Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

Provisioning and performance evaluation of parallel systems with output synchronization

WASIUR R. KHUDABUKHSH, Technische Universität Darmstadt, Germany
SOUNAK KAR, Technische Universität Darmstadt, Germany
AMR RIZK, Technische Universität Darmstadt, Germany
HEINZ KOEPPL, Technische Universität Darmstadt, Germany

Parallel server frameworks are widely deployed in modern large-data processing applications. Intuitively, splitting and parallel processing of the workload provides accelerated application response times and scaling flexibility. Examples of such frameworks include MapReduce, Hadoop, and Spark. For many applications, the dynamics of such systems are naturally captured by a Fork-Join (FJ) queuing model, where incoming jobs are split into tasks each of which is mapped to exactly one server. When all the tasks that belong to one job are executed, the job is reassembled and leaves the system. We consider this behavior at the output as a synchronization constraint.

In this paper, we study the performance of such parallel systems for different server properties, i.e., work-conservingness, phase-type behavior, and as suggested by recent evidence, for bursty input job arrivals. We establish a Large Deviations Principle (LDP) for the steady-state job waiting times in an FJ system based on Markov-additive processes. Building on that, we present a performance analysis framework for FJ systems and provide computable bounds on the tail probabilities of the steady-state waiting times. We validate our bounds using estimates obtained through simulations. In addition, we define and analyze provisioning, a flexible division of jobs into tasks, in FJ systems. Finally, we use this framework together with real-world traces to show the benefits of an adaptive provisioning system that adjusts the service within an FJ system based on the arrival intensity.

CCS Concepts: • **Mathematics of computing** → **Queueing theory**; *Markov processes*; • **General and reference** → Performance; • **Theory of computation** → MapReduce algorithms.

Additional Key Words and Phrases: Performance evaluation, queuing systems, Fork-Join queues, Markov additive processes, Parallel systems

ACM Reference Format:
Wasiur R. KhudaBukhsh, Sounak Kar, Amr Rizk, and Heinz Koeppl. 2019. Provisioning and performance evaluation of parallel systems with output synchronization. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 1, 1 (July 2019), 30 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Recent infrastructural advancement of cloud computing and large-scale data processing has brought about massive deployment of parallel-server systems. Frameworks, such as MapReduce [21, 51], its

Authors' addresses: Wasiur R. KhudaBukhsh, Technische Universität Darmstadt, Rundeturmstrasse 12, 64283 Darmstadt, Germany, wasiur.khudabukhsh@bcs.tu-darmstadt.de; Sounak Kar, Technische Universität Darmstadt, Rundeturmstrasse 10, 64283 Darmstadt, Germany, sounak.kar@kom.tu-darmstadt.de; Amr Rizk, Technische Universität Darmstadt, Rundeturmstrasse 10, 64283 Darmstadt, Germany, amr.rizk@kom.tu-darmstadt.de; Heinz Koeppl, Technische Universität Darmstadt, Rundeturmstrasse 12, 64283 Darmstadt, Germany, heinz.koeppl@bcs.tu-darmstadt.de.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2376-3639/2019/7-ART $15.00
https://doi.org/0000001.0000001

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

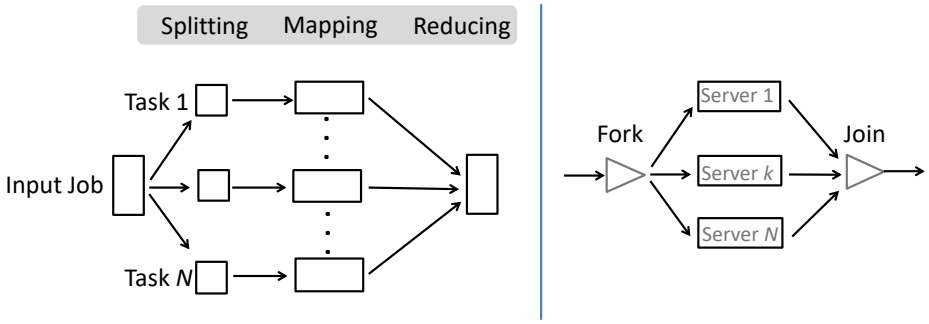

Fig. 1. MapReduce as an FJ system. Incoming jobs are first split into tasks and then "mapped" to N heterogeneous servers that work parallelly. A job leaves the system when *all* of its tasks are executed.

implementation Hadoop [32] and Spark [62] are dominant in today's world. Such systems seek to reap the benefits of parallelization. However, often they are also subject to a synchronization constraint, because the final output is composed of outputs from all the servers. This makes performance evaluation of such systems interesting. Fork-Join (FJ) queuing models naturally capture the dynamics of system parallelization under synchronization constraints [38, 55, 58].

In Figure 1, we present an FJ system abstraction of the MapReduce. Arriving jobs are first split into tasks each of which is mapped exactly to one server executing the *map* operation. A job leaves the system when all of its tasks are executed. We categorize the servers depending on whether they are work-conserving or not. Servers that start servicing the task of the next job, if available, immediately after finishing the current job, are labeled work-conserving. Servers that are not work-conserving, referred to as "blocking" servers hereinafter, wait until *all* servers finish servicing their current tasks before starting the task of the next job. Blocking systems, also known as split-merge systems [48], impose an additional synchronization barrier at the input. Nevertheless, split-merge systems can be treated as a special case of the work-conserving (non-blocking) system (see [36, 37, 48]). In particular, an FJ system with N blocking servers can be viewed as a hypothetical queuing system with just one work-conserving server whose service time distribution is the same as the distribution of the maximum order statistic of the individual service times of the N servers of the original FJ system. We shall use this observation for the purpose of performance evaluation of blocking systems using the tools developed for work-conserving systems.

As a metric of performance in an FJ system, we consider the waiting time, which we define as the amount of time a job waits until its *last* task starts being serviced from moment of its arrival. Its stochastic behavior is governed by the nature of inter-arrival times of the jobs, *i.e.*, the arrival process, and the service times of the servers. In the simplest case, one assumes a renewal arrival process, independent and identically distributed (iid) service times and mutually independent servers. However, recent evidences suggest that this assumption is untenable for various reasons. Arrival processes such as the input to a MapReduce system or datacenter traffic may not be renewal and may exhibit considerable burstiness [19, 33, 39, 61]. Moreover, the service times at different servers may also be inherently dependent, and may show phase-type behavior. The behavior of the inter-arrival times and the service times may change drastically depending on or being controlled by certain exogenous factors. For the purpose of mathematical abstraction, we use the term "environment" for these exogenous factors. In this paper, to account for the effects of changing environment, we present a Markov-additive process [35] model (see Figure 3), and show how particular application scenarios can be derived as special cases of it. In particular, we cover

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

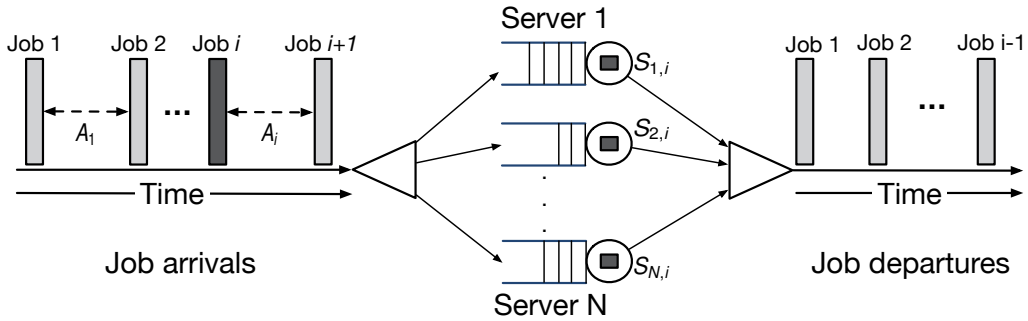

Fig. 2. Arrival and service processes of an FJ system. The random variable A_i denotes the inter-arrival time between the i-th and the $i+1$-th jobs. Each incoming job is split into N tasks and assigned to N heterogeneous servers. The service time at the n-th server for the task of the i-th job is denoted by $S_{n,i}$. A job leaves the system when *all* of its tasks are served.

three application scenarios: **(a)** non-renewal (Markov-modulated) arrivals, **(b)** servers showing phase-type behavior, and **(c)** Markov-modulated (MM) arrivals and service. We also bring in the notion of *provisioning*, an umbrella term used for a rule that decides on the FJ job division into tasks, or that regulates service rates either *reactively* or *proactively*. Proactive provisions anticipate the change of environment, and act accordingly, while reactive provisions only *react* to the current environment (*e.g.*, see [47]).

An exact analysis of an FJ system with more than two servers in a general setup remains elusive [6, 13] because the steady-state waiting time distribution is hard to obtain in closed form. One approach to circumvent this problem, which we take in this paper, is to bound the tail probabilities of the steady-state waiting times [55]. In [55], the authors provide upper bounds on the tail probabilities of the steady-state waiting and response times in homogeneous FJ systems with identical servers. They consider renewal and two-state Markov-modulated arrival processes with iid service times across N servers. In contrast to [55], we consider FJ systems in full generality. We consider a heterogeneous FJ system; allow the modulating Markov chain to lie in an arbitrary state space (even uncountable); and allow both arrival and service processes to be modulated and hence, correlated. Moreover, our approach here will be to first establish a Large Deviations Principle (LDP) (Theorem 1) for the steady-state waiting times, and thereby obtain a computable upper bound on the tail probability through further simplification (Theorem 2). By virtue of the generality of the model considered here, the bounds obtained in [55] can be recovered from Theorem 2 by choosing the state space and related probability distributions appropriately. We further use the bounds in Theorem 2 for performance evaluation purposes of provisioning. To give a concrete example, we also calibrate our model using a datacenter trace and devise a simplistic reactive provisioning.

Our contributions in this paper are: **(1)** A Markov-additive (MA) process model for a general FJ system, and a computable upper bound on the tail probabilities of the steady-state waiting times, obtained by means of an LDP. **(2)** Application of our result to three scenarios, namely, non-renewal (Markov-modulated) arrivals, servers showing phase-type behavior, and Markov-modulated arrivals and service. In the process, we also compare our theoretical bound against empirical Complementary Cumulative Distribution Functions (CCDFs) obtained through Monte Carlo simulations. **(3)** A formulation of (reactive and proactive) provisioning, a rule of flexible job division, in FJ systems. **(4)** A numerical study based on our model using a datacenter trace, and a corresponding example of reactive provisioning for the purpose of illustration.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

The paper, which is divided into three parts, is organized as follows: The theory part of the paper is covered in Section 2 and Section 3, whereas the application part consists of Section 4 through Section 7. We reserve Section 8 and Section 9 for background and discussion, which also constitute the third and the final part of the paper. Section 2 introduces the central mathematical model and presents the main results. In Section 3, we analyze the (N, r)-FJ systems with purging and the split-merge systems as a special case of the work-conserving (non-blocking) system. In Section 4, we apply our result to an FJ system with non-renewal input, followed by Section 5 where we describe an FJ system with dependent servers and introduce the notion of provisioning. The application to FJ systems with Markov-modulated arrivals and service is discussed in Section 6. Section 7 shows a trace-based evaluation before we discuss related work in Section 8. We conclude the paper with a discussion in Section 9.

PART A. THEORY

2 THE MODEL

In this section, we present our mathematical model for Fork-Join systems. The roadmap is as follows: We first establish a Large Deviations Principle for FJ systems based on a Markov-additive process representation. Based on the LDP, we provide computable bounds on the tail probabilities of the steady-state waiting times. The idea is to use these general results to obtain several special cases that are relevant for practical purposes. For the purpose of illustration, we compliment our main result with concrete application scenarios obtained as special cases in later sections.

2.1 Notational conventions

The following notational conventions are adhered to throughout the paper. We denote the set of natural numbers and the set of real numbers by $\mathbb{N}$ and $\mathbb{R}$ respectively. Let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For $N \in \mathbb{N}$, let $[N] := \{1, 2, \ldots, N\}$. For $F \subseteq \mathbb{R}^N$, we denote the Borel σ-field of subsets of F by $\mathcal{B}(F)$. For some $F \in \mathcal{B}(\mathbb{R}^N)$, the interior, the closure and the boundary of F are denoted by $\mathsf{Int}\, F$, $\mathsf{Cl}\, F$, and $\mathsf{Bnd}\, F$ respectively. For any extended real-valued function f, we denote the effective domain of f by $\mathcal{D}f$, *i.e.*, $\mathcal{D}f := \{x \in \mathbb{R} \mid f(x) < \infty\}$. For an event F, we denote the indicator function of F by $\mathbb{1}(F)$, taking value unity when F is true and zero otherwise.

2.2 System description

Consider a single stage FJ queuing system with N parallel servers as depicted in Figure 1 and Figure 2. Jobs arrive at the input station according to some process with inter-arrival time A_i between the i-th and $(i+1)$-th job, $i \in \mathbb{N}$. A job is split into N tasks each of which is assigned to exactly one server. The service time for the task of job i at the n-th server is denoted by the random variable $S_{n,i}$, where $n \in [N]$ (see Figure 2). Finally the job leaves the system when *all* of its tasks are served, imposing a synchronization constraint at the output. We assume the servers are work-conserving in the sense that a server immediately starts serving the next task, if available, upon finishing the current one.

In real applications, the behavior of the inter-arrival times and the service times may change drastically depending on certain exogenous factors. For example, during a heavy traffic period, the inter-arrival times are much shorter compared to those during a low traffic period. From considerations of energy conservation or cost, the service times may also be modulated externally to yield high or low efficiency. For instance, given a fixed monetary budget, the service rates of a cloud computing service such as the Amazon AWS [1] could be altered as the price changes to meet the budget constraint. For the purpose of mathematical abstraction, we use an umbrella term "environment" for these exogenous factors. To capture the effects of changing environment, we

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

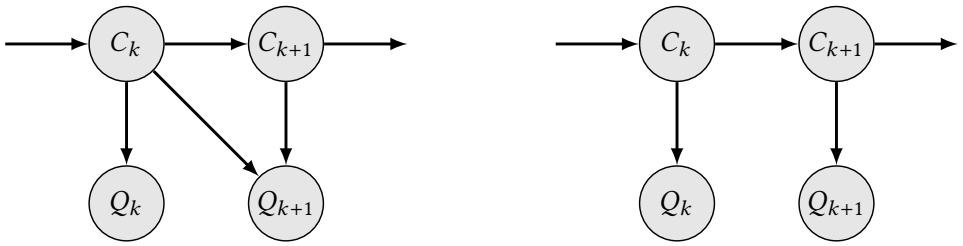

Fig. 3. Graphical representation of a Markov-additive process $\{C_k, Q_k\}_{k\in\mathbb{N}}$ **(Left)** and its special "uncoupled" case, the Markov modulated process **(Right)**. The nodes represent the variables and the arrows, the dependence structure. The process Q_k is an additive component, *i.e.*, $Q_{k+1} = Q_k + (X^A_{1,k+1}, X^A_{2,k+1}, \ldots, X^A_{N,k+1})$.

consider an underlying Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$ on some measure space $(\mathbb{E}, \mathcal{E})$, where $\mathbb{E}$ is assumed separable. Note that $\mathbb{E}$ need not be finite, or even countable. The Markov chain could capture the changes in job arrival rates, *i.e.*, modulate the arrival process; could decide the service rates of the servers, *i.e.*, modulate the service process; or both in which case it is said to modulate both the arrival as well as the service processes. Naturally, different choices of the state space $\mathbb{E}$ yield different types of modulation to suit different real-life applications. In Table 1, we present a glossary of examples of $\mathbb{E}$ capturing different modulation scenarios. Detailed examples will be provided in later sections.

