Computable Bounds in Fork-Join Queueing Systems

Amr Rizk
University of Warwick

Felix Poloczek
University of Warwick / TU Berlin

Florin Ciucu
University of Warwick

ABSTRACT

In a Fork-Join (FJ) queueing system an upstream fork station splits incoming jobs into N tasks to be further processed by N parallel servers, each with its own queue; the response time of one job is determined, at a downstream join station, by the maximum of the corresponding tasks' response times. This queueing system is useful to the modelling of multi-service systems subject to synchronization constraints, such as MapReduce clusters or multipath routing. Despite their apparent simplicity, FJ systems are hard to analyze.

This paper provides the first *computable* stochastic bounds on the waiting and response time distributions in FJ systems. We consider four practical scenarios by combining 1a) renewal and 1b) non-renewal arrivals, and 2a) non-blocking and 2b) blocking servers. In the case of non-blocking servers we prove that delays scale as $\mathcal{O}(\log N)$, a law which is known for first moments under renewal input only. In the case of blocking servers, we prove that the same factor of $\log N$ dictates the stability region of the system. Simulation results indicate that our bounds are tight, especially at high utilizations, in all four scenarios. A remarkable insight gained from our results is that, at moderate to high utilizations, multipath routing "*makes sense*" from a queueing perspective for two paths only, i.e., response times drop the most when $N = 2$; the technical explanation is that the resequencing (delay) price starts to quickly dominate the tempting gain due to multipath transmissions.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems; G.3 [**Mathematics of Computing**]: Probability and Statistics

Keywords

Fork-Join queue; Performance evaluation; Parallel systems; MapReduce; Multipath

SIGMETRICS'15, June 15–19, 2015, Portland, OR, USA.

ACM 978-1-4503-3486-0/15/06 ...$15.00.
http://dx.doi.org/10.1145/2745844.2745859.

1. INTRODUCTION

The performance analysis of Fork-Join (FJ) systems received new momentum with the recent wide-scale deployment of large-scale data processing that was enabled through emerging frameworks such as MapReduce [12]. The main idea behind these big data analysis frameworks is an elegant divide and conquer strategy with various degrees of freedom in the implementation. The open-source implementation of MapReduce, known as Hadoop [37], is deployed in numerous production clusters, e.g., Facebook and Yahoo [20].

The basic operation of MapReduce is depicted in Figure 1. In the *map phase*, a job is split into multiple tasks that are mapped to different workers (servers). Once a specific subset of these tasks finish their executions, the corresponding *reduce phase* starts by processing the combined output from all the corresponding tasks. In other words, the reduce phase is subject to a fundamental synchronization constraint on the finishing times of all involved tasks.

A natural way to model one reduce phase operation is by a *basic* FJ queueing system with N servers. Jobs, i.e., the input unit of work in MapReduce systems, arrive according to some point process. Each job is split into N (map) tasks (or *splits*, in the MapReduce terminology), which are simultaneously sent to the N servers. At each server, each task requires a random service time, capturing the variable task execution times on different servers in the map phase. A job leaves the FJ system when all of its tasks are served; this constraint corresponds to the specification that the reduce phase starts no sooner than when all of its map tasks complete their executions.

Concerning the execution of tasks belonging to different jobs on the same server, there are two operational modes. In the *non-blocking* mode, the servers are workconserving in the sense that tasks immediately start their executions once the previous tasks finish theirs. In the *blocking* mode, the mapped tasks of a job simultaneously start their executions, i.e., servers can be idle when their corresponding queues are not empty. The non-blocking execution mode prevails in MapReduce due to its conceivable efficiency, whereas the blocking execution mode is employed when the `jobtracker` (the node coordinating and scheduling jobs) waits for all machines to be ready to synchronize the configuration files before mapping a new job; in Hadoop, this can be enforced through the coordination service `zookeeper` [37].

In this paper we analyze the performance of the FJ queueing model in four practical scenarios by considering two broad arrival classes (driven by either renewal or non-renewal processes) and the two operational modes described above.

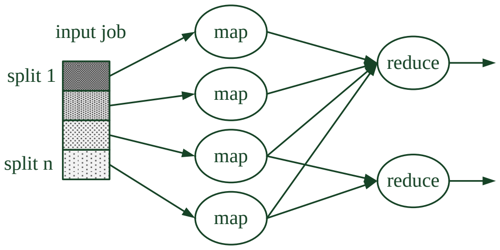


Figure 1: Schematic illustration of the basic operation of MapReduce.

The key contribution, to the best of our knowledge, are the first non-asymptotic and computable stochastic bounds on the waiting and response time distributions in the most relevant scenario, i.e., non-renewal (Markov modulated) job arrivals and the non-blocking operational mode. Under all scenarios, the bounds are numerically tight especially at high utilizations. This inherent tightness is due to a suitable martingale representation of the underlying queueing system, an approach which was conceived in [23] for the analysis of GI/GI/1 queues, and which was recently extended to address multi-class queues with non-renewal arrivals [11, 29]. The simplicity of the obtained stochastic bounds enables the derivation of scaling laws, e.g., delays in FJ systems scale as $\mathcal{O}(\log N)$ in the number of parallel servers N, for both renewal and non-renewal arrivals, in the non-blocking mode; more severe delay degradations hold in the blocking mode, and, moreover, the stability region depends on the same fundamental factor of $\log N$.

In addition to the direct applicability to the dimensioning of MapReduce clusters, there are other relevant types of parallel and distributed systems such as production or supply networks. In particular, by slightly modifying the basic FJ system corresponding to MapReduce, the resulting model suits the analysis of window-based transmission protocols over multipath routing. By making several simplifying assumptions such as ignoring the details of specific protocols (e.g., multipath TCP), we can provide a fundamental understanding of multipath routing from a queueing perspective. Concretely, we demonstrate that sending a flow of packets over two paths, instead of one, does generally reduce the steady-state response times. The surprising result is that by sending the flow over more than two paths, the steady-state response times start to increase. The technical explanation for such a rather counterintuitive result is that the $\log N$ resequencing price at the destination quickly dominates the tempting gain in the queueing waiting time due to multipath transmissions.

The rest of the paper is structured as follows. We first discuss related work on FJ systems and related applications. Then we analyze both non-blocking and blocking FJ systems with renewal input in Section 3, and with non-renewal input in Section 4. In Section 5 we apply the obtained results on the steady-state response time distributions to the analysis of multipath routing from a queueing perspective. Brief conclusions are presented in Section 6.

2. RELATED WORK

We first review analytical results on FJ systems, and then results related to the two application case studies considered in this paper, i.e., MapReduce and multipath routing.

The significance of the Fork-Join queueing model stems from its natural ability to capture the behavior of many parallel service systems. The performance of FJ queueing systems has been subject of multiple studies such as [4, 26, 35, 21, 24, 5, 7]. In particular, [4] notes that an exact performance evaluation of general FJ systems is remarkably hard due to the synchronization constraints on the input and output streams. More precisely, a major difficulty lies in finding an exact closed form expression for the joint steady-state workload distribution for the FJ queueing system. However, a number of results exist given certain constraints on the FJ system. The authors of [14] provide the stationary joint workload distribution for a two-server FJ system under Poisson arrivals and independent exponential service times. For the general case of more than two parallel servers there exists a number of works that provide approximations [26, 35, 24, 25] and bounds [4, 5] for certain performance metrics of the FJ system. Given renewal arrivals, [5] significantly improves the lower bounds from [4] in the case of heterogeneous phase-type servers using a matrix-geometric algorithmic method. The authors of [24] provide an approximation of the sojourn time distribution in a renewal driven FJ system consisting of multiple G/M/1 nodes. They show that the approximation error diminishes at extremal utilizations. Refined approximations for the mean sojourn time in two-server FJ systems that take the first two moments of the service time distribution are given in [21]; numerical evidence is further provided on the quality of the approximation for different service time distributions.

The closest related work to ours is [4], which provides computable lower and upper bounds on the expected response time in FJ systems under renewal assumptions with Poisson arrivals and exponential service times; the underlying idea is to artificially construct a more tractable system, yet subject to stochastic ordering relative to the original one. Our corresponding first order upper bound recovers the $\mathcal{O}(\log N)$ asymptotic behavior of the one from [4], and also reported in [26] in the context of an approximation; numerically, our bound is slightly worse than the one from [4] due to our main focus on computing bounds on the whole distribution (first order bounds are secondarily obtained by integration). Moreover, we show that the $\mathcal{O}(\log N)$ scaling law also holds in the case of Markov modulated arrivals. In a parallel work [22] to ours, the authors adopt a network calculus approach to derive stochastic bounds in a non-blocking FJ system, under a strong assumption on the input; for related constructions of such arrival models see [18].

