Fair workload distribution for multi-server systems with pulling strategies

Andrea Marin[a,*], Sabina Rossi[a]

[a] *Università Ca' Foscari Venezia, DAIS*
Via Torino, 155
Venice, Italy

Abstract

We consider systems with a single queue and multiple parallel servers. Each server fetches a job from the queue immediately after completing its current work. We propose a pulling strategy that aims at achieving a fair distribution of the number of processed jobs among the servers. We show that if the service times are exponentially distributed then our strategy ensures that in the long run the expected difference among the processed jobs at each server is finite while maintaining a reasonable throughput. We give the analytical expressions for the stationary distribution and the relevant stationary performance indices like the throughput and the system's balance. Interestingly, the proposed strategy can be used to control the join-queue length in fork-join queues and the analytical model gives the closed form expression of the performance indices in saturation.

1. Introduction

The problem of parallel job scheduling has been widely studied in the literature with the aim of improving some performance indices, such as the throughput, the response time, the fairness or a combination of these indices. One can distinguish two basic approaches to the problem depending on the phase at which the dispatcher is placed. In the model depicted in Figure 1-(A), the dispatcher decides how to assign a job to a server according to some scheduling discipline which takes into account the queueing state and other information about the servers. In contrast, the model depicted in Figure 1-(B) stores the jobs in a shared queue and these are assigned by the dispatcher to a server as soon as it becomes available. The scheduling policy can rely on a pushing strategy, i.e., the dispatcher takes the initiative to send a job to a specific server or on a pulling strategy, i.e., the servers decide autonomously to fetch a job from the shared queue. In queueing theory, these models have been widely investigated and the impact of the scheduling discipline on the performance indices is well-known.

*Corresponding author

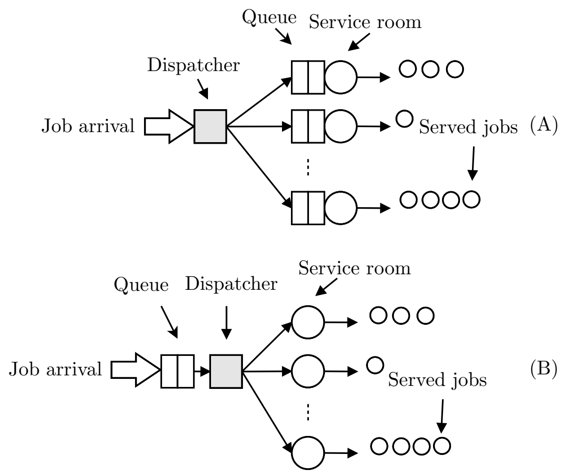

Figure 1: (A)-Dispatching before the queues. (B)-Dispatching after the queue.

In this paper we study the model depicted in Figure 1-(B) relying on a pulling strategy. In contrast to prior works, we focus on the balance of the total number of jobs served by a set of K identical servers. Informally, we propose a stateless scheduling discipline implemented by the servers so that the difference between the number of jobs served by each unity is finite in steady-state. We stress on the fact that we aim at maintaining finite the absolute difference among the jobs served by each server and not the difference of proportions of the total number of served jobs (which would be easily achieved by simple strategies like the random one). The proposed rate adaptation policy may find application in the regulation of the join-queue length in fork-join systems as discussed later on.

The contributions of the paper can be summarised as follows: *(a)* We propose a scheduling discipline based on a server rate-adaptation algorithm. Informally, each server maintains a variable to store the difference between the total number of jobs served by itself and a neighbour. Given just this piece of information, the server may decide to slow down its maximum service speed in order to reduce this difference. As soon as the server finishes the service of its job, it fetches a new one from the queue. *(b)* We study two rate-adaptation strategies. The first one, named *bimodal* strategy, uses only two distinct service rates: the highest is used when a server has served less jobs than its neighbour, while the slowest is used otherwise. The second rate-adaptation policy, named *proportional* strategy, requires a server to reduce a fixed maximum service rate in proportion to the number of extra jobs it has served with respect to its neighbour. *(c)* We propose a Markovian model for such a scheduling discipline and for both the rate-adaptation strategies described above. We show that, despite the little knowledge that each server has about the state of the system, we can derive a necessary and sufficient condition for the job balance index to have finite expectation in the bimodal strategy whereas in the proportional strategy the job balance is unconditionally finite. Finally, we compare the performance indices obtained by the application of the two rate-adaptation policies. *(d)* We derive the exact expressions for two relevant performance indices: the system's throughput and the balance index. The latter measures the differences among the total number of jobs processed by each server, hence low values imply a well balanced system. Although the Markov process underlying the models has an infinite state space, these expressions involve finite sums derived from the evaluation of hypergeometric functions.

Our findings show that maintaining reasonable low values for the balance index reduces the throughput at around 70% of the maximum. More interestingly, numerical evidences show that this value scales slowly with the number of servers, which means that the rate adaptation policy scales well with the system's size. From a theoretical point of view, to analyse our model we resort to the notion of *ρ-reversibility.* Indeed, we show that although the model is in general *not* reversible, it satisfies the Kolmogorov's criteria for the ρ-reversibility [1] (and also the dynamic reversibility [2]) allowing us to derive an analytical product-form expression for the invariant measure. This expression differs from the common product-form since the associated process is not obtained by composition of simpler components as, e.g., in [3, 4, 5].

Related work. There are several recent papers addressing the problem of parallel job scheduling which can be classified according to the system structure that they consider and the performance indices that they aim to optimise. In many cases, the scheduling problem aims at minimising the response or queueing time such as in [6, 7, 8] or optimising more complicated indices which may involve the notion of job-value as in [9]. In contrast, we propose a solution which aims at minimising the difference between the total number of jobs served by each server in steady-state. This result can be used to regulate the join-queue length in fork-join queues. Fork-join queueing systems have been widely studied for the performance evaluation of distributed systems and operating systems [10, 11, 12, 13, 14]. Models with numerically tractable stationary distributions are mostly based on product-form stochastic Petri nets (see, e.g., [15, 16, 17]) while several other works have addressed the problem of providing approximations or bounds for these queueing models (see, e.g., [11, 17]). In fork-join stations a job is split into several tasks which are served by independent servers. The job is considered served once all the parallel servers have completed their tasks and their resulting computations are joined. Examples of such a computational approach include the parallel processing of Big Data within the MapReduce framework [18], RAID disks, parallel processing systems with horizontal decomposition [19]. Moreover, differently from [20, 8], our results are not based on an asymptotic analysis of the model but they are exact for any finite number $K \geq 2$ of servers. In [21] we proposed a rate adaptation policy for fork-join queues that corresponds to the *proportional* strategy illustrated here. For the sake of comparing the bimodal and the proportional models in Section 5 we just state the performance indices of the proportional strategy. Notice that we also give an expression for the invariant measure of the underlying Markov process for any rate adaptation policy, which is a novel contribution.

Structure of the paper. In Section 2 we describe the scheduling algorithm and in Section 3 we propose a Markovian model for its analysis. In Sections 4 and 5 we study two possible implementations of the proposed scheduling strategy and derive their performance indices. In Section 6 we compare the performance indices obtained from the analytical models with the simulation outcomes obtained by relaxing some of the hypothesis. Section 7 concludes the paper.

2. Algorithm description

We consider the system depicted in Figure 1-(B) consisting of K identical servers fetching jobs from a shared unbounded queue. Our goal is to define an algorithm to dynamically regulate the service rate of each server in order to balance the total number of jobs served by each unit.

The algorithm we propose is stateless, in the sense that each node has a very limited information about the global system. Let us label each of the K servers by a positive natural number $k \in [1, K]$ and define the successor k^+ and the predecessor k^- of a server k as:

$$k^+ = \begin{cases} k+1 & \text{if } k < K \\ 1 & \text{if } k = K \end{cases} \qquad k^- = \begin{cases} k-1 & \text{if } k > 1 \\ K & \text{if } k = 1 \end{cases}.$$

Each server k maintains a state variable n_k which is incremented by one when it fetches a job and decremented by one when its successor k^+ fetches a job. Hence, each server takes into account only the jobs performed by itself and its successor and n_k denotes the difference between these two values. In practice, in implementing this algorithm by means of message passing, each server only informs its predecessor of its activity thus containing the traffic overhead. We assume that each server can regulate its own service rate depending on the local state n_k. In Sections 4 and 5 we give two possible implementations of this rate-adaptation algorithm.

