI/G/m$ queue. *Production and Operations Management*, 2(2):114–161, 1993.

[154] W. Whitt. A diffusion approximation for the G/GI/n/m queue. *Operations Research*, 52:922–941, 2004.

[155] W. Whitt. Heavy-traffic limits for the $G/H_2^*/n/m$ queue. *Math. Oper. Res.*, 30(1):1–27, 2005.

[156] A. Wierman, L. L. H. Andrew, and A. Tang. Power-aware speed scaling in processor sharing systems. *INFOCOM*, 2009.

[157] W. Winston. Optimality of the shortest line discipline. *J. Appl. Prob.*, 14:181–189, 1977.

[158] R. W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.

[159] D. Yao. Refining the diffusion approximation for the $M/G/m$ queue. *Operations Research*, 33:1266–1277, 1985.

[160] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced cpu energy. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:374, 1995.

[161] S. F. Yashkov. Processor-sharing queues: some progress in analysis. *Queueing Systems Theory Appl.*, 2(1):1–17, 1987.

[162] J. Zhang and B. Zwart. Steady state approximations of limited processor sharing queues in heavy traffic. *Submitted for publication.*

[163] A. P. Zwart and O. J. Boxma. Sojourn time asymptotics in the $M/G/1$ processor sharing queue. *Queueing Systems Theory Appl.*, 35(1-4):141–166, 2000.