2.2.1 Waiting times. In this work, we consider the waiting time as a performance metric. We adopt the definition of waiting times from [55]. For the first job to arrive, there is no waiting time. For subsequent jobs, we define the waiting time to be the amount of time between the arrival of the job and the time when its *last* task starts getting serviced. That is, a job *waits* until its last task starts being serviced from the time of its arrival. Formally, for an FJ queuing system with N work-conserving servers, we define the waiting time W_j for the j-th job as 0 for $j = 1$ and $\max\{0, \max_{k\in[j-1]}\{\max_{n\in[N]}\{\sum_{i=1}^{k} S_{n,j-i} - \sum_{i=1}^{k} A_{j-i}\}\}\}$, for $j > 1$. To simplify the notations, define the difference process Q_k (sometimes called the drift process) on $\left(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N)\right)$ as follows

$$Q_k := \left(X_{1,k}, X_{2,k}, \ldots, X_{N,k}\right) \text{ with } X_{n,k} := \sum_{i=1}^{k} X^A_{n,i}, \tag{2.1}$$

where $X^A_{n,i} = S_{n,i} - A_i$ for all $i \in \mathbb{N}$ and set $X_{n,0} := 0$, for each $n \in [N]$. We are interested in the steady-state waiting times. It can be showed that the steady-state waiting time W has the following distributional representation (see [55]),

$$W =_{\mathrm{D}} \max_{k\in\mathbb{N}_0} \max_{n\in[N]} X_{n,k}, \tag{2.2}$$

where $=_{\mathrm{D}}$ denotes equality in distribution. Despite this simple representation, getting closed-form expression of the probability distribution of W is hard under general settings [6, 13]. We can, nevertheless, obtain information about its asymptotic behavior such as an LDP [22, 59] from which we can achieve computable bounds on the tail probabilities of W. An LDP is important in that it quantifies probabilities of rare events (whose probabilities of occurance are exponentially small). The rate function associated with an LDP is also unique. In the next section, we establish an LDP for the waiting times under mild assumptions following [23, 35, 49].

2.3 Large deviations of the waiting times

We assume that the process $\{(C_k, Q_k)\}_{k\in\mathbb{N}_0}$ is a Markov-additive process on $(\mathbb{E}\times\mathbb{R}^N, \mathcal{E}\times\mathcal{B}(\mathbb{R}^N))$.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

Modulation	State space	Scenario
Only arrivals	$\mathbb{E} = \{0, 1\}$	Markov-modulated high-low (or on-off) arrivals.
	$\mathbb{E} = \{1, 2, \ldots, d\}$	Finite state modulation of the arrivals.
	$\mathbb{E} = \mathbb{N}$	Countable state modulation of the arrivals.
	$\mathbb{E} = \mathbb{E}^A \subseteq \mathbb{R}$	Modulation of the arrivals on an uncountable state space such as $[0, 1]$.
		Real-life example: bursty input at MapReduce clusters.
Only service	$\mathbb{E} = \{0, 1\}$	All servers are Markov high-low modulated.
	$\mathbb{E} = \{1, 2, \ldots, d\}$	All servers are Markov modulated on a finite set.
	$\mathbb{E} = \{0, 1\} \times \{0, 1\} \times \ldots \times \{0, 1\}$	All servers are Markov high-low modulated, but by separate chains that may or may not be independent.
	$\mathbb{E} = \{0, 1\} \times \ldots \times \{0, 1\} \times \{1, 2, \ldots, d\}$	All but the N-th server are Markov high-low modulated by separate chains and the N-th server is Markov modulated on a finite set.
	$\mathbb{E} = \{1\} \times \{1\} \times \ldots \times \{0, 1\}$	Only the N-th server is Markov high-low modulated.
	$\mathbb{E} = \mathbb{E}_1^S \times \mathbb{E}_2^S \times \ldots \times \mathbb{E}_N^S$	The n-th server is modulated on its own state space $\mathbb{E}_n^S \subseteq \mathbb{R}$, for $n \in [N]$.
		Real-life example: switching between cloud service machines such as Amazon AWS under a monetary budget constraint, as the prices change over time; provisioning such as round-robbin in MapReduce clusters.
Both arrivals and service	$\mathbb{E} = \mathbb{E}^A \times \mathbb{E}_1^S \times \mathbb{E}_2^S \times \ldots \times \mathbb{E}_N^S$	The arrival process is modulated on state space $\mathbb{E}^A \subseteq \mathbb{R}$ and the n-th server is modulated on its own state space $\mathbb{E}_n^S \subseteq \mathbb{R}$, for $n \in [N]$. The modulating chains need not be independent.
		Real-life example: adaptive provisioning (both proactive and reactive) in parallel systems such as MapReduce clusters; modulation in Multi-path Transmission Control Protocol (Multi-path TCP).
No modulation	$\mathbb{E} = \{1\}$	Reduces to the renewal case.

Table 1. Table showing different choices for the state space for different application scenarios.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

Definition 1. (Markov-additive process) The processes $\{(C_k, Q_k)\}_{k\in\mathbb{N}}$ is a Markov-additive process on $(\mathbb{E}\times\mathbb{R}^N, \mathcal{E}\times\mathcal{B}(\mathbb{R}^N))$ if

(1) The process $\{(C_k, Q_k)\}_{k\in\mathbb{N}}$ is a Markov process on $(\mathbb{E}\times\mathbb{R}^N, \mathcal{E}\times\mathcal{B}(\mathbb{R}^N))$.
(2) The following holds for $c\in\mathbb{E}, s\in\mathbb{R}^N, F\in\mathcal{E}, G\in\mathcal{B}(\mathbb{R}^N)$,

$$\begin{aligned}&\mathsf{P}((C_{k+1}, Q_{k+1})\in F\times(G+s)\mid (C_1,Q_1)=(c,s))\\ &=\mathsf{P}((C_{k+1}, Q_{k+1})\in F\times G\mid (C_1,Q_1)=(c,0))\\ &=\mathsf{P}((C_{k+1}, Q_{k+1})\in F\times G\mid C_1=c).\end{aligned}$$

The Markov chain C_k is endowed with an additive component Q_k, the difference process in our queuing system defined in (2.1). Note that the difference process Q_k is indeed additive in the sense that $Q_{k+1}=Q_k+(X^A_{1,k+1}, X^A_{2,k+1},\ldots,X^A_{N,k+1})$. Intuitively, the environment captured by the Markov chain C_k modulates the inter-arrival and service times (through their difference) not only for the current job but also for the next arriving job (see Figure 3). Accordingly, define the transition kernel

$$L(c, F\times G) := \mathsf{P}((C_1,Q_1)\in F\times G\mid Q_0=c), \tag{2.3}$$

where $c\in\mathbb{E}, F\in\mathcal{E}$ and $G\in\mathcal{B}(\mathbb{R}^N)$. Exponential transforms play a vital role in the study of large deviations [22, 59]. In fact, the exponential transform of the transition kernel together with its largest eigenvalue eventually yield an LDP [35]. Therefore, define the following exponential transform of the transition kernel defined in (2.3), for all $c\in\mathbb{E}$, $F\in\mathcal{E}$, and $s\in\mathbb{R}^N$,

$$\tilde{L}(c,F;s) := \int_{\mathbb{R}^N} L(c, F\times \mathrm{d}y)\exp(sy). \tag{2.4}$$

Our strategy is to first establish an LDP for $\{(C_k,Q_k)\}_{k\in\mathbb{N}_0}$ making use of standard results from probability theory and then, use that to arrive at an LDP for the waiting times in the queuing system via the contraction principle of large deviations. Therefore, we introduce the following notation that we make use of while applying the contraction principle. For $y\in\mathbb{R}$, define

$$\Upsilon_N(y) := \cup_{F\in\{S\subseteq[N]:S\neq\emptyset\}} G_F, \tag{2.5}$$

where

$$G_F := B_1\times B_2\times\ldots\times B_N \text{ such that } B_i=\begin{cases}\{y\} & \text{if } i\in F,\\ \mathbb{R}\setminus[y,\infty) & \text{if } i\in[N]\setminus F\end{cases}.$$

The set $\Upsilon_N(y)$ is the union of all N-fold Cartesian products of sets at least one of which is $\{y\}$ and all others are $(-\infty, y)$. For example,

$$\Upsilon_2(y)=\{y\}\times(-\infty,y)\bigcup(-\infty,y)\times\{y\}\bigcup\{(y,y)\}.$$

Note that, for each $y\in\mathbb{R}$, the set $\Upsilon_N(y)$ is a Borel set. We need to make some additional technical assumptions. We list them below before presenting Theorem 1.

Technical Assumptions

A1 (Recurrence) The process $\{C_k\}_{k\in\mathbb{N}_0}$ is an aperiodic, irreducible Markov chain with respect to some maximal irreducibility measure and there exists a probability measure ν on $(\mathbb{E}\times\mathbb{R}^N, \mathcal{E}\times\mathcal{B}(\mathbb{R}^N))$, an integer m, and real numbers $0<b_0\le b_1<\infty$ such that

$$b_0\nu(F\times G)\le L^m(x, F\times G)\le b_1\nu(F\times G),$$

where $L^m(x, F\times G) := \mathsf{P}((C_m,Q_m)\in F\times G\mid C_1=x)$, for each $x\in\mathbb{E}, F\in\mathcal{E}$ and $G\in\mathcal{B}(\mathbb{R}^N)$.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

A2 **(Exponential transform and openness of its effective domain)** Consider the exponential transform of ν,

$$\tilde{\nu}(F, s) \coloneqq \int_{\mathbb{R}^N} \nu(F \times \mathrm{d}y) \exp(sy). \tag{2.6}$$

We assume that $\mathcal{D} \coloneqq \mathcal{D}\tilde{\nu}(\mathbb{E}, .)$ is open, treating $\tilde{\nu}(\mathbb{E}, .)$ as a function on $\mathbb{R}^N$. The openness renders analyticity and essential smoothness to the logarithm of the maximal, simple eigenvalue of the transformed kernel $\tilde{L}$ in (2.4).

A3 **(Stability)** For stability of the queuing system, we assume $\max_{n\in[N]} \mathsf{E}[X_{n,1}] < 0$.

A4 **(Existence of cumulants)** Allowing possibly infinite values, define, for $s \in \mathbb{R}, n \in [N]$,

$$\lambda_k^{(n)}(s) \coloneqq k^{-1} \log \mathsf{E}[\exp(sX_{n,k})],$$
$$\lambda^{(n)}(s) \coloneqq \lim_{k\to\infty} k^{-1} \log \mathsf{E}[\exp(sX_{n,k})].$$

To exclude pathological cases, we assume that the effective domains of $\lambda_k^{(n)}$ and $\lambda^{(n)}$ include common open interval containing 0. This moment condition is required for the establishment of an LDP.

Theorem 1 (Large Deviations Principle)**.** *Assume **A1** (uniform recurrence), **A2** (openness of the effective domain of the exponential transform), **A3** (stability of the system), and **A4** (existence of cumulants) listed above. Then, for each $\theta \in \mathcal{D}$ defined in **A2**, the transformed kernel $\tilde{L}$ in (2.4) has a maximal, real, simple eigenvalue $\lambda(\theta)$. Moreover, the waiting times W_k satisfy a large deviations principle with a good rate function $J : \mathbb{R} \to \mathbb{R}$,*

$$\limsup_{k\to\infty} k^{-1} \log \mathsf{P}(W_k \in B) \le -\inf_{y\in \mathrm{Cl}\, B} J(y) \tag{2.7}$$
$$\liminf_{k\to\infty} k^{-1} \log \mathsf{P}(W_k \in B) \ge -\inf_{y\in \mathrm{Int}\, B} J(y), \tag{2.8}$$

for all $B \in \mathcal{B}(\mathbb{R})$, where

$$J(y) \coloneqq \inf_{x\in\Upsilon_N(y)} \Lambda^*(x), \text{ and } \Lambda^*(x) \coloneqq \sup_{z\in\mathbb{R}^N} \{zx - \log\lambda(z)\}.$$

The proof of Theorem 1 follows by first establishing an LDP for $\{(C_k, Q_k)\}_{k\in\mathbb{N}_0}$ using [35, 49] and then applying the contraction principle. For the sake of completeness we provide it in Appendix A. Once an LDP for $\{(C_k, Q_k)\}_{k\in\mathbb{N}_0}$ has been established, the idea is to treat the waiting times in the FJ system as a continuous mapping of the Markovian sample paths $\{(C_k, Q_k)\}_{k\in\mathbb{N}_0}$. Hence, the use of the contraction principle. The contraction principle is crucial because it captures the FJ-inherent synchronization constraint. Theorem 1 provides estimates of probabilities of rare events such as the waiting times making large deviations from its mean value. In particular, for events B that are P-continuous (*i.e.*, $\mathsf{P}(\mathrm{Bnd}\, B) = 0$), as many events of practical interest are, we can straightforwardly approximate their probabilities by the precise exponential estimates of the form $\exp\left(-k \inf_{y\in B} J(y)\right)$. Moreover, the rate function J is unique and therefore, uniquely characterizes the asymptotic behavior of the waiting times [22, 28, 59]. Remarkable that it is possible to estimate probabilities of rare events under mild technical conditions **A1**, **A2**, **A3** and **A4**. For practical purposes, however, the computation of the rate function J involves the joint distribution of Q_k, which, in turn, involves the joint distribution of the inter-arrival times and the service times at different servers. This computation may not be easy to perform for arbitrary choices of probability distributions of the inter-arrival times and the services times. Therefore, in the next section, we make a few simplifying assumptions for the sake of computability, and provide a computable

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

upper bound on the tail probabilities of the steady-state waiting times. The bound is derived as a by-product of the large deviations result.