3. Markovian model

In this section we propose a Markovian model for the algorithm described in Section 2 which is based on two assumptions: (1) The system has always waiting jobs, i.e., the throughput depends only on the service rates of the servers; (2) The job sizes are modelled by independent and identically distributed exponential random variables. As a consequence, since each server can dynamically adjust its own service rate depending on its state n_k, the service times at each server k are independent and exponentially distributed random variables with a state-dependent parameter $\lambda(n_k)$. Let $K \geq 2$ denote the number of servers of the analyzed system. For any $k \in \{1 \ldots, K\}$, let $N_k(t)$ be the stochastic process denoting the number of jobs which are fetched by the server k in the time interval $[0, t]$ and let $n_k(t)$ be defined as $N_k(t) - N_{k^+}(t)$. Clearly, we have that

$n_k(t) \in \mathbb{Z}$ and $\sum_{k=1}^{K} n_k(t) = 0$ for all $t \geq 0$. We are interested in studying the stochastic process $X_K(t) = (n_1(t), \ldots, n_K(t))$ on the state space

$$\mathcal{S} = \{\mathbf{n} = (n_1, \ldots, n_K) : n_i \in \mathbb{Z} \wedge \sum_{k=1}^{K} n_k = 0\}\,.$$

$X_K(t)$ is a continuous time Markov chain (CTMC) defined as follows: for $h \to 0^+$ and $t > 0$,

$$\begin{aligned} &Pr\{X_K(t+h) = \mathbf{n} - \mathbf{e}_{k^-} + \mathbf{e}_k \mid X_K(t) = \mathbf{n}\} \\ &\quad = \lambda(n_k)h + o(h) \end{aligned} \tag{1}$$

$$\begin{aligned} &Pr\{X_K(t+h) = \mathbf{n} \mid X_K(t) = \mathbf{n}\} \\ &\quad = 1 - \left(\sum_{k=1}^{K} \lambda(n_k)\right) h + o(h) \end{aligned} \tag{2}$$

where $\mathbf{e}_k$ is a K-dimensional vector with all zeros with the exception of component k which is 1. Equation (1) holds for all $k \in \{1 \ldots, K\}$ and describes the fetching of a job from server k: this occurs according to an exponentially distributed time with state-dependent rate $\lambda(n_k)$. Equation (2) models the state residence time. We will refer to such a Markov chain as $X_K(t)$. It is easy to notice that $X_K(t)$ is reversible if and only if $K = 2$ and that if we do not apply any rate adaptation policy the process is not ergodic [21].

3.1. ρ-reversibility

A stationary CTMC $Y(t)$ is said to be *reversible* if it is stochastically indistinguishable from $Y(\tau - t)$ for all $\tau, t \in \mathbb{R}$. In [2] the theory of time reversibility is presented and some important applications in the context of loss networks and classical exponential queueing systems are described. *ρ-reversibility* [1, 2, 22] extends the notion of reversibility by requiring that the time-reversed process $Y(\tau - t)$ is stochastically indistinguishable from $Y(t)$ when we apply a renaming ρ to its states. Clearly, when ρ is the identity the two definitions coincide.

Definition 1 (ρ-reversibility [23, 1]). *Let $Y(t)$ be a stationary CTMC with state space $\mathcal{S}$, and $\rho : \mathcal{S} \to \mathcal{S}$ be a bijection. Then, $Y(t)$ is* ρ-reversible *if it is stochastically indistinguishable from $Y(\tau - t)$ for all $t, \tau \in \mathbb{R}$ modulo the state renaming ρ.*

When the renaming function ρ is an involution, i.e., $\rho(\rho(s)) = s$ for all $s \in \mathcal{S}$, then the definition of ρ-reversibility is equivalent to that of *dynamic reversibility* [2, 22]. Given the renaming function ρ, proving that $Y(t)$ is ρ-reversible can be structurally done by means of the Kolmogorov's criteria or by showing that the system of detailed balance equations admits a non-trivial solution [22]. However, despite the analogies between the notions of reversibility and ρ-reversibility, we stress the fact that the main difficulty in proving that a chain is ρ-reversible consists in finding a suitable definition for the renaming ρ.

For $s, s' \in \mathcal{S}$, we denote by $q(s, s')$ the transition rate from s to s'. The proofs of Lemmas 1 and 2 can be found in [1] and in [23].

Lemma 1 (Kolmogorov's criteria). *Let $Y(t)$ be a stationary CTMC with state space $\mathcal{S}$ and ρ be a bijection on $\mathcal{S}$. Then, $Y(t)$ is ρ-reversible with respect to ρ if and only if: (K1) for each $s \in \mathcal{S}$, $\sum_{s' \in \mathcal{S}} q(s, s') = \sum_{s' \in \mathcal{S}} q(\rho(s), s')$, and (K2) for any finite sequence of states $s_1, \ldots, s_n$ with $s_i \in \mathcal{S}$, we have:*

$$
\begin{aligned}
&q(s_1, s_2)q(s_2, s_3) \cdots q(s_{n-1}, s_n)q(s_n, s_1) = \\
&\qquad\qquad q(\rho(s_1), \rho(s_n))q(\rho(s_n), \rho(s_{n-1})) \cdots q(\rho(s_2), \rho(s_1)) .
\end{aligned}
$$

Informally, (K1) requires that the residence time in a state and in its renaming are stochastically identical, while (K2) requires that given any cycle of transitions in the CTMC, the product of its rates equals the product of the rates of the inverse cycle in the renamed CTMC. Analogously to standard reversibility, there exists an efficient way for computing the stationary distribution of ρ-reversible chains.

Lemma 2 (Stationary distribution). *Let $Y(t)$ be a ρ-reversible CTMC with state space $\mathcal{S}$, π its stationary distribution and let $r, s \in \mathcal{S}$. Then, for each sequence of transitions taking the chain from state r to s*

$$
r \equiv s_1 \xrightarrow{q(s_1,s_2)} s_2 \xrightarrow{q(s_2,s_3)} \cdots \xrightarrow{q(s_{n-1},s_n)} s_n \equiv s ,
$$

we have:

$$
\pi(s) = \pi(r) \frac{\prod_{i=1}^{n-1} q(\rho(s_{i+1}), \rho(s_i))}{\prod_{i=1}^{n-1} q(s_i, s_{i+1})} .
$$

3.2. Stationary analysis of $X_K(t)$

In order to derive the stationary performance indices for the process $X_K(t)$ when $K > 2$ we resort to the notion of ρ-reversibility (recall that for $K = 2$ the process is reversible and the analysis is trivial). We adopt a constructive proof technique by deriving the expression for the stationary distribution from the properties of ρ-reversible CTMCs.

Theorem 1. *Let $K > 2$. If $X_K(t)$ is ergodic then it is ρ-reversible with respect to the renaming:*

$$
\rho(\mathbf{n}) = \mathbf{n}^R \tag{3}
$$

where $\mathbf{n}^R = (n_K, n_{K-1}, \ldots, n_1)$.

Notice that since ρ is an involution, $X_K(t)$ is also dynamically reversible. Before proving Theorem 1 we study some properties of the random walks in $X_K(t)$. Let u be a path starting from $\mathbf{n}$ and characterised by the arrivals of jobs at servers $(c_1, c_2, \dots, c_T)$ with $c_i \in \{1 \dots K\}$ and $T \in \mathbb{N}^+$:

$$u : \mathbf{n} \xrightarrow{c_1} \mathbf{n} + \mathbf{e}_{c_1} - \mathbf{e}_{c_1^-} \xrightarrow{c_2} \cdots \xrightarrow{c_t} \mathbf{n} + \sum_{w=1}^{t} \mathbf{e}_{c_w} - \sum_{w=1}^{t} \mathbf{e}_{c_w^-} \cdots$$
$$\xrightarrow{c_T} \mathbf{n} + \sum_{w=1}^{T} \mathbf{e}_{c_w} - \sum_{w=1}^{T} \mathbf{e}_{c_w^-} . \quad (4)$$

The following proposition allows us to state that for each path u we can define a reversed path u^R according to the renaming specified in Theorem 1.

