BOUND PERFORMANCE MODELS
OF HETEROGENEOUS PARALLEL PROCESSING SYSTEMS

S.Balsamo*, L.Donatiello° and N. Van Dijk•

* Dipartimento di Matematica e Informatica, University of Udine, Udine, Italy
° Dipartimento di Scienze dell'Informazione , University of Bologna, Bologna, Italy
• Department of Econometrics, University of Amsterdam, Amsterdam, The Netherlands

Abstract - Systems of heterogeneous parallel processing are studied such as arising in parallel programs executed on distributed systems. A lower and an upper bound model are suggested to obtain secure lower and upper bounds on the performance of these systems. The bounding models are solved by using a matrix-geometric algorithmic approach.
Formal proofs of the bounds are provided along with error bounds on the accuracy of the bounds. These error bounds in turn are reduced to simple computational expressions. Numerical results are included. The results are of interests for application to arbitrary fork-join models with parallel heterogeneous processors and synchronization.

Index Terms - parallel systems, queueing networks, fork and join, system performance evaluation

1. INTRODUCTION

The design of parallel processing systems requires the development of performance models for the quantitative evaluation of such systems. Performance models can be used in the design of parallel processing systems such as the evaluation of scheduling and resource allocation policies, speedup and efficiency evaluation of parallel programs and applications. Queueing networks represent a natural way to model parallel processing systems; system structure can be modelled as a queueing system while programs or applications, consisting of tasks with some precedence constraints, can be modelled as precedence graphs whose nodes are the tasks and edges correspond to precedence constraints. In a precedence graph we identify fork nodes when more than one edge leaves the node, and join nodes when more than one edge enters the node. Fork and join nodes represent, respectively, the starting point of parallel execution of tasks and the synchronisation of tasks. A program is completely executed when all its tasks have been completed. Figure 1 shows a simple example of precedence graph consisting of a single fork node, N parallel tasks and a single join node. Parallel programs including fork and join or parbegin/parend constructs, and parallel operations of write requests in a distributed database system can be represented by such queueing models.
Models of parallel processing systems can be homogeneous or heterogeneous. The latter represent the more general class of parallel processing systems composed by different processing units and different parallel tasks. The performance indices of interest include job and task response time, synchronization delay, queue length distribution and throughput.

Cuncurrency and synchronisation make the solution of such performance models more complex with respect to the classical queueing network analysis [12]. Exact analysis has been carried out by Flatto and Hahn [7] who consider programs with a fork node, two parallel tasks and a join node, and a system with two heterogeneous processing units, each having its own queue. Each incoming job is split into two tasks which are allocated to the processing units. A task, after it has been processed, waits for its siblings in a join queue before leaving the system. Under exponential assumption for the interarrival and service time distribution they obtain the generating function of the system state probabilities. Some limit results on the conditioned queue length are shown by Flatto [8]. Brun and Fajolle [4] obtain the Laplace transform of the response time distribution for the same model and an approximate solution has been proposed by Rao and Posner [17]. When the system consists of N≥2 homogeneous exponential processing units and incoming jobs are formed by N parallel tasks, Nelson and Tantawi [14] present approximate solutions for the mean job and task response time. Bounds on the average job response time for a system with general service time and inter-arrival time distributions have been proposed by Varma and Makowski [18]. A comparison between different parallel processing models in terms of mean response time is presented by Nelson, Towsley and Tantawi in [15]. A more general model with N≥2 heterogeneous servers and general arrival service time distribution is considered by Baccelli and Makowski [1] who provide bounds for the job mean response time, while Kim and Agrawala [9] obtain the transient and steady-state solution of the virtual waiting time. More complex systems where processing units are connected in series and parallel have been analysed by Baccelli et al. [2] deriving bounds on response time, while Duda and Czachórski [5,6] present approximate solutions for performance indices. Bounds on response time for systems with parallel dependent task have been derived by Kumar and Shorey [13]. Heidelberger and Trivedi [10,11] propose different approximate solution methods for models both with and without synchronisation.

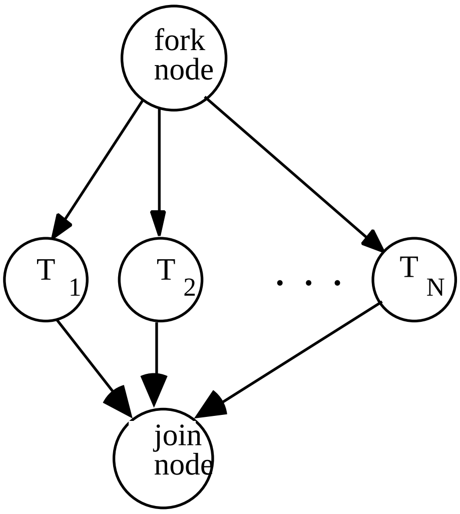

Fig. 1 - Example of precedence graph.

Unfortunately, the main drawback of approximate but not bound methods proposed is the lack of information on the introduced error, while bound methods do not allow an iterative process to improve the bound accuracy, i.e. to reduce the spread of bounds. The main contribution of this paper is the proposal of a method for the performance analysis of a class of fork and join queueing networks. The method has two main characteristics; first it provides an algorithm for the approximate solution of the steady-state probability distribution of the joint queue length for heterogeneous systems. The other feature of the proposed method is the following. Unlike other bound methods presented in literature [1,2,13,18], it allows us to provide bounds on the queue length distribution, beside other performance indices, and to control the spread of bounds to meet a given accuracy.

We consider a fork and join queueing system with N≥2 heterogeneous processing units and N parallel tasks. We present two models which provide respectively upper and lower bounds on performance indices and whose solution is obtained by applying an algorithm approach, both in terms of stationary state probability distribution (i.e., the number of tasks in each queue) and other performance indices such as job and task mean response time, synchronisation delay and speed up. The two proposed models are defined by considering appropriate state space partitions and reductions which enable us to apply a matrix-geometric approach [16]. We prove that the two models provide upper and lower bounds on a set of performance indices of the original fork and join model, respectively. Moreover we derive an expression of the bound width for the average performance indices. By comparing the results obtained by the proposed method with both the exact numerical solution and other approximate and bound solutions, we observe a good accuracy of the proposed bounds. Moreover we show the improvement of the approximation accuracy, i.e. the spread of bounds, by choosing the appropriate value of the modified model parameters.

The paper is organized as follows. In the next Section the model is introduced. Sections 3 and 4 present the upper and lower bound models, respectively defined by considering two different state space reductions. The algorithmic approach, its computational complexity and the bound computation are presented in Section 5. In Section 6 numerical examples are presented to compare the proposed bound solution with other solution methods. Finally, Section 7 summarises the results and future research.

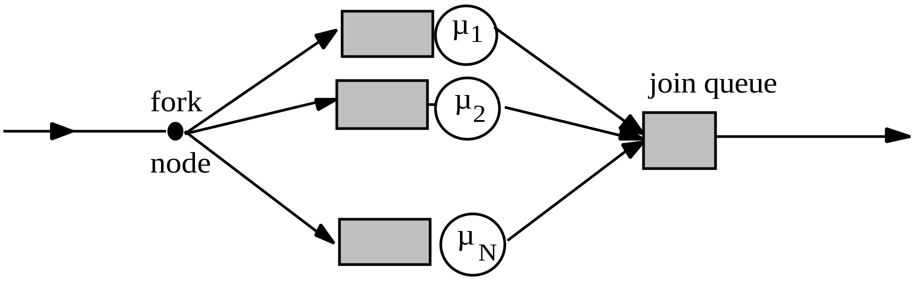

Fig. 2 - Fork and join N-server queueing system.

2. THE MODEL

Consider an open fork and join queueing system with $N \geq 2$ heterogeneous service centers, as shown in Figure 2. A service center consists of a single server and an infinite capacity queue with FCFS discipline. Arrival times of jobs at the systems are assumed to be statistically independent random variable having the same probability distribution A(t). Upon arrival a job splits into N tasks denoted by T_1, $T_2, \ldots, T_N$. Each server is dedicated to execute specific tasks, i.e., task T_i is always executed by service center i, $1 \leq i \leq N$; service times of task T_i are independent random variable with probability distribution $B_i(t)$. Tasks wait for their siblings in the join queue until the whole job is completed. We assume that probability distributions A(t) and $B_i(t)$, $1 \leq i \leq N$, have a Coxian or a phase-type representation [12]. Hereafter, for the sake of simplicity, we consider exponential distributions. However, the same approach can be used to analyse systems with more general inter-arrival and service time distributions. A more detailed study of the method for the case of more general distribution is out of the scope of the paper. Arrival rate is denoted by λ and service rate of center i is denoted by μ_i, $1 \leq i \leq N$. Without loss of generality let $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_N$. By assuming that the stability condition holds, i.e., $\lambda < \min_i \mu_i$, we analyze the system in steady-state condition.

System state is defined as $\mathbf{n}=(n_1,\ldots,n_N)$, where n_i denotes the number of tasks in service center i, $1 \leq i \leq N$. The number of tasks waiting in the join queue can be computed as

$\Sigma_{1\le i\le N}$ (n^*-n_i), where $n^*=\max_{1\le i\le N}$ n_i. It is easy to verify that n* also represents the number of jobs in the system. The system evolution can be modelled by an homogeneous discrete-space continuous-time Markov process with infinite state space

$$E =\{n=(n_1,\dots,n_N),\ n_i\ge 0,\ 1\le i\le N\}$$

and transition rate matrix $\mathbf{Q}=||q_{\mathbf{n},\mathbf{n}'}||$, $\mathbf{n},\mathbf{n}'$ E, defined as follows :

$$q_{\mathbf{n},\mathbf{n}'} = \lambda \qquad \text{if } \mathbf{n}'=(n_1+1,\ n_2+1,\dots,n_N+1) \qquad (1.1)$$

$$q_{\mathbf{n},\mathbf{n}'} = \mu_i \qquad \text{if } \mathbf{n}'=(n_1,\dots,\ n_{i-1},\ n_i-1,\ n_{i+1},\dots,n_N) \text{ and } n_i>0,\ 1\le i\le N \qquad (1.2)$$

$$q_{\mathbf{n},\mathbf{n}'} = 0 \qquad \text{otherwise} \qquad (1.3)$$

for $\mathbf{n}\neq\mathbf{n}'$ and

$$q_{\mathbf{n},\mathbf{n}} = -\Sigma_{\mathbf{n}'\neq\mathbf{n}}\ q_{\mathbf{n},\mathbf{n}'}\ .$$

Formula (1.1) corresponds to the arrival of a job at the system, while formula (1.2) corresponds to a completion of a task by server i.