2.4 Simplifications for computability: Probabilistic bounds on waiting times

In addition to **A1**, **A2**, **A3** and **A4**, we assume that conditional on $\{C_k = c\}$, the servers act independently. This entails that the processes $\{(C_k, X_{n,k})\}_{k\in\mathbb{N}}$, for each $n \in [N]$ are Markov-additive processes on $(\mathbb{E}\times\mathbb{R}, \mathcal{E}\times\mathcal{B}(\mathbb{R}))$. Their transition kernels are defined as,

$$K_n(c, F\times G) := \mathsf{P}((C_1, X_{n,1}) \in F\times G \mid C_0 = c), \tag{2.9}$$

for $n \in [N]$, where $c \in \mathbb{E}$, $F \in \mathcal{E}$ and $G \in \mathcal{B}(\mathbb{R})$. Note the difference to (2.3). Also, define the corresponding exponential transforms

$$\tilde{K}_n(c, F; s) := \int_{\mathbb{R}} K_n(c, F\times \mathrm{d}x)\exp(sx), \ \forall\, n \in [N]. \tag{2.10}$$

We construct martingales using the largest eigenvalues of the transformed kernels, and then apply the celebrated Doob's martingale inequality on each of $X_{n,k}$ for $n \in [N]$. This step essentially yields bounds on server-specific waiting times. Coupled with the assumption of conditional independence of the servers, we obtain an upper bound on the tail probability of the steady-state waiting time of the entire queueing system. These ideas are made precise in the proof of the following theorem providing upper bound on the tail probability of the steady-state waiting times in a Fork-Join system with N heterogeneous work-conserving servers.

Theorem 2 (Upper bound on the tail probabilities of the steady-state waiting time)**.** *Consider an FJ system with N parallel work-conserving servers, as described in Section 2.2. Then, we have*

(1) For all $n \in [N]$ and $s \in \mathcal{D}\lambda^{(n)}$, $\exp\left(\lambda^{(n)}(s)\right)$ is the simple maximal eigenvalue of $\tilde{K}_n$, and the corresponding right eigenfunction $\{r_n(c, s);\ c \in \mathbb{E}\}$ satisfying

$$\exp\left(\lambda^{(n)}(s)\right) r_n(c, s) = \int_{\mathbb{R}} \tilde{K}_n(c, \mathrm{d}\tau; s) r_n(\tau, s),$$

is positive and bounded above.

(2) The tail probabilities of the steady-state waiting times defined in (2.2) are bounded above by

$$\mathsf{P}(W \geq w) \leq \sum_{n\in[N]} \phi_n(\theta_n)\exp(-\theta_n w), \tag{2.11}$$

where $\theta_n := \sup\{s > 0 \mid \lambda^{(n)}(s) \leq 0\}$ and $\phi_n(s) := \operatorname{ess\,sup}\{\mathbb{1}(X_{n,1} > 0)/r_n(C_1, s)\}$, after having normalized $r_n(., \theta_n)$ so that $\mathsf{E}[r_n(C_0, \theta_n)] = 1$, for each $n \in [N]$.

Note that the existence of the simple maximal eigenvalue is guaranteed by [30, Chapter III, Theorem 10.1]. The proof of Theorem 2 follows by extending results for Markov-additive processes from probability literature (see, *e.g.*, [35, Lemma 3.1 and 3.2] and also [23]). However, for the sake of completeness, it is provided in Appendix B. Essentially, we first manufacture server-specific martingales exploiting the exponential transforms $\tilde{K}_n$. The martingales capture the average behavior of $X_{n,k}$ for each $n \in [N]$ as a stochastic process in k. The normalization of the right eigenfunction r_n is done to ensure the martingales have mean unity. Finally, the upper bound on the tail probability of the steady-state waiting time is obtained by combining the server-specific martingales and applying Doob's maximal inequality. Theorem 2 is central to all the application scenarios that we consider in this paper. The quantity θ_n is called the decay rate of the n-th server, and the quantity $\tilde{\theta} := \min_{n\in[N]}\theta_n$ is defined to be the decay rate of the system. The latter definition is motivated from the principle of largest exponent in large deviations theory [22, Lemma 1.2.15], which roughly states that, on an exponential scale, the effective rate of a sum of finitely many

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

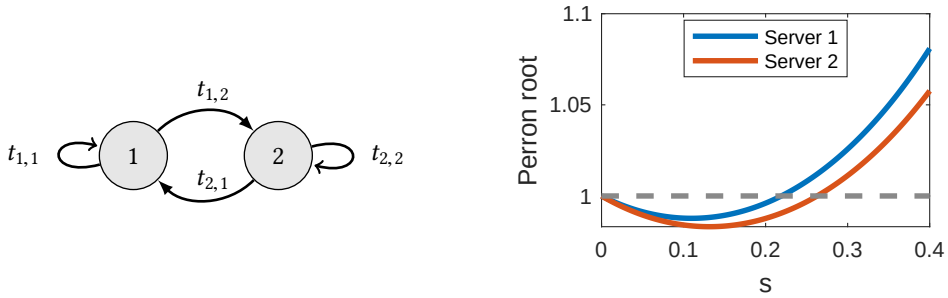

Fig. 4. **(Left)** The modulating Markov chain C_k in Example 1 has state space $\mathbb{E} = \{1, 2\}$. The arrows depict possible transitions between the states of the Markov chain. The numbers $t_{i,j}$'s are the corresponding transition probabilities. **(Right)** The largest eigenvalue or the Perron root of the transformed matrices in (2.12) as a function of s, the free parameter of the eigenvalues. The server-specific decay rates θ_1 and θ_2 are found by checking where the Perron root crosses the horizontal straight line corresponding to unity.

sequences is governed by the maximum of them. This supports the intuition that the system is constrained by the weakest (slowest) of the servers. The quantities ϕ_n's are called prefactors. Since the decay rate captures all information about the performance of the system, the prefactors here have not been optimized and are conservatively chosen.

The bound provided in Theorem 2 is computable. An interesting observation is that, given the transition kernel T of the Markov chain C_k alone, one can view the transformation defined in (2.10) as a transformation of T also. This point of view is useful for computational purposes. In the following, we provide two illustrations.

Example 1 (Correlated exponential inter-arrival and service times)**.** Suppose there are two heterogeneous servers labeled 1 and 2. We are interested in modeling two different environments, *i.e.*, we set $\mathbb{E} = \{1, 2\}$ (see Figure 4). In keeping with Figure 3, we assume the inter-arrival times and the services times at the n-th server are exponentially distributed with rates $\lambda_{i,j}$ and $\mu_{i,j}^{(n)}$ respectively, when the underlying Markov chain C_k transitions from state i to state j, for $i, j, n = 1, 2$. The λ's and the μ's are taken to be strictly positive to avoid trivialities. Assume the inter-arrival times and the service times are independent, conditional on the Markov chain. Let $T := ((t_{i,j}))_{i,j=1,2}$ denote the transition probability matrix of C_k. Then, for $n = 1, 2$, the random variable $X_{n,1}$ is a difference of two exponential random variables, and therefore, $K_n(c_i, \{c_j\} \times B) = t_{i,j} \int_B f_n(i, j; y)\, \mathrm{d}y$ (see Figure 3), where

$$f_n(i,j;y) := \begin{cases} \left(\frac{1}{\mu_{i,j}^{(n)}} + \frac{1}{\lambda_{i,j}}\right)^{-1} \exp\left(\lambda_{i,j} y\right) & \text{if } y \le 0, \\ \left(\frac{1}{\mu_{i,j}^{(n)}} + \frac{1}{\lambda_{i,j}}\right)^{-1} \exp\left(-\mu_{i,j}^{(n)} y\right) & \text{if } y > 0. \end{cases}$$

The transformed kernel is the conditional Moment Generating Function (MGF) of the random variable $X_{n,1}$. Simple calculation yields the exponentially transformed kernels $\tilde{K}_n(c_i, \{c_j\}; s) = t_{i,j} \left(\frac{\mu_{i,j}^{(n)}}{\mu_{i,j}^{(n)} - s}\right) \left(\frac{\lambda_{i,j}}{\lambda_{i,j}+s}\right)$. The decay rates θ_n's are then obtained by computing the largest eigenvalues

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

of the transformed transition matrices

$$T = \begin{pmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \end{pmatrix} \longmapsto \begin{pmatrix} t_{1,1}\left(\frac{\mu_{1,1}^{(n)}}{\mu_{1,1}^{(n)}-s}\right)\left(\frac{\lambda_{1,1}}{\lambda_{1,1}+s}\right) & t_{1,2}\left(\frac{\mu_{1,2}^{(n)}}{\mu_{1,2}^{(n)}-s}\right)\left(\frac{\lambda_{1,2}}{\lambda_{1,2}+s}\right) \\ t_{2,1}\left(\frac{\mu_{2,1}^{(n)}}{\mu_{2,1}^{(n)}-s}\right)\left(\frac{\lambda_{2,1}}{\lambda_{2,1}+s}\right) & t_{2,2}\left(\frac{\mu_{2,2}^{(n)}}{\mu_{2,2}^{(n)}-s}\right)\left(\frac{\lambda_{2,2}}{\lambda_{2,2}+s}\right) \end{pmatrix} \text{ for } n = 1, 2. \tag{2.12}$$

In Figure 4, we plot θ_1 and θ_2 as functions of s and show how the decay rates are obtained. Let r_1 and r_2 denote the corresponding right eigenvectors after having carried out the normalization to get $\mathsf{E}[r_1(C_0, \theta_1)] = 1$, and $\mathsf{E}[r_2(C_0, \theta_2)] = 1$. Denoting the initial distribution of the chain C_k by $\pi = (\pi_1, \pi_2)$, the normalization amounts to setting $r_1(1, \theta_1)\pi_1 + r_1(2, \theta_1)\pi_2 = 1$ and $r_2(1, \theta_2)\pi_1 + r_2(2, \theta_2)\pi_2 = 1$. Because of the exponential assumption, the prefactors are given by $\phi_1(\theta_1) = \max(1/r_1(1, \theta_1), 1/r_1(2, \theta_1))$, and $\phi_2(\theta_2) = \max(1/r_2(1, \theta_2), 1/r_2(2, \theta_2))$. Finally, following (2.11), we arrive at the bound $\mathsf{P}(W \geq w) \leq \phi_1(\theta_1) \exp(-\theta_1 w) + \phi_2(\theta_2) \exp(-\theta_2 w)$.

Parameters chosen in Figure 4 are as follows (shown up to four decimal points):

Parameter	Value
T	$\begin{pmatrix} 0.6307 & 0.3693 \\ 0.7668 & 0.2332 \end{pmatrix}$
π	$(0.0772, 0.9228)$
$(\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}, \lambda_{2,2})$	$(0.8842, 0.8784, 0.8930, 0.8338)$
$(\mu_{1,1}^{(1)}, \mu_{1,2}^{(1)}, \mu_{2,1}^{(1)}, \mu_{2,2}^{(1)})$	$(1.0782, 1.0635, 1.1632, 1.1629)$
$(\mu_{1,1}^{(2)}, \mu_{1,2}^{(2)}, \mu_{2,1}^{(2)}, \mu_{2,2}^{(2)})$	$(1.1578, 1.1011, 1.1705, 1.1271)$

All parameters are chosen randomly satisfying the technical assumptions **A1**, **A2**, **A3**, and **A4**. In this example, the first server is weaker than the second. Therefore, θ_1 (= 0.2192) is smaller than θ_2 (= 0.2628). Having obtained $(\phi_1, \phi_2) = (1.0014, 1.0005)$, the bound is given by $\mathsf{P}(W \geq w) \leq 1.0014 \exp(-0.2192w) + 1.0005 \exp(-0.2628w)$. ■

Example 2 (Modulation on an uncountable state space $\mathbb{E}$)**.** Similar to Example 1, let us assume there are two heterogenous servers labeled 1 and 2. However, in contrast to Example 1, we assume the Markov chain has an uncountable state space, *e.g.*, an interval $[a, b]$. Conforming to the dependence structure dictated by the graphical model shown in Figure 3, we assume the inter-arrival times and the services times at the n-th server are exponentially distributed with strictly positive rate functions $\lambda(x, y)$ and $\mu^{(n)}(x, y)$ respectively, when the underlying Markov chain C_k transitions from state x to state y, for $n = 1, 2$ and $x, y \in [a, b]$. For simplicity, we also assume the inter-arrival times and the service times are independent, conditional on the Markov chain. The transition kernel of C_k is denoted by T, as before. The choices of the rate functions λ and $\mu^{(n)}$, and the transition kernel T depend on the specific application scenario. For instance, if the environment in question does not vary drastically for two consecutive incoming jobs, we may choose a Gaussian kernel with a small variance or a Laplace kernel with a small scale parameter, both restricted to $[a, b]$. We can control how rapidly the environment changes via the variance parameter of the Gaussian kernel or the scale parameter of the Laplace kernel. In this example, let us take T to be the Laplace kernel with scale parameter σ. Then, doing similar calculation as in Example 1, we get

$$\tilde{K}_n(x, F; s) = \frac{1}{u(x)} \int_F \exp\left(-\frac{|y - x|}{\sigma}\right)\left(\frac{\mu^{(n)}(x, y)}{\mu^{(n)}(x, y) - s}\right)\left(\frac{\lambda(x, y)}{\lambda(x, y) + s}\right) \mathrm{d}y,$$

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

where $u(x) = \int_a^b \exp\left(-\frac{|y-x|}{\sigma}\right) \mathrm{d}y$, and $x \in [a, b]$. Given the choices of the rate functions λ and $\mu^{(n)}$, we find the maximal eigenvalue and the corresponding right eigenfunction of $\tilde{K}_n$ to obtain the bound given in (2.11). The eigenvalue and the right eigenfunction are usually found as a solution to the integral equation mentioned in Theorem 2. Note that finding closed-form expressions may be infeasible for arbitrary choices of the rate functions λ and $\mu^{(n)}$. In such a situation, we resort to numerical methods [3, 54]. A standard approach is to approximate the integral using samples [8].

For the sake of simplicity, let us assume that the environment only modulates the arrival process. In particular, when the Markov chain is in state x, the inter-arrival times are assumed to be exponentially distributed with rate x, *i.e.*, $\lambda(x, y) = x$. The task of finding the maximal eigenvalue of the transformed kernel $\tilde{K}_n$ is equivalent to solving the following integral equation for $\lambda^{(n)}$, and r_n,

$$\int_a^b \exp\left(-\frac{|x-y|}{\sigma}\right) r_n(x,s)\,\mathrm{d}x = U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s),$$

where the conditional MGF accounting for the service process as well as the constants have been absorbed into the function $U_n(y,s) = \left(1+\frac{s}{y}\right)\left(1-\frac{s}{\mu^{(n)}}\right)u(y)$. To solve the above integral equation, we differentiate it twice with respect to y to obtain the following differential equation,

$$r_n''(y,s) + 2\frac{U_n'(y,s)}{U_n(y,s)} r_n'(y,s) + \left(\frac{U_n''(y,s)}{U_n(y,s)} - \frac{1}{\sigma^2}\left(1 - \frac{2\sigma\exp\left(-\lambda^{(n)}(s)\right)}{U_n(y,s)}\right)\right) r_n(y,s) = 0. \tag{2.13}$$

The derivation of (2.13) is provided in Appendix B. The nonlinear differential equation (2.13) can then be solved numerically. After doing necessary normalization to get $\mathsf{E}[r_n(C_0, \theta_n)] = 1$, for $n = 1, 2$, we obtain the bound using Theorem 2. ■

For ease of computation, in the following we shall consider what is referred to as the "uncoupled" MA process in [35]. This essentially refers to a process with Markov-modulated increments (see Figure 3 and refer to [23]). This is an important class from a practical perspective, specially in the light of recent empirical evidences of burstiness in clusters running MapReduce [19, 33, 39, 61].