Proposition 1. *For each transition $\mathbf{n} \xrightarrow{c} \mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-}$ in the transition graph of $X_K(t)$ there exists an* inverse *transition $(\mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-})^R \xrightarrow{d} \mathbf{n}^R$ where $d = K - c^- + 1$.*

Proof. The proof is trivial, indeed it is sufficient to observe that the inverse transition adds one unit in position $K - c^- + 1$ and removes one from position $K - c + 1$. $\square$

We denote by u^R the inverse path of u from state $(\mathbf{n}+\sum_{w=1}^{T} \mathbf{e}_{c_w} - \sum_{w=1}^{T} \mathbf{e}_{c_w^-})^R$ to state $\mathbf{n}^R$ with the arrivals of jobs at servers $(K - c_T^- + 1, K - c_{T-1}^- + 1, \dots, K - c_1^- + 1)$. We define $\psi(u)$ as the product of the transition rates which appear in u. Lemma 3 plays an important role in proving Theorem 1.

Lemma 3. *Let u and v be two paths from the same state $\mathbf{n}$ such that the arrival sequence in v is a permutation of that of u. Then:*

$$\frac{\psi(u)}{\psi(u^R)} = \frac{\psi(v)}{\psi(v^R)} .$$

Proof. Let $(c_1, c_2, \dots, c_T)$ be the sequence of servers fetching the jobs in an arbitrary path u. We proceed by induction on T. If $T = 1$ then $u = v$ and the proof is trivial. Let us now consider the case $T = 2$ and let (c_1, c_2) be the arrival order in u. If the permutation is the identity then $u = v$ and the result is trivial, hence we assume the arrivals in v to be (c_2, c_1). Hence:

$$u : \mathbf{n} \xrightarrow{c_1} \mathbf{n} + \mathbf{e}_{c_1} - \mathbf{e}_{c_1^-} \xrightarrow{c_2} \mathbf{n} + \mathbf{e}_{c_1} - \mathbf{e}_{c_1^-} + \mathbf{e}_{c_2} - \mathbf{e}_{c_2^-}$$

where the first transition occurs with rate $\lambda(n_{c_1})$ and the second one with rate $\lambda(n_{c_2} - \delta_{c_2 = c_1^-})$. The inverse of u is:

$$u^R : \mathbf{n}^R + \mathbf{e}_{K-c_1+1} - \mathbf{e}_{K-c_1^-+1} + \mathbf{e}_{K-c_2+1} - \mathbf{e}_{K-c_2^-+1}$$
$$\xrightarrow{K-c_2^-+1} \mathbf{n}^R + \mathbf{e}_{K-c_1+1} - \mathbf{e}_{K-c_1^-+1} \xrightarrow{K-c_1^-+1} \mathbf{n}^R \quad (5)$$

where the first transition occurs with rate $\lambda(n_{c_2^-} + \delta_{c_2^- = c_1} - 1)$ and the second one with rate $\lambda(n_{c_1^-} - 1)$. Analogously, we derive the transition rates in v where the indices c_1 and c_2 are swapped with respect to those derived for u. Hence, we have to prove that:

$$\frac{\lambda(n_{c_1})\lambda(n_{c_2}-\delta_{c_2=c_1^-})}{\lambda(n_{c_2^-}+\delta_{c_2^-=c_1}-1)\lambda(n_{c_1^-}-1)} = \frac{\lambda(n_{c_2})\lambda(n_{c_1}-\delta_{c_1=c_2^-})}{\lambda(n_{c_1^-}+\delta_{c_1^-=c_2}-1)\lambda(n_{c_2^-}-1)} .$$

If $c_2^- \neq c_1$ and $c_1^- \neq c_2$, the proof is trivial. If we assume $c_2^- = c_1$ (and hence $c_1^- \neq c_2$ since $K > 2$) we obtain:

$$\frac{\lambda(n_{c_1})\lambda(n_{c_2})}{\lambda(n_{c_2^-})\lambda(n_{c_1^-}-1)} = \frac{\lambda(n_{c_2})\lambda(n_{c_1}-1)}{\lambda(n_{c_1^-}-1)\lambda(n_{c_2^-}-1)}$$

which is an identity since $\lambda(n_{c_1})/\lambda(n_{c_2^-}) = \lambda(n_{c_1}-1)/\lambda(n_{c_2^-}-1) = 1$. The case $c_1^- = c_2$ is symmetric.

Now consider the case $T > 2$ and let c, d be the last servers fetching a job in u and v, respectively. We can decompose u and v as follows:

$$u : \mathbf{n} \overset{u_1}{\Longrightarrow} \mathbf{n}_1 \xrightarrow{c} \mathbf{n}_F , \qquad \text{and} \qquad v : \mathbf{n} \overset{v_1}{\Longrightarrow} \mathbf{n}_2 \xrightarrow{d} \mathbf{n}_F ,$$

where $\mathbf{n}_F$ denotes the final state of the paths u and v. By inductive hypothesis, if we reorder the arrivals in u and v so that the last servers are d and c (resp., c and d), we obtain

$$u : \mathbf{n} \overset{u_2}{\Longrightarrow} \mathbf{n}_3 \xrightarrow{d} \mathbf{n}_1 \xrightarrow{c} \mathbf{n}_F$$

and

$$v : \mathbf{n} \overset{v_2}{\Longrightarrow} \mathbf{n}_4 \xrightarrow{c} \mathbf{n}_2 \xrightarrow{d} \mathbf{n}_F .$$

Clearly, $\mathbf{n}_3 = \mathbf{n}_4$ because they see the same servers fetching jobs. Hence, we can apply the inductive hypothesis to the paths u_2 and v_2 since they start from and arrive at the same states obtaining $\psi(u_2)/\psi(u_2^R) = \psi(v_2)/\psi(v_2^R)$ and to the remaining two transitions labelled (c, d) and (d, c):

$$\frac{\psi(u)}{\psi(u^R)} = \frac{\psi(u_2)\psi((d,c))}{\psi(u_2^R)\psi((d,c)^R)} = \frac{\psi(v_2)\psi((c,d))}{\psi(v_2^R)\psi((c,d)^R)} = \frac{\psi(v)}{\psi(v^R)}$$

where with an abuse of notation we write $\psi(c, d)$ ($\psi(d, c)$) to denote the product of the rates in the last two transitions of the paths. □

We are now in position to prove Theorem 1.

Proof of Theorem 1. In order to prove that $X_K(t)$ is ρ-reversible with respect to the renaming given by Equation (3) we have to prove that the conditions (K1) and (K2) of Lemma 1 are satisfied.

Condition (K1) is easy to verify. Indeed the residence time in $\mathbf{n} \in \mathcal{S}$ is exponentially distributed with rate $\sum_{k=1}^{K} \lambda(n_k)$. Since $\rho(\mathbf{n})$ has the same components of $\mathbf{n}$ but with a different order, then the condition is trivially satisfied. Let us now verify Condition (K2). First, we notice that every cycle consists of the arrivals of the same amount V of jobs for each server, therefore a cycle has length VK. Assume $V = 1$ (simple cycle) and let us compute $\psi(u)/\psi(u^R)$ where u is a cycle in which we have the arrival of jobs at servers $(K, K-1, \dots 1)$ from an arbitrary state $\mathbf{n}$. Then we have:

$$\psi(u) = \lambda(n_K) \prod_{k=1}^{K-1} \lambda(n_k - 1)\,.$$

Let us consider u^R. By Proposition 1, the servers fetching a job are $(1, K, K-1, K-2, \dots, 2)$, therefore we obtain:

$$\psi(u^R) = \lambda(n_1^R) \prod_{k=2}^{K} \lambda(n_k^R - 1) = \lambda(n_K) \prod_{k=1}^{K-1} \lambda(n_k - 1) = \psi(u)\,.$$

By Lemma 3, $\psi(u) = \psi(u^R)$ for any simple cycle. Assume now, $V > 1$. In this case, again by Lemma 3 we can rearrange the order of servers fetching a job so that we decompose the cycle u into V simple cycles without changing the ratio $\psi(u)/\psi(u^R) = 1$, hence proving the theorem. □

As a consequence of Theorem 1 we can easily derive the expressions of the invariant measures of $X_K(t)$. The irreducibility of $X_K(t)$ implies the fact that all the invariant measures differ by a positive constant and that if all the states are positive recurrent then we can derive the stationary distribution by normalisation.