Under irreducibility assumption there exists the stationary probability distribution of system state, denoted by vector π, whose component $\pi(\mathbf{n})$ is the probability of state $\mathbf{n}$, and $\Sigma_{\mathbf{n}\ \ E}\ \pi(\mathbf{n}) =1$. Probability distribution π can be computed as the solution of the following linear system:

$$\pi Q=\mathbf{0}\ , \qquad \text{with } \pi\ \mathbf{1} =1 \qquad (2)$$

where **0** and **1** are the column vector with all zeros and all ones, respectively.

From vector π the following performance indices can also be evaluated:

- mean job response time
- mean response time of task T_i ,$1\le i\le N$
- join queue length distribution
- mean synchronisation delay
- speedup, defined as the mean job response time using N processors divided by the mean job response time using one processor.

In order to solve linear system (2) a numerical technique cannot be applied because of the infinite state space E and it is not trivial to extend to N>2 the derivation of the generating function of the state probability proposed for N=2 in [7]. On the other hand, classical closed form solutions [3] do not hold for such models, because of the presence of fork and join constructs. We shall now propose a bound solution.

In the next two sections, we introduce two modified models based on two different state space reductions of the original model which lead to an upper bound and a lower bound model. For both the models an algorithmic approach is applied to evaluate the stationary state distribution and average performance indices.

The proposed solutions are based on the matrix-geometric algorithmic method for solving Markov processes having a special structure known as quasi-birth-death processes (QBD) [16]. We shall now recall the matrix-geometric algorithm for a Markov process with state space E* and transition rate matrix Q*. By defining an appropriate partition of the state space E*, we assume that process matrix Q* can be rewritten as shown in Figure 3, where submatrices B and A_i, i=0,1,2, are square matrices of order a, with a>0. If stability conditions are verified, then it is possible to efficiently compute steady-state probability vector π^* through the following algorithmic approach [16].

$$Q^* = \begin{bmatrix} B & A_0 & & & & \\ A_2 & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & & & \\ & & & & & \end{bmatrix}$$

Fig. 3 - Quasi-Birth-Death Markov process matrix.

Let vector π^* be partitioned as $\pi^*=(\pi^*_0,\pi^*_1,\pi^*_2,...)$ where subvectors π^*_i, $i\geq 0$, have dimension a. Let $A=A_0+A_1+A_2$ be the infinitesimal generator of a finite Markov process which is assumed to be irreducible. Therefore there exists the steady-state probability vector $\mathbf{x}$ defined by $\mathbf{x}\ A=\mathbf{0}$, with $\mathbf{x}\ \mathbf{1}=1$. Neuts proved the following theorem [16, chap. 1]:

Theorem 1. The Markov process with infinitesimal generator Q* is positive recurrent if and only if $\mathbf{x}\ A_2\ \mathbf{1} > \mathbf{x}\ A_0\ \mathbf{1}$. In this case there exists a non-negative matrix R, with spectral radius less than 1, which is the unique non-negative solution of the matrix quadratic equation

$$A_0 + R\ A_1 + R^2\ A_2 = \emptyset$$

Steady-state probability π^* is given by

$$\pi^*_i = \pi^*_0\ R^i \qquad i \geq 1 \qquad (3.1)$$

and

$$\pi^*_0\ (B + R\ A_2) = \emptyset \qquad (3.2)$$

with

$$\pi^*_0\ (I - R)^{-1}\ \mathbf{1} = 1.$$

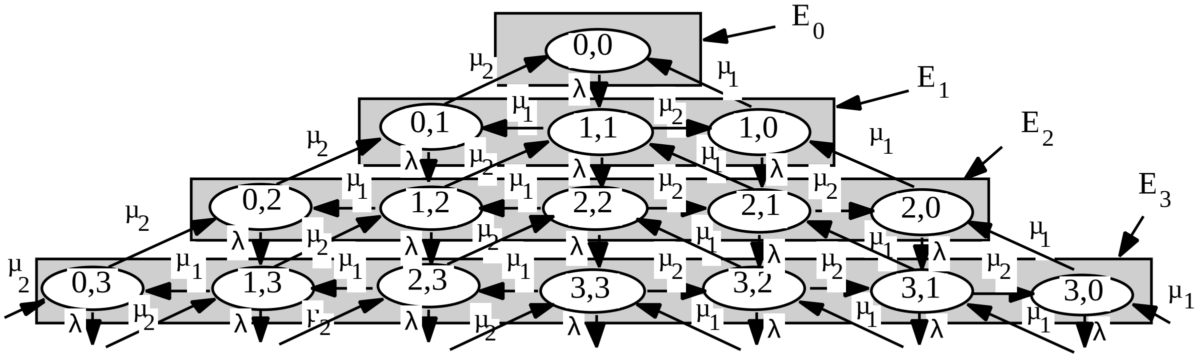

Fig. 4 - State transition diagram of a 2-node heterogeneous fork and join system: first state space partition.

An iterative approach can be used to compute R as follows:

$$R(0) = \emptyset$$
$$R(n) = - A_0\ A_1^{-1} + R^2(n)\ A_0\ A_1^{-1} \qquad n\geq 0 \qquad (4)$$

and it can be shown that R(n) monotonically converges to R, as n goes to infinity [16].

3. THE UPPER BOUND MODEL

In this section we construct a new model whose solution provides an upper bound on the performance of the original fork and join queuing system. To this end, in order to apply the matrix geometric method, we inspect the structure of process matrix Q defined by formulas (1). State space E can be partitioned as follows :

$$E = \bigcup_{k \geq 0} E_k$$

$$E_k = \{\mathbf{n} = (n_1, n_2, \quad , n_N) \quad E, \quad i : n_i = k, \; n_j \leq k, j \neq i, \; 1 \leq i, j \leq N\}, \; k \geq 0$$

where E_k contains all the states with k jobs in the system. For the simple case of the fork and join system with N=2, the state transition diagram and the corresponding partition is shown in Fig. 4.
By reordering system states according to this partition, matrix Q can be rewritten as follows:

$$Q = \begin{array}{c} E_0 \\ E_1 \\ E_2 \\ \vdots \\ E_k \\ \vdots \\ \\ \end{array} \begin{bmatrix} Q_{00} & Q_{01} & & & & & & \\ Q_{10} & Q_{11} & Q_{12} & & & & & \\ & Q_{21} & Q_{22} & Q_{23} & & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & Q_{kk-1} & Q_{kk} & Q_{kk+1} & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & & & \end{bmatrix} \qquad (5)$$

The block tridiagonal structure of matrix Q derives from transition rates definition (1) and from the space state partition. In other words, the only non-zero transitions from a state $\mathbf{n} \quad E_k$, for k≥0, are to states belonging either to subset E_{k+1} or E_{k-1} (if k>0) or to E_k itself. Unfortunately Q does not show a QBD structure (as the matrix in Fig. 3) because subset E_k cardinality increases with k, which also implies that submatrices Q_{kj} dimensions, k-1≤j≤k+1, grow with k. However, by choosing an appropriate ordering of system's states, it is possible to show that matrix Q_{kj} is also submatrix of Q_{k+1j+1}, k-1≤j≤k+1. In this case we can define a reduced state space E^U and a partition into subsets $E^U_k \quad E_k$, k≥0, to obtain a new process matrix which has a QBD structure. However, we can consider various possible definitions of a new reduced process with a QBD structure and whose state space partition does not necessarily satisfy condition $E^U_k \quad E_k$, k≥0. Then we choose a different and simpler definition of the reduced state space E^U and of its partition which leads to a simple QBD structure of the associated Markov process.
The reduced state space E^U is defined as follows:

$$E^U = \{\mathbf{n} \quad E : \; -U_{ji} \leq n_i - n_j \leq U_{ij}, \; 1 \leq i, j \leq N\} \qquad (6)$$

where U_{ij} are positive constant, 1≤i,j≤N, and

$$E^U = \bigcup_{k \geq 0} E^U_k \quad ,$$

$$E^U_k = \{ n \quad E^U : \min_{1 \leq i \leq N} n_i = k \} \quad , \quad \left| E^U_k \right| = k_U \qquad k \geq 0.$$

Therefore the corresponding new transition rate matrix Q^U has the QBD structure shown in Fig. 3, where $A_0 = Q^U{}_{01} = \lambda\, \mathbf{I}_{k_U}$, $A_1 = Q^U{}_{11}$, $A_2 = Q^U{}_{10}$, where each $Q^U{}_{kj}$ is derived by formulas (1.1) through (1.3) from the corresponding submatrix of Q by considering only the rows related to states of $E^U{}_k$ and the columns related to states of $E^U{}_j$, for k,j≥0, except for the diagonal elements in the diagonal submatrices $Q^U{}_{kk}$, k≥0, which are given by