2.5 The "uncoupled" case

Suppose the distributions of increments, $X^A_{n,k+1}$, for each $n \in [N]$, do not depend on C_k, conditional on C_{k+1} (see Figure 3). This allows us to find conditional distributions $Q_n(c, B) := \mathsf{P}(X^A_{n,1} \in B \mid C_1 = c)$, for each $n \in [N]$ and for each $c \in \mathbb{E}$ and $B \in \mathcal{B}(\mathbb{R})$. Then, the transformed kernels in (2.10) simplify as follows

$$\tilde{K}_n(c, \mathrm{d}\tau; s) = T(c, \mathrm{d}\tau)\int_{\mathbb{R}} Q_n(\tau, \mathrm{d}z)\exp(sz) = T(c, \mathrm{d}\tau)\mathsf{E}_\tau\left(\exp\left(sX^A_{n,1}\right)\right).$$

Here we use the shorthand notation $\mathsf{E}_\tau\left(\exp\left(sX^A_{n,1}\right)\right)$ to denote $\mathsf{E}[\exp\left(sX^A_{n,1}\right) \mid C_1 = \tau]$, the moment generating function of $X^A_{n,1}$ conditioned on $\{C_1 = \tau\}$, the event that underlying Markov chain is in state $\tau \in \mathbb{E}$ for the first arrival. We can further simplify the formulas if we make following assumptions[1].

U1 We assume that the service times and the arrival times are independent, conditioned on $\{C_k = c\}$. This yields

$$\tilde{K}_n(c, \mathrm{d}\tau; s) = T(c, \mathrm{d}\tau)\mathsf{E}_\tau\left(\exp\left(sS_{n,1}\right)\right)\mathsf{E}_\tau\left(\exp\left(-sA_1\right)\right). \tag{2.14}$$

[1]These assumptions are only for the sake of simplification of computation, and are not necessary for the bounds of the general case.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

U2 Further, if the increments $X^A_{n,1}$ take positive values with non-zero probability for any conditioning of C_k, then the essential supremums in Theorem 2 simplify to

$$\phi_n(s) = \sup_{c\in\mathbb{E}}\{1/r_n(c,s)\}. \tag{2.15}$$

With these simplifications the computation of the bound on the tail probabilities of the waiting times is easier. We present the procedure in the form of pseudocode 1 for ease of understanding and implementation. Note that pseudocode 1 requires numerical solution methods when closed-form analytic expressions are difficult to obtain.

ALGORITHM 1: Pseudocode for work-conserving systems

Input : Transition kernel T, and the MGFs $\mathsf{E}_\tau\left(\exp\left(sS_{n,1}\right)\right), \mathsf{E}_\tau\left(\exp\left(-sA_1\right)\right)$
Output: The decay rates θ_n and the prefactors ϕ_n

if *A1 and A2 and A3 and A4* **then**
  for $n \in [N]$ **do**
    Transform T to get $\tilde{K}_n(c, \mathrm{d}\tau; s)$ (see (2.14));
    $\exp\left(\lambda^{(n)}(s)\right) \leftarrow$ maximal eigenvalue of $\tilde{K}_n(c, \mathrm{d}\tau; s)$;
    $\theta_n \leftarrow \sup\{s > 0 \mid \lambda^{(n)}(s) \le 0\}$;
    Normalize $r_n(., \theta_n)$ so that $\mathsf{E}[r_n(C_0, \theta_n)] = 1$;
    $\phi_n(\theta_n) \leftarrow \sup_{c\in\mathbb{E}}\{1/r_n(c, \theta_n)\}$;
  end
end

2.6 Renewal Processes as a special case

Several previously known results on FJ systems where a renewal arrival process was assumed (*e.g.*, the renewal cases in [42, 55]) can be retrieved by simply setting $\mathbb{E} = \{1\}$. In this case, following Algorithm 1, the bounds turn out to be

$$\mathsf{P}(W \ge w) \le \sum_{n\in[N]} \exp\left(-\theta_n w\right), \tag{2.16}$$

where

$$\theta_n = \sup\{s > 0 \mid \mathsf{E}[\exp\left(sS_{n,1}\right)]\mathsf{E}[\exp\left(-sA_1\right)] \le 1\}.$$

The technique used in [42, 55] to prove inequalities like (2.16) is also based on a martingale construction and an application of the Doob's inequality. In fact, their construction can be seen as a special case of the martingale in (A.2) for general MA processes. However, it should be noted that it does not immediately generalize to an LDP for our MA process setting without careful handling of additional technicalities.

Remark 1. The bound in (2.11) can also be used to derive an upper bound on the mean waiting time for the work-conserving system as follows

$$\mathsf{E}[W] \le \sum_{n\in[N]} \frac{\phi_n(\theta_n)}{\theta_n}. \tag{2.17}$$

So far we have considered only work-conserving servers. However, there are situations when the assumption of work-conservingness is not tenable. In particular, there are many real-life application scenarios where the servers are so called "blocking" in nature. Such a server waits for (some of

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. Provisioning and performance evaluation of parallel systems with output synchronization. *In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.*

the) other servers to finish servicing the tasks of the current job before taking up the next job. This entails forced idleness resulting in higher waiting times. In the next section, we show that our framework is applicable to (partially) blocking systems as well.

3 PURGING (N, r)-FJ SYSTEMS AS HYPOTHETICAL SINGLE-SERVER QUEUES

Redundancy techniques have become increasingly popular over the last few years as a tool to decrease latency. Such techniques typically create redundant tasks for each job with the hope of achieving smaller response times as the creation of redundant jobs mitigates the synchronization constraint at the output (see Figure 2) either entirely (in case of full replication) or partially (in case of partial replication, *e.g.*, (n, k) Fork-Join in [38]). In this section, we show that our Markov-additive framework for a general FJ system developed in Section 2 can be applied to study a purging (N, r) replication strategy in an FJ system.

An (N, r)-replication strategy with purging assigns tasks of each incoming job to each of the N available servers (one task per server). The tasks are created in such a way that a job leaves the system as soon as *any* $r \in [N]$ of its N tasks are executed (see [38]). Therefore, there is only a partial synchronization constraint at the output (there is no synchronization if $r = 1$). Purging enforces that as soon as the first r servers execute their tasks, all other servers immediately discontinue their tasks at that time and take up the task of the next job. In that sense, the servers are partially blocking. Only the first $r - 1$ servers that complete the tasks of a given job assigned to them wait until one more server completes its task of the current job (at which point r tasks of the current job are completed) and then take up the task of the next job[2]. An (N, r)-FJ system with a purging replication strategy can be viewed as a hypothetical work-conserving system with a single server whose service times are now distributed as $\tilde{S}_i =_{\mathrm{D}} \iota_r\{S_{n,i} \mid n \in [N]\}$. The symbol ι_r denotes the r-th order statistic. Therefore, the steady-state waiting time has the following representation

$$\tilde{W} =_{\mathrm{D}} \max_{k \in \mathbb{N}_0} Z_k \text{ with } Z_k := \sum_{i=1}^{k} Z_i^A, \tag{3.1}$$

where $Z_i^A := \iota_r\{S_{n,i} \mid n \in [N]\} - A_i$ for all $i \in \mathbb{N}$ and set $Z_0 := 0$. Also define

$$\rho_k(s) := k^{-1} \log \mathsf{E}[\exp(sZ_k)], \text{ and } \rho(s) := \lim_{k \to \infty} k^{-1} \log \mathsf{E}[\exp(sZ_k)].$$

The upper bound on the tail probabilities of the steady-state waiting times can then be derived directly from Theorem 2. Therefore, we have the following corollary to Theorem 2. The transformed kernel $\tilde{L}$ is calculated using (2.4).

Corollary 1 (Replication with purging)**.** *Consider an (N, r)-FJ system governed by a purging replication strategy. Then, we have*

(1) For all $s \in \mathcal{D}\rho$, $\exp(\rho(s))$ is the simple maximal eigenvalue of $\tilde{L}$ and the corresponding right eigenfunction $\{\tilde{r}(c, s);\ c \in \mathbb{E}\}$ satisfying

$$\exp(\rho(s))\, \tilde{r}(c, s) = \int_{\mathbb{R}} \tilde{L}(c, \mathrm{d}\tau; s)\tilde{r}(\tau, s),$$

is positive and bounded above.

[2]The case when the servers do not wait at all is not directly reducible to a hypothetical single-server system without modifying the definition of the waiting time in Section 2.2.1 unless $r = 1$. We do not analyse such systems in this paper. Nevertheless, one can argue that the waiting times in those work-conserving (N, r) systems are expected to be stochastically dominated by the waiting times in our (N, r)-system with purging. Therefore, at least intuitively, the upper bounds on the tail probabilities of the steady-state waiting times in our system can also be taken as upper bounds on the corresponding tail probabilities of the steady-state waiting times in a work-conserving (N, r)-system. However, the decay rate in our bound will not be optimal for those systems.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

(2) The tail probabilities of the steady-state waiting times are bounded above by

$$\mathsf{P}(\tilde{W} \geq w) \leq \phi(\theta) \exp(-\theta w), \tag{3.2}$$

where $\theta := \sup\{s > 0 \mid \rho(s) \leq 0\}$ *and* $\phi(s) := \operatorname{ess\,sup}\{\mathbb{1}(Z_1 > 0)/\tilde{r}(C_1, s)\}$ *after having normalized* $r(., \theta)$ *so that* $\mathsf{E}[\tilde{r}(C_0, \theta)] = 1$.

We provide illustrative examples for different choices of distributions in the following.

Example 3 (Arbitrary $r \in [N]$, Irwin-Hall inter-arrival times, arbitrary service distributions)**.** Let $\mathbb{E}$ be finite. Suppose at state j of the Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$, the inter-arrival times are Irwin-Hall distributed with parameter $\lambda_j \in \mathbb{N}_0$ and MGF $(\frac{\exp(s)}{s} - 1)^{\lambda_j}$, and the service times at the n-th server are distributed according to an absolutely continuous Cumulative Distribution Function (CDF) $F_{n,j}$. That is, both the inter-arrival times and the service times are modulated. We assume the service times are independent conditionally on $\{C_i = j\}$. Write $F^{(j)} := (F_{1,j}, F_{2,j}, \ldots, F_{N,j})^{\mathsf{T}}$ and $1 - F^{(j)} := (1 - F_{1,j}, 1 - F_{2,j}, \ldots, 1 - F_{N,j})^{\mathsf{T}}$. Then, the distribution of the r-th order statistic $\tilde{S}_i := \iota_r\{S_{n,i} \mid n \in [N]\}$ for the i-th job, conditionally on $\{C_i = j\}$, can be written in terms of the permanents of a matrix (see [10, Theorem 4.1]; also [11, 41] for applications)

$$\mathsf{P}(\tilde{S}_i \leq s \mid C_i = j) = \sum_{l=r}^{N} \frac{1}{l!(N-l)!} \operatorname{per}\left[\frac{F^{(j)}(s)}{l} \frac{1 - F^{(j)}(s)}{N-l}\right],$$

where $\left[\frac{F^{(j)}(s)}{l} \frac{1-F^{(j)}(s)}{N-l}\right]$ is the matrix whose first l columns are $F^{(j)}$ and the last $N - l$ columns are $1 - F^{(j)}$, evaluated at s. The permanent of an $N \times N$ real matrix $B = ((b_{i,j}))_{i,j\in[N]}$ is defined as

$$\operatorname{per} B := \sum_{\sigma \in \mathsf{Sym}([N])} \prod_{i=1}^{N} b_{i,\sigma(i)},$$

where $\mathsf{Sym}([N])$ denotes the class of all permutations of $[N]$. Then, the MGF of r-th order statistic $\tilde{S}_i$ conditionally on the Markov chain being in state j, *i.e.*, $\{C_i = j\}$, is given by the $\mathsf{m}_r[]$ operators acting on $F^{(j)}$ as follows

$$\mathsf{E}[\exp(s\tilde{S}_i) \mid C_i = j] = \mathsf{m}_r\left[F^{(j)}\right](s) := \sum_{k=N-r+1}^{N} (-1)^{k-(N-r-1)} \binom{k-1}{N-r} \mathsf{M}_k\left[F^{(j)}\right](s), \tag{3.3}$$

for $s > 0$, where the $\mathsf{M}_k[]$-operators, for $k \in [N]$, acting on the space of N-dimensional functions each component of which is a valid CDF are defined as

$$\mathsf{M}_k\left[F^{(j)}\right](s) := \sum_{G\in\{A\subseteq[N]:|A|=k\}} \int_0^\infty \left(\prod_{i\in G}\left(1 - F_{i,j}(\frac{1}{s}\ln x)\right)\right) \mathrm{d}x. \tag{3.4}$$

The summation runs over all subsets of $[N]$ with cardinality k. See [41] for elaborate calculations involving order statistics. The required transformation of the transition matrix is given by

$$t_{ij} \to t_{ij} \left(\frac{1 - \exp(-s)}{s}\right)^{\lambda_j} \mathsf{m}_r\left[F^{(j)}\right](s).$$

Denote the largest eigenvalue of the transformed matrix by $\chi_{AS}^{(N,r)}$. The decay rate is found as

$$\theta = \sup\{s > 0 \mid \chi_{AS}^{(N,r)}(s) \leq 1\}. \tag{3.5}$$

After normalization of the right eigenvector, we compute the bounds on the tail probabilities of the steady-state waiting times using formulas in (3.2). ■

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

Example 4 ($r = 1$, hyperexponential service times)**.** Suppose the arrival process is renewal, *i.e.*, $\mathbb{E} = \{1\}$. Then, instead of solving an eigenvalue problem to find the decay rate, we solve a nonlinear equation involving the MGF and the Laplace transform of the service times and the inter-arrival times. We follow an $(N, 1)$-replication strategy with purging to maximize the benefit of replication (*i.e.*, to minimize the output synchronization constraint). Suppose the service times of the n-th server are independently hyperexponentially distributed with rates $\mu_{n,1}, \mu_{n,2}, \ldots, \mu_{n,k_n}$ and mixing probabilities $p_{n,1}, p_{n,2} \ldots, p_{n,k_n}$, for some $k_n \in \mathbb{N}_0$ and $n \in [N]$. Also assume the inter-arrival times are exponentially distributed with rate λ. Since the minimum of a finite collection of hyperexponential random variables is itself hyperexponentially distributed (see Remark 2 in Appendix C), the decay rate θ in Corollary 1 is found by solving the following equation

$$\left(\sum_{\pi\in[k_1]\times[k_2]\times\cdots\times[k_N]}\prod_{n\in[N]} p_{n,\pi_n}\left(\frac{\sum_{n\in[N]}\mu_{n,\pi_n}}{\sum_{n\in[N]}\mu_{n,\pi_n}-\theta}\right)\right)\left(\frac{\lambda}{\lambda+\theta}\right)=1.$$

The upper bound on the tail probabilities is then found by plugging in the solution θ in (3.2). ■

3.1 Split-merge systems

Split-merge [48] or blocking systems arise naturally in several real-life applications, for instance, when the dispatcher and the task collector in Figure 2 are one and the same unit that assigns new jobs only after the current job is executed. In a parallel computation scenario, the master node, upon arrival of a computation request, may assign intermediate tasks to a number of slave nodes, then wait for all the slave nodes to hand over their intermediate results back to the master node for further aggregation before assigning new computation tasks to the slave nodes. Blocking systems also arise when there needs to be a consensus among the servers regarding the job division before its tasks can be executed.