Corollary 1. *$X_K(t)$ has a product-form invariant measure given by:*

$$\pi(\mathbf{n}) = \frac{1}{G_K} \prod_{i=1}^{K} \frac{\prod_{n=0}^{n_i-1} \lambda(n)}{\prod_{n=n_i}^{-1} \lambda(n)} \tag{6}$$

which can be normalised on G_K to give the stationary distribution whenever $X_K(t)$ is ergodic.

Proof. Since $X_K(t)$ is ρ-reversible, we derive the expression of the invariant measure associated with state $\mathbf{n}$ with respect to a reference state $\mathbf{0}$ as given by Lemma 2. Let u be an arbitrary path from state $\mathbf{0}$ to state $\mathbf{n}$, and let u^R its reversed according to Proposition 1. Then we have:

$$\frac{\pi(\mathbf{n})}{\pi(\mathbf{0})} = \frac{\psi(u^R)}{\psi(u)}\,.$$

Consider an arbitrary state $\mathbf{n}$ and let T be the minimum number of arrivals that takes the model from state $\mathbf{n}$ to state $\mathbf{0}$. Notice that T is well-defined. In fact, since $X_K(t)$ represents the "profile" of $N(t)$ at time epoch t, T is the sum of all the arrivals that take each $N_i(t)$ to the same value of the maximum (see Figure 2). We proceed by induction on T. If $T = 1$ then $\mathbf{n} = \mathbf{0} - \mathbf{e}_c + \mathbf{e}_{c^-}$ for

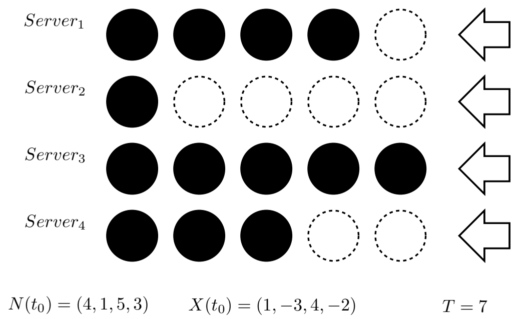

Figure 2: The example shows the minimum number of arrivals in case of $K = 4$ servers from state $(1, -3, 4, -2)$ to state $(0, 0, 0, 0)$.

some $1 \leq c \leq K$, and hence we have:

$$u : \mathbf{n} \xrightarrow{\lambda(-1)} \mathbf{0} \qquad u^R : \mathbf{0} \xrightarrow{\lambda(0)} \mathbf{n}^R \tag{7}$$

which verifies Equation (6). If $T > 1$, then by Lemma 3 we can allow any arrival to get one step closer to the reference state $\mathbf{0}$. We choose c such that $n_c < 0$ and $n_{c^-} \geq 0$. In this case, we have:

$$u \quad : \mathbf{n} \xrightarrow{\lambda(n_c)} \mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-} \tag{8}$$

$$u^R \quad : (\mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-})^R \xrightarrow{\lambda(n_{c^-}-1)} \mathbf{n}^R \,. \tag{9}$$

Hence, by inductive hypothesis, we have:

$$\pi(\mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-}) = \frac{1}{G_K} \prod_{i=1}^{K} \left(\frac{\prod_{n=0}^{n_i-1} \lambda(n)}{\prod_{n=n_i}^{-1} \lambda(n)} \right) \frac{\lambda(n_c)}{\lambda(n_{c^-} - 1)} \,. \tag{10}$$

By Lemma 2 we have that:

$$\frac{\pi(\mathbf{n})}{\pi(\mathbf{0})} = \frac{\lambda(n_{c^-} - 1)}{\lambda(n_c)} \frac{\pi(\mathbf{n} + \mathbf{e}_c - \mathbf{e}_{c^-})}{\pi(\mathbf{0})}$$

from which, by simplifying $\pi(\mathbf{0})$ on both sides and by using Equation (10), we finally obtain Equation (6). $\square$

3.3. Performance indices.

We are interested in the evaluation of two stationary performance indices. The first is the system throughput, i.e, the expected number of jobs that are fetched per unit of time. The second index measures how well-balanced is the model. For a state $\mathbf{n}$, we sum the positive components of $\mathbf{n}$ to measure the balancing of the model. Therefore, higher values for the expectation of this index denote a badly balanced system. Assume that $X_K(t)$ is stationary, the system's throughput T_K is defined as:

$$T_K = \sum_{\mathbf{n}\in\mathcal{S}} \left(\pi(\mathbf{n}) \sum_{j=1}^{K} \lambda(n_j) \right) \tag{11}$$

while the system's balance index B_K is:

$$B_K = \sum_{\mathbf{n}\in\mathcal{S}} \left(\pi(\mathbf{n}) \sum_{j=1}^{K} n_j \delta_{n_j>0} \right) . \tag{12}$$

Ideally, one would like to design a system in which T_K is large and B_K is small by choosing an appropriate rate transition function $\lambda(n_k)$.

4. The bimodal model

In this section we study the model whose underlying CTMC is $X_K(t)$ and the transition rates are defined as:

$$\lambda(n_k) = \begin{cases} \eta & \text{if } n_k \geq 0 \\ \mu & \text{if } n_k < 0 \end{cases} \tag{13}$$

where $k \in \{1, \dots, K\}$. Intuitively, each server k can work at two different rates corresponding to the cases in which it has served less or more jobs than the server k^+. We prove that for all finite K a necessary and sufficient condition for the ergodicity of $X_K(t)$ is that $\eta < \mu$, i.e., any server k has to work faster if it has processed less jobs than k^+.

4.1. Ergodicity and stationary analysis

The main result for the bimodal model is given by Theorem 2. The proof is in the Appendix.

Theorem 2. *Let $X_K(t)$ be the CTMC defined according to Equations (1)-(2) with $\lambda(n_k)$ defined in Equation (13). Then, the following properties hold:*

- *$X_K(t)$ is ergodic for all finite $K \geq 2$ if and only if $x = \eta/\mu < 1$;*
- *If $x < 1$, the normalising constant of Equation (6) that gives the unique stationary distribution is:*

$$G_K^b(x) = 1 + \sum_{j=1}^{K-1} \binom{K}{j} \binom{K-1}{j-1} (K-j)\beta(x, K-j, 1-K) , \tag{14}$$

where β denotes the incomplete Euler's Beta-function:

$$\beta(z, a, b) = \int_0^z u^{a-1}(1-u)^{b-1} du .$$

The evaluation of the incomplete β function is efficient and for small values of K it can be performed symbolically. The following corollary gives the expression for the normalising constant in terms of finite sums. The proof is given in appendix.

Corollary 2. *Given a finite $K \in \mathbb{N}$, $K \geq 2$, the expression of $G_K^b(x)$ is a rational function in x which can be computed as:*

$$G_K^b(x) = 1 + \sum_{j=1}^{K-1} \binom{K}{j} \cdot \binom{K-1}{j-1} (K-j) \left(\frac{x}{1-x}\right)^{k-j} \cdot \sum_{v=0}^{j-1} (-1)^v \binom{j-1}{v} \frac{1}{K-j+v} \left(\frac{x}{x-1}\right)^v$$

The throughput of the model in steady-state is defined as follows.

Lemma 4. *Let $X_K(t)$ be the CTMC defined according to Equations (1)-(2) with $\lambda(n_k)$ defined in Equation (13) and $\eta < \mu$. Then, the throughput of the model in steady-state is:*

$$T_K^b(\eta, \mu) = \frac{1}{G_K^b(x)} \Big(K\eta + \sum_{j=1}^{K-1} (j\eta + (K-j)\mu) \binom{K}{j} \cdot \binom{K-1}{j-1} (K-j) \beta(x, K-j, 1-K) \Big). \quad (15)$$

Proof. Note that for a state $\mathbf{n}$ the throughput depends only on the number of non-negative components j. Therefore, the proof follows the same lines of that proposed for Theorem 2. $\square$

The evaluation of $T_K^b(\eta, \mu)$ requires only the computation of finite sums as stated by the following corollary whose proof is analogue of that of Corollary 2.