Concerning concrete applications of FJ systems, in particular MapReduce, there are several empirical and analytical studies analyzing its performance. For instance, [39, 2] aim to improve the system performance via empirically adjusting its numerous and highly complex parameters. The targeted performance metric in these studies is the job response time, which is in fact an integral part of the business model of MapReduce based query systems such as [27] and time priced computing clouds such as Amazon's EC2 [1]. For an overview on works that optimize the performance of MapReduce systems see the survey article [28]. Using a similar idea as in [4], the authors of [32] derive asymptotic results on the response time distribution in the case of renewal arrivals; such results are further used to understand the impact of different scheduling models in the reduce phase of MapReduce. Using the model from [32] the work in [33] provides approximations for the number of jobs in a tandem system consisting of a map queue and a reduce queue in the heavy traffic regime. The work in [36] derives approximations of the mean response time in MapReduce systems using a mean value analysis technique and a closed FJ queueing system model from [34].

Concerning multipath routing, the works [3, 17] provided ground for multiple studies on different formulations of the underlying resequencing delay problem, e.g., [16, 38]. Factorization methods were used in [3] to analyze the disordering delay and the delay of resequencing algorithms, while the authors of [17] conduct a queueing theoretic analysis of an $M/G/\infty$ queue receiving a stream of numbered customers. In [16, 38] the multipath routing model comprises Bernoulli thinning of Poisson arrivals over N parallel queueing stations followed by a resequencing buffer. The work in [16] provides asymptotics on the conditional probability of the resequencing delay conditioned on the end-to-end delay for different service time distributions. For $N = 2$ and exponential interarrival and service times, [38] derives a large deviations result on the resequencing queue size. Our work differs from these works in that we consider a model of the basic operation of window-based transmission protocols over multipath routing, motivated by the emerging application of multipath TCP [30]. We point out, however, that we do not model the specific operation of any particular multipath transmission protocol. Instead, we analyze a generic multipath transmission protocol under simplifying assumptions, in order to provide a theoretical understanding of the overall response times comprised of both queueing and resequencing delays.

Relative to the existing literature, our key theoretical contribution is to provide *computable* and non-asymptotic bounds on the *distributions* of the steady-state waiting and response times under both *renewal* and *non-renewal* input in FJ systems. The consideration of non-renewal input is particularly relevant, given recent observations that job arrivals are subject to temporal correlations in production clusters. For instance, [10, 19] report that job, respectively, flow arrival traces in clusters running MapReduce exhibit various degrees of burstiness.

3. FJ SYSTEMS WITH RENEWAL INPUT

We consider a FJ queueing system as depicted in Figure 2. Jobs arrive at the input queue of the FJ system according to some point process with interarrival times t_i between the i and $i+1$ jobs. Each job i is split into N tasks that are mapped through a bijection to N servers. A task of job i that is serviced by some server n requires a random service time $x_{n,i}$. A job leaves the system when all of its tasks finish their executions, i.e., there is an underlying synchronization constraint on the output of the system. We assume that the families $\{t_i\}$ and $\{x_{n,i}\}$ are independent.

In the sequel we differentiate between two cases, i.e., *a*) non-blocking and *b*) blocking servers. The first case corresponds to workconserving servers, i.e., a server starts servicing a task of the next job (if available) immediately upon finishing the current task. In the latter case, a server that finishes servicing a task is blocked until the corresponding job leaves the system, i.e., until the last task of the current job completes its execution. This can be regarded as an additional synchronization constraint on the input of the system, i.e., all tasks of a job start receiving service simultaneously. We will next analyze *a*) and *b*) for renewal arrivals.

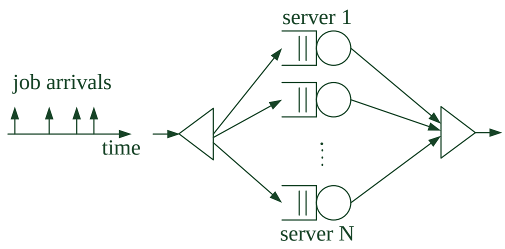


Figure 2: A schematic Fork-Join queueing system with N parallel servers. An arriving job is split into N tasks, one for each server. A job leaves the FJ system when all of its tasks are served. An arriving job is considered waiting until the service of the last of its tasks starts, i.e., when the previous job departs the system.

3.1 Non-Blocking Systems

Consider an arrival flow of jobs with renewal interarrival times t_i, and assume that the waiting time of the first job is $w_1 = 0$. Given N parallel servers, the *waiting time* w_j of the jth job is defined as

$$w_j = \max\left\{0, \max_{1\le k\le j-1}\left\{\max_{n\in[1,N]}\left\{\sum_{i=1}^{k} x_{n,j-i} - \sum_{i=1}^{k} t_{j-i}\right\}\right\}\right\} , \tag{1}$$

for all $j \ge 2$, where $x_{n,j}$ is the service time required by the task of job j that is mapped to server n. We count a job as waiting until its last task starts receiving service. Similarly, the *response times* of jobs, i.e., the times until the last corresponding tasks have finished their executions, are defined as $r_1 = \max_n x_{n,1}$ for the first job, and for $j \ge 2$ as

$$r_j = \max_{0\le k\le j-1}\left\{\max_{n\in[1,N]}\left\{\sum_{i=0}^{k} x_{n,j-i} - \sum_{i=1}^{k} t_{j-i}\right\}\right\} , \tag{2}$$

where by convention $\sum_{i=1}^{0} t_i = 0$; for brevity, we will denote $\max_n := \max_{n\in[1,N]}$.

We assume that the task service times $x_{n,j}$ are independent and identically distributed (iid). The stability condition for the FJ queueing system is given as $\mathsf{E}[x_{1,1}] < \mathsf{E}[t_1]$. By stationarity and reversibility of the iid processes $x_{n,j}$ and t_j, there exists a distribution of the steady-state waiting time w and steady-state response time r, respectively, which have the representations

$$w =_{\mathcal{D}} \max_{k\ge 0}\left\{\max_n\left\{\sum_{i=1}^{k} x_{n,i} - \sum_{i=1}^{k} t_i\right\}\right\} \tag{3}$$

and

$$r =_{\mathcal{D}} \max_{k\ge 0}\left\{\max_n\left\{\sum_{i=0}^{k} x_{n,i} - \sum_{i=1}^{k} t_i\right\}\right\} , \tag{4}$$

respectively. Here, $=_{\mathcal{D}}$ denotes equality in distribution. Note that the only difference in (3) and (4) is that for the latter the sum over the $x_{n,i}$ starts at $i = 0$ rather than at $i = 1$.

The following theorem provides stochastic upper bounds on w and r. The corresponding proof will rely on submartingale constructions and the Optional Sampling Theorem (see Lemma 6 in the Appendix).

THEOREM 1. (RENEWALS, NON-BLOCKING) *Given a FJ system with N parallel non-blocking servers that is fed by renewal job arrivals with interarrivals t_j. If the task service times $x_{n,j}$ are iid, then the steady-state waiting and response times w and r are bounded by*

$$\begin{aligned} \mathsf{P}[w \geq \sigma] &\leq Ne^{-\theta_{nb}\sigma} && (5) \\ \mathsf{P}[r \geq \sigma] &\leq N\mathsf{E}\left[e^{\theta_{nb}x_{1,1}}\right]e^{-\theta_{nb}\sigma} , && (6) \end{aligned}$$

where θ_{nb} (with the subscript 'nb' standing for non-blocking) is the (positive) solution of

$$\mathsf{E}\left[e^{\theta x_{1,1}}\right]\mathsf{E}\left[e^{-\theta t_1}\right] = 1 . \tag{7}$$

We remark that the stability condition $\mathsf{E}[x_{1,1}] < \mathsf{E}[t_1]$ guarantees the existence of a positive solution in (7) (see also [29]).

PROOF. Consider the waiting time w. We first prove that for each $n \in [1, N]$ the process

$$z_n(k) = e^{\theta_{nb}\sum_{i=1}^{k}(x_{n,i}-t_i)}$$

is a martingale with respect to the filtration

$$\mathcal{F}_k := \sigma\{x_{n,m}, t_m \,|\, m \leq k, n \in [1, N]\} .$$

The independence assumption of $x_{n,j}$ and t_j implies that

$$\begin{aligned} \mathsf{E}[z_n(k)\,|\,\mathcal{F}_{k-1}] &= \mathsf{E}\left[e^{\theta_{nb}\sum_{i=1}^{k}(x_{n,i}-t_i)}\,\middle|\,\mathcal{F}_{k-1}\right] \\ &= \mathsf{E}\left[e^{\theta_{nb}(x_{n,k}-t_k)}\right]e^{\theta_{nb}\sum_{i=1}^{k-1}(x_{n,i}-t_i)} \\ &= e^{\theta_{nb}\sum_{i=1}^{k-1}(x_{n,i}-t_i)} \\ &= z_n(k-1) , && (8) \end{aligned}$$

under the condition on θ_{nb} from the theorem. Moreover, $z_n(k)$ is obviously integrable by the condition on θ_{nb} from the theorem, completing thus the proof for the martingale property.