Example 5 ($r = N$, blocking system)**.** Let $\mathbb{E}$ be finite. Suppose at state j of the Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$, the inter-arrival times are exponentially distributed with parameter λ_j and accordingly, the service times at the n-th server are distributed exponentially with parameter $\mu_{n,j}$. Define $\mu^{(j)} := (\mu_{1,j}, \mu_{2,j}, \ldots, \mu_{n,j})$. Then, the required transformation for the blocking system is given by $t_{ij} \to t_{ij}\beta(\mu^{(j)}; s)\left(\frac{\lambda_j}{\lambda_j+s}\right)$, where β is the MGF of the maximum of N exponentially distributed random variables and is given by (see Remark 3 in Appendix C)

$$\beta(\mu; s) := \sum_{G\in\{A\subset[N]\,|\,A\neq\emptyset\}} (-1)^{|G|+1}\frac{(\sum_{i\in G}\mu_i)}{(\sum_{i\in G}\mu_i)-s}. \tag{3.6}$$

Denote the largest eigenvalue of the transformed matrix by χ^b_{AS}. The decay rate is found as

$$\theta = \sup\{s > 0 \mid \chi^b_{AS}(s) \leq 1\}. \tag{3.7}$$

After normalization of the right eigenvector, we compute the bounds on the tail probabilities of the steady-state waiting times using formulas in (3.2). ■

The equivalence between a blocking system and a system with one work-conserving server has been shown before (see [36, 37, 48]). While the previous works provide bounds on the mean response time, we provide upper bound on the tail probabilities of the steady-state waiting times under a more general set-up. In particular, we allow for changing environments via the Markov-additive process formulation and a broad class of inter-arrival and service time distributions.

Next, in Part B of the paper, we shall apply of our results to different Markov modulation scenarios and provide numerical examples.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

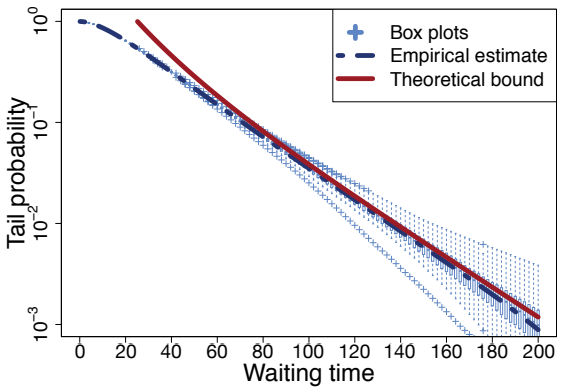

Fig. 5. Comparison of the theoretical bound with Monte Carlo box plots obtained from 10^3 independent simulation runs each with 10^6 jobs. We consider an FJ system with five work-conserving servers. The inter-arrival times are Markov modulated with $\mathbb{E} = \{1, 2, 3\}$. The inter-arrival times are Gamma distributed with randomly chosen parameters.

PART B. APPLICATIONS

4 FJ SYSTEM WITH NON-RENEWAL INPUT

In this section, we describe an FJ system with Markov-modulated inputs. This is principally motivated by recent empirical evidences that reveal burstiness in Internet traffic and also in inputs to MapReduce clusters [19, 33, 39, 61]. In general, to model burstiness, we can assume the inter-arrival times to be modulated by some Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$.

Example 6 (Numerical example: MM inter-arrival times)**.** Suppose the modulating Markov chain takes three distinct values (corresponding to different phases of arrival traffic). In state j of the chain, suppose the inter-arrival times are Gamma distributed with parameters λ_j and k_j. Also assume, the service times at the n-th server are exponentially distributed with parameter μ_n. Then, the transformation in (2.14) is simply

$$\begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{pmatrix} \longmapsto \begin{pmatrix} t_{1,1}(\frac{\lambda_1}{\lambda_1+s})^{k_1} & t_{1,2}(\frac{\lambda_2}{\lambda_2+s})^{k_2} & t_{1,3}(\frac{\lambda_3}{\lambda_3+s})^{k_3} \\ t_{2,1}(\frac{\lambda_1}{\lambda_1+s})^{k_1} & t_{2,2}(\frac{\lambda_2}{\lambda_2+s})^{k_2} & t_{2,3}(\frac{\lambda_3}{\lambda_3+s})^{k_3} \\ t_{3,1}(\frac{\lambda_1}{\lambda_1+s})^{k_1} & t_{3,2}(\frac{\lambda_2}{\lambda_2+s})^{k_2} & t_{3,3}(\frac{\lambda_3}{\lambda_3+s})^{k_3} \end{pmatrix}.$$

Having done the above transformation, the decay rates are found as

$$\theta_n = \sup\{s > 0 \mid \frac{\mu_n}{\mu_n - s}\chi_A(s) \leq 1\}, \tag{4.1}$$

where χ_A is the largest eigenvalue of the transformed matrix. After normalization of the right eigenvector, one obtains the bounds using (2.11). Please see Figure 5 to compare our bounds with empirical CCDFs obtained from Monte Carlo simulations. In addition to the empirical estimates, we also show box plots. As the tail probabilities decrease, the Monte Carlo estimates are based on fewer samples. Therefore, higher variance is observed for smaller tail probabilities. ■

5 PARALLEL SYSTEMS WITH DEPENDENT SERVERS

In this section, we consider an FJ system as described in Section 2 with correlated servers. To be precise, we assume that the service times are modulated by a Markov chain. The motivation behind this is the phase type behavior that service times show due to various exogenous effects. Before furnishing numerical examples, we mention some factors that might engender such a phase-type behavior.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

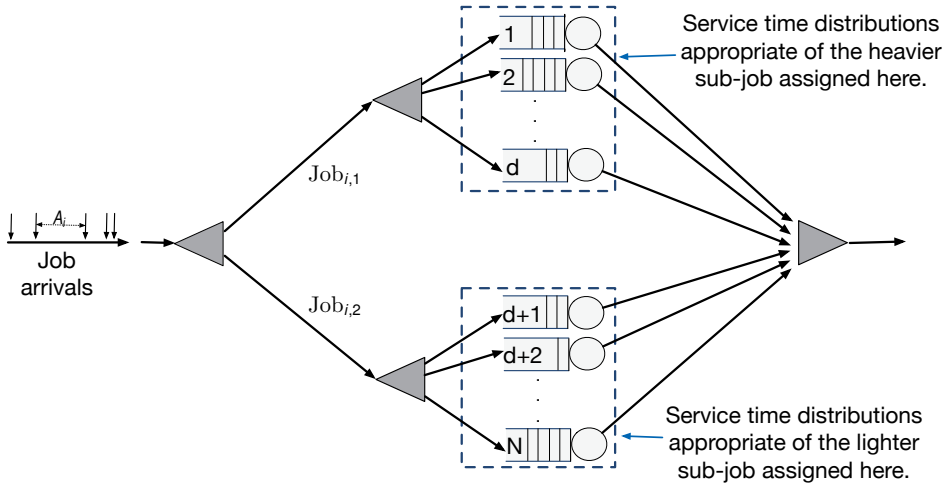

Fig. 6. Single-node FJ system with a provisioning where the heavier part of each incoming job is apportioned in a round robin fashion. Let $\text{Job}_i = (\text{Job}_{i,1}, \text{Job}_{i,2})$, where $\text{Job}_{i,1}$ denotes the heavier sub-job. For instance, for the first job Job_1, the sub-job $\text{Job}_{1,1}$ is allotted to servers $1, 2, \ldots, d$ and $\text{Job}_{1,2}$, to the rest. Then, $\text{Job}_{2,1}$ is allotted to servers $d+1, d+2, \ldots, 2d$ and $\text{Job}_{2,2}$ to the rest, and so on.

Unequal job sizes. Phase-type behavior may arise when the sizes of the incoming jobs are unequal enforcing a change of service time distribution across the servers. Intuitively, heavier jobs demand greater service times in total. This can be modeled by scaling up the service times or the parameters of their distributions whenever a heavier job arrives. For instance, in the context of MapReduce, the job sizes can be time varying. In the context of Multi-path TCP, the packet sizes are usually of different sizes. The modulating chain captures the different job sizes enforcing different service time distributions. The state space of the chain $\mathbb{E}$ can be chosen depending on the particular application under consideration.

Provisioning in MapReduce. The "irregular" service times may also arise due to provisioning, even when the job sizes are constant. Suppose that the incoming jobs are split unequally among the available servers. The rule that decides job division into tasks is termed *provisioning*. Such provisioning can be employed in MapReduce systems to influence waiting times. Consider the following example: Each job consists of two sub-jobs one of which is more demanding than the other. That is, $\text{Job}_i = (\text{Job}_{i,1}, \text{Job}_{i,2})$, where $\text{Job}_{i,1}$ can be assumed to be heavier (more time-consuming) without loss of generality. Now, in order to apportion the burden of the heavier job, devise a variant of the round robin mechanism such that for the first job Job_1, the sub-job $\text{Job}_{1,1}$ is allotted to servers $1, 2, \ldots, d$ and $\text{Job}_{1,2}$, to the rest $N - d$ servers. Then, $\text{Job}_{2,1}$ is allotted to servers $d+1, d+2, \ldots, 2d$ and $\text{Job}_{2,2}$ to the rest, and so on. Mathematically this is equivalent to having a modulating Markov chain that starts at state 1 where it assigns service time distributions appropriate of the heavier job (*e.g.*, scaled service times as explained before) to servers $1, 2, \ldots, d$ and the usual unscaled service time distribution, to the rest, and then jumps with probability one to state 2 where it assigns service time distributions appropriate of the heavier job to servers $d+1, d+2, \ldots, 2d$ and the usual, to the rest. See Figure 6 for a pictorial description of this provisioning.

Modulation in MPTCP. Packet scheduling or load-balancing mechanisms, *e.g.*, [27], could also give rise to correlated service times. The load-balancing algorithm typically decides on the amount of packets to send over each path with the objective of keeping congestion under control. Taking the

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

liberty of mathematical abstraction, we can model such a scenario with a Markov chain (representing the decisions of the load-balancer) that modulates only the service times of the system.

Efficiency differentiation. Servers may themselves have their own high and low efficiency periods that may or may not depend on the state of the other servers, *e.g.*, enforced by energy-saving routines [56]. The service rates may also be modulated by the user. For instance, given a fixed monetary budget, the user of a cloud computing service such as the Amazon AWS may be forced to switch to a less expensive machine (with inferior service rates) when the price of the current machines increase, to meet the budget constraint (*e.g.*, see [57]).

Example 7 (Numerical example: Markov-modulated service times)**.** Motivated by the above scenarios, we now elaborate the bound computation in (2.11). In this example, assume the arrival process is renewal and inter-arrival times are Gamma distributed with rate λ and shape l.

Suppose there are two servers each of which has two efficiency phases, high and low. We model this by two Markov chains modulating the servers, each on state space $\{0,1\}$. For the sake of simplicity, assume that server i is shifted exponentially distributed with rate μ_i and shift a_i or rate κ_i and shift b_i according as its modulating Markov chain is state 0 or 1. The two Markov chains may not be independent. Mathematically this is equivalent to having one single modulating Markov chain on state space $\{0,1\}\times\{0,1\}$. Since the set $\{0,1\}\times\{0,1\}$ has one-to-one correspondence with the set $\{1,2,3,4\}$, we can conveniently rename the states as $(0,0)\mapsto 1$, $(0,1)\mapsto 2$, $(1,0)\mapsto 3$, $(1,1)\mapsto 4$. For the 1st server, following (2.14), we transform

$$\begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4}\\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4}\\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4}\\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4}\end{pmatrix} \longmapsto \begin{pmatrix} t_{1,1}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{1,2}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{1,3}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s} & t_{14}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s}\\ t_{2,1}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{2,2}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{2,3}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s} & t_{24}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s}\\ t_{3,1}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{3,2}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{3,3}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s} & t_{34}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s}\\ t_{4,1}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{4,2}e^{a_1 s}\frac{\mu_1}{\mu_1-s} & t_{4,3}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s} & t_{44}e^{b_1 s}\frac{\kappa_1}{\kappa_1-s}\end{pmatrix}.$$

Transformation for the 2nd server is analogous. Denote the largest eigenvalues of these two transformed matrices by $\chi_S^{(1)}$ and $\chi_S^{(2)}$ respectively. Having done the above transformation, the decay rates are found as

$$\theta_n = \sup\{s>0 \mid \left(\frac{\lambda}{\lambda+s}\right)^l \chi_S^{(n)}(s)\le 1\}. \tag{5.1}$$

After normalization of the right eigenvectors, one finds the bounds on the tail probabilities of the steady-state waiting times using formula in (2.11). To see the quality of our bounds on a bigger state space, we simulated an FJ system with five heterogeneous servers being modulated by a chain having 32 states. See Figure 7 to compare our bounds against empirical CCDFs. ■

6 MARKOV MODULATED ARRIVALS AND SERVICE

In this section, we describe a system where service and inter-arrival times may be dependent. This is essentially a generalization of Sections 4 and 5. All the motivating examples listed in Sections 4 and 5 can be extended to this case to account for generalized application scenarios. While this allows us to endow service times of each server, and the arrival process, separate modulating Markov chains (which can be modeled by one single chain on the Cartesian product space as shown before), we can use this formalism to devise more advanced provisioning by taking into account the current job arrival rate (*i.e.*, set efficiency of servers to "high" during busy period and to "low" otherwise etc.). This paves way for what we call "reactive provisioning."