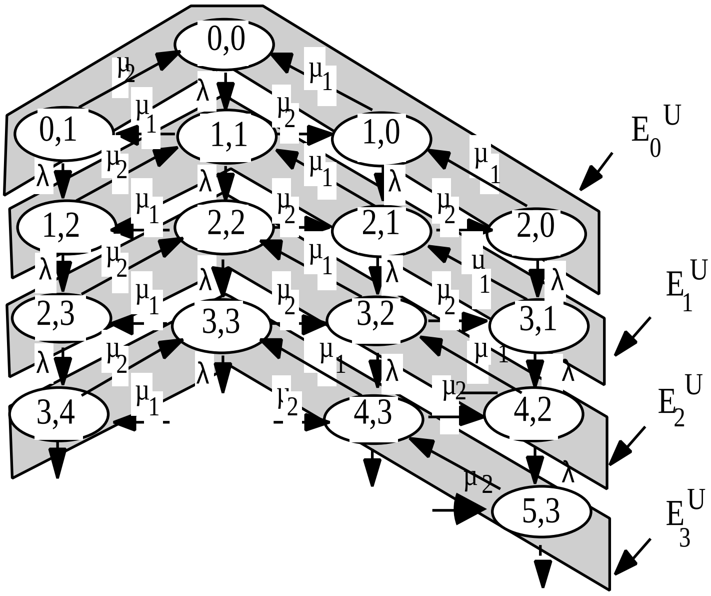

Fig. 5 - State transition diagram for reduction (6) with N=2, U_{12}=2, U_{21}=1 and k_U=4.

$$q^U_{\mathbf{n},\,\mathbf{n}} = -\sum_{\mathbf{n}'\quad E^U_{k-1} \cup E^U_k \cup E^U_{k+1}} q^U_{\mathbf{n},\,\mathbf{n}'}\ .$$

Figure 5 shows the reduced state space diagram E^U of a system with N=2, U_{12}=2 and U_{21} =1. By applying reduction (6) we define the reduced state space E^U by discarding an appropriate subset of system states so obtaining an approximate model for the fork and join system. This reduced model can be exactly evaluated in terms of steady-state probability distribution π_U by using the matrix-geometric technique. Vector π_U is the solution of the linear system $\pi_U Q^U=0$ with $\pi_U \mathbf{1}=1$ and is computed by applying the theorem, if the stability condition is verified.

3.1 Performance indices

By using the matrix-geometric solution of the upper bound model, we can directly compute other average performance indices such as the average number of jobs in the system and the average job response time, respectively denoted by L^U and W^U. One can derive the following expression:

$$L^U = \pi_{U,0}\,(I\text{-}R)^{-1}\,\alpha + \pi_{U,0}\,R\,(I\text{-}R)^{-2}\,\mathbf{1} \qquad (7)$$

where $\pi_{U,0}$ denotes the probability subvector of π_U corresponding to subset $E^U{}_0$, R is the matrix derived from the algorithm by formula (4) and vector α has the same number of components as vector $\pi_{U,0}$ and is defined as follows:

α (**n**) = max $_{1\leq i\leq k}$ n_i for each **n** $E^U{}_0$.

The derivation of formula (7) is given in Appendix A.
Note that if the stability condition holds, the throughputs of the new model and the original one are identical and equal to the arrival rate λ. Hence we can immediately derive the mean job response time as follows:

$$W^U = L^U / \lambda. \qquad (8)$$

The proposed reduction of state space E defined by (6) discards all those states for which the difference between queue lengths i and j is greater than U_{ij}, $1\leq i,j\leq N$. Thresholds U_{ij} are the minimum values such that, for a given $\varepsilon > 0$, Prob$\{n_i-n_j>U_{ij}\}<\varepsilon$, $1\leq i,j\leq N$. The value of ε represents an upper bound to the probabilities of the discarded states by the state space reduction. From the system's viewpoint the new model represents the following behavior: when $n_i = n_j + U_{ij}$ server j is blocked until a service is completed by node i. As soon as a departure occurs from node i, the server of node j starts again servicing the tasks. Therefore, roughly speaking, the mean number of jobs and the mean job response time of the new model are upper bounds on those obtained by the original fork and join model, because of the blocking of the servers.
We shall now formally prove that the new model provides upper bounds on a set of performance measures of the original model, by following the approach in [19, 20].

3.2 Proof of upper bound

In order to provide the proof of the bound we first transform the continuous-time Markov processes in corresponding discrete-time Markov process by uniformization [19].
Let $M = [\lambda + \sum_{i=1}^{N} \mu_i]$ and denote by P and P_U the corresponding uniformized Markov one-step transition matrices with

$$\begin{array}{lll}
p(\mathbf{n},\mathbf{n}')= \lambda\, M^{-1} & \text{if } \mathbf{n}'=(n_1+1,\, n_2+1,\dots,n_N+1) & (9.1)\\
p(\mathbf{n},\mathbf{n}')= \mu_i\, 1_{\{n_i>0\}} M^{-1} & \text{if } \mathbf{n}'=(n_1,\dots,\; n_{i-1},\, n_i\text{-}1,\, n_{i+1},\dots,n_N)\ 1\leq i\leq N & (9.2)\\
p(\mathbf{n},\mathbf{n}')=0 & \text{otherwise if } \mathbf{n}'\neq \mathbf{n} & (9.3)\\
p(\mathbf{n},\mathbf{n})=1- \sum_{\mathbf{n}'\neq\mathbf{n}} p(\mathbf{n},\mathbf{n}') & & (9.4)
\end{array}$$

and

$$\begin{array}{lll}
p_U(\mathbf{n},\mathbf{n}')= \lambda\, M^{-1} & \text{if } \mathbf{n}'=(n_1+1,\, n_2+1,\dots,n_N+1) & (10.1)\\
p_U(\mathbf{n},\mathbf{n}')= \mu_i\, 1_{\{n_i>0\}} M^{-1} & & (10.2)\\
 & \text{if } \mathbf{n}'=(n_1,\dots,\; n_{i-1},\, n_i\text{-}1,\, n_{i+1},\dots,n_N)\ 1\leq i\leq N & \\
p_U(\mathbf{n},\mathbf{n}')=0 & \text{otherwise if } \mathbf{n}'\neq \mathbf{n} & (10.3)\\
p_U(\mathbf{n},\mathbf{n})=1- \sum_{\mathbf{n}'\neq\mathbf{n}} p_U(\mathbf{n},\mathbf{n}') & (10.4) &
\end{array}$$

For a given reward rate function r(.) at E let the function $V^t(.)$ for t=0,1,2,… be defined by:

$$V^t(\mathbf{n}) = \sum_{k=0}^{t-1} M^{-1} \sum_{\mathbf{n}'} p^k(\mathbf{n},\mathbf{n}')\, r(\mathbf{n}') = r(\mathbf{n})\, M^{-1} + \sum_{\mathbf{n}'} p(\mathbf{n},\mathbf{n}')\, V^{t-1}(\mathbf{n}') \qquad (11)$$

Case	Measure	Reward rate
1	total number of jobs $n^* = \max_i n_i$	$r(\mathbf{n}) = n^* = \max_i n_i$
2	total number of tasks $n = n_1+n_2+\ldots+n_N$	$r(\mathbf{n}) = n = n_1+n_2+\ldots+n_N$
3	tail probability for number of jobs $\mathrm{Prob}\{n^*>t\}$	$r(\mathbf{n}) = 1_{\{n^*>t\}}$
4	arbitrary probability of joint task vector, provided $\mathbf{n}\ \ E^U \qquad \mathbf{n}+\mathbf{e}_i\ \ E\ \ i$ $\pi(\mathbf{n})$	$r(\mathbf{n}) = 1_{\{\mathbf{n}\ \ E\}}$

and similarly define $V^t_U(.)$ with P replaced by P_U. Then by standard Tauberian theorems, the following limit is well defined and independent of the initial distribution $\pi^0(\mathbf{n})$ at E.

$$G = \lim_{t\to\infty} \frac{M}{t} \sum_{\mathbf{n}} \pi^0(\mathbf{n})\, V^t(\mathbf{n}) \qquad (12)$$

This value represents the expected average reward per unit of time of the original model when using the reward rate r(.). Similarly, we define G^U for the new model.

The following lemmas 1 and 2 will relate the performance measures G and G^U. These lemmas are a direct application of results in [19, 20] tailored to the above models.

Lemma 1. Let f(.) be a function such that for any $\mathbf{n}\ \ E^U$ and $t \geq 0$:

$$\sum_{i=1}^{N} \mu_i\, 1_{\{n_i>0\}} 1_{\{\mathbf{n}-\mathbf{e}_i\ \ E^U\}} \left| V^t(\mathbf{n}-\mathbf{e}_i) - V^t(\mathbf{n}) \right| \leq f(\mathbf{n}) \qquad (13)$$

Then

$$\left| G - G^U \right| \leq \sum_{\mathbf{n}} \pi_U(\mathbf{n})\, f(\mathbf{n}) \qquad (14)$$

Proof. The proof is given in appendix B.

Lemma 2. $G \leq (\geq)\ G^U$ when

$$\sum_{i=1}^{N} \mu_i\, 1_{\{n_i>0\}} 1_{\{\mathbf{n}-\mathbf{e}_i\ \ E^U\}} \left[V^t(\mathbf{n}) - V^t(\mathbf{n}-\mathbf{e}_i) \right] \geq (\leq)\ 0 \qquad (15)$$

Proof. The proof is given in appendix B.

The following lemma 3 will enable us to apply the above two lemmas for a general class of performance measures G by appropriate choice of a reward rate r. Most notably it will apply for instance to the following steady-state performance measures:

(16)

Lemma 3. With $C = \max_i M / (\mu_i - \lambda)$ and arbitrary $r(\mathbf{n})$ such that

$$0 \leq r(\mathbf{n}+\mathbf{e}_i) - r(\mathbf{n}) \leq 1 \qquad (17)$$

for any $\mathbf{n}\ \ E^U$ and t≥0:

$$0 \ \le\ V^t(\mathbf{n}+\mathbf{e}_i) - V^t(\mathbf{n}) \ \le\ (n_i+1)\,C \qquad (18)$$

Proof. The proof is given in appendix B.