Next we prove that the process

$$z(k) = \max_n z_n(k) \tag{9}$$

is a submartingale w.r.t. $\mathcal{F}_k$. Given the martingale property of each of the z_n and the monotonicity of the conditional expectation we can write for $j \in [1, N]$:

$$\mathsf{E}\left[\max_n z_n(k)\,\middle|\,\mathcal{F}_{k-1}\right] \geq \mathsf{E}[z_j(k)\,|\,\mathcal{F}_{k-1}] = z_j(k-1) ,$$

where the inequality stems from $\max_n z_n(k) \geq z_j(k)$ for $j \in [1, N]$ a.s., whereas the subsequent equality stems from the martingale property (8) for $z_n(k)$ for all $n \in [1, N]$. Hence we can write

$$\mathsf{E}[z(k)\,|\,\mathcal{F}_{k-1}] \geq \max_n z_n(k-1) = z(k-1) , \tag{10}$$

which proves the submartingale property.

To derive a bound on the steady-state waiting time distribution, let $\sigma > 0$ and define the stopping time

$$K := \inf\left\{k \geq 0 \,\middle|\, \max_n \sum_{i=1}^{k}(x_{n,i} - t_i) \geq \sigma\right\} , \tag{11}$$

which is also the first point in time k where $z(k) \geq e^{\theta_{nb}\sigma}$. Note that with the representation of w from (3):

$$\{K < \infty\} = \{w \geq \sigma\} .$$

Now, using the Optional Sampling Theorem (see Lemma 6 from the Appendix) for submartingales with $k \geq 1$:

$$\begin{aligned} N &= \sum_{n\in[1,N]}\mathsf{E}\left[e^{\theta_{nb}\sum_{i=1}^{k}(x_{n,i}-t_i)}\right] \\ &\geq \mathsf{E}\left[\max_n e^{\theta_{nb}\sum_{i=1}^{k}(x_{n,i}-t_i)}\right] && (12) \\ &= \mathsf{E}[z(k)] \geq \mathsf{E}[z(K \wedge k)] \geq \mathsf{E}[z(K)1_{K<k}] \\ &\geq e^{\theta_{nb}\sigma}\mathsf{P}[K < k] , \end{aligned}$$

where we used the condition on θ_{nb} from the theorem in the first line, the union bound in the second line, and the submartingale property in the third line. In the last line we used the definition of the stopping time K; note that we use the notation $K \wedge n := \min\{K, n\}$. The proof completes by letting $k \to \infty$.

For the response time r, define the processes

$$\tilde{z}_n(k) = e^{\theta_{nb}\left(\sum_{i=0}^{k}x_{n,i}-\sum_{i=1}^{k}t_i\right)} ,$$

which differs from the z_n only in the range of the sum of the service times $x_{n,i}$. Then we proceed as for the derivation of the bound on the waiting time w. The only difference in the derivation is that inequality (12) translates to

$$N\mathsf{E}\left[e^{\theta_{nb}x_{1,1}}\right] \geq \mathsf{E}\left[\max_n e^{\theta_{nb}\left(\sum_{i=0}^{k}x_{n,i}-\sum_{i=1}^{k}t_i\right)}\right] .$$

□

Fixing the right hand sides in (5) and (6) to ε, we find that the corresponding quantiles on the waiting and response times grow with the number of parallel servers N as $\mathcal{O}(\log N)$, a law which was already demonstrated in the special case of Poisson arrival and exponential service times, and for first moments, in [26], and more generally in [4]. This scaling result is essential for dimensioning FJ systems such as MapReduce computing clusters, as it explains the impact of a MapReduce server pool size N on the job waiting/response times.

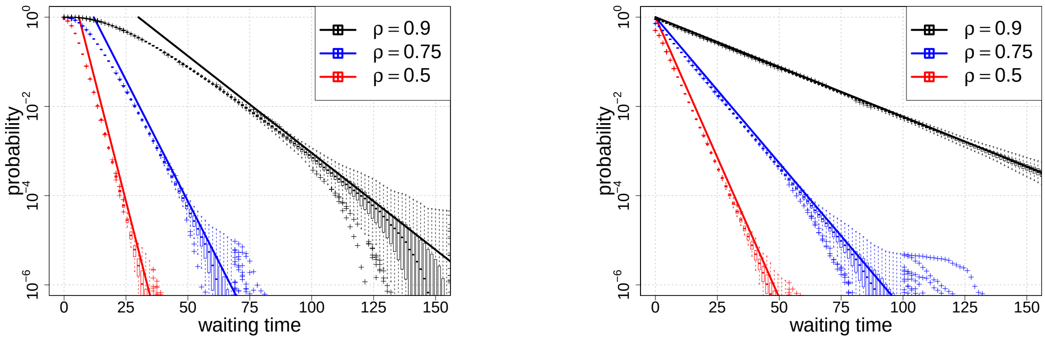
We note that the bound in Theorem 1 can be computed for different arrival and service time distributions as long as the MGF (moment generating function) and Laplace transform from (7) are computable. Given a scenario where the job interarrival process and the task size distributions in a MapReduce cluster are not known a priori, estimates of the corresponding MGF and Laplace transforms can be obtained using recorded traces, e.g., using the method from [15].

Next we illustrate two immediate applications of Theorem 1.

Example 1: Exponentially distributed interarrival and service times

Consider that the interarrival times t_i and service times $x_{n,i}$ are exponentially distributed with parameters λ and μ, respectively; note that when $N = 1$ the system corresponds to the M/M/1 queue. The corresponding stability condition becomes $\mu > \lambda$. Using Theorem 1, the bounds on the steady-state waiting and response time distributions are

$$\mathsf{P}[w \geq \sigma] \leq Ne^{-(\mu-\lambda)\sigma} \tag{13}$$



(a) Non-Blocking (b) Blocking

Figure 3: Bounds on the waiting time distributions vs. simulations (renewal input): (a) the non-blocking case (13) **and (b) the blocking case** (22)**. The system parameters are** $N = 20$, $\mu = 1$**, and three utilization levels** $\rho = \{0.9, 0.75, 0.5\}$ **(from top to bottom). Simulations include** 100 **runs, each accounting for** 10^7 **slots.**

and

$$\mathsf{P}[r \geq \sigma] \leq \frac{N}{\rho} e^{-(\mu-\lambda)\sigma} , \qquad (14)$$

where the exponential decay rate $\mu - \lambda$ follows by solving $\frac{\mu}{\mu-\theta}\frac{\lambda}{\lambda+\theta} = 1$, i.e., the instantiation of (7).

Next we briefly compare our results to the existing bound on the mean response time from [4], given as

$$\mathsf{E}[r] \leq \frac{1}{\mu-\lambda}\sum_{n=1}^{N}\frac{1}{n} . \qquad (15)$$

By integrating the tail of (14) we obtain the following upper bound on the mean response time

$$\mathsf{E}[r] \leq \frac{\log(N/\rho)+1}{\mu-\lambda} .$$

Compared to (15), our bound exhibits the same $\log N$ scaling law but is numerically slightly looser; asymptotically in N, the ratio between the two bounds converges to one. A key technical reason for obtaining a looser bound is that we mainly focus on deriving bounds on distributions; through integration, the numerical discrepancies accumulate.

For the numerical illustration of the tightness of the bounds on the waiting time distributions from (13) we refer to Figure 3.(a); the numerical parameters and simulation details are included in the caption.

Example 2: Exponentially distributed interarrival times and constant service times

We now consider the case of iid exponentially distributed interarrival times t_i with parameter λ, and deterministic service times $x_{n,i} = 1/\mu$, for all $i \geq 0$ and $n \in [1, N]$; note that when $N = 1$ the system corresponds to the M/D/1 queue.

The condition on the asymptotic decay rate θ_{nb} from Theorem 1 becomes

$$\frac{\lambda}{\lambda+\theta_{nb}} = e^{-\frac{\theta_{nb}}{\mu}} ,$$

which can be numerically solved; upper bounds on the waiting and response time distributions follow then immediately from Theorem 1.

3.2 Blocking Systems

Here we consider a blocking FJ queueing system, i.e., the start of each job is synchronized amongst all servers. We maintain the iid assumptions on the interarrival times t_i and service times $x_{n,i}$. The waiting time and response time for the jth job can then be written as

$$w_j = \max\left\{0, \max_{1\leq k\leq j-1}\left\{\sum_{i=1}^{k}\max_n x_{n,j-i} - \sum_{i=1}^{k} t_{j-i}\right\}\right\}$$
$$r_j = \max_{0\leq k\leq j-1}\left\{\sum_{i=0}^{k}\max_n x_{n,j-i} - \sum_{i=1}^{k} t_{j-i}\right\} .$$

Note that the only difference to (1) and (2) is that the maximum over the number of servers now occurs inside the sum.

It is evident that the blocking system is more conservative than the non-blocking system in the sense that the waiting time distribution of the non-blocking system is dominated by the waiting time distribution of the blocking system. Moreover, the stability region for the blocking system, given by $\mathsf{E}[t_1] > \mathsf{E}[\max_n x_{n,1}]$, is included in the stability region of the corresponding non-blocking system (i.e., $\mathsf{E}[t_1] > \mathsf{E}[x_{1,1}]$).

