Dynamic control of the join-queue lengths in saturated fork-join stations

Andrea Marin and Sabina Rossi

DAIS - Università Ca' Foscari, Venezia, Italy
{marin,srossi}@dais.unive.it

Abstract. The analysis of fork-join queueing systems has played an important role for the performance evaluation of distributed systems where parallel computations associated with the same job are carried out and a job is considered served only when all the parallel tasks it consists of are served and then joined. The fork-join nodes that we consider consist of $K \geq 2$ parallel servers each of which is equipped with two FCFS queues, namely the service-queue and the join-queue. The former stores the tasks waiting for being served while the latter stores the served tasks waiting for being joined. When the queueing station is saturated, i.e., the service-queues are never empty, we observe that the join-queue sizes tend to grow infinitely even if the expected service times at the servers are the same. In fact, this is due to the variance of the service time distribution. To tackle this problem, we propose a simple service-rate control mechanism, and show that under the exponential assumption on the service times, we can analytically study a set of relevant performance indices. We show that by selectively reducing the speed of some servers, significant energy saving can be achieved.

1 Introduction

Fork-join queueing stations have been extensively studied in the literature because of their wide applications in the context of distributed and parallel systems. Such queueing stations behave as follows: jobs arrive according to a certain arrival process and are forked into K tasks that are enqueued in the *service-queues* and then served by independent servers. Once a task is served, it is enqueued in the *join-queue* waiting for the service completions of all the other tasks of the job it belongs to. Once all the tasks of a job are served, the *join* operation is performed and the job leaves the system. In this work we assume that all the queues implement a First Come First Served (FCFS) discipline.

Fork-join queues have found applications in a wide variety of domains in computer science and telecommunication networks. For instance, in [21] the authors study the response times of multiprocessor systems by means of fork-join networks, in [10] the authors consider parallel communication systems and in [12] a RAID system is studied by simulating a fork-join station.

Unfortunately, despite their importance, few analytical results are known for fork-join stations. One of the reasons is the complexity of the model consisting

of two sets of queues, the service-queues and the join-queues, and no general decomposition result is available at the state of the art [1]. Many works have considered the fork-join station under heavy traffic (see, e.g., [13]) and provided approximations of the expected response time based on the analysis of the associated reflecting Brownian motion [18]. In this scenario we observe that when $K >> 2$ the join-queues tend to be very long because each served task has to wait for the completion of the slowest of its siblings (which may also be enqueued at their servers). In [20] the authors observe that such a system can be highly inefficient both because it handles long join-queues and because the servers work at maximum speed even if their join-queue length is very long. Significant energy saving can be obtained by slowing down the servers that have already served more tasks than others.

1.1 Contribution

In this work we introduce a rate control mechanism for the station's servers that allows us to control the join-queue lengths and to reduce the system's power consumption. The importance of containing the size of the output buffer and reducing the energy consumption is well-known in the literature, e.g., [22, 23, 20]. In contrast with [20], we do not require the estimation of the amount of work needed by a task, but we base our algorithm on a single state variable associated with each server. We assume that each server has a neighbour defined to form a circular dependency. For instance, the neighbour of server i can be server $(i \ mod \ K)+1$. If a server has completed less or equal tasks than its neighbour then it works at maximum speed, otherwise it reduces its speed by a certain factor. Therefore, each server has to maintain a single variable that is incremented by 1 at each local task completion, while it is decremented by 1 when a task completion occurs at its neighbour. Our contribution includes an analytically tractable model of such a rate control mechanism. We start by considering the Flatto-Hahn-Wright (FHW) model [8, 25] in saturation, i.e., the service times are modelled by independent and identically distributed (i.i.d.) exponential random variables, the join operation is instantaneous, and the service-queues are never empty. We show that even in the case of two servers ($K = 2$), the stochastic process modelling the join-queue lengths is unstable because of the variance in the service times. Conversely, by the introduction of our rate-control mechanism we show that, for any $K \geq 2$, the process underlying the join-queue lengths becomes stable and their expectation is finite. Moreover, we are able to derive an analytical expression for the system's throughput. The stationary probabilities, the marginal stationary probabilities and the throughput are expressed in terms of Kummer's confluent hypergeometric functions. In general, the evaluation of such functions can be done by numerical approximations, but in our case the evaluation points are such that a closed form expression is always known.

Finally, we study by simulation the behaviour of our algorithm when the service times are not exponentially distributed and show the impact of the service times' coefficients of variation (CV) on the performance indices.

1.2 Related work

In [9] the authors extend their previous work on fork-join queueing networks in order to include join nodes and apply an approximate analysis to study their stationary performance indices based on a decomposition technique or an iterative solution of tractable models. In [8, 25] the authors introduce the so called Flatto-Han-Wright model [18] consisting of only two exponential servers. They derive the stability conditions and propose an approximate analysis as well as some exact results on the conditional join-queue lengths. In [17] the authors provide the exact expression of the mean response time for the FHW model, when $K = 2$ and the service times are i.i.d. exponential random variables. They also give an approximation technique to study the models with $K > 2$. In [2, 3] the authors study the stability conditions for a set of fork-join queueing networks. In [18] the author applies the method based on the heavy traffic assumption that lead to important results in queueing network analysis for studying the fork-join queueing nodes. Order statistics has been used to solve a class of fork-join queues with block-regular structure in [7].