Corollary 3. *Given a finite $K \in \mathbb{N}$, $K \geq 2$, the expression of $T_K^b(\eta, \mu)$ is a rational function in $x = \eta/\mu$ which can be computed as:*

$$T_K^b(\eta, \mu) = \frac{1}{G_K^b(x)} \Bigg(K\eta + \sum_{j=1}^{K-1} (j\eta + (K-j)\mu) \binom{K}{j} \binom{K-1}{j-1} (K-j) \cdot \left(\frac{x}{1-x}\right)^{k-j} \cdot \sum_{v=0}^{j-1} (-1)^v \binom{j-1}{v} \frac{1}{K-j+v} \left(\frac{x}{x-1}\right)^v \Bigg).$$

Finally, we give the expression for the balance index. The proof is given in the Appendix.

Lemma 5. *Let $X_K(t)$ be the CTMC defined according to Equations (1)-(2) with $\lambda(n_k)$ defined in Equation (13) and $\eta < \mu$. The balance index of the model in steady-state is:*

$$B^b_K(x) = \frac{1}{G^b_K(x)} \left(\frac{x}{1-x}\right)^K \cdot \sum_{j=1}^{K-1} \binom{K}{j}\binom{K-1}{j-1}(K-j)\frac{1}{x^j}. \tag{16}$$

4.2. Numerical evaluation.

We study the bimodal model for different values of K and by varying $x = \eta/\mu$. We are interested in studying the throughput and the balance index normalised by the number of servers. Figures 3 and 4 show the normalised throughput and balance index, respectively, for $K = 2, 4, 6, 10, 20$. Notice that the balance index of the model grows quickly for values of x greater than 0.7 as it clearly appears from the plot in Figure 5.

Example 1. *Let us consider a system with $K = 5$ servers and without loss of generality let us assume $\mu = 1$. We aim at finding η^* which satisfies:*

$$\eta^* = argmax_\eta (T^b_5(\eta, 1)^2 / B^b_5(\eta)) \, .$$

Informally, η^* represents the optimal operating point when the requirement for high throughput is considered more important than the balance index. Notice that since both $T^b_5(\eta, 1)$ and $B^b_5(\eta)$ are rational functions we can compute $d(\eta) = \partial(T^b_5(\eta, 1)^2/B^b_5(\eta))/\partial\eta$ analytically. Now, finding the zero of $d(\eta)$ requires to find the roots of a polynomial in η which can be numerically performed by several computationally efficient approaches. In our case the plot of $T^b_5(\eta, 1)^2/B^b_5(\eta)$ is shown in Figure 6 and we have $\eta^* \simeq 0.3513$ and

$$\lambda(n_k) = \begin{cases} \eta^* & \text{if } n_k \geq 0 \\ 1 & \text{if } n_k < 0 \end{cases}$$

with $T^b_5(\eta^*, 1) \simeq 2.903$ and $B^b_5(\eta^*) = 3.5798$.

5. The proportional model

We consider the case in which each server $k \in \{1, \dots, K\}$ can fetch a job from the queue with a maximum rate ζ but it may decide to slow down its service rate according to its internal state, i.e, the value of n_k. We study the case in which function $\lambda(n_k)$ is defined as:

$$\lambda(n_k) = \begin{cases} \frac{\zeta}{(n_k+1)} & \text{if } n_k \geq 0 \\ \zeta & \text{if } n_k < 0 \end{cases} . \tag{17}$$

The Markov chain underlying this model has been previously considered in [21], hence we omit the proofs and just state the main results for the sake of comparing the performance indices of the bimodal and the proportional models in the following section.

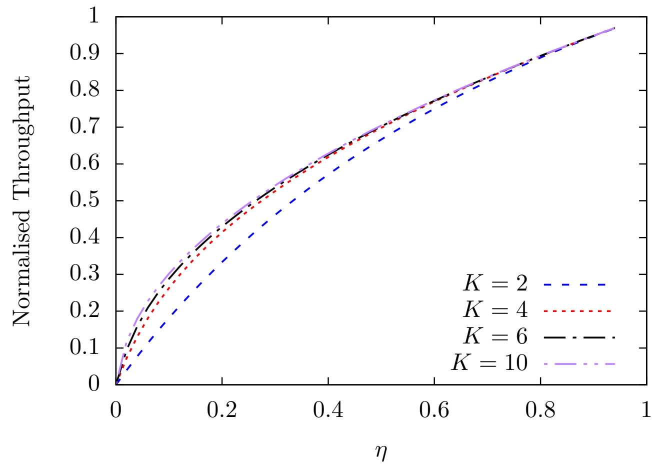

Figure 3: Throughput for each server in the bimodal model as a function of η assuming $\mu = 1$. The line $K = 20$ is graphically overlapped to the one for $K = 10$ and is omitted.

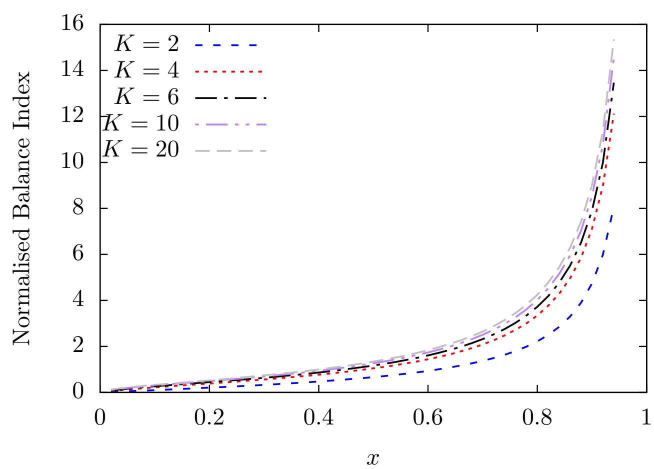

Figure 4: Balance index for each server in the bimodal model as a function of $x = \eta/\mu$.

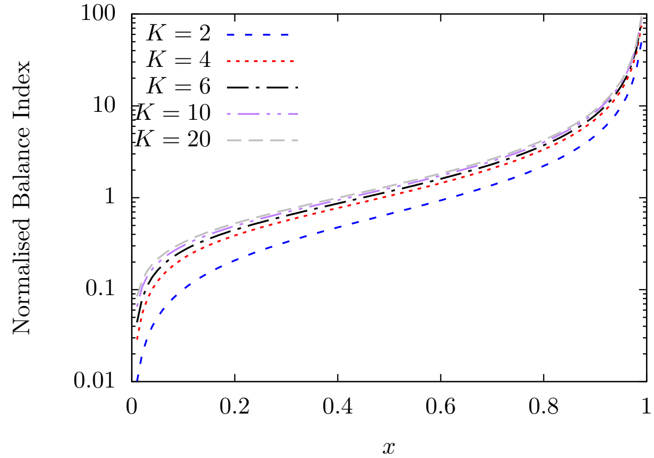

Figure 5: Balance index for each server in the bimodal model as a function of $x = \eta/\mu$ in log-scale.

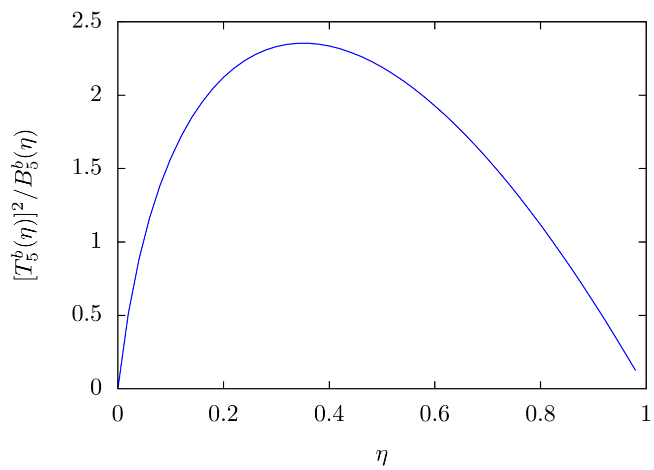

Figure 6: Objective function of the optimisation problem of Example 1.