Analogously to (3), the steady-state waiting and response times w and r have now the representations

$$w =_{\mathcal{D}} \max_{k\geq 0}\left\{\sum_{i=1}^{k}\max_n x_{n,i} - \sum_{i=1}^{k} t_i\right\} \qquad (16)$$
$$r =_{\mathcal{D}} \max_{k\geq 0}\left\{\sum_{i=0}^{k}\max_n x_{n,i} - \sum_{i=1}^{k} t_i\right\} . \qquad (17)$$

The following theorem provides upper bounds on w and r.

THEOREM 2. (RENEWALS, BLOCKING) *Given a FJ queueing system with N parallel blocking servers that is fed by renewal job arrivals with interarrivals t_j and iid task service times $x_{n,j}$. The distributions of the steady-state waiting and response times are bounded by*

$$\mathsf{P}[w \geq \sigma] \leq e^{-\theta_b\sigma} \qquad (18)$$
$$\mathsf{P}[r \geq \sigma] \leq \mathsf{E}\left[e^{\theta_b x_{1,1}}\right] e^{-\theta_b\sigma} ,$$

where θ_b (with the subscript 'b' standing for blocking) is the (positive) solution of

$$\mathsf{E}\left[e^{\theta \max_n x_{n,1}}\right] \mathsf{E}\left[e^{-\theta t_1}\right] = 1 \ . \tag{19}$$

Before giving the proof we note that, in general, (19) can be numerically solved. Moreover, for small values of N, θ_b can be analytically solved.

PROOF. Consider the waiting time w. We proceed similarly as in the proof of Theorem 1. Letting $\mathcal{F}_k$ as above, we first prove that the process

$$y(k) = e^{\theta_b \sum_{i=1}^{k}\left(\max_n x_{n,i} - t_i\right)}$$

is a martingale w.r.t. $\mathcal{F}_k$ using a technique from [23]. We write

$$\begin{aligned}
\mathsf{E}\left[y(k) \mid \mathcal{F}_{k-1}\right] &= \mathsf{E}\left[e^{\theta_b \sum_{i=1}^{k}\left(\max_n x_{n,i} - t_i\right)} \,\middle|\, \mathcal{F}_{k-1}\right] \\
&= e^{\theta_b \sum_{i=1}^{k-1}\left(\max_n x_{n,i} - t_i\right)} \mathsf{E}\left[e^{\theta_b\left(\max_n x_{n,k} - t_k\right)}\right] \\
&= e^{\theta_b \sum_{i=1}^{k-1}\left(\max_n x_{n,i} - t_i\right)} \\
&= y(k-1) \ ,
\end{aligned}$$

where we used the independence and renewal assumptions for $x_{n,i}$ and t_i in the second line, and finally the condition on θ_b from (19).

In the next step we apply the Optional Sampling Theorem (37) to derive the bound from the theorem. We first define the stopping time K by

$$K := \inf\left\{k \geq 0 \,\middle|\, \sum_{i=1}^{k}\left(\max_n x_{n,i} - t_i\right) \geq \sigma\right\} \ . \tag{20}$$

Recall that $\mathsf{P}\left[K < \infty\right] = \mathsf{P}\left[w \geq \sigma\right]$. We can next write for every $k \in \mathbb{N}$

$$\begin{aligned}
1 &= \mathsf{E}\left[y(0)\right] \\
&= \mathsf{E}\left[y(K \wedge k)\right] \\
&\geq \mathsf{E}\left[y(K \wedge k) 1_{K<k}\right] \\
&= \mathsf{E}\left[e^{\theta_b \sum_{i=1}^{K}\left(\max_n x_{n,i} - t_i\right)} 1_{K<k}\right] \\
&\geq e^{\theta_b \sigma} \mathsf{P}\left[K < k\right] \ .
\end{aligned}$$

Taking $k \to \infty$ completes the proof. The proof for the response time r is analogous. $\square$

Example 3: Exponentially distributed interarrival and service times

Consider interarrival and service times t_i and $x_{n,i}$ that are exponentially distributed with parameters λ and μ, respectively. In [31] it was shown that

$$\max_n L_n =_{\mathcal{D}} \sum_{n=1}^{N} \frac{L_n}{n}$$

for iid exponentially distributed random variables L_n, so that the stability condition $\mathsf{E}\left[t_1\right] > \mathsf{E}\left[\max_n x_{n,1}\right]$ becomes

$$\frac{1}{\lambda} > \frac{1}{\mu} \sum_{n=1}^{N} \frac{1}{n} \ . \tag{21}$$

By applying Theorem 2, the bounds on the steady-state waiting and response time distributions are

$$\mathsf{P}\left[w \geq \sigma\right] \leq e^{-\theta_b \sigma} \tag{22}$$

and

$$\mathsf{P}\left[r \geq \sigma\right] \leq \frac{\mu}{\mu - \theta_b} e^{-\theta_b \sigma} \ ,$$

where θ_b can be numerically solved from the condition

$$\prod_{n=1}^{N} \frac{n\mu}{n\mu - \theta_b} \frac{\lambda}{\lambda + \theta_b} = 1 \ .$$

For quick numerical illustrations we refer back to Figure 3.(b).

The interesting observation is that the stability condition from (21) depends on the number of servers N. In particular, as the right hand side grows in $\log N$, the system becomes unstable (i.e., waiting times are infinite) for sufficiently large N. This shows that the optional blocking mode from Hadoop should be judiciously enabled.

Example 4: Exponentially distributed interarrival and constant times

If the service times are deterministic, i.e., $x_{n,i} = 1/\mu$ for all $i \geq 0$ and $n \in [1, N]$, the representations of w and r from (16) and (17) match their non-blocking counterparts from (3) and (4) and hence the corresponding stability regions and stochastic bounds are equal to those from Example 2.

4. FJ SYSTEMS WITH NON-RENEWAL INPUT

In this section we consider the more realistic case of FJ queueing systems with non-renewal job arrivals. This model is particularly relevant given the empirical evidence that clusters running MapReduce exhibit various degrees of burstiness in the input [10, 19]. Moreover, numerous studies have demonstrated the burstiness of Internet traces, which can be regarded in particular as the input to multipath routing.

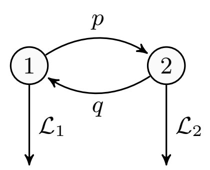


Figure 4: Markov modulating chain c_k for the job interarrival times.

We model the interarrival times t_i using a Markov modulated process. Concretely, consider a two-state modulating Markov chain c_k, as depicted in Figure 4, with a transition matrix T given by

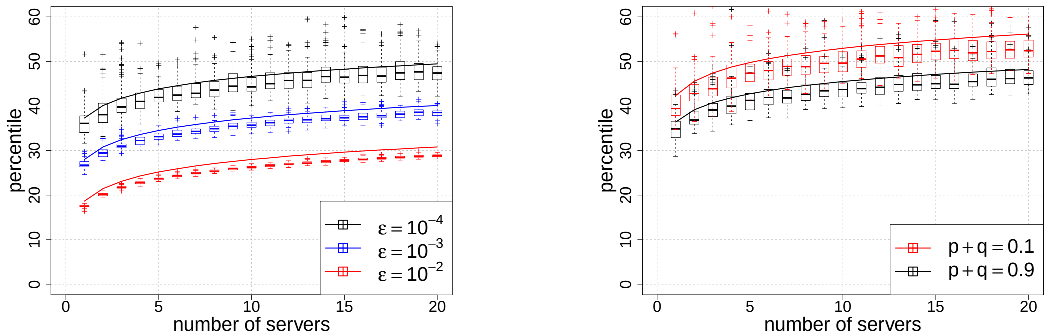
$$T = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix} \ , \tag{23}$$

for some values $0 < p, q < 1$. In state $i \in \{1, 2\}$ the interarrival times are given by iid random variables L_i with distribution $\mathcal{L}_i$. Without loss of generality we assume that L_1 is stochastically smaller than L_2, i.e.,

$$\mathsf{P}\left[L_1 \geq t\right] \leq \mathsf{P}\left[L_2 \geq t\right] \ ,$$

for any $t \geq 0$. Additionally, we assume that the Markov chain c_k satisfies the burstiness condition

$$p < 1 - q \ , \tag{24}$$



(a) Impact of ε (b) Impact of the burstiness factor $p+q$

Figure 5: The $\mathcal{O}(\log N)$ scaling of waiting time percentiles w^{ε} for Markov modulated input (the non-blocking case (25)). The system parameters are $\mu = 1$, $\lambda_2 = 0.9$, $\rho = 0.75$ (in both (a) and (b)) $p = 0.1$, $q = 0.4$ (in (a)), three violation probabilities ε (in (a)), $\varepsilon = 10^{-4}$ and only two burstiness parameters $p+q$ (in (b)) (for visual convenience). Simulations include 100 runs, each accounting for 10^7 slots.

i.e., the probability of jumping to a different state is less than the probability of staying in the same state.