The work that is probably closer to the one proposed here is [20] where the authors propose to reduce the energy consumption of a fork-join station by slowing down the servers that work on tasks with lower needs. They devise a scheduling algorithm and prove an optimality property. However, in contrast to what we propose here, the method requires the estimation of the tasks' service demands which is not always possible. In [22, 23] the authors propose an approach based on the order statistics that introduces deterministic delays at the servers aiming at reducing the task dispersion. The delays are determined so that the 100αth percentile of variability of the distributions obtained once the delays are inserted is minimised.

1.3 Structure of the paper

The paper is structured as follows. In Section 2 we introduce the problem that we aim to address and describe the algorithm that we propose. In Section 3 we provide an analytical model for the performance evaluation of the algorithm under the assumptions of saturated station and exponential service time distributions. Section 4 studies the performance of the rate-control algorithm by using the results of the previous section and the stochastic simulation. Finally, Section 5 gives some concluding remarks.

2 Rate-control algorithm

In this section we formally introduce the problem we are studying and the rate-control algorithm that we propose. In the following sections we study the performance of such an algorithm in terms of throughput and energy saving.

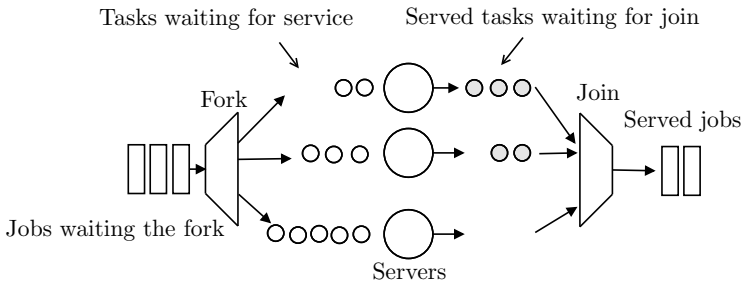


Fig. 1: Fork-join queueing station with $K = 3$ servers

2.1 Problem statement

Let us consider a fork-join queueing system with K servers as depicted in Figure 1. We consider a saturated model, i.e., there is always a job waiting to be processed. As a consequence the service-queues always contain at least one task. The service times are modelled by i.i.d. continuous time random variables and we initially assume that the join operation occur immediately after all the tasks belonging to the same job are served. All the queues follow a FCFS discipline. Clearly, if the expected service time at the servers is not the same, and if a rate-control mechanism is not applied, then the join-queue length of the fastest server tend to grow infinitely as time $t \to \infty$. Less obvious is the case in which all the service times are independent and identically distributed, i.e., with the same mean. In these cases, the variance of the service time causes an unbounded growth of the join-queue population, i.e., the expected join-queue lengths at the servers tend to infinity as $t \to \infty$. In Figure 2 we show a transient simulation of the saturated model with three service time distributions: Erlang-2, hyperexponential and exponential. The confidence intervals have been build on 15 independent executions of the simulation with a confidence of 95%. The plot supports the intuition that higher coefficient of variations in the service times make the expected queue lengths grow faster. We formally prove the model instability if the service times are exponentially distributed.

Proposition 1. *In the long run, the saturated fork-join model with $K \geq 2$, i.i.d. exponential service times, immediate join, has an infinite expectation of the join-queue length.*

Proof. For brevity, we give the proof for $K = 2$. The state space of the model is

$$\mathcal{S} = \{(n_1, n_2) : n_1 = 0 \vee n_2 = 0, n_i \in \mathbb{N}\},$$

where n_i denotes the join-queue length of server i. The transitions are from state $(0, n_2)$ to $(0, n_2 + 1)$ or to $(0, n_2 - 1)$ and from state $(n_1, 0)$ to $(n_1 + 1, 0)$ or $(n_1 - 1, 0)$. Since the service times are exponentially distributed, then the stochastic process is a continuous time Markov chain, and specifically it is a random walk

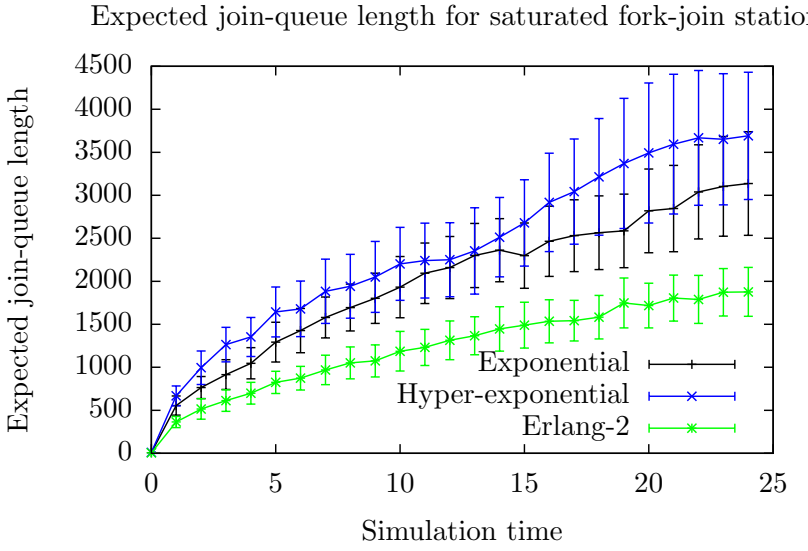


Fig. 2: Growth of the expected join-queue length for $K = 20$ servers, exponential ($CV = 1$), Erlang-2 ($CV = \sqrt{2}/2$), Hyper-exponential ($CV = 1.31$)