5.1. Ergodicity and stationary analysis.

In this case we can express both the normalising constant and the throughput in terms of the regularized Kummer's confluent hypergeometric function $\mathbf{M}(a, b, z)$ defined as follows (the first equality shows an alternative common notation):

$$\mathbf{M}(a, b, z) = {}_1\tilde{F}_1(a; b; z) = \frac{1}{\Gamma(b)} M(a, b, z) \qquad b \in \mathbb{N}^+ \tag{18}$$

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

$$M(a, b, z) = {}_1F_1(a; b; z) = \sum_{k=0}^{\infty} \frac{(a)_k}{(b)_k} \frac{x^k}{k!} \qquad b \in \mathbb{N}^+. \tag{19}$$

In [21] we proved that, when the transition rates are defined according to Definition (17), the performance indices are expressed as rational functions of K as stated below.

Theorem 3. *Let $X_K(t)$ be the CTMC defined according to Equations (1)-(2) with $\lambda(n_k)$ defined in Equation (17). Then, the following properties hold:*

1. *$X_K(t)$ is ergodic and hence admits a unique stationary distribution.*
2. *The normalising constant G_K^p of Equation (6) that gives the unique stationary distribution is:*

$$G_K^p = 1 + \sum_{j=1}^{K-1} \binom{K}{j} j^{K-j} \mathbf{M}(K-j, K-j+1, j)\,. \tag{20}$$

3. *The throughput T_K^p of the model in steady-state is:*

$$\begin{aligned} T_K^p = \frac{\zeta}{G_K^p} \Bigg(K + \sum_{j=1}^{K-1} \binom{K}{j} j \Big(& j^{K-j+1}\ \mathbf{M}(K-j, K-j+2, j) \\ & - (j-1)^{K-j+1}\ \mathbf{M}(K-j, K-j+2, j-1) \\ & + (K-j) j^{K-j-1} \mathbf{M}(K-j, K-j+1, j) \Big) \Bigg). \end{aligned} \tag{21}$$

4. *The balance index of the model in steady-state is:*

$$B_K^p = \frac{1}{G_K^p} \sum_{j=1}^{K-1} \binom{K}{j} \frac{j^{K-j}}{\Gamma(K-j)}\ e^j \tag{22}$$

The numerical evaluations of G_K^b and T_K^p are based 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

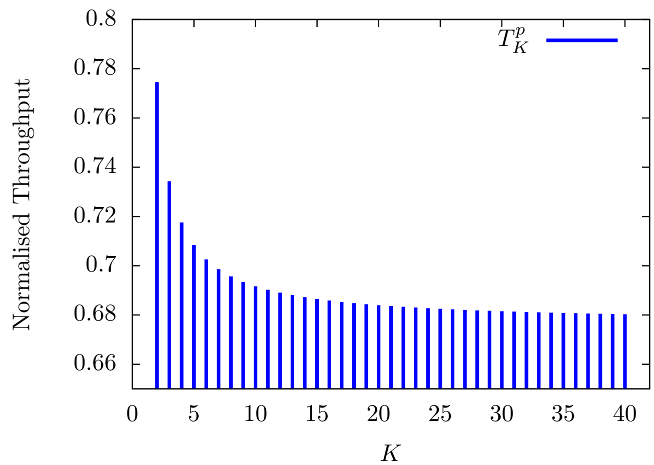

Figure 7: Throughput for each server in the proportional model as a function of K.

Table 1: Comparison between the two models.

K	T_K^*/K	x	$B_K^b(x)/K$	B_K^p/K
2	0.774	0.632	1.052	0.613
4	0.718	0.526	1.135	0.822
6	0.702	0.500	1.170	0.883
8	0.696	0.489	1.191	0.913
10	0.692	0.482	1.201	0.930
12	0.689	0.478	1.210	0.943
14	0.687	0.475	1.216	0.951
16	0.685	0.472	1.217	0.957
18	0.685	0.471	1.222	0.962
20	0.684	0.470	1.226	0.966

converges for all finite x to values that are computable by means of finite sums (see [24]). In particular, for $b > a$, $\mathbf{M}(a,b,z)$ converges to [24]:

$$\mathbf{M}(a,b,z) = \left(e^z \sum_{k=0}^{a-1} \frac{(1-a)_k \ (-z)^k}{k! \ (2-b)_k} \right. \\ \left. - \sum_{k=0}^{b-a-1} \frac{(1-b+a)_k \ z^k}{k! \ (2-b)_k} \right) \frac{(2-b)_{a-1} \ z^{1-b}}{(a-1)!} . \quad (23)$$

5.2. Numerical evaluation

In Figure 7 and 8 we show the throughput and the balance index per server in the proportional model. We observe that for $20 < K < 40$ the normalised

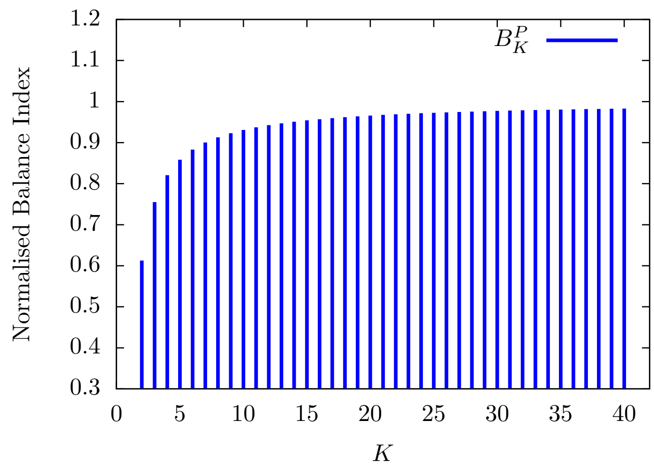

Figure 8: Balance index for each server in the proportional model as a function of K.

Table 2: Service time distribution used for the sensitivity analysis. The last column shows the coefficient of variation.

Scenario	Distribution	Parameters	CV
S1	Lognormal(μ, σ^2)	$\mu = -0.1$, $\sigma^2 = 0.2$	0.471
S2	Erlang$(2, \lambda)$	$\lambda = 2$	0.705
S3	Exponential(μ)	$\mu = 1$	1
S4	HyperExponential$(\mathbf{p}, \boldsymbol{\lambda})$	$\mathbf{p} = (0.2, 0.8)$, $\boldsymbol{\lambda} = (1.5, 0.875)$	1.22
S5	Lognormal(μ, σ^2)	$\mu = -2$, $\sigma^2 = 4$	7.321

throughput tends to assume a value of approximately 0.68, while the normalised balance index is kept below 1. In Table 1 we compare the two rate adaptation strategies, for the bimodal one we choose the ratio $x = \eta/\mu$ which reproduces the same throughput of the proportional. We assume $\mu = \zeta = 1$. Observe that although the proportional model outperforms the bimodal, its throughput is around 70% of the maximum achievable and is not adjustable as is the case for the bimodal.

6. Sensitivity analysis

In this section we compare the performance measures obtained from the analytical models based on the Flatto-Hahn-Wright assumptions (i.e., the arrivals follow a Poisson distribution and the service times are i.i.d. exponential random variable) with the simulation outcomes obtained by relaxing some of the hypothesis. Specifically, we are interested in studying the impact of the service time distribution on the performance indices. All the simulations consist of 15 independent runs and the warm up period is removed according to the Welch's method. We have built confidence intervals of 98% which are shown in the plots only when their sizes are compatible with the scale of the graph. We simulate the bimodal model with the service time distributions shown in Table 2 and $\eta = 0.8$. Notice that the expected service time is 1 for all the scenarios. The plots showing the normalised throughput and the balance index as functions of the number of servers are shown in Figure 9 and 10. As expected, the distribution of the service time affects the performance indices since these depend on its variance. Nevertheless, we see that the analytical bimodal model is a lower bound for the throughput and the balance index when the service time CV is greater than 1 whereas it is an upper bound in the opposite case. It is worth of notice that when the coefficient of variation is much larger than 1 (S5), the bimodal rate control algorithm maintains a finite balance index but very high, i.e., the system is not well-balanced.

In Figures 11 and 12 we show the simulation outcomes for the same scenarios of Table 2 for the proportional rate control algorithm. Also for these simulations the performance measures depend on the service time distribution but in this case the analytical proportional model is an upper bound for the throughput when the coefficient of variation is greater than 1 and a lower bound otherwise. Moreover, the balance index for the scenario S5 is much lower than that observed for the bimodal rate control algorithm.