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

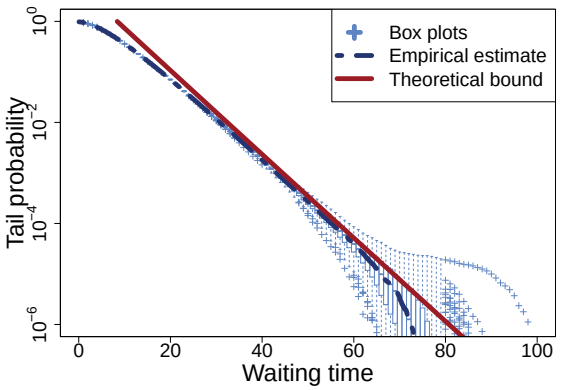

Fig. 7. Comparison of the theoretical bound with Monte Carlo box plots obtained from 10^3 independent simulation runs each with 10^6 jobs. We consider an FJ system with five work-conserving servers. The inter-arrival times are Gamma distributed. The service times are distributed according to shifted exponential distributions, the parameters of which are Markov modulated. The modulating Markov chain takes values in the set $\mathbb{E} = \{1, 2, \ldots, 32\}$. All the parameters randomly chosen.

6.1 Reactive provisioning

We propose to take into account information on the current FJ system environment, *e.g.*, estimates of the arrival intensities, and then modulate, *i.e.*, set service rates accordingly. Such a provisioning is reactive in nature and hence the nomenclature. The changing environment is essentially captured through the modulating Markov chain for the arrivals in this case.

Example 8 (Numerical example: MM inter-arrival and service times)**.** Consider a Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$ capturing the changing environment in the sense that at state j of the Markov chain, the inter-arrival times are Gamma distributed with rate λ_j and shape l_j, and accordingly, the service times at the n-th server are distributed according to a shifted exponential distribution with rate $\mu_{n,j}$ and shift $a_{n,j}$. Then, the required transformation for work-conserving systems is $t_{ij} \to t_{ij} \exp(a_{n,j}s) \left(\frac{\mu_{n,j}}{\mu_{n,j}-s}\right) \left(\frac{\lambda_j}{\lambda_j+s}\right)^{l_j}$, for the n-th server. Let us denote the largest eigenvalue of the transformed matrix for the n-th server by $\chi_{AS}^{(n)}$. Therefore, the decay rates are found as

$$\theta_n = \sup\{s > 0 \mid \chi_{AS}^{(n)}(s) \leq 1\}. \tag{6.1}$$

After normalization of the right eigenvectors, we compute the bounds on the tail probabilities of the steady-state waiting times using the formula in (2.11). To see the quality of our bounds, we simulated the system with the modulating chain having 64 states. See Figure 8 to compare our bounds against empirical CCDFs. ■

7 TRACE-BASED EVALUATION

In this section, we describe a trace-based evaluation of an arrival-aware server provisioning strategy for a MapReduce cluster. In contrast to an arrival-agnostic random strategy, we show that an arrival-aware strategy that adapts the service rates to the intensity of the arriving job stream yields lower job waiting times. To this end, we characterize the arrival process based on a datacenter trace [16].

7.1 Description of the dataset

The datacenter traces used in this work are from Google cluster management software and systems that are publicly available [16]. The traces provide job time-stamps along with other relevant usage data from a Google compute cell, recorded in 2011. We use *job-events* files and subsequently pick the job arrival times by looking at *job ID* field. Further, we randomly select a starting time-stamp and take the subsequent 10^4 consecutive time-stamps as input.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

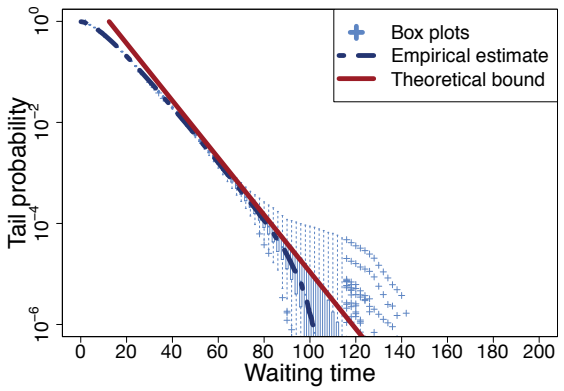

Fig. 8. Comparison of the theoretical bound with Monte Carlo box plots obtained from 10^3 independent simulation runs each with 10^6 jobs. We consider an FJ system with five work-conserving servers. The inter-arrival times as well as the service times are Markov modulated. The modulating Markov chain takes values in the set $\mathbb{E} = \{1, 2, \ldots, 64\}$. The inter-arrival times are Gamma distributed and the service times are distributed according to shifted exponential distributions. All parameters are randomly chosen.

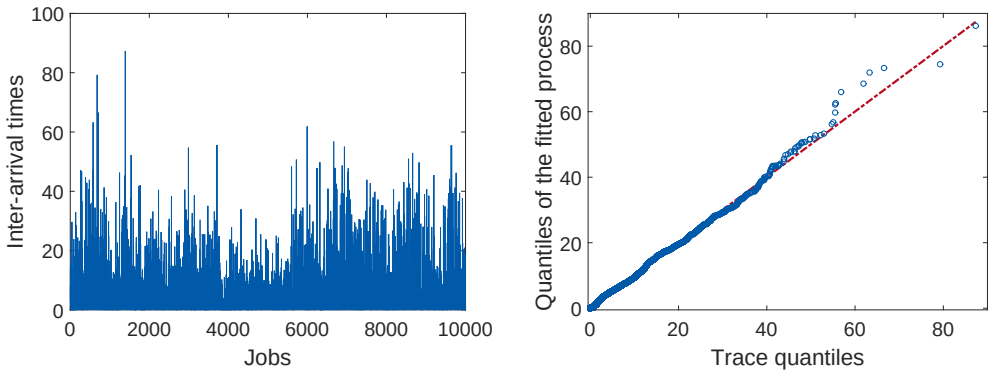

Fig. 9. Numerical study based on recorded traces from [16]. **(Left)** Job inter-arrival times against job numbers. **(Right)** Q-Q plot of the data trace versus simulations of the fitted process.

7.2 Estimation Procedure

We model the given inter-arrival times as time interval between successive jumps of a Markov-modulated Poisson Process (MMPP). In this work, we employ the Baum-Welch algorithm [12, Section 23.3, p. 505] for the maximum likelihood estimation in a Hidden Markov Model to arrive at the estimates for state intensities of the underlying exponential variables and the transition matrix. We set the number of states, which describes the modulation of the arrival process, to 3 to model low, medium and high arrival intensities. The mean arrival times corresponding to these states are estimated to be $(\lambda_1, \lambda_2, \lambda_3) = (9.7638, 3.6215, 0.0054)$ seconds.

7.3 FJ System with arrival-aware Provisioning

Now, we present a numerical test case where we assume a cluster of twenty five heterogeneous servers that are fed by Markov-modulated arrivals with the estimated transition probability matrix and mean inter-arrival times mentioned above. Further, suppose each server can operate in three service rate efficiency settings: high, medium, and low, which may correspond to how the servers are shared with other applications. We assume the servers have exponential service times with different rates that are generated randomly, satisfying the stability conditions. In order to design an

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

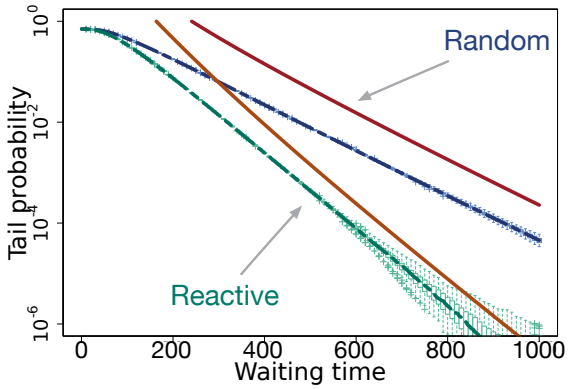

Fig. 10. An FJ system with 25 servers fed by a trace-based fitted arrival process. Service is exponentially distributed with rates randomly assigned. An arrival-aware provisioning (reactive) is designed and compared against a random assignment, which is arrival-agnostic. The arrival-aware provisioning rearranges the service rates of each of the servers so as to have a high service rate when the arrival rate is high. This leads to significantly lower waiting times.

arrival-aware provisioning, we need a rule of assigning these rates to the servers, *i.e.*, modulating the efficiency settings of the servers, when the modulating Markov chain makes a transition. In reality, the modulating chain is unobservable except perhaps for some special cases (*e.g.*, distinguishable job types being represented by the chain). Therefore, to design an arrival-aware provisioning, one needs to estimate the hidden state from observable inter-arrival times. Machine learning techniques can be used to achieve this objective. But in light of **U1**, we do not attempt to do that here. For the sake of demonstration, we devise an illustrative arrival-aware provisioning assuming the chain is visible. A simple arrival-aware provisioning would try to match the service rates (or equivalently, the efficiency settings) with the arrival rates. That is, the provisioning simply assigns the highest service rate when the arrival rate is the highest, assigns the lowest, when the arrival rate is the lowest, and medium otherwise. Mathematically, it just rearranges the service rates so that $\mu_{n,i} \geq \mu_{n,j}$ whenever $i > j$ for all $n \in [N]$, because the arrival rates satisfy $\lambda_1 > \lambda_2 > \lambda_3$. This provisioning, which is reactive in nature, is compared against an arrival-agnostic random assignment in Figure 10. Here, we show simulation box plots along with our bounds on the tail probabilities of the steady-state waiting times, represented by solid lines. Visibly, in case of arrival-aware provisioning, the probability of long waiting times is significantly lowered.

PART C. BACKGROUND AND DISCUSSION

8 RELATED WORK

The works most relevant to ours are [35, 49], where the authors establish a large deviations principle for uniformly recurrent Markov-additive processes. Later on, several queuing theoretic results such as the bounds on the tail probabilities of the steady-state waiting times in [23] have been derived based on [35]. Further, inequalities for the stationary waiting times in *GI/G/k* queues were first shown in [44]. Martingale techniques have been used to derive exponential upper bounds on the tail probabilities of the queue length by means of maximal inequalities in [14, 23]. The authors in [52] also provide exponential bounds on the tail probabilities of the stationary queue size and the virtual delay, the amount of time a data unit would have stayed in the system had it departed at a fixed time, using martingale techniques. Markov fluid traffic models for a single-node constant service-rate queuing system are revisited in [20], where the authors provide upper and lower bounds for the queue size distribution. In [45], the authors apply the theory of Markov-additive processes to estimate the decay rate of the probability that, in a two-node tandem Jackson network, the content of the second buffer exceeds some predefined level before becoming empty. In a recent

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

paper [15], the authors consider a d-node $G/G/1$ tandem queue with renewal input and independent, iid service processes, and characterize the decay rate for the probability of reaching a total of N customers during a busy cycle of the system. Note that these results are not directly applicable to our setup because of the inherent synchronization at the output.

An exact analysis of Fork-Join systems [5, 58] in a general setup is hard [6, 13], and could be carried out for a handful of special cases. Useful bounds have been provided in [6, 9, 55] using probabilistic techniques. Stochastic network calculus has also been used to derive performance upper bounds for FJ systems in [24, 40]. Transient and steady-state solutions of the FJ queue in terms of virtual waiting times are obtained in [43]. Results for the special case of FJ systems with two servers that have exponential service times under Poissonian job arrivals are shown in [26]. The authors in [26] convert equilibrium joint probabilities for the queue lengths into functional equations and derive their asymptotic limits. More recently, homogeneous FJ systems have been analyzed, and an upper bound solution has been provided in [18] using dynamic bubblesort technology [17]. Stability conditions and bounds on the network response times are obtained for acyclic FJ networks based on notions of stochastic ordering in [7]. Linear algebraic tools such as a matrix exponential representation of the maximum order statistic of the service times in split-merge queues have been used to derive approximate results for queue lengths and response times in FJ systems in [25]. In [53], the authors consider a K-node homogeneous FJ system and approximate the response time distribution for a Markovian arrival process and phase-type processing times. Instead of the actual queue lengths in the FJ system, they keep track of queue length differences with respect to the shortest queue rendering a state space reduction. They show this approximation yields accurate results when $K = 2$, but becomes intractable for a larger K. For large values of K, they propose further approximations based on the theory of order statistics and extreme values. In [50], the authors present a product-form approximation of closed FJ systems with interfering requests using the stochastic Petri nets (referred to as RB-n-m replication blocks in [50]). They consider both full and partial forking. Their main tool to approximate the FJ system is the Reversed Compound Agent Theorem (RCAT) for cooperating Markovian processes from [31].

The authors in [60] study the limiting behavior of finite buffer FJ systems. They study how the throughput of a general FJ system with blocking servers behaves as the number of nodes increases to infinity while the processing speed and buffer space of each node remain unaltered. On another note, FJ networks with non-exchangeable tasks under a heavy traffic regime are studied in [2], where the authors show asymptotic equivalence between this network and its corresponding assembly network with exchangeable tasks. The authors of [64] provide necessary and sufficient conditions for throughput scalability for FJ systems with blocking servers as they grow in size.

From the perspective of scheduling, [29] presents various policies in a distributed server system and suggests optimal ones for different situations. Similarly, [34] attempts to quantify the benefits of parallelization in a dispatching system, where jobs, arriving in batches, are assigned to single-server FCFS queues. In [65], the authors study the distributed resource allocation problem in a processing network where the processor nodes are allowed to involve a combination of FJ semantics. They propose a unified modeling framework, and formulate the resource allocation problem as a convex optimization problem. The work in [38] models a cloud computing scenario as an FJ system with identical servers and analyzes different redundancy techniques with a view to reducing latency in a cost-efficient manner. The authors find that the log-concavity of the task service times is crucial for the success of redundancy techniques. Note that the underlying premises in these works are quite dissimilar among themselves and from ours. For instance, the authors in [38] consider expected latency and the mean computing cost as performance metrics; we, on the contrary, focus on tail probabilities of steady-state waiting times under a more general distributional setup and changing environments.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

The works [6, 42, 55] are close to ours and share similar objectives. The seminal work in [6] provides computable bounds for the expected response times under renewal Poissonian arrival and exponential service times. The work in [55] considers two-state Markov-modulated arrival processes, and homogeneous, independent servers. Further, the work in [42] proposes stochastic scheduling in FJ systems to determine how many servers out of a set of given ones should optimally be chosen. In addition, [42] optimizes waiting and response time bounds for heterogeneous FJ systems under a renewal setup. In this work, we provide results for the general case of Markov-additive processes while allowing heterogeneous servers and arbitrary state space. Moreover, we introduce provisioning, a flexible division of jobs into tasks in an FJ system. Finally, we provide examples of FJ systems with Markov-modulated arrivals and service, as well as, a study of an adaptive FJ system that uses estimates of the modulating Markov chains that control the arrivals and the service processes to reactively adapt the task provisioning. The bounds presented in both [55] and [42] can be seen as special cases of Theorem 2 in this paper because they can be obtained by choosing state space $\mathbb{E}$ and the transition kernel (2.3) appropriately.