Subsequent derivations will exploit the following exponential transform of the transition matrix T defined as

$$T_\theta := \begin{pmatrix} (1-p)\,\mathsf{E}\left[e^{-\theta L_1}\right] & p \;\; \mathsf{E}\left[e^{-\theta L_2}\right] \\ q \;\; \mathsf{E}\left[e^{-\theta L_1}\right] & (1-q)\,\mathsf{E}\left[e^{-\theta L_2}\right] \end{pmatrix} ,$$

for some $\theta > 0$. Let $\Lambda(\theta)$ denote the maximal positive eigenvalue of T_θ, and the vector $h = (h(1), h(2))$ denote a corresponding eigenvector. By the Perron-Frobenius Theorem, $\Lambda(\theta)$ is equal to the spectral radius of T_θ such that h can be chosen with strictly positive components.

As in the case of renewal arrivals, we will next analyze both non-blocking and blocking FJ systems.

4.1 Non-Blocking Systems

We first analyze a non-blocking FJ system fed with arrivals that are modulated by a stationary Markov chain as in Figure 4. We assume that the task service times $x_{n,j}$ are iid and that the families $\{t_i\}$ and $\{x_{n,i}\}$ are independent. Note that both the definition of w_j from (1) and the representation of the steady-state waiting time w in (3) remain valid, due to stationarity and reversibility; the same holds for the response times.

The next theorem provides upper bounds on the steady-state waiting and response time distributions in the non-blocking scenario with Markov modulated interarrivals.

Theorem 3. (Non-Renewals, Non-Blocking) *Given a FJ queueing system with N parallel non-blocking servers, Markov modulated job interarrivals t_j according to the Markov chain depicted in Figure 4 with transition matrix (23), and iid task service times $x_{n,j}$. The steady-state waiting and response time distributions are bounded by*

$$\mathsf{P}[w \geq \sigma] \;\leq\; N e^{-\theta_{nb}\sigma} \qquad (25)$$

$$\mathsf{P}[r \geq \sigma] \;\leq\; N\,\mathsf{E}\left[e^{\theta_{nb} x_{1,1}}\right] e^{-\theta_{nb}\sigma} , \qquad (26)$$

where θ_{nb} is the (positive) solution of

$$\mathsf{E}\left[e^{\theta x_{1,1}}\right] \Lambda(\theta) = 1 .$$

(Recall that $\Lambda(\theta)$ was defined as a spectral radius.)

We remark that the existence of a positive solution θ_{nb} is guaranteed by the Perron-Frobenius Theorem, see, e.g., [29].

Proof. Consider the filtration

$$\mathcal{F}_k := \sigma\left\{x_{n,m}, t_m, c_m \,|\, m \leq k, n \in [1, N]\right\} ,$$

that includes information about the state c_k of the Markov chain. Now, we construct the process $z(k)$ as

$$\begin{aligned} z(k) &= h(c_k) e^{\theta_{nb}\left(\max_n \sum_{i=1}^{k} x_{n,i} - \sum_{i=1}^{k} t_i\right)} \\ &= \left(e^{\theta_{nb}\left(\max_n \sum_{i=1}^{k} x_{n,i} - kD\right)}\right)\left(h(c_k) e^{\theta_{nb}\left(kD - \sum_{i=1}^{k} t_i\right)}\right) \end{aligned} \qquad (27)$$

with the deterministic parameter

$$D := \theta_{nb}^{-1} \log\left(\mathsf{E}\left[e^{\theta_{nb} x_{1,1}}\right]\right) .$$

Note the similarity of $z(k)$ to (9) except for the additional function h. Roughly, the function h captures the correlation structure of the non-renewal interarrival time process.

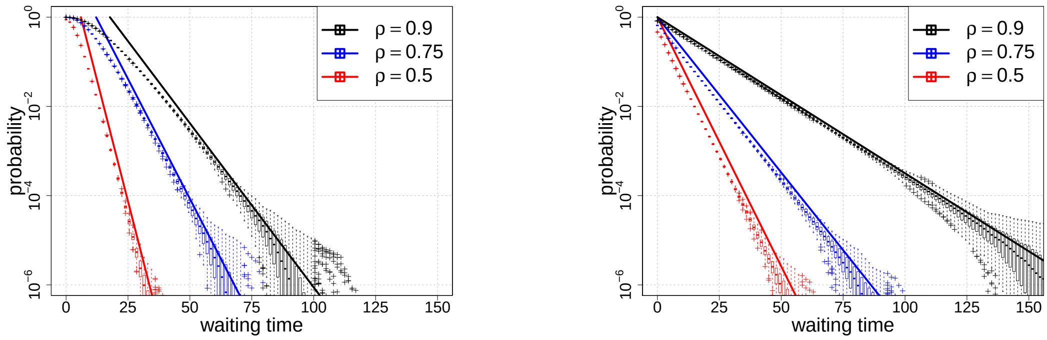
Next we show that both terms of (27) are submartingales. In the first step we note that by the definition of D:

$$\mathsf{E}\left[e^{\theta_{nb}\left(\sum_{i=1}^{k} x_{n,i} - kD\right)} \,\middle|\, \mathcal{F}_{k-1}\right] = e^{\theta_{nb}\left(\sum_{i=1}^{k-1} x_{n,i} - (k-1)D\right)} ,$$

hence, following the line of argument in (10) the left factor of (27), which accounts for the additional $\max_n$, is a submartingale. The second step is similar to the derivations in [9, 13]. First, note that

$$\begin{aligned} \mathsf{E}\left[h(c_k) e^{\theta_{nb}(D - t_k)} \,\middle|\, \mathcal{F}_{k-1}\right] &= e^{\theta_{nb} D} T_{\theta_{nb}} h(c_{k-1}) \\ &= e^{\theta_{nb} D} \Lambda(\theta_{nb}) h(c_{k-1}) \\ &= h(c_{k-1}) , \end{aligned} \qquad (28)$$

where the last line is due to the definitions of D and θ_{nb}. Now, multiplying both sides of (28) by $e^{\theta_{nb}\left((k-1)D - \sum_{i=1}^{k-1} t_i\right)}$ proves the martingale and hence the submartingale property of the right factor in (27). As the process $z(k)$ is a product of two independent submartingales, it is a submartingale itself w.r.t. $\mathcal{F}_k$.



(a) Non-Blocking (b) Blocking

Figure 6: Bounds on the waiting time distributions vs. simulations (non-renewal input): (a) the non-blocking case (25) and (b) the blocking case (31). The parameters are $N = 20, \mu = 1, p = 0.1, q = 0.4, \lambda_1 \in \{0.4, 0.72, 0.72\}$ and $\lambda_2 \in \{0.9, 0.9, 1.62\}$ leading to utilizations $\rho \in \{0.5, 0.75, 0.9\}$. Simulations include 100 runs, each accounting for 10^7 slots.

Next we derive a bound on the steady-state waiting time distribution using the Optional Stopping Theorem. Here we use the stopping time K defined in (11). Recall that $\mathsf{P}[K < \infty] = \mathsf{P}[w \geq \sigma]$. On the one hand we can write for every $k \in \mathbb{N}$

$$\begin{aligned} \mathsf{E}[z(k)] &\geq \mathsf{E}[z(K \wedge k)] \\ &\geq \mathsf{E}[z(K \wedge k) 1_{K<k}] \\ &= \mathsf{E}\left[\max_n h(c_K) e^{\theta_{nb}\left(\sum_{i=1}^K x_{n,i} - \sum_{i=1}^K t_i\right)} 1_{K<k}\right] \\ &\geq e^{\theta_{nb}\sigma} \mathsf{E}[h(c_K) 1_{K<k}] \\ &= e^{\theta_{nb}\sigma} \mathsf{E}[h(c_K) \,|\, K < k] \, \mathsf{P}[K < k] \ . \end{aligned} \qquad (29)$$

On the other hand we can upper bound the term

$$\begin{aligned} \mathsf{E}[z(k)] &= \mathsf{E}\left[\max_n e^{\theta_{nb}\left(\sum_{i=1}^k x_{n,i} - kD\right)}\right] \mathsf{E}\left[h(c_k) e^{\theta_{nb}\left(kD - \sum_{i=1}^k t_i\right)}\right] \\ &\leq N \mathsf{E}[h(c_1)] \ . \end{aligned}$$

Letting $k \to \infty$ in (29) leads to

$$\mathsf{P}[K < \infty] \leq \frac{\mathsf{E}[h(c_1)]}{\mathsf{E}[h(c_K) \,|\, K < \infty]} N e^{-\theta_{nb}\sigma} \ . \qquad (30)$$

In Lemma 7 it is shown that the distribution of the random variable $(c_K \mid K < k)$ is stochastically smaller than the stationary distribution of the Markov chain. Given the burstiness condition in (24) and that the function h is monotonically decreasing [8], we can further upper bound the prefactor in (30) as

$$\frac{\mathsf{E}[h(c_1)]}{\mathsf{E}[h(c_K) \,|\, K < \infty]} \leq 1 \ ,$$

which completes the proof. The proof for the response time r is analogous. □

Remark: Note that, if the burstiness condition (24) is not fulfilled then we can still upper bound the prefactor in (30) using the trivial upper bound

$$\frac{\mathsf{E}[h(c_1)]}{\mathsf{E}[h(c_K) \,|\, K < \infty]} \leq \frac{\mathsf{E}[h(c_1)]}{\min_k h(c_k)} \ .$$

Figure 5 displays the bounds on the waiting time percentiles w^ε, for various violation probabilities ε, in the FJ system with non-renewal input. The bounds closely match the corresponding simulation results, shown as box-plots, while also exhibiting the $\mathcal{O}(\log N)$ scaling behavior (which can be also derived from both (25) and (26), as in Section 3).