By combination of lemmas 1, 2 and 3 the following result can be established.

***Result 1**. With $C = \max_i M / (\mu_i - \lambda)$ and $r(\mathbf{n})$ satisfying (17):*

$$0 \ \le\ G^U - G \ \le\ C \sum_{\mathbf{n}\ \tilde{E}^U} \pi_U(\mathbf{n}) \sum_i n_i\, \mu_i = \Delta_U \qquad (19)$$

where $\tilde{E}^U = \{\mathbf{n}\ \ E^U |\ \mathbf{n}-\mathbf{e}_i\ \ E^U\ \text{ for some i}\}$. *Particularly it applies to any of the measures form (16) and with the average response time of a job we obtain by Little's law:*

$$0 \ \le\ W^U - W \ \le\ \frac{\Delta_U}{\lambda} \qquad (20)$$

Remark. Note that the condition in case 4 of (16) includes as a special case any set E of the form E={$\mathbf{n}$ | $n_i > t_i$ for 1≤i≤N}. In other words, the error bound Δ_U also applies to arbitrary tail probabilities of the joint population vector and thus also the detailed joint probability distribution.

4. THE LOWER BOUND MODEL

Though relations (19) and (20) are of some practical interest as one can recursively solve the upper bound model, they still contain the complication that this upper bound model is infinite. In this section we therefore also consider a third but finite model. This model will not only provide lower performance bounds on the performance of the original fork and join queuing system, but it provides in addition computational error bounds.

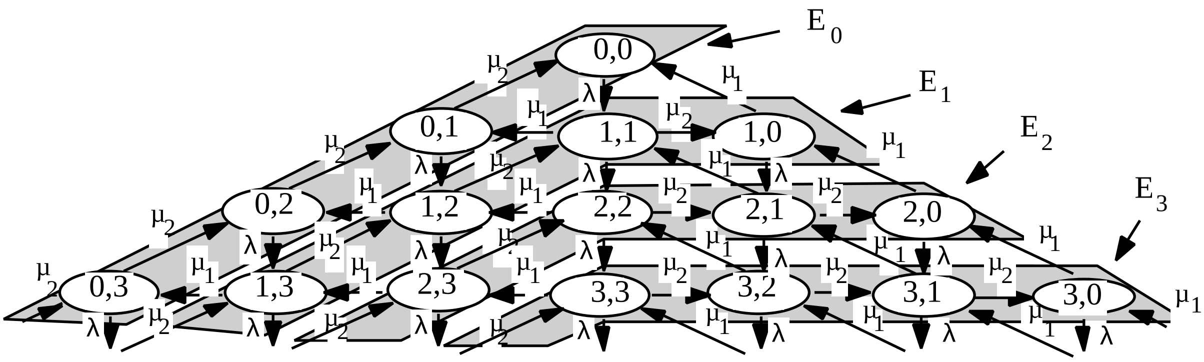

Fig. 6 - State transition diagram of a 2-node heterogeneous fork and join system: second state space partition.

Similarly to the previous section we define a new appropriate state space partition for the original model as follows:

$$E = \bigcup_{k\ge 0} E_k \qquad E_k = \{\mathbf{n}\ \ E : n_1 = k\},\ k \ge 0$$

where E_k includes all the states with k jobs in the first queue. Figure 6 shows the state space partition on the state transition diagram for the fork and join system with N=2.

By rewriting the transition rate matrix Q according to this new state space partition and by considering an appropriate state ordering, we obtain the structure shown in (5) where $Q_{k+1k}=\mu_1\mathbf{I}$, for k≥0, **I** denotes the identity matrix,

$$Q_{kk+1} = \lambda \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & & \\ & & & \end{bmatrix} \quad \text{for } k \geq 0,$$

Q_{kk} are identical for k≥1 and $Q_{00}=Q_{10}+Q_{11}$.

In order to define a new process having a QBD structure we define the following state space reduction E^L of state space E:

$$E^L = \{ \mathbf{n} \in E : n_i \leq U_i,\ 2 \leq i \leq N \} \qquad (21)$$

$$E^L = \bigcup_{k\geq 0} E^L_k\ ,\ E^L_k \subset E_k \quad \text{for } k\geq \max_{2\leq i\leq N} U_i \quad \text{and} \left|E^L_k\right| = k_L \quad \text{for } k\geq 0$$

$$E^L_k = \{ \mathbf{n} \in E^L : \mathbf{n} \in E_k \} \quad \text{for } k\geq \max_{2\leq i\leq N} U_i$$

where U_i are positive constant, 2≤i≤N, and all the subsets E^L_k have identical cardinality $k_L = \prod_{i=2}^{N}(U_i + 1)$.

Therefore the corresponding new transition rate matrix Q^L has the QBD structure shown in Fig. 3, where $A_0= Q^L_{01}$, $A_1= Q^L_{11}$, $A_2= Q^L_{10} = \mu_1\mathbf{I}_{k_L}$, for k≥0, where each Q^L_{kj} is derived by formulas (1.1) through (1.3) from the corresponding submatrix of Q by considering only the rows of states of E^L_k and the columns of states of E^L_j, for k,j≥0 except for the diagonal elements in the diagonal submatrices Q^L_{kk}, k≥0, which are given by

$$q^L_{\mathbf{n},\,\mathbf{n}} = - \sum_{\mathbf{n}' \in E^L_{k-1}\cup E^L_k \cup E^L_{k+1}} q^L_{\mathbf{n},\,\mathbf{n}'} \,.$$

The stationary solution of the new model, denoted by π_L is derived by the solution of linear system $\pi_L Q^L=0$ with $\pi_L\mathbf{1}=1$ and is computed by applying the theorem. Note that if the original system is stable, this also guarantees that the stability condition of the new model is always verified.

4.1 Performance indices

By using the matrix-geometric solution we derive, as given in Appendix A, the following expression for the average number of jobs in the systems, denoted by L^L:

$$L^L = \sum_{k=0}^{\max_i U_i -1} k \sum_{\substack{\mathbf{n}\ E^L \\ \max_i n_i = k}} \pi_L(\mathbf{n}) + \max_i U_i \sum_{\substack{\mathbf{n}\ E^L_k \\ n_1 < \max_i U_i , \\ j\neq 1 :\, n_j = \max_i U_i}} \pi_L(\mathbf{n}) + \qquad (22)$$

$$+ \rho_1^{\max_i U_i} \left(\frac{\rho_1}{1-\rho_1} + \max_i U_i \; (1-\rho_1) \right)$$

where $\pi_{L,0}$ denotes the probability subvector of π_L corresponding to subset $E^L{}_0$. Note that probabilities $\pi_L(\mathbf{n})$ in the first summation of formula (22) belong to subvectors $\pi_{L,k}$ for $0\leq k\leq(\max_i U_i)-1$, and those in the second summation belong to subvector $\pi_{L,k}$ with $k=\max_i U_i$.

The throughputs of the new model, denoted by X^L can be immediately computed by the job arrival rate λ and the probability that a job is lost, denoted by P_{loss}, as follows

$$X^L = \lambda\ (1- P_{loss}) \qquad (23)$$

where

$$P_{loss} = \pi_{L,0} \ (I - R)^{-1} \ \mathbf{1} \ \gamma \qquad (24)$$

where vector γ is defined as follows: $\gamma(\mathbf{n})=1$ if $\ i : n_i = U_i$, $2\leq i\leq N$, $\gamma(\mathbf{n})=0$ otherwise, for each $\mathbf{n}\ \ E^U{}_0$.

The derivation of formula (24) is given in Appendix A.

The mean job response time, W^L, is given by the Little's theorem again as:

$$W^L = L^L / X^L. \qquad (25)$$

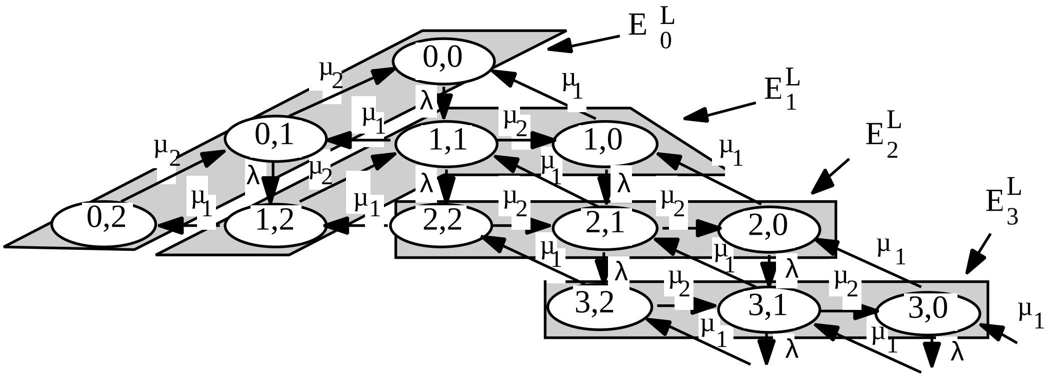

Fig. 7 - State transition diagram for reduction (21) with N=2 and U_2=2.