Performance of MapReduce has been analyzed in [21, 63]. The authors in [21] present MapReduce as a programming model and show that many real world tasks are expressible in this model. On the other hand, [63] points out that Hadoop's performance depends heavily on its task scheduler, which implicitly assumes homogeneous cluster nodes, and that it is adversely impacted in a heterogeneous setup. To address this issue, they propose Longest-Approximate-Time-to-End (LATE) scheduling. Similar optimization problems are surveyed in [32, 51]. Further, in [46] the authors point out MapReduce efficiency issues, especially I/O costs, which still need to be addressed. The efficiency of a MapReduce system, in general, requires tuning a number of parameters. In [4], the author proposes an out-of-the-box automation technique to avoid manual tuning of the parameters. As opposed to our theoretical standpoint, these articles provide a complimentary view from a practical implementation perspective.

9 DISCUSSION AND CONCLUSIONS

In this paper, we provided computable upper bounds on the tail probabilities of the steady-state waiting times based on a Large Deviations Principle, for general FJ queuing systems using a Markov-additive process model. We applied our results to three specific application areas, and also presented a formulation of provisioning, a flexible job division rule. Further, we demonstrated the usefulness of this model by means of a numerical example where we applied our results to a real-world datacenter trace. In this closing section, we highlight the strength of our model by mentioning another way in which our results can be utilized.

9.1 Design of Proactive Mechanisms

Markov-additive processes are capable of modeling not only reactive but also proactive systems. In Section 6, we modeled the changing environment with a Markov chain $\{C_k\}_{k\in\mathbb{N}_0}$ and devised a reactive mechanism. For many applications, reactive mechanisms may be expensive, and it is profitable to be able to anticipate the changes in the environment and act accordingly (*e.g.*, set the service rates). Our Markov-additive process framework allows for such a proactive provisioning (see Figure 3). In this coupled model, the distribution of the increments $X^A_{n,k+1}$, for each $n \in [N]$, will also depend on C_k (as assumed in Example 1). Such a provisioning is promising as it allows for a notion of agility and adaptation in parallel server systems. The preparedness aimed for in proactive provisions could potentially reduce cost and yield a smoother transition.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

A APPENDIX A

Proof of Theorem 1. In the light of **A1**, **A2**, **A3**, and **A4**, the following statements are immediate from known results on large deviations of Markov additive process [35, 49],

B1 For all $\theta \in \mathcal{D}$, the transformed kernel $\tilde{L}$ in (2.4) has a maximal, real, simple eigenvalue $\lambda(\theta)$.

B2 The corresponding right eigenfunction $\{r(c,\theta);\, c \in \mathbb{E}\}$ satisfying

$$\lambda(\theta) r(c,\theta) = \int_{\mathbb{R}} \tilde{L}(c, \mathrm{d}\tau; \theta) r(\tau, \theta),$$

is positive and bounded above.

B3 $\mathcal{D}\lambda = \mathcal{D}\tilde{\nu} = \mathcal{D}$.

B4 Define the filtration

$$\mathcal{F}_k := \sigma(\{(C_i, Q_i)\}_{i\in[k]}), \tag{A.1}$$

the σ-algebra generated by the history of the process $\{(C_i, Q_i)\}_{i\in[k]}$ till and including time point k. Define

$$M_k(\theta) := \exp\left(\alpha Q_k - k\Lambda(\theta)\right) r(C_k, \theta), \tag{A.2}$$

where $\Lambda(\theta) := \log \lambda(\theta)$. The process $M_k(\theta)$ is a martingale with respect to the filtration $\mathcal{F}_k$.

B5 $\lambda(\theta) \to \infty$ as $\theta \to \mathrm{Bnd}\, \mathcal{D}$ or $\|\theta\| \to \infty$. This further implies essential smoothness of Λ. This is important for the application of Ellis' theorem to establish an LDP.

Note that **B1** and **B2** are generalizations of the well known *Perron-Frobenius* theorem for real matrices with positive entries. However, when the state space $\mathbb{E}$ is not finite, one could still obtain similar results. The existence, and properties **B1** and **B2** follow from [30, 35]. Define $\pi : \mathcal{E} \to [0,1]$ to be the invariant probability measure for L defined in (2.3). The following LDP holds [35] for the sequence of probability measures $\{L^k(x, F \times .)\}_{k\in\mathbb{N}_0}$ on $(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$,

$$\limsup_{k\to\infty} k^{-1} \log L^k(x, F \times kG) \le -\inf_{y \in \mathrm{Cl}\, G} \Lambda^*(y), \tag{A.3}$$

$$\liminf_{k\to\infty} k^{-1} \log L^k(x, F \times kG) \ge -\inf_{y \in \mathrm{Int}\, G} \Lambda^*(y), \tag{A.4}$$

for $x \in \mathbb{E}, F \in \mathcal{E}, G \in \mathcal{B}(\mathbb{R}^N)$, where $\Lambda^*(y) := \sup_{z\in\mathbb{R}^N}\{zy - \Lambda(z)\}$ and F is such that $\pi(F) > 0$.

To derive an LDP for the waiting times for our queuing system defined in (2.2), consider the following map $f : \mathbb{R}^N \to \mathbb{R}$ defined as

$$f(s) := \max\{s_1, s_2, \ldots, s_N\}, \tag{A.5}$$

where $s := (s_1, s_2, \ldots, s_N) \in \mathbb{R}^N$. Note that f is a continuous map on $\mathbb{R}^N$ with respect to the topology endowed by the Borel open sets. Therefore, by the contraction principle for continuous maps [22, Theorem 4.2.1] , $\{f(Q_k)\}_{k\in\mathbb{N}_0}$ satisfies an LDP with good rate function

$$J(y) := \inf_{x \in f^{-1}(y)} \Lambda^*(x) = \inf_{x \in \Upsilon_N(y)} \Lambda^*(x), \tag{A.6}$$

where Υ_N is defined in (2.5). Notice that $f(Q_k)$ is simply $W_k := \max(X_{1,k}, X_{2,k}, \ldots, X_{N,k})$ with $W =_{\mathrm{D}} \max_{k\in\mathbb{N}_0} W_k$. Therefore, by virtue of the contraction principle, we get

$$\limsup_{k\to\infty} k^{-1} \log \mathrm{P}(W_k \in B) \le -\inf_{y \in \mathrm{Cl}\, B} J(y)$$

$$\liminf_{k\to\infty} k^{-1} \log \mathrm{P}(W_k \in B) \ge -\inf_{y \in \mathrm{Int}\, B} J(y),$$

for all $B \in \mathcal{B}(\mathbb{R})$. This completes the proof. □

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

B APPENDIX B

Derivation of (2.13). We wish to solve the following integral equation for $\lambda^{(n)}$, and r_n,

$$\int_a^b \exp\left(-\frac{|y-x|}{\sigma}\right) r_n(x,s)\,\mathrm{d}x = U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s),$$

where $U_n(y,s) = \left(1+\frac{s}{y}\right)\left(1-\frac{s}{\mu^{(n)}}\right)u(y)$ and $u(y) = \int_a^b \exp\left(-\frac{|x-y|}{\sigma}\right)\mathrm{d}x$. Our strategy is to differentiate the above integral equation with respect to y twice and then get a nonlinear Ordinary Differential Equation (ODE), which can be solved numerically. Therefore, separating the integral into two parts we get

$$\exp\left(-\frac{y}{\sigma}\right)\int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x + \exp\left(\frac{y}{\sigma}\right)\int_y^b \exp\left(-\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x$$
$$= U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s).$$

Differentiating once with respect to y, we get

$$-\frac{1}{\sigma}\exp\left(-\frac{y}{\sigma}\right)\int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x + \frac{1}{\sigma}\exp\left(\frac{y}{\sigma}\right)\int_a^y \exp\left(-\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x$$
$$= U_n'(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s) + U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n'(y,s).$$

Differentiating once again with respect to y, we get

$$\frac{1}{\sigma^2}\left(\exp\left(-\frac{y}{\sigma}\right)\int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x + \exp\left(\frac{y}{\sigma}\right)\int_y^b \exp\left(-\frac{x}{\sigma}\right) r_n(x,s)\mathrm{d}x - 2\sigma r_n(y,s)\right)$$
$$= U_n''(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s) + 2U_n'(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n'(y,s) + U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n''(y,s).$$

Since the left hand side is $\frac{1}{\sigma^2}U_n(y,s)\exp\left(\lambda^{(n)}(s)\right) r_n(y,s)$, after rearrangement of terms, we get

$$r_n''(y,s) + 2\frac{U_n'(y,s)}{U_n(y,s)}r_n'(y,s) + \left(\frac{U_n''(y,s)}{U_n(y,s)} - \frac{1}{\sigma^2}\left(1 - \frac{2\sigma\exp\left(-\lambda^{(n)}(s)\right)}{U_n(y,s)}\right)\right) r_n(y,s) = 0.$$

□

Proof of Theorem 2. In the light of **A1**, **A2**, **A3**, and **A4**, the following statements are immediate from known results in probability theory, such as [35, 49],

C1 For all $n \in [N]$ and $\theta \in \mathcal{D}\lambda^{(n)}$, $\exp\left(\lambda^{(n)}(\theta)\right)$ is the simple maximal eigenvalue of $\tilde{K}_n$.

C2 The corresponding right eigenfunction $\{r_n(c,\theta);\, c \in \mathbb{E}\}$ satisfying

$$\exp\left(\lambda^{(n)}(\theta)\right) r_n(c,\theta) = \int_{\mathbb{R}} \tilde{K}_n(c,\mathrm{d}\tau;\theta) r_n(\tau,\theta),$$

is positive and bounded above.

C3 For all $n \in [N]$, the functions $\lambda^{(n)}$ and $\lambda_k^{(n)}$, $k \in \mathbb{N}$ are both strictly convex and essentially smooth.

C4 Recall the filtration $\mathcal{F}_k$ defined in (A.1). For each $n \in [N]$, define

$$M_k^{(n)}(s) := \exp\left(sX_{n,k} - k\lambda^{(n)}(s)\right) r_n(C_k,s). \tag{B.1}$$

Then, $M_k^{(n)}(s)$ is a martingale with respect to the filtration $\mathcal{F}_k$.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

The existence, and properties **C1** and **C2** follow from [30, 35]. The statements **C3** and **C4** are proved in [35]. Also, see [23]. In the following, we normalize $r_n(.,\theta)$ so that $\mathsf{E}[r_n(C_0,\theta)] = 1$, for each $n \in [N]$.

Having constructed the martingales $M_k^{(n)}(s)$, we can apply Doob's maximal inequality to obtain

$$\mathsf{P}(\max_{k\in\mathbb{N}_0} X_{n,k} \geq w) \leq \phi_n(s)\exp(-sw), \tag{B.2}$$

for all $s \in \mathcal{D}\lambda^{(n)}$, following Theorem 3 of [23]. In particular, we get

$$\mathsf{P}(\max_{k\in\mathbb{N}_0} X_{n,k} \geq w) \leq \phi_n(\theta_n)\exp(-\theta_n w), \tag{B.3}$$

where $\theta_n := \sup\{s > 0 \mid \lambda^{(n)}(s) \leq 0\}$ and $\phi_n(s) := \operatorname{ess\,sup}\{\mathbb{1}(X_{n,1} > 0)/r_n(C_1,s)\}$, after having normalized $r_n(.,\theta)$ so that $\mathsf{E}[r_n(C_0,\theta)] = 1$, for each $n \in [N]$. The final bound is obtained as follows

$$\mathsf{P}(W \geq w) \leq \sum_{n\in[N]} \mathsf{P}(\max_{k\in\mathbb{N}_0} X_{n,k} \geq w) \leq \sum_{n\in[N]} \phi_n(\theta_n)\exp(-\theta_n w).$$

This completes the proof. □

C APPENDIX C

Remark 2 (Minimum of hyperexponential random variables)**.** Consider a collection of independent random variables $U_1, U_2, \ldots, U_N$, where U_n is distributed according to a hyperexponential distribution with parameters $\mu_{n,1}, \mu_{n,2}, \ldots, \mu_{n,k_n}$ and mixing probabilities $p_{n,1}, p_{n,2} \ldots, p_{n,k_n}$, *i.e.*,

$$\mathsf{P}(U_n \leq u) = 1 - \sum_{i=1}^{k_n} p_{n,i}\exp\left(-\mu_{n,i}u\right).$$

Then, $V = \min_{n\in[N]} U_n$ is also hyperexponentially distributed. To see this, note that

$$\mathsf{P}(V > v) = \prod_{n\in[N]}\sum_{i=1}^{k_n} p_{n,i}\exp\left(-\mu_{n,i}v\right) = \sum_{\pi\in[k_1]\times[k_2]\times\cdots\times[k_N]}\left(\prod_{n\in[N]} p_{n,\pi_n}\right)\exp\left(-\left(\sum_{n\in[N]}\mu_{n,\pi_n}\right)v\right).$$

Therefore, the MGF of V is given by

$$\mathsf{E}[\exp(sV)] = \sum_{\pi\in[k_1]\times[k_2]\times\cdots\times[k_N]}\left(\prod_{n\in[N]} p_{n,\pi_n}\right)\left(\frac{\sum_{n\in[N]}\mu_{n,\pi_n}}{\sum_{n\in[N]}\mu_{n,\pi_n} - s}\right). \tag{B.4}$$

Remark 3 (Maximum of exponential random variables)**.** For the computation of the MGF for the blocking system, we make use of the following statistical result. Consider a finite collection of independent random variables $\{U_n\}_{n\in[N]}$ such that U_n is exponentially distributed with rate μ_n, for each $n \in [N]$. Write $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$. Then, the MGF of $V := \max_{n\in[N]} U_n$ is given by

$$\mathsf{E}[\exp(sV)] = \beta(\mu;s) := \sum_{S\in\{A\subset[N]\mid A\neq\emptyset\}} (-1)^{|S|+1}\frac{(\sum_{i\in S}\mu_i)}{(\sum_{i\in S}\mu_i) - s}. \tag{B.5}$$

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

Proof of Remark 3. The CDF of V is given by $\mathrm{P}(V \leq z) = \prod_{i\in[N]}(1-\exp(-\mu_i z))$, whence we derive the Probability Density Function (PDF) of V as

$$\begin{aligned} f_V(z) &= \sum_{j\in[N]} \mu_j \exp(-\mu_j z)\left[\prod_{i\in[N]\setminus\{j\}}(1-\exp(-\mu_i z))\right] \\ &= \sum_{j\in[N]} \mu_j \sum_{S\in\{A\subset[N]\setminus\{j\}\}} (-1)^{|S|}\exp\left(-z\sum_{i\in S\cup\{j\}}\mu_i\right) \\ &= \sum_{j\in[N]} \mu_j \sum_{S\in\{A\subset[N]|j\in A\}} (-1)^{|S|+1}\exp\left(-z\sum_{i\in S}\mu_i\right) \\ &= \sum_{S\in\{A\subset[N]|A\neq\emptyset\}} (-1)^{|S|+1}\Big(\sum_{i\in S}\mu_i\Big)\exp\left(-z\sum_{i\in S}\mu_i\right). \end{aligned}$$

Therefore, the MGF of V is given by

$$\mathrm{E}[\exp(\theta V)] = \int_0^\infty \exp(\theta z) f_V(z)\,\mathrm{d}z = \sum_{S\in\{A\subset[N]|A\neq\emptyset\}} (-1)^{|S|+1}\frac{(\sum_{i\in S}\mu_i)}{(\sum_{i\in S}\mu_i)-\theta}.$$

This completes the proof. □

ACKNOWLEDGMENTS

This work has been funded by the German Research Foundation (DFG) as part of projects C3 and B4 within the Collaborative Research Center (CRC) 1053 – MAKI. Computational facilities provided by the Lichtenberg - High Performance Computer at TU Darmstadt are gratefully acknowledged. The authors also thank the anonymous reviewers whose constructive comments helped us improve the paper significantly.