4.2 Blocking Systems

Now we turn to the blocking variant of the FJ system that is fed by the same non-renewal arrivals as in the previous section. Without loss of generality we consider exponential distributions $\mathcal{L}_m$ for $m \in [1, 2]$. The main result is:

Theorem 4. (Non-Renewals, Blocking) *Given a FJ system with N blocking servers, Markov modulated job interarrivals t_j, and iid task service times $x_{n,j}$. The steady-state waiting and response time distributions are bounded by*

$$\begin{aligned} \mathsf{P}[w \geq \sigma] &\leq e^{-\theta_b\sigma} \\ \mathsf{P}[r \geq \sigma] &\leq \mathsf{E}\left[e^{\theta_b x_{1,1}}\right] e^{-\theta_b\sigma} \ , \end{aligned} \qquad (31)$$

where θ_b is the (positive) solution of

$$\mathsf{E}\left[e^{\theta \max_n x_{n,1}}\right] \Lambda(\theta) = 1 \ .$$

We remark that the positive solution for θ_b is guaranteed under the stronger stability condition $\mathsf{E}[t_1] > \mathsf{E}[\max_n x_{n,1}]$ and the Perron-Frobenius Theorem.

Proof. Let $D := \theta_b^{-1} \log \mathsf{E}\left[e^{\theta_b \max_n x_{n,1}}\right]$ and define the process y by:

$$\begin{aligned} y(k) &= h(c_k) e^{\theta_b\left(\sum_{i=1}^k \max_n x_{n,i} - \sum_{i=1}^k t_i\right)} \\ &= \left(e^{\theta_b\left(\sum_{i=1}^k \max_n x_{n,i} - kD\right)}\right)\left(h(c_k) e^{\theta_b\left(kD - \sum_{i=1}^k t_i\right)}\right) \ . \end{aligned}$$

Similarly to the proofs of Theorem 2 and Theorem 3 one can show that both the first and second factor of y are martingales, and hence y is a martingale. We use the stopping time K in (20) and write

$$\begin{aligned} \mathsf{E}[h(c_1)] &= \mathsf{E}[y(0)] \\ &\geq \mathsf{E}[y(K \wedge k)] \\ &\geq \mathsf{E}[y(K \wedge k) 1_{K<k}] \\ &= \mathsf{E}\left[e^{\theta_b\left(\sum_{i=1}^K \max_n x_{n,i} - \sum_{i=1}^K t_i\right)} h(c_K) 1_{K<k}\right] \\ &\geq e^{\theta_b\sigma} \mathsf{E}[h(c_K) \,|\, K < \infty] \, \mathsf{P}[K < k] \ . \end{aligned}$$

Taking $k \to \infty$ we obtain the bound

$$\mathsf{P}[K < \infty] \le \frac{\mathsf{E}[h(c_1)]}{\mathsf{E}[h(c_K) \mid K < \infty]} e^{-\theta_b \sigma} \le e^{-\theta_b \sigma} ,$$

where we used Lemma 7 for the last inequality. The proof for r is analogous. □

A close comparison of the waiting time bound in the non-renewal case (31) to the corresponding bound in the renewal case (18) reveals that the decay factors θ_b depend on similar conditions, whereby the MGF of the interarrival times in (18) is replaced by the spectral radius of the modulating Markov chain in (31). Moreover, given the ergodicity of the underlying Markov chain, the blocking system with non-renewal input is subject to the same degrading stability region (in $\log N$) as in the renewal case (recall (21)).

For quick numerical illustrations of the tightness of the bounds on the waiting time distributions in both the non-blocking and blocking cases we refer to Figure 6.

So far we have contributed stochastic bounds on the steady-state waiting and response time distributions in FJ systems fed with either renewal and non-renewal job arrivals. The key technical insight was that the stochastic bounds in the non-blocking model grow as $\mathcal{O}(\log N)$ in the number of parallel servers N under non-renewal arrivals, which extends a known result for renewal arrivals [26, 4]. The same fundamental factor of $\log N$ was shown to drive the stability region in the blocking model. A concrete application follows next.

5. APPLICATION TO WINDOW-BASED PROTOCOLS OVER MULTIPATH ROUTING

In this section we slightly adapt and use the non-blocking FJ queueing system from Section 3.1 to analyze the performance of a *generic* window-based transmission protocol over multipath routing. While this problem has attracted much interest lately with the emergence of multipath TCP [30], it is subject to a major difficulty due to the likely overtaking of packets on different paths. Consequently, packets have to additionally wait for a *resequencing delay*, which directly corresponds to the synchronization constraint in FJ systems. We note that the employed FJ non-blocking model is subject to a convenient simplification, i.e., each path is modelled by a single server/queue only.

As depicted in Figure 7, we consider an arrival flow containing l batches of N packets, with $l \in \mathbb{N}$, at the fork node A. In practice, a *packet* as denoted here may represent an entire train of consecutive datagrams. The incoming packets are sent over multiple paths to the destination node B, where they need to be eventually reordered. We assume that the batch size corresponds to the transmission window size of the protocol, such that one packet traverses a single path only. For example, the first path transmits the packets $\{1, N+1, 2N+1, \dots\}$, i.e., packets are distributed in a round-robin fashion over the N paths. We also assume that packets on each path are delivered in a (locally-)FIFO order, i.e., there is no overtaking on the same path.

In analogy to Section 3.1, we consider a batch waiting until its last packet starts being transmitted. When the transmission of the last packet of batch j begins, the previous batch has already been received, i.e., all packets of the batch $j-1$ are *in order* at node B.

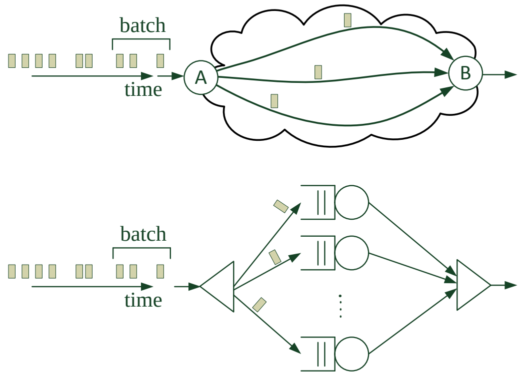


Figure 7: A schematic description of the window-based transmission over multipath routing; each path is modelled as a single server/queue.

We are interested in the response times of the batches, which are upper bounded by the largest response time of the packets therein. The arrival time of a batch is defined as the latest arrival time of the packets therein, i.e., when the batch is entirely received. Formally, the response time of batch $j \in \{lN+1 \mid l \in \mathbb{N}\}$ can be given by slightly modifying (2), i.e.,

$$r_j = \max_{0 \le k \le j-1} \left\{ \max_n \left\{ \sum_{i=0}^{k} x_{n,j-i} - \sum_{i=1}^{k} t_{n,j-i} \right\} \right\} .$$

The corresponding steady-state response time has the modified representation

$$r =_{\mathcal{D}} \max_{k \ge 0} \left\{ \max_n \left\{ \sum_{i=0}^{k} x_{n,i} - \sum_{i=1}^{k} t_{n,i} \right\} \right\} .$$

The modifications account for the fact that the packets of each batch are asynchronously transmitted on the corresponding paths (instead, in the basic FJ systems, the tasks of each job are simultaneously mapped). In terms of notations, the $t_{n,i}$'s now denote the interarrival times of the packets transmitted over the same path n, whereas $x_{n,i}$'s are iid and denote the transmission time of packet i over path n; as an example, when the arrival flow at node A is Poisson, $t_{n,i}$ has an Erlang E_N distribution for all n and i.

We next analyze the performance of the considered multipath routing for both renewal and non-renewal input.

Renewal Arrivals

Consider first the scenario with renewal interarrival times. Similarly to Section 3.1 we bound the distribution of the steady-state response time r using a submartingale in the time domain $j \in \{lN+1 | l \in \mathbb{N}\}$. Following the same steps as in Theorem 1, the process

$$z_n(k) = e^{\theta\left(\sum_{i=0}^{k} x_{n,i} - \sum_{i=1}^{k} t_{n,i}\right)}$$

is a martingale under the condition

$$\mathsf{E}\left[e^{\theta x_{1,1}}\right] \mathsf{E}\left[e^{-\theta t_{1,1}}\right] = 1 ,$$

where we used the filtration

$$\mathcal{F}_k := \sigma\{x_{n,m}, t_{n,m} | m \le k, n \in [1, N]\} .$$

Note that $\mathsf{E}\left[e^{-\theta t_{1,1}}\right]$ denotes the Laplace transform of the interarrival times of packets transmitted over each path. The proof that $\max_n z_n(k)$ is a submartingale follows a similar argument as in (10). Hence, we can bound the distribution of the steady-state response time as

$$\mathsf{P}\left[r \geq \sigma\right] \quad \leq \quad N\mathsf{E}\left[e^{\theta x_{1,1}}\right] e^{-\theta\sigma} \ , \tag{32}$$

with the condition on θ from above.