on the line. In this CTMC all the rates are equal and hence the states are not positive recurrent. Therefore, let Q be the random variable associated with the join-queue length for one of the two servers at a time t_0, with $t_0 \to \infty$, then $E[Q] = \infty$. If $K > 2$ the proof is similar but the CTMC is multidimensional. □

We devise an algorithm that dynamically controls the service rates (e.g., by scaling the operating frequency of the processors) with the following aims:

- Having a finite expectation of the join-queue lengths;
- Maintaining the throughput at reasonable high levels;
- Reducing the overall energy consumption by controlling the servers' rates.

Moreover, we will see that if the service rates are exponentially distributed, then a Markovian model with analytically tractable solution exists, therefore one can tackle problems of optimisation or capacity planning that would be expensive to address by stochastic simulation.

2.2 The rate-control algorithm

The main idea of the algorithm is to slow down the servers that have already completed their work on many tasks whereas the servers that have served less tasks will work at maximum speed. Since it would be unrealistic to assume that each server can take a decision about its own speed by knowing the global state of

the system, we introduce a policy that implements a rate-control strategy by just maintaining a single integer state variable. Let us label each of the K servers with integer numbers in $\{1, \ldots, K\}$ and define the following neighbourhood relation: for each server k we define its neighbour $ne(k)$ as:

$$ne(k) = \begin{cases} k+1 & \text{if } k < K \\ 1 & \text{if } k = K \end{cases} .$$

Let n_k denote the state variable of each server. When server k completes a task, then n_k is increased by 1, while when its neighbour completes a task n_k is decreased by 1. In other words, n_k maintains the difference between the join-queue length of server k and $ne(k)$. Let $\mu(n_k)$ be the local state dependent service rate at a server (recall that they are all stochastically identical), then:

$$\mu(n_k) = \begin{cases} \frac{\mu}{n_k+1} & \text{if } n_k \geq 0 \\ \mu & \text{otherwise} \end{cases} . \tag{1}$$

Intuitively, when a server k has completed less or the same number of tasks than $ne(k)$ then it works at its full service speed, otherwise it slows down in a proportional way with the number of exceeding jobs. Notice that for server k, the key point for regulating the join-queue length is to consider the difference in the queue lengths of the servers rather than the total length of its join queue. Indeed this latter value could be high because of some delay in the join operation, while the mechanism that we propose is based on balancing the number of tasks served by each server.

3 Analytical model for the rate-control mechanism

In this section we consider the FHW model equipped with our rate control mechanism, i.i.d. exponentially distributed service times, immediate join and in saturation. Let us consider the vector $\mathbf{n} = (n_1, \ldots, n_K)$ of the state variables of each server, and observe that at each time epoch we have $\sum_{k=1}^{K} n_k = 0$. We aim at studying the stochastic process $\mathbf{n}(t)$ on the state space $\mathcal{S} = \{\mathbf{n} = (n_1, \ldots, n_K) : n_k \in \mathbb{Z},\ \sum_{k=1}^{K} n_k = 0\}$. Since the service rates are the only events that cause a state change, from the fact that they are exponentially distributed we conclude that $\mathbf{n}(t)$ is a homogeneous CTMC. Although we will derive a product-form expression for the invariant measure of $\mathbf{n}(t)$, it is worth of notice that $\mathbf{n}(t)$ is *not* reversible for $K > 2$. In fact, consider state $(0, 0, 0)$ and assume that server 2 completes a task taking the state of the process to $(-1, 1, 0)$. It should be clear that there does not exist any transition bringing back the model to $(0, 0, 0)$. One path that brings back the model state to $(0, 0, 0)$ is that consisting of a sequence of transitions associated with one task completion at servers 1 and 3.

Before proceeding with the analysis we have to introduce the regularized Kummer's confluent hypergeometric function $\mathbf{M}(a, b, x)$ defined as follows (the

first equality shows an alternative common notation):

$$\mathbf{M}(a,b,x) = {}_1\tilde{F}_1(a;b;x) = \frac{1}{\Gamma(b)} M(a,b,x) \qquad a,b \in \mathbb{N}^+ , \tag{2}$$

where $\mathrm{M}(a,b,x)$ is the Kummer's confluent hypergeometric function defined by the series

$$M(a,b,x) = {}_1F_1(a;b;x) = \sum_{k=0}^{\infty} \frac{(a)_k}{(b)_k} \frac{x^k}{k!} \qquad a,b \in \mathbb{N}^+ , \tag{3}$$

Γ is the Euler's Gamma function and $(y)_k$ is the Pochhammer's symbol, i.e., $(y)_k = y(y+1)\cdots(y+k-1)$.