The new model is defined by the state space reduction (21) by assuming a limited capacity U_i of queue length i, for $2\leq i\leq N$. The first service center, which has the slowest service rate, has infinite queue length. Threshold U_i can be defined as the maximum value such that $\text{Prob}\{n_i > U_i\}<\varepsilon$, $1\leq i\leq N$, given $\varepsilon>0$. From the system's viewpoint the new model represents the following behavior: when any of the queue lengths i, for $2\leq i\leq N$, reaches its

maximum capacity, i.e., $n_i = U_i$, the job arrival process is “turned off” (or blocked) until a departure occurs from server i. Because of the exponential inter-arrival time distribution, we can also assume that an arriving job that find the system in a state **n** with at least one full queue ($2 \le i \le N$), is not accepted by the system and is lost.
Figure 7 shows an example state diagram for this state space reduction of a system with $N=2$, $U_2 = 2$.
Informally, since the new model has a lower effective arrival rate than the original system, due to the loss of jobs when one of the queues is full, it provides a lower bound on the average response time of a job in the fork and join system.

As in the previous section again we will prove that the proposed model provides performance bounds, in this case lower bounds, on a set of performance measures of the original system. Moreover we obtain an expression of the bound width for the average performance indices.

4.2 Proof of lower bound

As before, first consider the discrete-time Markov process obtained by uniformization of the continuous-time Markov process.
For the reduction (21), Lemma 3 can still be applied and leads to the following result in place of result 1 given by (19), as derived in appendix B.

Result 2. *With $C = \max_i M / (\mu_i - \lambda)$ and $r(\mathbf{n})$ satisfying (16):*

$$0 \le G - G^L \le C \sum_{\mathbf{n}\ \tilde{E}^L} n\ \pi_L(\mathbf{n}) + C\ N\ \pi_L(\tilde{E}^L) = \Delta_L \qquad (26)$$

where

$$\tilde{E}^L = \{\mathbf{n} \quad E^L \mid \quad i : 2 \le i \le N ,\ n_i = U_i \}$$

and $\pi(\tilde{E}^L)$ *is the probability of subset* $\tilde{E}^L$.
In particular it applies to any of the measures form (16) and with the average response time of a job:

$$0 \le W - W^L \le \frac{\Delta_L}{\lambda} \ . \qquad (27)$$

Bound Δ_L from (26) can be expressed as follows:

$$\Delta_L = C \{ \sum_{\mathbf{n}\ \tilde{E}^L} (n+N)\ \pi_L(\mathbf{n}) \} = C \{ \sum_{\substack{\mathbf{n}\ \tilde{E}^L \\ n_1 < U_1}} (n+N)\ \pi_L(\mathbf{n}) + \sum_{\substack{\mathbf{n}\ \tilde{E}^L \\ n_1 \ge U_1}} (n+N)\ \pi_L(\mathbf{n}) \}$$

for a given constant $U_1 > 0$. The evaluation of this bound Δ_L can be performed by direct computation of the first summation and by using the following *a priori* bound for the second summation:

$$\sum_{\substack{\mathbf{n}\ \tilde{E}^L \\ n_1 \ge U_1}} (n_1 + n_2 + \quad + n_N + N)\ \pi_L(\mathbf{n}) \le \sum_{i \ge 2} U_i \sum_{\substack{\mathbf{n}\ \tilde{E}^L \\ n_1 \ge U_1}} \pi_L(\mathbf{n}) + \sum_{\substack{\mathbf{n}\ \tilde{E}^L \\ n_1 \ge U_1}} (n_1 + N)\ \pi_L(\mathbf{n}) \le$$

$$\le \rho_1^{n_1+U_1}\left\{\left[\rho_1^{U_1}-\rho_1^{U_1+1}\right]+\rho_1^{U_1+1}\right\}/\left[1-\rho_1^{U_1+1}\right] \tag{28}$$

The derivation of expression (28) is given in appendix C.

Remark. (Computation of bound) Note that the lower bound solution with the explicit computation of bound Δ_L as per (28) can be used to derive both an upper and a lower bound on the performance of the fork and join model as by:

$$G^L \le G \le G^L + \Delta_L \tag{29}$$

As Δ_L can be computed by a finite summation of terms $\pi_L(\mathbf{n})$ up to the explicit error bound (28), the form (29) is most appropriate for computational purposes.

5. SOLUTION ALGORITHM

The algorithm to evaluate the stationary joint queue length probability and average performance indices of the fork and join model can be summarized as follows, given the approximation bound ε.

1. Choose thresholds U_{ij} and U_i in definitions (6) and (21), $1 \le i,j \le N$, as function of ε. In case of definition (6), bounds U_{ij} require the computation of probabilities Prob$\{n_i - n_j > U_{ij}\}$, i.e., the joint distribution of queue lengths i and j, $1 \le i,j \le N$. This can be obtained by considering the isolated fork and join system with only two service centers (i and j) and by using the result by [7]. In case of definition (21), bounds U_i require the computation of probabilities Prob$\{n_i > U_i\}$ which can be easily calculated by considering the isolated service center i which is an M/M/1 system with arrival rate l and service rate μ_i, $2 \le i \le N$. Therefore one can write $U_i = \left\lceil \log_{\rho_i} \varepsilon \right\rceil$ where $\rho_i = \lambda / \mu_i$.
2. Define the reduced matrices Q^U and Q^L as given in Section 3 and 4, respectively.
3. Apply the matrix-geometric method (formulas (3) and (4)) to the QBD matrices Q^U and Q^L to compute steady-state probability vectors π_U and π_L, respectively.
4. Derive the average performance indices form formulas (7), (8) and (22)-(25), respectively.

The simple case of the two node fork and join system shall serve as example of model solution. Let us apply the state space reduction defined by (21) to the original state space. Matrix Q^L has the QBD structure shown in Figure 3 with $k_L = U_2$ and where submatrices are defined as described in Section 4 as $A_2 = \mu_1 \mathbf{I}_3$, and

$$A_1 = \begin{bmatrix} u & & & \\ \mu_2 & v & & \\ & & & \\ & \mu_2 & v & \\ & & \mu_2 & w \end{bmatrix} \qquad A_0 = \lambda \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & & \\ & & 0 & 1 \\ & & & 0 \end{bmatrix}$$

with $u = -(\lambda + \mu_1)$, $v = -(\lambda + \mu_1 + \mu_2)$, $w = -(\mu_1 + \mu_2)$, $k \geq 0$. All the submatrices are square of order U_2+1.

Note that matrix $A=A_0+A_1+A_2$ is the infinitesimal generator of the M/M/1/U_2 queue with parameters λ and μ_2, for which the steady-state solution $\mathbf{x}$ can be immediately obtained under the stability condition, as observed in [17]. However, note that it is not even necessary to check the stability condition, since this model has been proved to be a lower bound for the original model and hence it is stable if the original model is stable.
Moreover bound Δ_L on the difference between the lower bound and the original model can be evaluated by using (28) so providing also an upper bound on the original fork-join model.

5.1 Computational complexity

The computational complexity of the proposed method is strictly related to the dimension of submatrices A_i $(0 \leq i \leq 2)$ and B which are square matrices of order a, where $a = k_U$ and $a = k_L$ for the upper and lower bound model, respectively.

In order to compute steady-state probability π we have to compute matrix R through the iterative approach (4) with a computational cost estimated as $O(m\, a^3)$, where m is the number of required iterations. Once matrix R has been obtained, probability subvectors π_0 is computed by solving linear system (3), which requires a computational cost of $O(a^3)$. Finally, probability subvector π_i, $i>1$, can be recursively computed as $\pi_i = \pi_{i-1} R$, whose complexity is $O(a^2)$. Therefore the overall computational cost can be estimated as $O(m\, a^3)$.

In conclusion the computational cost of the method is related to the dimension a of the submatrices, which can be determined as a function of system parameters as the number of service centers N and the required approximation bound ε. For instance, in the case of N=2 service centers for the upper and lower state space reduction, respectively, we find $a=U_{12}+U_{21}+1$, $b \leq (k_U)^2$, and $a=U_2+1$. For the general fork and join system with N>2 it can be proved that the computational complexity is low polynomial with U_i and U_{ij} for the two models, but combinatorial in the number of servers.

6. NUMERICAL RESULTS

In this Section, we present some numerical examples to show the effectiveness of the proposed method, in the evaluation of the bounds on the stationary probability distribution of system state and average performance indices.

We consider both homogeneous and heterogeneous fork and join systems. We assume for each numerical example the arrival rate $\lambda=1$.
The first example is a homogeneous fork and join model with two servers. Since both the joint queue length probability distribution and the mean response time of this model can be exactly computed [7, 15], then it is possible to test the accuracy of the proposed method.

We consider the system utilization ρ varying from 0.1 to 0.9, which corresponds to various values of the service rates of the two servers. Table I shows the average job response time for various values of thresholds U_1, U_{12} and U_{21}. The table contains the exact values, the lower (LB) and the upper (UB) bounds on the average job response time. The first two thresholds U_{12} and U_{21} for the upper bound model have been calculated by assuming ε=0.5 10^{-2}, while threshold U_1 for the lower bound model is obtained by ε=10^{-2}. Note that for this homogeneous model U_{12} = U_{21}. Table I shows the difference between the upper and the lower bounds and the percentage error, which is defined as follows:

$$\max\left\{\frac{\text{Exact} - \text{LB}}{\text{Exact}}, \frac{\text{UB} - \text{Exact}}{\text{Exact}}\right\} 100\%$$

We observe that the bounds are very tight.