Non-Renewal Arrivals

Next, consider a scenario with non-renewal interarrival times t_i of the packets arriving at the fork node A in Figure 7, as described in Section 4. On every path $n \in [1, N]$ the interarrivals are given by a sub-chain $(c_{n,k})_k$ that is driven by the N-step transition matrix $T^N = (\alpha_{i,j})_{i,j}$ for T given in (23). Similarly as in the proof of Theorem 3, we will use an exponential transform $(T^N)_\theta$ of the transition matrix that describes each path n, i.e.,

$$(T^N)_\theta := \begin{pmatrix} \alpha_{1,1}\beta_1 & \alpha_{1,2}\beta_2 \\ \alpha_{2,1}\beta_1 & \alpha_{2,2}\beta_2 \end{pmatrix} \ ,$$

with $\alpha_{i,j}$ defined above and β_1, β_2 being the elements of a vector β of conditional Laplace transforms of N consecutive interarrival times t_i. The vector β is given by

$$\beta := \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \mathsf{E}\left[e^{-\theta\sum_{i=1}^{N} t_i} \,\middle|\, c_1 = 1\right] \\ \\ \mathsf{E}\left[e^{-\theta\sum_{i=1}^{N} t_i} \,\middle|\, c_1 = 2\right] \end{pmatrix} \ ,$$

and can be computed given the transition matrix T from (23) via an exponential row transform [9] (Example 7.2.7) denoted by

$$\tilde{T}_\theta := \begin{pmatrix} (1-p)\mathsf{E}\left[e^{-\theta L_1}\right] & p\mathsf{E}\left[e^{-\theta L_1}\right] \\ \\ q\mathsf{E}\left[e^{-\theta L_2}\right] & (1-q)\mathsf{E}\left[e^{-\theta L_2}\right] \end{pmatrix} \ ,$$

yielding $\beta = (\tilde{T}_\theta)^N \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Denote $\Lambda(\theta)$ and $h = (h(1), h(2))$ as the maximal positive eigenvalue of the matrix $(T^N)_\theta$ and the corresponding right eigenvector, respectively. Mimicking the proof of Theorem 3, one can show for every path n that the process

$$z_n(k) = h(c_{n,k}) e^{\theta\left(\sum_{i=0}^{k} x_{n,i} - \sum_{i=1}^{k} t_{n,i}\right)}$$

is a martingale under the condition on (positive) θ

$$\mathsf{E}\left[e^{\theta x_{1,1}}\right] \Lambda(\theta) = 1 \ . \tag{33}$$

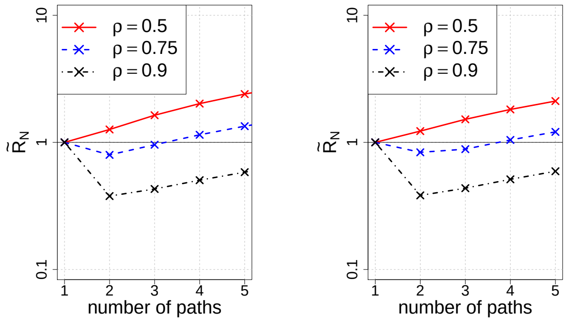
Given the martingale representation of the processes $z_n(k)$ for every path n, the process

$$z(k) = \max_n z_n(k)$$

is a submartingale following the line of argument in (10). We can now use (30) and the remark at the end of Section 4.1 to bound the distribution of the steady-state response time r as

$$\mathsf{P}\left[r \geq \sigma\right] \leq \frac{\mathsf{E}\left[h(c_{1,1})\right]}{h(2)} N\mathsf{E}\left[e^{\theta x_{1,1}}\right] e^{-\theta\sigma} \ , \tag{34}$$

where we also used that h is monotonically decreasing and θ as defined in (33).



(a) Renewal (b) Non-renewal

Figure 8: Multipath routing reduces the average batch response time when $\tilde{R}_N < 1$; smaller $\tilde{R}_N$ corresponds to larger reductions. Baseline parameter $\mu = 1$ and non-renewal parameters: $p = 0.1, q = 0.4$, $\lambda_1 = \{0.39, 0.7, 0.88\}, \lambda_2 = 0.95$, yielding the utilizations $\rho = \{0.5, 0.75, 0.9\}$ (from top to bottom).

As a direct application of the obtained stochastic bounds (i.e., (32) and (34)), consider the problem of optimizing the number of parallel paths N subject to the batch delay (accounting for both queueing and resequencing delays). More concretely, we are interested in the number of paths N minimizing the overall average batch delay. Note that the path utilization changes with N as

$$\rho = \frac{\lambda}{N\mu} \ ,$$

since each path only receives $\frac{1}{N}$ of the input. In other words, the packets on each path are delivered much faster with increasing N, but they are subject to the additional resequencing delay (which increases as $\log N$ as shown in Section 3.1).

To visualize the impact of increasing N on the average batch response times we use the ratio

$$\tilde{R}_N := \frac{E\left[r_N\right]}{E\left[r_1\right]} \ ,$$

where, with abuse of notation, $E\left[r_N\right]$ denotes a bound on the average batch response time for some N, and $E\left[r_1\right]$ denotes the corresponding baseline bound for $N = 1$; both bounds are obtained by integrating either (32) or (34) for the renewal and the non-renewal case, respectively.

In the renewal case, with exponentially distributed interarrival times with parameter λ, and homogenous paths/servers where the service times are exponentially distributed with parameter μ, we obtain

$$\tilde{R}_N = \left(\frac{\log(N\mu/(\mu-\theta)) + 1}{\log(1/\rho) + 1}\right)\left(\frac{\mu - \lambda}{\theta}\right) \ , \tag{35}$$

where θ is the solution of

$$\frac{\mu}{\mu - \theta}\left(\frac{\lambda}{\lambda + \theta}\right)^N = 1 \ .$$

In the non-renewal case we obtain the same expression for $\tilde{R}_N$ as in (35) except for the additional prefactor $\frac{\mathsf{E}[h(c_1(1))]}{h(2)}$ prior to N; moreover, θ is the implicit solution from (33).

Figure 8 illustrates $\tilde{R}_N$ as a function of N for several utilization levels ρ for both renewal (a) and non-renewal (b) input; recall that the utilization on each path is $\frac{\rho}{N}$. In both cases, the fundamental observation is that at small utilizations (i.e., roughly when $\rho \leq 0.5$), multipath routing increases the response times. In turn, at higher utilizations, response times benefit from multipath routing but only for 2 paths. While this result may appear as counterintuitive, the technical explanation (in (a)) is that the waiting time in the underlying $E_N/M/1$ queue quickly converges to $\frac{1}{\mu}$, whereas the resequencing delay grows as $\log N$; in other words, the gain in the queueing delay due to multipath routing is quickly dominated by the resequencing delay price.

6. CONCLUSIONS

In this paper we have provided the first computable and non-asymptotic bounds on the waiting and response time distributions in Fork-Join queueing systems. We have analyzed four practical scenarios comprising of either workconserving or non-workconserving servers, which are fed by either renewal or non-renewal arrivals. In the case of workconserving servers, we have shown that delays scale as $\mathcal{O}(\log N)$ in the number of parallel servers N, extending a related scaling result from renewal to non-renewal input. In turn, in the case of non-workconserving servers, we have shown that the same fundamental factor of $\log N$ determines the system's stability region. Given their inherent tightness, our results can be directly applied to the dimensioning of Fork-Join systems such as MapReduce clusters and multipath routing. A highlight of our study is that multipath routing is reasonable from a queueing perspective for two routing paths only.

Acknowledgement

This work was partially funded by the DFG grant Ci 195/1-1.

7. REFERENCES

[1] Amazon Elastic Compute Cloud EC2. `http://aws.amazon.com/ec2`.

[2] S. Babu. Towards automatic optimization of MapReduce programs. In *Proc. of ACM SoCC*, pages 137–142, 2010.

[3] F. Baccelli, E. Gelenbe, and B. Plateau. An end-to-end approach to the resequencing problem. *J. ACM*, 31(3):474–485, June 1984.

[4] F. Baccelli, A. M. Makowski, and A. Shwartz. The Fork-Join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds. *Adv. in Appl. Probab.*, 21(3):629–660, Sept. 1989.

[5] S. Balsamo, L. Donatiello, and N. M. Van Dijk. Bound performance models of heterogeneous parallel processing systems. *IEEE Trans. Parallel Distrib. Syst.*, 9(10):1041–1056, Oct. 1998.

[6] P. Billingsley. *Probability and Measure*. Wiley, 3rd edition, 1995.

[7] O. Boxma, G. Koole, and Z. Liu. Queueing-theoretic solution methods for models of parallel and distributed systems. In *Proc. of Performance Evaluation of Parallel and Distributed Systems. CWI Tract 105*, pages 1–24, 1994.