Theorem 1. *Given the CTMC $\mathbf{n}(t)$, we have that:*

1. *$\mathbf{n}(t)$ is ergodic, i.e., it admits a unique stationary distribution $\pi_K(\mathbf{n})$;*
2. *The stationary distribution is given by the following expression:*

$$\pi_K(\mathbf{n}) = \frac{1}{G_K} \frac{1}{\prod_{i=1}^{K} (n_i \delta_{n_i>0})!} \tag{4}$$

where we assume that empty products are equal to 1 and δ_P is 1 if proposition P is true, 0 otherwise and

$$G_K = 1 + \sum_{j=1}^{K-1} \binom{K}{j} j^{K-j} \mathbf{M}(K-j, K-j+1, j) . \tag{5}$$

We base the proof of the theorem on few lemmas: first we assume the ergodicity and derive the model's product-form expression. Then, we show that the normalising constant G_K is finite (thanks to the properties of the Kummer's confluent hypergeometric function) for finite K and hence the CTMC must be ergodic.

Lemma 1. *Assume that $\mathbf{n}(t)$ is ergodic and hence admits a unique stationary distribution. Then, its expression is that of Equation (4) where:*

$$G_K = \sum_{\mathbf{n} \in \mathcal{S}} \frac{1}{\prod_{i=1}^{K} (n_i \delta_{n_i>0})!} . \tag{6}$$

Proof. The proof can be obtained by substitution of Equation (4) in the system of global balance equations of the CTMC or by noticing that the process is dynamically reversible [11, 15, 14, 16]. Let $\mathbf{n} = (n_1, \ldots, n_K)$ and let its renaming be $\rho(\mathbf{n}) = (n_K, \ldots, n_1)$, then by [11, Thm 1.14] we have to prove that Equation (4) satisfies:

$$\pi(\mathbf{n})\mu(n_k) = \pi(\rho(\mathbf{n} + \mathbf{1}_k - \mathbf{1}_{k-1}))\mu(n_{k-1} - 1) ,$$

where $\mathbf{1}_k$ is a K-size vector with a 1 in the k-th position and zeros elsewhere and we assumed $\mathbf{1}_0 = \mathbf{1}_K$ and $n_0 = n_K$. □

Notice that since $\mathcal{S}$ is an infinite set, at the moment the fact that G_K is finite, i.e., the infinite series (6) converges, depends on the assumption of ergodicity. We now algebraically prove that (6) and (5) are equivalent and converge. As a consequence the CTMC $\mathbf{n}(t)$ is ergodic.

Lemma 2. *The series (6) is equivalent to the expression given by Equation (5) which is finite for any $K \in \mathbb{N}$, $K \geq 2$.*

Proof. Let $\mathcal{P}(\mathbf{n})$ be the multiset with all the non-negative components of $\mathbf{n}$, i.e., $\mathcal{P}(\mathbf{n}) = \{n_i : n_i \geq 0\}$ and observe that for all the states $\mathbf{n}'$ such that $\mathcal{P}(\mathbf{n}') = \mathcal{P}(\mathbf{n})$ the expression under the sum symbol of Equation (6) is the same. Let $1 \leq j \leq K-1$ and $(x_1, \ldots, x_j)$ be a tuple such that $x_i \geq 0$ for all $i = 1, \ldots, j$ and $\sum_{i=1}^{j} x_j = n$, with $n \geq 0$. Basically, j denotes the number of non-negative components in a state and n their sum. Notice that, given j and n we can count how many states have exactly j non-negative components whose sum is n. This is given by the product of the number of non-negative solutions of the Diophantine's equation $y_1 + \ldots + y_j = n$ multiplied by the number of strictly positive solutions of the Diophantine's equation $y_1 + \ldots + y_{K-j} = n$ (since the sum of all the state components is 0), i.e., we can rewrite the normalising constant as:

$$G_K = 1 + \sum_{j=1}^{K-1} \sum_{n=K-j}^{\infty} \sum_{\mathbf{x}: x_1+\ldots+x_j=n} \frac{1}{\prod_{t=1}^{j} x_t!} \binom{K}{j} \cdot \binom{n-1}{K-j-1} = 1 + \sum_{j=1}^{K-1} \binom{K}{j} \sum_{n=K-j}^{\infty} \frac{j^n}{n!} \binom{n-1}{K-j-1},$$

where the last equality follows from the multinomial theorem. Notice that the boundaries of j in the external summatory start from 1 (there cannot be any state with all negative components) and terminate at $K-1$. Indeed, the only state with all non-negative components is $\mathbf{0}$ that we take into account by summing 1 at the beginning of the right-hand-side.
We can rewrite Equation (2) as:

$$\mathbf{M}(a, b, x) = \sum_{k=0}^{\infty} \frac{(a)_k}{\Gamma(b+k)} \frac{x^k}{k!} \qquad b \in \mathbb{N}^+. \tag{7}$$