To test the accuracy of the method in the evaluation of joint queues length probability distribution, we calculated for the upper, lower and exact model, and for each value of ρ, the steady-state probability on a subset Z of state space, such that it guarantees that the following conditions hold:

$$\sum_{(n_1,n_2)\ Z} \pi_L(n_1,n_2) \geq 0.99 \qquad (30)$$

Utilization	U_1	LB	Exact	UB	U_{12}	Spread of Bounds	Percentage Error
0.1	2	0.161811	0.165278	0.165505	2	3.70 10^{-3}	2.1
0.2	3	0.361654	0.368750	0.371428	2	9.78 10^{-3}	1.92
0.3	4	0.613625	0.626786	0.640465	3	2.70 10^{-2}	2.1
0.4	5	0.940366	0.966667	0.978392	3	3.81 10^{-2}	2.72
0.5	7	1.404751	1.437500	1.439488	6	3.52 10^{-2}	2.27
0.6	9	2.074521	2.137500	2.152190	6	7.71 10^{-2}	2.94
0.7	13	3.198019	3.295833	3.306008	10	1.08 10^{-1}	2.97
0.8	21	5.434605	5.600000	5.623753	15	1.89 10^{-1}	2.95
0.9	44	12.08890	12.48750	12.50644	38	4.17 10^{-1}	3.19

Table I: Homogeneous two servers model: first set of experiments.

$$\sum_{(n_1,n_2)\ Z} \pi_U(n_1,n_2) \geq 0.99 \qquad (31)$$

$$\sum_{(n_1,n_2)\ Z} \pi(n_1,n_2) \geq 0.99 \qquad (32)$$

Experimental results can be summarized as follows.
The maximum discrepancy between exact and approximate results has been observed for state (0,0) for the lower bound model and for each value of ρ. For the upper bound model the maximum difference between exact and approximate results has been observed for states (0, U_{12}) and (U_{21}, 0). These results are perfectly consistent with respect to the definition of the two approximate models. The most significant approximation errors have

been observed for state (0, 0) and are presented in Table II for various values of system utilization.

In order to illustrate the trade-off between computational cost and accuracy of the proposed method to obtain the two bounds, we have solved the homogeneous fork and join model by varying thresholds U_1, U_{12} and U_{21}. Threshold U_1 has been calculated by assuming $\varepsilon=10^{-3}$, while U_{12} and U_{21} have been calculated by assuming $\varepsilon=0.5\ 10^{-3}$. Numerical results for the joint queue length distribution are shown in Table III, for various values of system utilization ρ, by considering conditions (30), (31) and (32). Similarly to the first set of experiments the maximum percentage error has been observed for state (0,0) for the lower bound model and for states (0, U_{12}) and (U_{21}, 0) for the upper bound model. The maximum percentage error has been observed for state (0, U_{12}) and (U_{21}, 0). Numerical results are shown in Table IV.

By comparing Tables I and II with Tables III and IV, respectively, we observe how the improvement of the approximation accuracy affects both the computational cost, which is related to thresholds U_1 and U_{12}, and the percentage error. Like the previous case, we observe the most significant approximation error for state (0, 0).

The second example is a fork and join model with two heterogeneous servers. We consider system utilization $\rho = \lambda/\mu_1$ varying from 0.1 to 0.9. For this model the joint queue length distribution can be exactly computed [7], while there are no exact results for the mean job response time. The presented numerical examples allow us to make the following observations: first we compare the bounds obtained by proposed method with those obtained by applying the bounding technique proposed in [1], in terms of spread of bounds

Utilization	U_2	$\pi_L(0,0)$ Lower Bound	$\pi(0,0)$ Exact	Percentage Error
0.1	2	0.854823	0.853672	0.135
0.2	3	0.716855	0.715541	0.181
0.3	4	0.587271	0.584724	0.435
0.4	5	0.466900	0.464758	0.461
0.5	7	0.355083	0.353553	0.433
0.6	9	0.254676	0.252982	0.670
0.7	13	0.165542	0.164316	0.746
0.8	21	0.090164	0.089442	0.807
0.9	44	0.031923	0.031622	0.952

Table II: Homogeneous two servers model, first set of experiments: lower bound approximation for state (0,0).

Utilization	U_1	LB	Exact	UB	U_{12}	Spread of Bounds	Percentage Error
0.1	3	0.164837	0.165278	0.165288	3	$4.50\ 10^{-4}$	0.26
0.2	4	0.367032	0.368750	0.368781	4	$1.74\ 10^{-3}$	0.46
0.3	6	0.621365	0.626786	0.627195	4	$5.83\ 10^{-3}$	0.86
0.4	8	0.964247	0.966667	0.966878	6	$2.26\ 10^{-3}$	0.25
0.5	10	1.437026	1.437500	1.437757	8	$7.31\ 10^{-4}$	0.33
0.6	14	2.130400	2.137500	2.138956	9	$8.50\ 10^{-3}$	0.32
0.7	19	3.284116	3.295833	3.297013	14	$1.28\ 10^{-2}$	0.35
0.8	31	5.574794	5.600000	5.603072	22	$2.82\ 10^{-2}$	0.45
0.9	66	12.43011	12.48750	12.48951	52	$5.94\ 10^{-2}$	0.46

Table III: Homogeneous two servers model: second set of experiments.

Utilization	U_1	$\pi_L(0,0)$ Lower Bound Model	$\pi(0,0)$ Exact	Percentage Error
0.1	3	0.853911	0.853672	0.0239
0.2	4	0.715796	0.715541	0.0254
0.3	6	0.585800	0.584724	0.1842
0.4	8	0.464888	0.464758	0.2810
0.5	10	0.353564	0.353553	0.0322
0.6	14	0.253108	0.252982	0.0125
0.7	19	0.164413	0.164316	0.0588
0.8	31	0.089517	0.089442	0.0836
0.9	66	0.031651	0.031622	0.0917

Table IV: Homogeneous two servers model, second set of experiments: lower bound approximation for state (0, 0).

of the job mean response time. Then we study how the service rate of the second server affects the spread of bounds (note that by assumption $\mu_1 \le \mu_2$). To this aim we have performed, for each different value of ρ, three experiments varying the service rate of the second server as follows: $\mu_2=1.5\mu_1$, $\mu_2=2.0\mu_1$ and $\mu_2=3.0\mu_1$.

Table V shows the numerical results for the combination of service rate values. For each utilization ρ we consider the thresholds already determined for the first set of experiments of the homogeneous model.

Table V shows the results obtained by the lower (LB) and upper bound (UB) of the proposed method and the bounds proposed by Baccelli and Makowski in [1], which are shown in column 6 (BM-LB) and 7 (BM-UB).

We observe that the proposed method provides tighter bounds than those obtained with the technique proposed in [1]. In particular, the approximation accuracy of the approach proposed in this paper is very good even for high system utilization (i.e., $\rho \rightarrow 1$). On the other hand, the computational cost of the method proposed in [1] is negligible with respect to that characterizing the method proposed in this paper. However, note that the proposed method provides both the average job response time and the joint queue length distribution.

In order to illustrate the trade-off between computational cost and accuracy of the bounds, we analyze the heterogeneous fork and join model by varying thresholds U_1, U_{12} and U_{21}. U_1 has been calculated by assuming $\varepsilon=10^{-3}$, while U_{12} and U_{21} with $\varepsilon=0.5\ 10^{-3}$. Table VI shows the numerical results of the proposed bounds for various combinations of service rates.

The joint queue length probability distribution has been evaluated for this heterogeneous model and the experimental results confirm the behaviour observed for the homogeneous model.

Finally, the third set of experiments is a fork and join model with three homogeneous servers. System utilization $\rho = \lambda/\mu_1$ varies from 0.1 to 0.8. To the best of our knowledge, no exact method has been proposed for this model to evaluate the stationary state probability distribution, while there are several methods to calculate approximate job mean response time. We compare our bounds on the mean job response time with those obtained by the recently proposed method by Varma and Makowski [18].

Table VII shows the numerical results for various values of system utilization. The table includes the average job response time obtained by the lower (LB), the upper bound (UB)

Utilization	μ_2	LB	UB	Spread of Bounds	BM-LB	BM-UB	Spread of Bounds
0.1	1.5 μ_1	0.136934	0.138341	$1.41\ 10^{-3}$	0.126670	0.139061	$1.24\ 10^{-2}$
0.1	2.0 μ_1	0.126810	0.127570	$7.60\ 10^{-4}$	0.116671	0.128028	$1.13\ 10^{-2}$
0.1	3.0 μ_1	0.118726	0.119064	$3.38\ 10^{-4}$	0.108337	0.119278	$1.10\ 10^{-2}$
0.2	1.5 μ_1	0.303229	0.306183	$2.95\ 10^{-3}$	0.254562	0.308608	$5.40\ 10^{-2}$
0.2	2.0 μ_1	0.281280	0.282611	$1.33\ 10^{-4}$	0.234586	0.284188	$4.96\ 10^{-2}$
0.2	3.0 μ_1	0.264699	0.265137	$4.38\ 10^{-4}$	0.217984	0.265873	$4.79\ 10^{-2}$
0.3	1.5 μ_1	0.509666	0.517257	$7.60\ 10^{-3}$	0.391676	0.520677	0.138
0.3	2.0 μ_1	0.474273	0.477505	$3.23\ 10^{-3}$	0.361511	0.480042	0.118
0.3	3.0 μ_1	0.449214	0.450178	$3.86\ 10^{-3}$	0.337015	0.451447	0.114
0.4	1.5 μ_1	0.774955	0.781258	$6.30\ 10^{-3}$	0.551928	0.795009	0.243
0.4	2.0 μ_1	0.724836	0.726733	$1.90\ 10^{-3}$	0.510560	0.734885	0.224
0.4	3.0 μ_1	0.691624	0.692180	$5.56\ 10^{-4}$	0.478712	0.695513	0.217
0.5	1.5 μ_1	1.136201	1.138330	$2.13\ 10^{-3}$	0.755432	1.166667	0.441
0.5	2.0 μ_1	1.068470	1.068805	$3.25\ 10^{-4}$	0.701211	1.083333	0.382
0.5	3.0 μ_1	1.028042	1.028067	$2.50\ 10^{-5}$	0.662636	1.033333	0.371
0.6	1.5 μ_1	1.658744	1.661672	$2.92\ 10^{-3}$	1.038192	1.705128	0.667
0.6	2.0 μ_1	1.574211	1.574531	$3.20\ 10^{-4}$	0.969377	1.595238	0.626
0.6	3.0 μ_1	1.528754	1.528767	$1.30\ 10^{-5}$	0.925132	1.535714	0.610
0.7	1.5 μ_1	2.457140	2.467553	$1.40\ 10^{-2}$	1.481257	2.571969	1.090
0.7	2.0 μ_1	2.407225	2.407244	$1.90\ 10^{-5}$	1.397166	2.434295	1.037
0.7	3.0 μ_1	2.360258	2.360266	$8.00\ 10^{-6}$	1.349487	2.368450	1.020
0.8	1.5 μ_1	4.168701	4.168805	$1.04\ 10^{-4}$	2.326987	4.253968	1.930
0.8	2.0 μ_1	4.064488	4.064490	$2.00\ 10^{-6}$	2.231347	4.095238	1.860
0.8	3.0 μ_1	4.021976	4.021977	$1.00\ 10^{-6}$	2.185053	4.030303	1.850
0.9	1.5 μ_1	9.124710	9.124870	$1.60\ 10^{-4}$	4.802873	9.214286	4.410
0.9	2.0 μ_1	9.041762	9.041926	$1.64\ 10^{-4}$	4.715021	9.068182	4.350
0.9	3.0 μ_1	9.012592	9.013258	$6.66\ 10^{-4}$	4.680464	9.019480	4.330