REFERENCES

[1] Amazon.com, Inc. 2017. Amazon Web Services (AWS). https://aws.amazon.com/. Accessed: 11-12-2017.
[2] R. Atar, A. Mandelbaum, and A. Zviran. 2012. Control of Fork-Join Networks in Heavy Traffic. In *Allerton*. 823–830.
[3] Kendall E. Atkinson. 2008. *The Numerical Solution of Integral Equations of the Second Kind.* Cambridge University Press.
[4] Shivnath Babu. 2010. Towards Automatic Optimization of MapReduce Programs. In *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10)*. ACM, 137–142.
[5] F. Baccelli and A. M. Makowski. 1989. Queueing models for systems with synchronization constraints. *Proc. IEEE* 77, 1 (Jan 1989), 138–161.
[6] Francois Baccelli, Armand M Makowski, and Adam Shwartz. 1989. The Fork-Join Queue and Related Systems with Synchronization Constraints: Stochastic Ordering and Computable Bounds. *Advances in Applied Probability* (1989), 629–660.
[7] François Baccelli, William A. Massey, and Don Towsley. 1989. Acyclic Fork-join Queuing Networks. *J. ACM* 36, 3 (July 1989), 615–642.
[8] Christopher T. H. Baker. 1977. *The Numerical Treatment of Integral Equations.* Oxford University Press.
[9] S. Balsamo, L. Donatiello, and N. M. Van Dijk. 1998. Bound Performance Models of Heterogeneous Parallel Processing Systems. *IEEE Transactions on Parallel and Distributed Systems* 9, 10 (Oct 1998), 1041–1056.
[10] R. B. Bapat and M. I. Beg. 1989. Order Statistics for Nonidentically Distributed Variables and Permanents. *Sankhya Ser A* 51, 1 (1989), 79–93.
[11] H. M. Barakat and Y. H. Abdelkader. 2004. Computing the moments of order statistics from nonidentical random variables. *Stat. Method. and Appl.* 13, 1 (2004), 15–26.
[12] D. Barber. 2012. *Bayesian Reasoning and Machine Learning.* Cambridge University Press.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

[13] Onno Johan Boxma, Ger Koole, and Zhen Liu. 1994. *Queueing-theoretic Solution Methods for Models of Parallel and Distributed Systems.* Centrum voor Wiskunde en Informatica, Department of Operations Research, Statistics, and System Theory.
[14] E Buffet and NG Duffield. 1994. Exponential Upper Bounds via Martingales for Multiplexers with Markovian Arrivals. *Journal of Applied Probability* (1994), 1049–1060.
[15] Anne Buijsrogge, Pieter-Tjerk de Boer, Karol Rosen, and Werner Scheinhardt. 2017. Large deviations for the total queue size in non-Markovian tandem queues. *Queueing Systems* 85, 3 (01 Apr 2017), 305–312.
[16] Joseph Hellerstein Charles Reiss, John Wilkes. 2013. Google cluster-usage traces. https://github.com/google/cluster-data
[17] Ray Jinzhu Chen. 2001. A hybrid solution of fork/join synchronization in parallel queues. *IEEE Transactions on Parallel and Distributed Systems* 12, 8 (Aug 2001), 829–845.
[18] R. J. Chen. 2011. An Upper Bound Solution for Homogeneous Fork/Join Queuing Systems. *IEEE Transactions on Parallel and Distributed Systems* 22, 5 (May 2011), 874–878.
[19] Yanpei Chen, Sara Alspaugh, and Randy Katz. 2012. Interactive Analytical Processing in Big Data Systems: A Sross-industry Study of MapReduce Workloads. *Proceedings of the VLDB Endowment* 5, 12 (2012), 1802–1813.
[20] F. Ciucu, F. Poloczek, and J. Schmitt. 2016. Stochastic Upper and Lower Bounds for General Markov Fluids. In *2016 28th International Teletraffic Congress (ITC 28)*, Vol. 01. 184–192.
[21] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM* 51, 1 (2008), 107–113.
[22] Amir Dembo and Ofer Zeitouni. 2010. *Large deviations techniques and applications.* Springer-Verlag Berlin Heidelberg.
[23] N. G. Duffield. 1994. Exponential Bounds for Queues with Markovian Arrivals. *Queueing Systems* 17, 3 (1994), 413–430.
[24] M. Fidler and Y. Jiang. 2016. Non-Asymptotic Delay Bounds for (k,l) Fork-Join Systems and Multi-Stage Fork-Join Networks. In *Proceedings of IEEE INFOCOM.*
[25] P. M. Fiorini. 2015. Analytic approximations of fork-join queues. In *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 2. 966–971.
[26] Leopold Flatto and S Hahn. 1984. Two Parallel Queues Created by Arrivals with Two Demands I. *SIAM Journal on Applied Mathematics* 44, 5 (1984), 1041–1053.
[27] Alexander Frömmgen, Amr Rizk, Tobias Erbshäußer Max Weller, Boris Koldehofe, Alejandro Buchmann, and Ralf Steinmetz. 2017. A Programming Model for Application-defined Multipath TCP Scheduling. In *Proc. 18th ACM/IFIP/USENIX Middleware Conf.* 134–146.
[28] Ayalvadi J. Ganesh, Neil O'Connell, and Damon J. Wischik. 2004. *Big Queues.* Springer-Verlag Berlin Heidelberg.
[29] Mor Harchol-Balter, Mark E Crovella, and Cristina D Murta. 1999. On Choosing a Task Assignment Policy for a Distributed Server System. *Journal of Parallel and Distributed Computing* 59, 2 (1999), 204–228.
[30] Theodore Edward Harris. 1963. *The Theory of Branching Processes.* Springer-Verlag Berlin Heidelberg.
[31] Peter G. Harrison. 2003. Turning back time in Markovian process algebra. *Theoretical Computer Science* 290, 3 (2003), 1947 – 1986.
[32] Ibrahim Abaker Targio Hashem, Nor Badrul Anuar, Abdullah Gani, Ibrar Yaqoob, Feng Xia, and Samee Ullah Khan. 2016. MapReduce: Review and Open Challenges. *Scientometrics* (2016), 1–34.
[33] H. Heffes and D. Lucantoni. 1986. A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *IEEE Journal on Selected Areas in Communications* 4, 6 (Sep 1986), 856–868.
[34] Esa Hyytiä and Samuli Aalto. 2013. To Split or Not to Split: Selecting the Right Server with Batch Arrivals. *Operations Research Letters* 41, 4 (2013), 325 – 330.
[35] I Iscoe, P Ney, and E Nummelin. 1985. Large Deviations of Uniformly Recurrent Markov Additive Processes. *Advances in Applied Mathematics* 6, 4 (1985), 373 – 412.
[36] G. Joshi, Y. Liu, and E. Soljanin. 2014. On the Delay-Storage Trade-Off in Content Download from Coded Distributed Storage Systems. *IEEE Journal on Selected Areas in Communications* 32, 5 (May 2014), 989–997.
[37] Gauri Joshi, Emina Soljanin, and Gregory Wornell. 2015. Efficient replication of queued tasks for latency reduction in cloud systems. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on.* IEEE, 107–114.
[38] Gauri Joshi, Emina Soljanin, and Gregory Wornell. 2017. Efficient Redundancy Techniques for Latency Reduction in Cloud Systems. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 2, 2, Article 12 (April 2017), 30 pages.
[39] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, and Ronnie Chaiken. 2009. The Nature of Data Center Traffic: Measurements & Analysis. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference.* ACM, New York, NY, USA, 202–208.
[40] G. Kesidis, Y. Shan, B. Urgaonkar, and J. Liebeherr. 2015. Network Calculus for Parallel Processing. *ACM SIGMETRICS Perform. Eval. Rev.* 43, 2 (Sept. 2015), 48–50.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Wasiur R. Khudabukhsh, Sounak Kar, Amr Rizk, Heinz Koeppl. *Provisioning and performance evaluation of parallel systems with output synchronization.* In ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 4 Issue 1, March 2019.

[41] Wasiur R. KhudaBukhsh, Bastian Alt, Sounak Kar, Amr Rizk, and Heinz Koeppl. 2018. Collaborative Uploading in Heterogeneous Networks: Optimal and Adaptive Strategies. In *IEEE International Conference on Computer Communications (INFOCOM)*. https://ieeexplore.ieee.org/document/8486310

[42] Wasiur R. KhudaBukhsh, Amr Rizk, Alexander Frömmgen, and Heinz Koeppl. 2017. Optimizing Stochastic Scheduling in Fork-Join Queueing Models: Bounds and Applications. In *IEEE International Conference on Computer Communications (INFOCOM)*.

[43] C. Kim and A. K. Agrawala. 1989. Analysis of the Fork-join Queue. *IEEE Transactions on Computers* 38, 2 (Feb 1989), 250–255.

[44] JFC Kingman. 1970. Inequalities in the Theory of Queues. *Journal of the Royal Statistical Society. Series B (Methodological)* (1970), 102–110.

[45] D. P. Kroese and V. F. Nicola. 1999. Efficient simulation of a tandem Jackson network. In *Simulation Conference Proceedings, 1999 Winter*, Vol. 1. 411–419 vol.1.

[46] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. 2012. Parallel Data Processing with MapReduce: A Survey. *ACM SIGMOD Rec.* 40, 4 (Jan. 2012), 11–20.

[47] Andrea Marin and Sabina Rossi. 2017. Power control in saturated fork-join queueing systems. *Performance Evaluation* 116 (2017), 101 – 118.

[48] R. Nelson and A. N. Tantawi. 1988. Approximate Analysis of Fork/join Synchronization in Parallel Queues. *IEEE Transactions on Computers* 37, 6 (Jun 1988), 739–743.

[49] P. Ney and E. Nummelin. 1987. Markov Additive Processes II. Large Deviations. *The Annals of Probability* 15, 2 (04 1987), 593–609.

[50] Rasha Osman and Peter G. Harrison. 2015. Approximating closed fork-join queueing networks using product-form stochastic Petri-nets. *Journal of Systems and Software* 110 (2015), 264 – 278.

[51] Ivanilton Polato, Reginaldo Ré, Alfredo Goldman, and Fabio Kon. 2014. A Comprehensive View of Hadoop Research-A Systematic Literature Review. *Journal of Network and Computer Applications* 46 (2014), 1–25.

[52] Felix Poloczek and Florin Ciucu. 2014. Scheduling Analysis with Martingales. *Performance Evaluation* 79 (2014), 56–72.

[53] Zhan Qiu, Juan F. Pérez, and Peter G. Harrison. 2015. Beyond the mean in fork-join queues: Efficient approximation for response-time tails. *Performance Evaluation* 91 (2015), 99 – 116. Special Issue: Performance 2015.

[54] Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

[55] Amr Rizk, Felix Poloczek, and Florin Ciucu. 2015. Computable Bounds in Fork-Join Queueing Systems. *ACM SIGMETRICS Perform. Eval. Rev.* 43, 1 (June 2015), 335–346.

[56] Navin Sharma, Dilip Kumar Krishnappa, David E. Irwin, Michael Zink, and Prashant J. Shenoy. 2013. GreenCache: augmenting off-the-grid cellular towers with multimedia caches. In *Multimedia Systems Conference 2013, MMSys '13, Oslo, Norway, February 27 - March 01, 2013*. 271–280.

[57] S. Shastri, A. Rizk, and D. Irwin. 2016. Transient Guarantees: Maximizing the Value of Idle Cloud Capacity. In *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*. 992–1002.

[58] Alexander Thomasian. 2014. Analysis of Fork/Join and Related Queueing Systems. *ACM Comput. Surv.* 47, 2, Article 17 (Aug. 2014), 71 pages.

[59] S. R. S. Varadhan. 2016. *Large deviations*. American Mathematical Soceity.

[60] Cathy H. Xia, Zhen Liu, Don Towsley, and Marc Lelarge. 2007. Scalability of Fork/Join Queueing Networks with Blocking. *ACM SIGMETRICS Perform. Eval. Rev.* 35, 1 (June 2007), 133–144.

[61] Tadafumi Yoshihara, Shoji Kasahara, and Yutaka Takahashi. 2001. Practical Time-Scale Fitting of Self-Similar Traffic with Markov-Modulated Poisson Process. *Telecommunication Systems* 17, 1 (2001), 185–211.

[62] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *USENIX HotCloud*.

[63] Matei Zaharia, Andy Konwinski, Anthony D Joseph, Randy H Katz, and Ion Stoica. 2008. Improving MapReduce Performance in Heterogeneous Environments. In *USENIX OSDI*, Vol. 8. 7.

[64] Yun Zeng, Augustin Chaintreau, Don Towsley, and Cathy H. Xia. 2016. A Necessary and Sufficient Condition for Throughput Scalability of Fork and Join Networks with Blocking. *SIGMETRICS Perform. Eval. Rev.* 44, 1 (June 2016), 25–36.

[65] Haiquan (Chuck) Zhao, Cathy H. Xia, Zhen Liu, and Don Towsley. 2010. A Unified Modeling Framework for Distributed Resource Allocation of General Fork and Join Processing Networks. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '10)*. ACM, New York, NY, USA, 299–310.

Received February 2018; revised August 2018; accepted December 2018

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.