So we have:

$$
\begin{aligned}
G_K &= 1 + \sum_{j=1}^{K-1} \binom{K}{j} \sum_{w=0}^{\infty} \frac{j^{w+K-j}}{(w+K-j)!} \binom{w+K-j-1}{K-j-1} \\
&= 1 + \sum_{j=1}^{K-1} \binom{K}{j} \sum_{w=0}^{\infty} \frac{j^{w+K-j}}{(w+K-j)!} \frac{(K-j)_w}{w!} \\
&= 1 + \sum_{j=1}^{K-1} \binom{K}{j} j^{K-j} \sum_{w=0}^{\infty} \frac{j^w}{\Gamma(w+K-j+1)} \frac{(K-j)_w}{w!} \\
&= 1 + \sum_{j=1}^{K-1} \binom{K}{j} j^{K-j} \mathbf{M}(K-j, K-j+1, j)
\end{aligned}
$$

where the last equality follows from Equation (7) with $a = K-j$, $b = K-j+1$ and $x = j$. Finally, we observe that $1 < G_K < \infty$ since its definition does not involve any infinite sum and function $\mathbf{M}$ evaluated at the specified integer parameters is always finite and non-negative. □

Proof of Theorem 1. The theorem follows straightforwardly by Lemmas 1 and 2. □

In order to derive the expression for the marginal distribution of the join-queue lengths we have to consider that although the state space of each single queue ranges from $-\infty$ to $+\infty$, the joint state space is not the Cartesian product of the single state spaces. Therefore, the knowledge of G_K is not sufficient to obtain the marginal distribution. A similar situation arises when studying closed queueing networks. However, while for closed product-form queueing networks several algorithms have been proposed, e.g., [5, 4, 6], in our case we are able to express the marginal distributions in terms of (regularized) Kummer's hypergeometric functions evaluated in points whose closed-form solution is known.

Let us consider the definition of G_K given by Equation (6), and let G_k^N be the normalising constant defined as:

$$
G_k^N = \sum_{\mathbf{n} \in \mathcal{S}_k^N} \frac{1}{\prod_{i=1}^{k} (n_i \delta_{n_i>0})!} ,
$$

where $\mathcal{S}_k^N = \{(n_1, \ldots, n_k) : \sum_{i=1}^{k} n_i = N\}$. Note that $G_K = G_K^0$. Then, we can write the marginal distribution as:

$$
\pi_K^*(n) = \frac{1}{(n\delta_{n>0})!} \frac{G_{K-1}^{-n}}{G_K^0} . \tag{8}
$$

The following Lemma gives the expression for G_k^N for arbitrary $k \geq 1$ and $N \in \mathbb{Z}$.

Lemma 3. *The expression for G_k^N is:*

– *If* $N \geq 0$*:*

$$G_k^N = \frac{(k\mu)^N}{N!} + \mu^N \sum_{j=1}^{k-1} \binom{k}{j} j^{N+k-j} \, \mathbf{M}(k-j, N+k-j+1, j)\,.$$

– *If* $N < 0$ *and* $2 \leq k \leq -N$*:*

$$G_k^N = \binom{-N-1}{k-1} \mu^N + \mu^N \sum_{j=1}^{k-1} \binom{k}{j} \binom{-N-1}{k-j-1} M(-N, -N-k+j+1, j)\,.$$

– *If* $N < 0$ *and* $k > -N$*:*

$$\begin{aligned} G_k^N = \mu^N & \sum_{j=1}^{k+N-1} \binom{k}{j} j^{N+k-j} \, \mathbf{M}(k-j, N+k-j+1, j) \\ & + \mu^N \sum_{j=k+N}^{K-1} \binom{k}{j} \binom{-N-1}{k-j-1} M(-N, -N-k+j+1, j) \end{aligned}$$

– *If* $k = 1$*:*

$$G_1^N = \begin{cases} \mu^N/N! & \text{if } N \geq 0 \\ \mu^N & \text{if } N < 0 \end{cases}$$

Proof. The proof is based on hypergeometric function manipulations.

In Figure 3b we show the distribution of $\pi_K^*(n)$ for $K = 2, 5, 10, 15$. Notice that while for $K = 2$ the distribution is symmetric with respect to $n = 0$, this is not true for $K > 2$. Moreover, by increasing the value of K, numerical evidences suggest that there may exist a limiting distribution for the marginal probabilities (and hence for the throughput and the power consumption). Another important aspect is the observation that the expression of π_K and π_K^* in terms of (regularized) Kummer's confluent functions allows us to have a symbolic expression for the stationary probabilities as shown in Figure 3a for $K = 3$.

One of the most important performance indices for a rate-control algorithm is the throughput, i.e., the number of join performed by the station per unit of time. In fact, by slowing down some servers we surely decrease the system's throughput. We are able to provide an analytical expression for the station's throughput that depends on the number of servers K and the service rate μ.