Table V: Heterogeneous two servers model: bound comparison.

Utilization	μ_2	LB	UB	Spread of Bounds
0.1	$1.5\ \mu_1$	0.138127	0.138240	$1.13\ 10^{-4}$
0.1	$2.0\ \mu_1$	0.127470	0.127515	$4.50\ 10^{-5}$
0.1	$3.0\ \mu_1$	0.119030	0.119043	$1.30\ 10^{-5}$
0.2	$1.5\ \mu_1$	0.304757	0.305053	$2.29\ 10^{-4}$
0.2	$2.0\ \mu_1$	0.281952	0.282046	$9.40\ 10^{-5}$
0.2	$3.0\ \mu_1$	0.264921	0.264940	$1.90\ 10^{-5}$
0.3	$1.5\ \mu_1$	0.511872	0.512084	$2.12\ 10^{-4}$
0.3	$2.0\ \mu_1$	0.475017	0.475066	$4.90\ 10^{-5}$
0.3	$3.0\ \mu_1$	0.449381	0.449388	$7.60\ 10^{-6}$
0.4	$1.5\ \mu_1$	0.778222	0.778258	$3.60\ 10^{-5}$
0.4	$2.0\ \mu_1$	0.725677	0.725694	$1.70\ 10^{-5}$
0.4	$3.0\ \mu_1$	0.691954	0.691958	$4.00\ 10^{-6}$
0.5	$1.5\ \mu_1$	1.138053	1.138161	$1.08\ 10^{-4}$
0.5	$2.0\ \mu_1$	1.068764	1.068774	$1.00\ 10^{-5}$
0.5	$3.0\ \mu_1$	1.028061	1.028063	$2.00\ 10^{-6}$
0.6	$1.5\ \mu_1$	1.660599	1.660662	$6.50\ 10^{-5}$
0.6	$2.0\ \mu_1$	1.574390	1.574395	$5.00\ 10^{-6}$
0.6	$3.0\ \mu_1$	1.528763	1.528766	$3.00\ 10^{-6}$
0.7	$1.5\ \mu_1$	2.507407	2.507408	$1.00\ 10^{-6}$
0.7	$2.0\ \mu_1$	2.407231	2.407243	$2.00\ 10^{-6}$
0.7	$3.0\ \mu_1$	2.360258	2.360266	$9.20\ 10^{-5}$
0.8	$1.5\ \mu_1$	4.168771	4.168779	$8.00\ 10^{-6}$
0.8	$2.0\ \mu_1$	4.064488	4.064490	$2.00\ 10^{-5}$
0.8	$3.0\ \mu_1$	4.021967	4.021977	$1.00\ 10^{-5}$
0.9	$1.5\ \mu_1$	9.124710	9.124861	$1.50\ 10^{-4}$
0.9	$2.0\ \mu_1$	9.041765	9.041915	$1.50\ 10^{-4}$
0.9	$3.0\ \mu_1$	9.012594	9.013247	$6.53\ 10^{-4}$

Table VI: Heterogeneous two servers model: second set of experiments.

and the approximation proposed in [18] (VM-APP), and the spread of bounds. Note that the bounds proposed in [18] provide results at a negligible computational cost, but only for homogeneous systems and the method does not provide the joint queue length probability

Utilization	U_i	LB	VM-APP	U_{ij}	UB	Spread of Bounds
0.1	2	0.194853	0.206322	2	0.215790	0.020937
0.2	3	0.433482	0.459715	3	0.475947	0.042464
0.3	4	0.731201	0.78933	4	0.827475	0.096273
0.4	5	1.111711	1.229813	5	1.317926	0.206215
0.5	7	1.656398	1.847650	7	1.989013	0.332614
0.6	9	2.426575	2.822750	9	3.198924	0.772366
0.7	11	3.592916	4.324856	12	5.010544	1.417628
0.8	13	5.467226	7.425760	13	9.264357	3.797130

Table VII : Homogeneous three servers model.

distribution. For this set of experiments we observe a very good accuracy of the proposed method. Note that for high utilization the results can be improved by choosing larger values of parameters U_i and U_{ij} in order to obtain higher accuracy.

7. CONCLUSIONS

An algorithmic approach for the performance evaluation of a fork and join system with synchronisation has been presented based on two models which provide upper and lower bounds on the system performance. The solution model is given in terms of steady-state joint queue length probability distributions from which other performance indices, such as synchronisation delay, job and task response time, can be derived. The proposed algorithm shows a low polynomial computational complexity.
The two models have been proved to provide lower and upper bounds on the system performance. Moreover computation error bounds have been derived.
A number of extensions seem possible such as to job and task response time probability distributions and other synchronization conditions for parallel processors.

REFERENCES

[1] F.Baccelli and A.M. Makowski “Simple computable bounds for the fork-join queue”, in Proc. John Hopkins Conf. Inf. Sci., John Hopkins Univ., 1985.

[2] F.Baccelli, W.A. Massey and D.Towsley “Acyclic fork-join queueing network”, J. of ACM, Vol. 36, n. 3, July 1989.

[3] F.Baskett, K.M.Chandy, R.R.Muntz and J.Palacios “Open, closed and mixed networks of queues with different classes of customers” J. of ACM, vol. 22, 2, pp. 248-260, Apr. 1975.

[4] M.A.Brun and G.Fajolle “The distribution of the transaction processing time in a simple fork-join system” 2nd Int. Workshop on Math. and Comp. Performance and Reliability, Rome, May 25-29, 1987, (North Holland).

[5] A.Duda and T.Czachòrski “Performance evaluation of fork and join synchronisation primitives” Acta Informatica, vol. 24, pp. 525-553, 1987.

[6] A.Duda “Approximate performance analysis of parallel systems” 2nd Int. Workshop on Math. and Comp. Performance and Reliability, Rome, May 25-29, 1987, (North Holland).

[7] L.Flatto and S.Hahn “Two parallel queues created by arrivals with two demands I” SIAM J. Appl. Math., vol. 44, pp. 1041-1053, Oct. 1984.

[8] L.Flatto “Two parallel queues created by arrivals with two demands II” SIAM J. Appl. Math., vol. 45, pp. 861-878, Oct. 1985.

[9] C.Kim and A.K.Agrawala “Analysis of the fork-join queue” IEEE Trans. on Comp., vol. 38, n.2, Feb. 1989.

[10] P.Heidelberger and K.Trivedi “Queueing network models for parallel processing with asynchronous tasks” IEEE Trans. on Comp., vol. 31, pp.1099-1109, Nov. 1982.

[11] P.Heidelberger and K.Trivedi “Analytic queueing models for programs with internal concurrency” IEEE Trans. on Comp., vol. 32, pp. 73-82, Jan. 1983.

[12] P.Heidelberger and S.S.Lavenberg “Computer Performance Evaluation Methodology” IEEE Trans. on Computers, vol. 33, pp. 1195-1220, 1984.

[13] A.Kumar and R. Shorey “Performance Analysis and Scheduling of Stochastic Fork-Join Jobs in a Mutilcomputer System” IEEE Trans. on Parallel and Distr. Systems, vol. 4, pp. 1147-1164, 1993.

[14] R.Nelson and A.N.Tantawi “Approximate analysis of fork-join synchronization in parallel queues” IEEE Trans. on Computers., vol. 37, 6, June 1988, pp. 739-743.

[15] R.Nelson, D.Towsley and A.N.Tantawi “Performance analysis of parallel processing systems” IEEE Trans. on Software Eng., vol. 14, 4, Apr. 1988, pp. 532-540.

[16] M.Neuts ‘Matrix Geometric Solution in Stochastic Models’, John Hopkins University Press, Baltimore, Md., 1981.

[17] B.M.Rao and M.J.M.Posner “Algorithmic approximation analysis of the split and match queue” Comm. Statis. Stochastic Models, 1 (3), pp. 433-456, 1985.

[18] S.Varma and A.M.Makowski “Interpolation Approximations for Symmetric Fork-Join Queues” Proc. Performance ‘93, Rome, Italy, Sept. 29-Oct. 1, pp. 245-273, 1993.

[19] N.van Dijk “On the Importance of Bias-terms for error bounds and comparison results” Proc. First Internat. Conference on Numerical Solution of Markov chains, January, 1990.