[8] E. Buffet and N. G. Duffield. Exponential upper bounds via martingales for multiplexers with Markovian arrivals. *J. Appl. Probab.*, 31(4):1049–1060, Dec. 1994.

[9] C. Chang. *Performance Guarantees in Communication Networks*. Springer, 2000.

[10] Y. Chen, S. Alspaugh, and R. Katz. Interactive analytical processing in big data systems: A cross-industry study of mapreduce workloads. *Proc. VLDB Endow.*, 5(12):1802–1813, Aug. 2012.

[11] F. Ciucu, F. Poloczek, and J. Schmitt. Sharp per-flow delay bounds for bursty arrivals: The case of FIFO, SP, and EDF scheduling. In *Proc. of IEEE INFOCOM*, pages 1896–1904, April 2014.

[12] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[13] N. Duffield. Exponential bounds for queues with Markovian arrivals. *Queueing Syst.*, 17(3–4):413–430, Sept. 1994.

[14] L. Flatto and S. Hahn. Two parallel queues created by arrivals with two demands I. *SIAM J. Appl. Math.*, 44(5):1041–1053, Oct. 1984.

[15] R. J. Gibbens. Traffic characterisation and effective bandwidths for broadband network traces. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 1996.

[16] Y. Han and A. Makowski. Resequencing delays under multipath routing - Asymptotics in a simple queueing model. In *Proc. of IEEE INFOCOM*, pages 1–12, April 2006.

[17] G. Harrus and B. Plateau. Queueing analysis of a reordering issue. *IEEE Trans. Softw. Eng.*, 8(2):113–123, Mar. 1982.

[18] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.

[19] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: Measurements & analysis. In *Proc. of ACM IMC*, pages 202–208, 2009.

[20] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan. An analysis of traces from a production MapReduce cluster. In *Proc. of IEEE/ACM CCGRID*, pages 94–103, May 2010.

[21] B. Kemper and M. Mandjes. Mean sojourn times in two-queue Fork-Join systems: Bounds and approximations. *OR Spectr.*, 34(3):723–742, July 2012.

[22] G. Kesidis, B. Urgaonkar, Y. Shan, S. Kamarava, and J. Liebeherr. Network calculus for parallel processing. *CoRR*, abs/1409.0820, 2014.

[23] J. F. C. Kingman. Inequalities in the theory of queues. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 32(1):102–110, 1970.

[24] S.-S. Ko and R. F. Serfozo. Sojourn times in G/M/1 Fork-Join networks. *Naval Res. Logist.*, 55(5):432–443, May 2008.

[25] A. S. Lebrecht and W. J. Knottenbelt. Response time approximations in Fork-Join queues. In *Proc. of UKPEW*, July 2007.

[26] R. Nelson and A. Tantawi. Approximate analysis of Fork/Join synchronization in parallel queues. *IEEE Trans. Computers*, 37(6):739–743, June 1988.

[27] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel analysis with Sawzall. *Sci. Program.*, 13(4):277–298, Oct. 2005.

[28] I. Polato, R. Ré, A. Goldman, and F. Kon. A comprehensive view of Hadoop research - a systematic literature review. *J. Netw. Comput. Appl.*, 46(0):1 – 25, Nov. 2014.

[29] F. Poloczek and F. Ciucu. Scheduling analysis with martingales. *Perform. Evaluation*, 79:56–72, Sept. 2014.

[30] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath TCP. *SIGCOMM Comput. Commun. Rev.*, 41(4):266–277, Aug. 2011.

[31] A. Rényi. On the theory of order statistics. *Acta Mathematica Academiae Scientiarum Hungarica*, 4(3–4):191–231, 1953.

[32] J. Tan, X. Meng, and L. Zhang. Delay tails in MapReduce scheduling. *SIGMETRICS Perform. Eval. Rev.*, 40(1):5–16, June 2012.

[33] J. Tan, Y. Wang, W. Yu, and L. Zhang. Non-work-conserving effects in MapReduce: Diffusion limit and criticality. *SIGMETRICS Perform. Eval. Rev.*, 42(1):181–192, June 2014.

[34] E. Varki. Mean value technique for closed Fork-Join networks. *SIGMETRICS Perform. Eval. Rev.*, 27(1):103–112, May 1999.

[35] S. Varma and A. M. Makowski. Interpolation approximations for symmetric Fork-Join queues. *Perform. Eval.*, 20(1–3):245–265, May 1994.

[36] E. Vianna, G. Comarela, T. Pontes, J. Almeida, V. Almeida, K. Wilkinson, H. Kuno, and U. Dayal. Analytical performance models for MapReduce workloads. *Int. J. Parallel Prog.*, 41(4):495–525, Aug. 2013.

[37] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2009.

[38] Y. Xia and D. Tse. On the large deviation of resequencing queue size: 2-M/M/1 case. *IEEE Trans. Inf. Theory*, 54(9):4107–4118, Sept. 2008.

[39] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica. Improving MapReduce performance in heterogeneous environments. In *Proc. of USENIX OSDI*, pages 29–42, Dec. 2008.

APPENDIX

We assume throughout the paper that all probabilistic objects are defined on a common filtered probability space $\left(\Omega, \mathcal{A}, (\mathcal{F}_n)_n, \mathsf{P}\right)$. All processes $(X_n)_n$ are assumed to be *adapted*, i.e., for each $n \geq 0$, the random variable X_n is $\mathcal{F}_n$-measurable.

DEFINITION 5. *(*MARTINGALE*) An integrable process $(X_n)_n$ is a* martingale *if and only if for each $n \geq 1$*

$$\mathsf{E}\left[X_n \mid \mathcal{F}_{n-1}\right] = X_{n-1} \ . \tag{36}$$

Further, X is said to be a sub-(super-)martingale if in (36) we have $\geq$ ($\leq$) instead of equality.

The key property of (sub, super)-martingales that we use in this paper is described by the following lemma:

LEMMA 6. *(*OPTIONAL SAMPLING THEOREM*) Let $(X_n)_n$ be a martingale, and K a bounded stopping time, i.e., $K \leq n$ a.s. for some $n \geq 0$ and $\{K = k\} \in \mathcal{F}_k$ for all $k \leq n$. Then*

$$\mathsf{E}\left[X_0\right] = \mathsf{E}\left[X_K\right] = \mathsf{E}\left[X_n\right] \ . \tag{37}$$

If X is a sub-(super-)martingale, the equality sign in (37) is replaced by $\leq$ ($\geq$).

PROOF. See, e.g., [6]. □

Note that for *any* (possibly unbounded) stopping time K, the stopping time $K \wedge n$ is always bounded. We use Lemma 6 with the stopping times $K \wedge n$ in the proofs of Theorems 1 – 4.

LEMMA 7. *Let c_k be the Markov chain from Figure 4 and K be the stopping time from* (11)*. Then the distribution of $(c_K \mid K < \infty)$ is stochastically smaller than the steady-state distribution of c_k, i.e.,*

$$\mathsf{P}\left[c_K = 2 \mid K < \infty\right] \leq \mathsf{P}\left[c_1 = 2\right] \ ,$$

or, equivalently,

$$\mathsf{E}\left[h(c_K) \,|\, K < \infty\right] \geq \mathsf{E}\left[h(c_k)\right] \ ,$$

for all monotonically decreasing functions h on $\{1, 2\}$.

PROOF. Using Bayes' rule and the stationarity of the process c_k, it holds:

$$\begin{aligned}\mathsf{P}\left[c_K = 2 \mid K < \infty\right] &= \sum_{k=1}^{\infty} \mathsf{P}\left[c_k = 2 \mid K = k\right] \mathsf{P}\left[K = k\right] \\ &= \sum_{k=1}^{\infty} \mathsf{P}\left[K = k \mid c_k = 2\right] \mathsf{P}\left[c_k = 2\right] \\ &= \mathsf{P}\left[c_1 = 2\right] \sum_{k=1}^{\infty} \mathsf{P}\left[K = k \mid c_k = 2\right] \ .\end{aligned}$$

Since L_1 is stochastically smaller than L_2, we have for any $k \geq 1$

$$\begin{aligned}&\mathbb{P}[K = k \mid c_k = 2] \\ &= \mathsf{P}\left[t_k \leq \max_n \sum_{i=1}^{k} x_{n,i} - \sum_{i=1}^{k-1} t_i - \sigma, \max_n \sum_{i=1}^{k-1} (x_{n,i} - t_i) < \sigma \,\middle|\, c_k = 2\right] \\ &\leq \mathsf{P}\left[t_k \leq \max_n \sum_{i=1}^{k} x_{n,i} - \sum_{i=1}^{k-1} t_i - \sigma, \max_n \sum_{i=1}^{k-1} (x_{n,i} - t_i) < \sigma\right] \\ &= \mathsf{P}\left[K = k\right] \ .\end{aligned}$$

Hence $\sum_{k=1}^{\infty} \mathsf{P}\left[K = k \mid c_k = 2\right] \leq 1$, which completes the proof. □