Lemma 4. *The throughput* $X_K(\mu)$ *of the model in steady-state is:*

$$\begin{aligned} X_K(\mu) = \frac{\mu}{K G_K} \Bigg(K + \sum_{j=1}^{K-1} \binom{K}{j} j \Big(& j^{K-j+1} \, \mathbf{M}(K-j, K-j+2, j) \\ & - (j-1)^{K-j+1} \, \mathbf{M}(K-j, K-j+2, j-1) \\ & + (K-j) j^{K-j-1} \mathbf{M}(K-j, K-j+1, j) \Big) \Bigg). \end{aligned} \tag{9}$$

K	$X_K(\mu)$
2	$\frac{4\mu(e-1)}{K(2e-1)}$
3	$\frac{9\mu(e^2-e+1)}{K(1+3e^2)}$
4	$\frac{8\mu(2e^3+3e-2)}{K(4e^3+6e^2+2e-1)}$
5	$\frac{25\mu(6e^4+12e^3-11e+6)}{2K(15e^4+60e^35e+30E^2-5e+3)}$
6	$\frac{6\mu(24e^5+120e^4+120e^3-40e^2+53e-24)}{K(24e^5+180e^4+200e^3+20e^2+9e-4)}$
7	$\frac{147\mu(40e^6+360e^5+600e^4+100e^3+120e^2-103e+40)}{4K(210e^6+2520e^5+5250e^4+2100e^3+210e^2-77e+30)}$

Table 1: Analytical expression of the throughput for the FHW model with K severs.

Proof. The proof is based on hypergeometric function manipulations.

In Table 1 we show the analytical expression of the throughput for some values of K.

The numerical evaluations of both G_k^N and of $X_K(\mu)$ rely on the computation of the confluent hypergeometric function $\mathbf{M}(a,b,z)$ with parameters $a \in \mathbb{N}^+$, $b \in \mathbb{N}^+$ and $b > a$. Indeed, if a and b are non-negative integers, then the series (3) converges for all finite x. In particular, for $b > a$, $\mathbf{M}(a,b,z)$ converges to [19]:

$$\mathbf{M}(a,b,x) = \left(e^x \sum_{k=0}^{a-1} \frac{(1-a)_k \ (-x)^k}{k! \ (2-b)_k} - \sum_{k=0}^{b-a-1} \frac{(1-b+a)_k \ x^k}{k! \ (2-b)_k} \right) \frac{(2-b)_{a-1} \ x^{1-b}}{(a-1)!} . \quad (10)$$

4 Numerical evaluation

In this section we study the sensitivity of the throughput, the expected join-queue length and the power consumption with respect to the distribution of the service times. Then, we study the performance in terms of throughput and energy consumption of the model implementing the rate-control algorithm under the assumptions introduced in Section 3. We consider three important performance

indices: the system throughput, the expected join-queue lengths and the power consumption. While for the first index Lemma 4 gives us its analytical expression, for the latter two indices we rely on the stochastic simulation and on the bounded approximation described in Section 4.1, respectively.

4.1 The power consumption

Since our rate-control mechanism reduces the computation speed of the severs, this can be interpreted as a reduction of the operating frequency leading to a reduction of the overall server power consumption. Clearly, the minimum power consumption with maximum throughput corresponds to a situation in which the servers work at a constant maximum rate, but we have already discussed that the drawback of this approach is the infinite growth of the join-queue length in saturated models.

Under the assumptions of Section 3 we know the analytical expression of the marginal stationary distribution for each server (see Equation (8) and Lemma 3). This allows us to define a lower and upper bound of the energy consumption by truncation of the probabilities. Given an integer $E > 0$, the expected power consumption in steady-state $\overline{P}_K$ is bounded by:

$$\sum_{i=-E}^{-1} \pi_K^*(i) + \sum_{i=0}^{E-1} \pi_K^*(i) \frac{1}{(i+1)^3} < \overline{P}_K < \sum_{i=-E}^{-1} \pi_K^*(i) + \sum_{i=0}^{E-1} \pi_K^*(i) \frac{1}{(i+1)^3} + (1 - \sum_{i=-E}^{E-1} \pi_K^*(i)) ,$$

where we have assumed that the sever at maximum speed consumes 1 unit of energy for unit of time, and that the power consumption depends on the cube of the operating frequency, i.e.:

$$\overline{P}_K = \sum_{i=-\infty}^{-1} \pi_K^*(i) + \sum_{i=0}^{\infty} \pi_K^*(i) \frac{1}{(i+1)^3} .$$

Clearly, more accurate models of the relation between operating frequency and power consumption can be considered, but this is out of the scope of this paper, especially because this relation depends on the intrinsic characteristics of the processors [20]. It is important to notice that with small values of $E \simeq 10$ we obtain tight bounds for the energy consumption as shown in Figure 3c.

4.2 Sensitivity analysis

The analytical model proposed in Section 3 requires that the service times are state dependent i.i.d. exponential random variables. Under this assumption, and by considering a saturated model with immediate join, we proved the stability of the process modelling the join-queue lengths. Clearly, we expect to find a

sensitivity of the performance indices on the distribution of the service times, because it is its variance the cause of the join-queue length growth in the model without the rate-control mechanism. Figures 3d-3f show the three considered performance indices for a saturated model with immediate join. The indices with exact or approximated analytical expression have not been simulated, while the others have been obtained via stochastic simulation. For each scenario we run 15 independent experiments and considered the confidence interval of 95%. The widths of the confidence intervals are all below 1% of the measure and are too small to be visible in the plots. The warm up periods have been removed by using the Welch's method [24]. The service time distributions have mean 1 and the Erlang 2 has a coefficient of variations of $\sqrt{2}/2$ while the Hyper-Exponential has a coefficient of variation of 1.31.