In conclusion, we can say that both the performance measures of the rate adaptation algorithms are sensitive to the service time distributions but the analytical models studied in Section 4 and 5 can be used for giving upper/lower bounds. Moreover, the simulations have shown that the proportional rate control algorithm is more effective in maintaining a low balance index for large coefficient of variations at the cost of a high reduction in the system's throughput.

7. Conclusion

In this paper we have proposed an algorithm for balancing the total number of jobs performed by each of a set of K identical servers. The servers use a small amount of information to adapt their service rates in order to maintain the difference between the total number of served jobs small. We have defined a ρ-reversible CTMC to study the stationary distribution and the performance indices for two rate-adaptation policies, named bimodal and proportional. For both these strategies we have derived finite expressions for the performance indices which in the case of the bimodal model are, for a fixed number of servers, rational functions of the system's parameters. As a consequence, the problem of optimising the system can be tackled efficiently. In comparing the performance of the two rate-adaptation strategies, we make some important observations. For given throughput and number of servers, the proportional strategy gives always a lower balance index, i.e., the system is better balanced than in the bimodal strategy. However, we must consider that the bimodal strategy gives

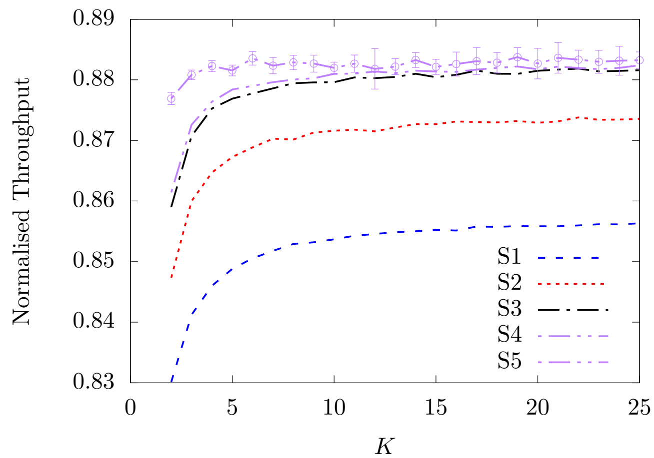

Figure 9: Normalised throughput of the bimodal model.

the possibility to control the model behaviour through an additional parameter $x = \eta/\mu$ which allows us to control the throughput and the balance index. In practice, when high throughput is required, we should adopt the bimodal strategy with $0.7 < x < 1$ and bear with a badly balanced system. Instead, when the system's balance is pivotal, the proportional strategy achieves high levels of system balancing while maintaining a higher throughput than that of the bimodal approach. We finally remark that, according to the numerical evidences of the proposed experiments (see Figures 3, 4, 7 and 8), the throughput and the balance index normalised by the number of servers K worsen slowly with the growth of K and hence both the strategies can be used to tackle systems with large number of servers. Future works include the application of the approach proposed here to balance the energy consumptions in wireless sensor networks (see e.g., [25, 26, 27]) with the aim of augmenting the network life expectation.

References

[1] A. Marin, S. Rossi, On the relations between Markov chain lumpability and reversibility, Acta Inf. (2016) Available online.

[2] F. Kelly, Reversibility and stochastic networks, Wiley, New York, 1979.

[3] F. Baskett, K. M. Chandy, R. R. Muntz, F. G. Palacios, Open, closed, and mixed networks of queues with different classes of customers, J. of the ACM 22 (2) (1975) 248–260.

[4] E. Barbierato, G. Dei Rossi, M. Gribaudo, M. Iacono, A. Marin, Exploiting product forms solution techniques in multiformalism modeling, Electr. Notes Theor. Comput. Sci. 296 (2013) 61–77.

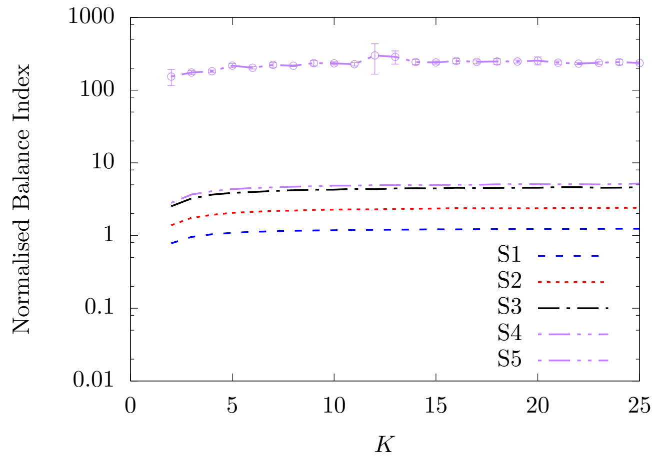

Figure 10: Normalised balance index of the bimodal model.

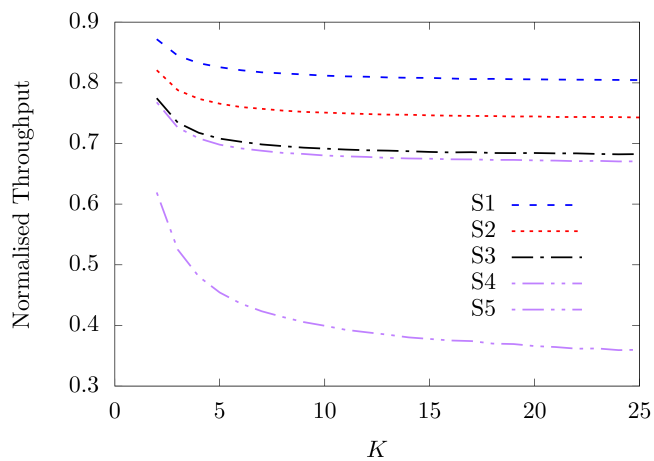

Figure 11: Normalised throughput of the proportional model.

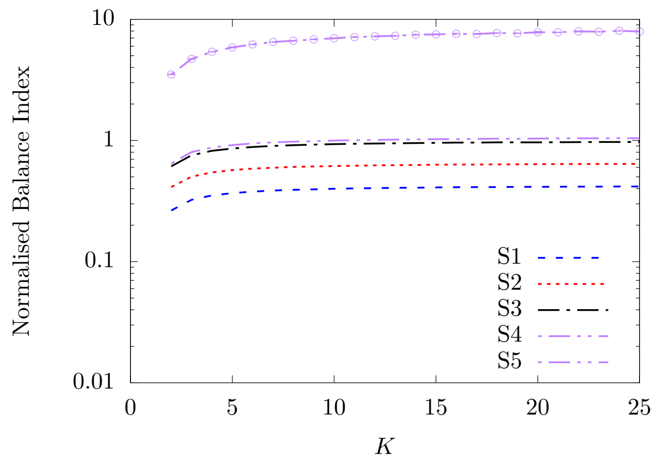

Figure 12: Normalised balance index of the proportional model.

[5] J. Hillston, A. Marin, C. Piazza, S. Rossi, Contextual lumpability, in: Proc. of Valuetools 2013 Conf., ACM Press, 2013.

[6] M. Harcol-Balter, Task assignment with unknown duration, J. of the ACM 49 (2) (2002) 260–288.

[7] E. Hyytiä, J. Virtamo, S. Aalto, A. Penttinen, M/M/1-PS queue and size-aware task assignment, Perf. Eval. 68 (11) (2011) 1136–1148, special Issue: Performance 2011.

[8] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. Larus, A. Greenberg, Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services, Perf. Eval. 68 (11) (2011) 1056–1071, special Issue: Performance 2011.

[9] S. Doroudi, E. Hyytia, M. Harchol-Balter, Value driven load balancing, Perf. Eval. 79 (2014) 306–327.

[10] F. Alomari, D. A. Menascé, Efficient response time approximations for multiclass fork and join queues in open and closed queuing networks, IEEE Trans. on Parallel and Distributed Systems 25 (6) (2014) 1437–1446.

[11] R. Chen, An upper bound solution for homogeneous fork/join queueing systems, IEEE Trans. on Parallel and Distributed Systems 22 (5) (2011) 874–878.

[12] R. J. Chen, A hybrid solution of fork/join synchronization in parallel queues, IEEE Trans. on Parallel and Distributed Systems 12 (8) (2001) 829–845.