4.3 Performance of the algorithm as function of the number of servers

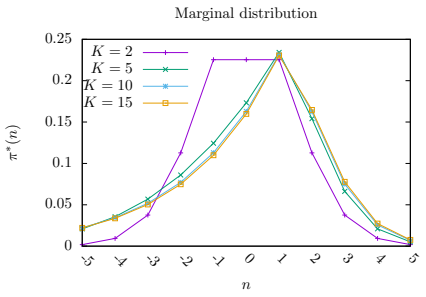
In this section we focus on the saturated FHW model with immediate join and study the impact of the number of servers K on the performance indices. Figures 3d, 3e and 3f show the system's throughput, the expected queue length for the join-queues and the power consumption for each server when the maximum service rate is $\mu = 1$. Notice that the expected queue length is for each server and is obtained by stochastic simulation. We notice that while the throughput decreases very slowly with the growth of the number of servers (e.g., for $K = 150$ servers we compute a throughput of 0.677), the expected join-queue lengths tend to grow with the number of servers and hence for large models the benefits of the rate-adaptation algorithm are lower. As for the power consumption, the power consumption is significantly lower than the reference value of the model without rate-control, 1. For instance for $K = 6$ the throughput is $X_K(1) \simeq 0.70$ while the power consumption $\overline{P}_K \simeq 0.54$.

5 Conclusion

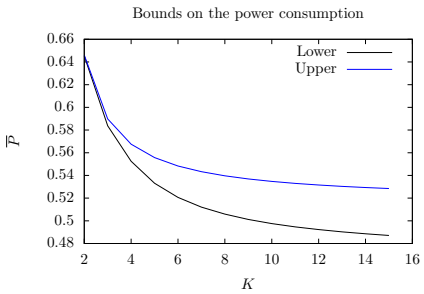
In this paper we have proposed a rate-control mechanism for fork-join stations designed to maintain the join-queue lengths finite in the long run, even when the station is saturated. We observed that the variance in the service time distribution causes an unbounded increase of the join-queue lengths. Informally, the idea behind our rate-control mechanism is to reduce the operating speed of the servers that have served more customers while maintaining at the maximum level the speed of the other servers. Each server maintains a state variable which is incremented at a local service completion event and is decremented at a service completion event occurring at a neighbour server. The servers maintain their maximum speed if the state variable is not positive, otherwise they reduce their speed. This allows for both a control of the join-queue length and a reduction on the system's power consumption. However, we also observed a reduction in the system's throughput. Despite the few analytical results available for fork-join

n	$\pi_3^*(n)$	n	$\pi_3^*(n)$
-3	$2\frac{e-2}{3e^2+1}$	1	$2\frac{e}{3e^2+1}$
-2	$\frac{2e-3}{3e^2+1}$	2	$\frac{2e+1}{6e^2+2}$
-1	$2\frac{e-1}{3e^2+1}$	3	$\frac{e+1}{9e^2+3}$
0	$\frac{2e-1}{3e^2+1}$	4	$\frac{2e+3}{72e^2+24}$

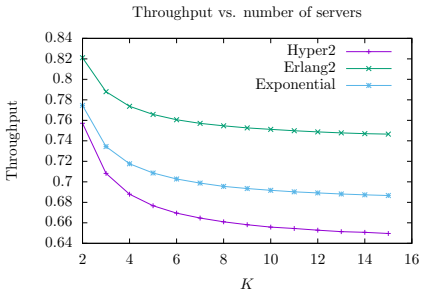
(a) Marginal distribution for $K = 3$



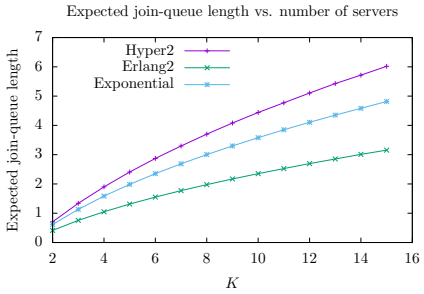
(b) Plot of marginal distributions



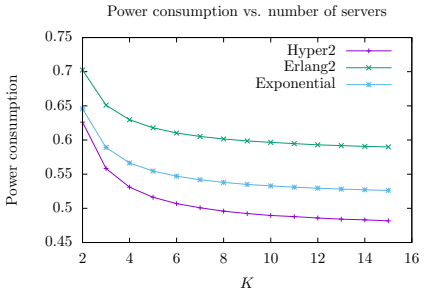
(c) Bounds of the power consumption



(d) Throughput as function of K with $\mu = 1$



(e) Expected join-queue length as function of K with $\mu = 1$



(f) Power consumption as function of K with $\mu = 1$

Fig. 3: Numerical Evaluation.

stations, we have provided the analytical expression for the steady-state distribution of the rate-control model and derived the marginal distributions for each server and the system's throughput under the FHW assumptions. The stationary distributions and the performance indices are expressed in terms of Kummer's confluent hypergeometric functions which are evaluated at special points that require the computations of finite sum. We resorted to the simulation for studying the impact of the rate-control algorithm on stations with different service time distributions and the experiments have supported the intuition that the performance degrades with the increase of the variance in the service time distribution. The main strengths of the proposed mechanism are the easiness of implementation, since the algorithm is basically stateless and does not require nor the estimation of the jobs' service times as in [20], neither the knowledge of the service time distributions as in [22, 23], and the effectiveness in drastically reducing the expected join-queue lengths with respect to the models not implementing any rate-control mechanism for the servers.

With respect to a solution which addresses the problem of containing the join-queue length based on a rate adaptation mechanism that considers for each server its associated join-queue length, our approach has the advantage that its implementation is independent of the system's parameters since it aims at balancing the total work performed by each server. Conversely, the join-queues may be long because the join operation's rate is close to the system's throughput and hence considering only its instantaneous state for deciding the service rate can be counter-productive.

Future work includes the derivation of the analytical expression for other performance indices in the case of the saturated FHW model. Moreover, we aim at introducing a parameterisation of the algorithm so that we can control the servers' speed more accurately, e.g., by reducing the service rate for positive states n by $\alpha n + 1$, where $0 < \alpha < 1$ is a parameter that regulates the trade-off between the throughput and the expected join-queue length. However, at the moment, no analytical solution for such a model is known.

References

1. F. Baccelli and M. A. Makowski. Queueing models for systems with synchronization constraints. *Proceedings of the IEEE*, 77(1):138–161, 1989.
2. F. Baccelli and Z. Liu. On the execution of parallel programs on multiprocessor systems: a queuing theory approach. *J. ACM*, 37(2):373–414, 1990.
3. F. Baccelli, W. A. Massey, and D. Towsley. Acyclic fork-join queuing networks. *J. ACM*, 36(3):615–642, 1989.
4. S. C. Bruell, G. Balbo, and P. V. Afshari. Mean Value Analysis of mixed, multiple class BCMP networks with load dependent service stations. *Perf. Eval.*, 4:241–260, 1984.
5. J. P. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Commun. ACM*, 16(9):527–531, 1973.
6. G. Casale. A generalized method of moments for closed queueing networks. *Perform. Eval.*, 68(2):180–200, 2011.

7. P. M. Fiorini and L. Lipsky. Exact analysis of some split-merge queues. *SIGMETRICS Perform. Eval. Rev.*, 43(2):51–53, 2015.
8. L. Flatto and S. Hahn. Two parallel queues created by arrivals with two demands. *SIAM J. on Applied Mathematics*, 44(5):1041–1053, 1984.
9. P. Heidelberger and K. Trivedi. Analytic queueing models for programs with internal concurrency. *IEEE Trans. Comput.*, C-32:73–82, 1983.
10. G.J. Hoekstra, R.D. van der Mei, and S. Bhulai. Optimal job splitting in parallel processor sharing queues. *Stochastic models*, 28:144–166, 2012.
11. F. Kelly. *Reversibility and stochastic networks*. Wiley, New York, 1979.
12. A. S. Lebrecht, N. J. Dingle, and W. J. Knottenbelt. Modelling zoned RAID systems using fork-join queueing simulation. In *Proc. of 6th European Performance Engineering Workshop, EPEW 2009 London, UK, July 9-10, 2009 Proceedings*, pages 16–29. Springer, 2009.
13. H. Lu and G. Pang. Gaussian limits for a fork-join network with nonexchangeable synchronization in heavy traffic. *Mathematics of Operations Research*, 41(2):560–595, 2016.
14. A. Marin and S. Rossi. On discrete time reversibility modulo state renaming and its applications. In *8th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS*, pages 1–8, 2014.
15. A. Marin and S. Rossi. On the relations between lumpability and reversibility. In *Proc. of the IEEE 22nd Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'14)*, pages 427–432, 2014.
16. A. Marin and S. Rossi. On the relations between Markov chain lumpability and reversibility. *Acta Informatica*, Available online:1–39, 2016.
17. R. Nelson and A. N. Tantawi. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Trans. on Computers*, 37(6), 1986.
18. V. Nguyen. Processing networks with parallel and sequential tasks: heavy traffic analysis and Browinian limits. *Annals of Applied Probability*, 3(1):28–55, 1993.
19. F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
20. T. Rauber and G. Rünger. Energy-aware execution of fork-join-based task parallelism. In *Proc. of the 20th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, (MASCOTS)*, pages 231–240, 2012.
21. D. Towsley, G. Romel, and J. Astankovic. Analysis of fork-join program response times on multiprocessors. *IEEE Trans. on Parallel and Distributed Systems*, 1(3):286–303, 1990.
22. I. Tsimashenka, W. Knottenbelt, and P.G. Harrison. Controlling variability in split-merge systems. In *Proc. of Analytical and Stochastic Modeling Techniques and Applications (ASMTA)*, pages 165–177, 2012.
23. I Tsimashenka, W.J. Knottenbelt, and P.G. Harrison. Controlling variability in split-merge systems and its impact on performance. *Annals of Oper. Res.*, page Available online, 2014.
24. P. D. Welch. On the problem of the initial transient in steady-state simulations. Technical report, IBM Watson Research Center, Yorktown Heights, NY, 1981.
25. Paul E. Wright. Two parallel processors with coupled inputs. *Advances in Applied Probability*, 24:986–1007, 1992.