[13] J. C. S. Lui, R. R. Muntz, D. Towsley, Computing performance bounds of fork-join parallel programs under a multiprocessing environment, IEEE Trans. on Parallel and Distributed Systems 9 (3) (1998) 295–311.

[14] T. Rauber, G. Rünger, Energy-aware execution of fork-join-based task parallelism, in: Proc. of the IEEE 20th Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'12), 2012, pp. 231–240.

[15] S. Balsamo, P. G. Harrison, A. Marin, Methodological Construction of Product-form Stochastic Petri-Nets for Performance Evaluation, J. of System and Software 85 (7) (2012) 1520–1539.

[16] A. Marin, S. Balsamo, P. Harrison, Analysis of stochastic Petri nets with signals, Perf. Eval. 85 (7) (2012) 1520–1539.

[17] R. Osman, P. G. Harrison, Approximating closed fork-join queueing networks using product-form stochastic Petri-nets, J. of System and Software 110 (2015) 264–278.

[18] J. Dean, S. Ghemawat, Mapreduce: Simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[19] H. Chen, J. Yin, C. Pu, Performance analysis of parallel processing systems with horizontal decomposition, in: Proc. of IEEE Cluster Cumputing, 2012, pp. 220–229.

[20] M. Bramson, Y. Lu, B. Prabhakar, Randomized load balancing with general service time distributions, SIGMETRICS Perform. Eval. Rev. 38 (1) (2010) 275–286, special Issue: SIGMETRICS 2010.

[21] A. Marin, S. Rossi, Dynamic control of the join-queue lengths in saturated fork-join stations, in: Proc. of Int. Conf. on Quantitative Evaluation of Systems, (QEST'16), 2016, pp. 123–138.

[22] P. Whittle, Systems in Stochastic Equilibrium, John Wiley & Sons, Inc., New York, NY, USA, 1986.

[23] A. Marin, 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), 2014, pp. 427–432.

[24] F. W. Olver, D. W. Lozier, R. F. Boisvert, C. W. Clark, NIST Handbook of Mathematical Functions, 1st Edition, Cambridge University Press, New York, NY, USA, 2010.

[25] L. Gallina, S. Hamadou, A. Marin, S. Rossi, A probabilistic energy-aware model for mobile ad-hoc networks, in: Proc. of Analytical and Stochastic Modeling Techniques and Applications (ASMTA), 2011, pp. 316–330.

[26] M. Bugliesi, L. Gallina, A. Marin, S. Rossi, S. Hamadou, Interference-sensitive preorders for manets, in: Proc. of Int. Conf. on Quantitative Evaluation of Systems, QEST, 2012, pp. 189–198.

[27] M. Bugliesi, L. Gallina, S. Hamadou, A. Marin, S. Rossi, Behavioural equivalences and interference metrics for mobile ad-hoc networks, Perform. Eval. 73 (2014) 41–72.

Appendix

Proof of Theorem 2. Let us recall the Taylor's expansion of the incomplete Beta-function:

$$\beta(x,a,b) = x^a \sum_{n=0}^{\infty} \frac{(1-b)_n}{n!(a+n)} x^n \,,$$

where $(y)_n$ is the Pochhammer's symbol:

$$(y)_n = y(y+1)\cdots(y+n-1)\,.$$

By replacing the definition of $\lambda(n_k)$ of Equation (13) in Equation (6), we have:

$$G_K^b(x) = \sum_{n\in\mathcal{S}} x^{\sum_{i=1}^{K} n_i \delta_{n_i>0}} \qquad \text{for } x = \frac{\eta}{\mu}\,.$$

This expression can be rewritten as:

$$G_K^b(x) = 1 + \sum_{j=1}^{K-1} \sum_{n=K-j}^{\infty} \binom{n+j-1}{j-1} \binom{n-1}{K-j-1} \binom{K}{j} x^n \,.$$

We can interpret the equation as follows:

- j is the number of non-negative components in a state. j ranges from 1 to $K-1$ since there must be at least one non negative component. The case $j=K$ identifies only state $\mathbf{0}$ which is considered by adding the unity before the sums.
- n is the sum of the positive components of the state. Since there are $K-j$ negative components, the starting value for n is $K-j$.
- The first binomial coefficient counts the number of non-negative solutions of the equation: $y_1 + \ldots + y_j = n$, i.e., the number of possible values that can be assumed by the non-negative state components conditioned on the fact that their sum is n.
- The second binomial coefficient counts the number of strictly positive solutions of the equation $y_1 + \ldots + y_{K-j} = n$ since all the positive recurrent states of $X_K(t)$ have components that sum to 0.
- The third binomial coefficient counts the way of assigning the j non-negative components to the K state components.

Henceforth, the proof is purely algebraic:

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

from which we derive the desired Taylor's expansion of the incomplete Beta-function, i.e., we obtain Equation (14). Notice that since $1-K<0$ the integral defining the incomplete Beta-function is finite if and only if $x<1$, i.e., $\eta<\mu$. $\square$

Proof of Corollary 2. We can express the incomplete-Beta function in terms of the hyper-geometric function ${}_2F_1$ as follows:

$$\beta(x, K-j, 1-K) = \frac{x^{K-j}}{K-j}\ {}_2F_1(K-j, K, K-j+1, x)\,,$$

and then by using the transformation rule [24]:

$${}_2F_1(a, b, b-n, z) = (1-z)^{-a}\ {}_2F_1\left(-n, a, b-n, \frac{z}{z-1}\right)\,,$$

for $n \in \mathbb{N}$, we obtain:

$$\begin{aligned}\beta(x, K-j, 1-K) = \left(\frac{x}{1-x}\right)^{K-j} & \frac{1}{K-j} \\ & {}_2F_1\left(-j+1, K-j, K-j+1, \frac{x}{x-1}\right)\,.\end{aligned}$$

Since $-j+1 \leq 0$ the hyper-geometric function can be reduced to a polynomial according to the following transformation rule [24] for the function ${}_2F_1$

$${}_2F_1(-m, b, c, z) = \sum_{n=0}^{m} (-1)^m \binom{m}{n} \frac{(b)_n}{(c)_n} z^n\,, \quad m \in \mathbb{N}$$

and the corollary follows after few algebraic steps. $\square$

Proof of Lemma 5. Let j and n be the number of non-negative components in a state and the sum of the positive components, respectively. Then we can write:

$$B_K^b(x) = \frac{1}{G_K^b(x)}\left(\sum_{j=1}^{K-1}\sum_{n=K-j}^{\infty} n x^n \binom{K}{j}\binom{n+j-1}{j-1}\binom{n-1}{K-j-1}\right)\,,$$

where the interpretation of the formula follows the line of that given in the proof of Theorem 2. The right-hand side can be rewritten as:

$$
\begin{aligned}
& \frac{1}{G_K^b(x)} \sum_{j=1}^{K-1} \binom{K}{j} \sum_{n=K-j}^{\infty} n x^n \frac{(n+j-1)!}{(j-1)!n!} \frac{(n-1)!}{(K-j-1)!} \frac{1}{(n-K+j)!} \\
= \; & \frac{1}{G_K^b(x)} \sum_{j=1}^{K-1} \binom{K}{j} \sum_{w=0}^{\infty} x^{w+K-j} \frac{(w+K-1)!}{(j-1)!} \frac{1}{(K-j-1)!w!} \\
= \; & \frac{1}{G_K^b(x)} \sum_{j=1}^{K-1} \binom{K}{j} x^{K-j} \frac{1}{(j-1)!} \frac{1}{(K-j-1)!} \sum_{w=0}^{\infty} \frac{x^w}{w!} (w+K-1)! \\
= \; & \frac{1}{G_K^b(x)} \sum_{j=1}^{K-1} \binom{K}{j} x^{K-j} \frac{1}{(j-1)!} \frac{(K-1)!}{(K-j-1)!} \sum_{w=0}^{\infty} x^w \binom{w+K-1}{w} \\
= \; & \frac{1}{G_K^b(x)} \sum_{j=1}^{K-1} \binom{K}{j} x^{K-j} \frac{1}{(j-1)!} \frac{(K-1)!}{(K-j-1)!} \frac{1}{(1-x)^K} \, .
\end{aligned}
$$

Now, Lemma 5 follows straightforwardly. □