[20] N.van Dijk “An error bound theorem for approximate Markov chains” Probability in Engineering and Informational Science, 6, 413-424.

APPENDIX A

Performance indices of the upper bound model

The average number of jobs in the upper bound model is defined as follows:

$$L^U = \sum_{\mathbf{n} \in E^U} \pi_U(\mathbf{n})(\max_i n_i) = \sum_{k=0}^{\infty} \pi_{U,k}\,(\alpha + k\,\mathbf{1}) =$$

$$= \Big(\sum_{k=0}^{\infty} \pi_{U,k}\Big)\,\alpha + \sum_{k=0}^{\infty} k\ \pi_{U,k}\ \mathbf{1} =$$

$$= \pi_{U,0}\Big(\sum_{k=0}^{\infty} R^k\Big)\,\alpha + \pi_{U,0}\ R\ (I- R)^{-2}\mathbf{1} =$$

$$= \pi_{U,0}(I-R)^{-1}\,\alpha + \pi_{U,0}\ R\ (I-R)^{-2}\mathbf{1}$$

where vector α has the same number of components as vector $\pi_{U,0}$ and is defined as follows:

$$\alpha\,(\mathbf{n}) = \max_{1\le i\le N}\ n_i \qquad \text{for each } \mathbf{n} \in E^U{}_0.$$

We assume the same state ordering within each subset $E^U{}_k$, $k\ge 0$. Note that subvector $\pi_{U,0}$ is obtained by the solution of linear system (3.2) for the upper bound model.

Performance indices of the lower bound model

The average number of jobs in the lower bound model is defined as follows:

$$L^L = \sum_{\mathbf{n}\ E^L} \pi_L(\mathbf{n})\,(\max_i n_i) = \sum_{k=0}^{\infty} k \sum_{\substack{\mathbf{n}\ E^L\\ \max_i n_i = k}} \pi_L(\mathbf{n}) =$$

$$= \sum_{k=0}^{\max_i U_i - 1} k \sum_{\substack{\mathbf{n}\ E^L\\ \max_i n_i = k}} \pi_L(\mathbf{n}) + \max_i U_i \sum_{\substack{\mathbf{n}\ E^L_k\\ n_1 < \max_i U_i,\\ j\neq 1\,:\, n_j = \max_i U_i}} \pi_L(\mathbf{n}) +$$

$$+ \max_i U_i \sum_{\substack{\mathbf{n}\ E^L\\ n_1 = \max_i U_i}} \pi_L(\mathbf{n}) + \sum_{k=\max_i U_i + 1}^{\infty} k \sum_{\substack{\mathbf{n}\ E^L_k\\ \max_i n_i = k}} \pi_L(\mathbf{n}) \tag{A.1}$$

The first and the second summations include only elements of the subvectors $\pi_{L,k}$ for $k=0,\ldots,\max_i U_i-1$, each corresponding to a subset $E^L{}_k$. The fourth summation in expression (A.1) can be rewritten as follows:

$$\sum_{k > \max_i U_i} k \sum_{\substack{\mathbf{n}\ E^L_k\\ n_1 = k}} \pi_L(\mathbf{n}) \tag{A.2}$$

where the internal summation is the marginal probability of k tasks in node 1 and it can be easily computed by the M/M/1 queue length distribution with arrival rate $\lambda(1-P_{loss})$, where P_{loss} denotes the probability that a job is lost, and service rate μ_1. Let $\rho_1=\lambda(1-P_{loss})/\mu_1$. Hence, the third and fourth summations in expression (A.1) can be rewritten as follows:

$$\rho_1^{\max_i U_i}\left(\frac{\rho_1}{1-\rho_1} + \max_i U_i\ (1-\rho_1)\right)$$

which by substitution in (A.1) leads to formula (22).

Therefore the computation of L^L only requires the first $\max_i U_i$ subvectors $\pi_{L,k}$ from $\pi_{L,0}$.

The probability P_{loss} that a job is lost is defined as follows:

$$P_{loss} = \sum_{\substack{\mathbf{n}\ \ E^L \\ i\,:\,2 \le i \le N,\ n_i = U_i}} \pi_L(\mathbf{n}) =$$

$$= \sum_{k=0}^{\infty} \sum_{\substack{\mathbf{n}\ \ E^L \\ n_1 = k,\quad i\,:\,2 \le i \le N,\ n_i = U_i}} \pi_L(\mathbf{n}) =$$

$$= \sum_{k=0}^{\infty} \sum_{\mathbf{n}\ \ E^L_k} \pi_L(\mathbf{n})\, 1_{\{\ i\,:\,2 \le i \le N,\ n_i = U_i\}} =$$

$$= \sum_{k=0}^{\infty} \pi_{L,k}\ \gamma = \pi_{L,0}\ \Big(\sum_{k=0}^{\infty} R^k \mathbf{1}\Big)\ \gamma = \pi_{L,0}\ (I - R)^{-1}\mathbf{1}\ \gamma$$

where vector γ is defined as follows:

$$\gamma(\mathbf{n}) = 1_{\{\ \ i\,:\,n_i = U_i\,,\ 2 \le i \le N\}} \qquad \text{for each } \mathbf{n}\ \ E^U_0$$

which is the indicator function that at least one queue i, 2≤i≤N, is full. We assume the same state ordering within each subset $E^L{}_k$, k≥0. And this completes the proof.

APPENDIX B

Proof of Lemma 1

By using the second relation from (11) and the fact that P_U remains restricted to E^U which is a subset of E, for arbitrary state $\mathbf{n}\ \ E^U$ we can write:

$$(V^t_U - V^t)(\mathbf{n}) = (P_U V^{t-1}_U - P\, V^{t-1})(\mathbf{n}) = (P_U - P)\, V^{t-1}(\mathbf{n}) + P_U (V^{t-1}_U - V^{t-1})(\mathbf{n}) =$$

$$= \quad = \sum_{k=0}^{t-1} P^k_U\, [(P_U - P) V^{t-k-1}](\mathbf{n}) \qquad \text{(A.3)}$$

where the latter equality follows by iteration and the fact that $V^0(.)=V_U{}^0(.)=0$.
Further, by substituting (9) and (10) and $h=M^{-1}$, for any s we have for $\mathbf{n}\ \ E^U$:

$$(P_U - P)V^s(\mathbf{n}) =$$

$$= \Big\{\sum_i h\,\mu_i\, 1_{\{n_i>0\}}\, 1_{\{\mathbf{n}-\mathbf{e}_i\ \ E^U\}}\, V^s(\mathbf{n}-\mathbf{e}_i) + [1 - h\,\lambda - \sum_i h\,\mu_i 1_{\{n_i>0\}}\, 1_{\{\mathbf{n}-\mathbf{e}_i\ \ E^U\}}]\, V^s(\mathbf{n})\Big\}$$

$$- \Big\{\sum_i h\,\mu_i\, 1_{\{n_i>0\}}\, V^s(\mathbf{n}-\mathbf{e}_i) + [1 - h\,\lambda - \sum_i h\,\mu_i 1_{\{n_i>0\}}]\, V^s(\mathbf{n})\Big\} =$$

$$= \sum_i h\,\mu_i\, 1_{\{n_i>0\}}\, 1_{\{\mathbf{n}-\mathbf{e}_i\ \ E^U\}}\, [V^s(\mathbf{n}-\mathbf{e}_i) - V^s(\mathbf{n})] \qquad \text{(A.4)}$$

Now by choosing $\pi^0(.)=\pi_U{}^0(.)=\pi_U(.)$, the steady state distribution of the reduced model in order to apply formula (12). Then by substituting (13) and (A.4) and since the transition matrix P_U leaves its steady state distribution unchanged, we obtain from (A.3):

$$\left|\sum_{\mathbf{n}} \pi_U(\mathbf{n})\,[V_U^t - V^t](\mathbf{n})\right| =$$

$$= \sum_{k=0}^{t-1} \sum_{\mathbf{n}} \pi_U(\mathbf{n}) \sum_{\mathbf{m}} P_U^t(\mathbf{n},\mathbf{m}) \left|(P_U - P)V^{t-k-1}(\mathbf{m})\right| =$$

$$= \sum_{k=0}^{t-1} \sum_{\mathbf{m}} \pi_U(\mathbf{m}) \left|(P_U - P)V^{t-k-1}(\mathbf{m})\right| \leq t\,h \sum_{\mathbf{m}} \pi_U(\mathbf{m})\, f(\mathbf{m})$$

Substituting $h=M^{-1}$ and employing (12) completes the proof. □

Proof of Lemma 2

Directly by substituting (A.4) in (A.3) and observing that the matrix P_U is nonnegative so that $P_U\, g \geq 0$ if $g \geq 0$ componentwise. □

Proof of Lemma 3

The proof will follow by induction in t.
Clearly, (18) holds for t=0 as $V^0(.)=0$. Suppose that (18) holds for t=k. Then for t=k+1 we obtain by applying (14) in state $\mathbf{n}+\mathbf{e}_i$ and $\mathbf{n}$:

$$V^{k+1}(\mathbf{n}+\mathbf{e}_i) - V^{k+1}(\mathbf{n}) =$$

$$= \Big\{ r(\mathbf{n}+\mathbf{e}_i) + h\,\lambda\, V^k(\mathbf{n}+\mathbf{e}_i+\mathbf{1}) \; + $$
$$+ \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} V^k(\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j) + h\,\mu_i V^k(\mathbf{n}) +$$
$$+ [1 - h\,\lambda - \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} - h\,\mu_i\,]\, V^k(\mathbf{n}+\mathbf{e}_i) \Big\} -$$

$$- \Big\{ r(\mathbf{n}) + h\,\lambda\, V^k(\mathbf{n}+\mathbf{1}) \; +$$
$$+ \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} V^k(\mathbf{n}-\mathbf{e}_j) + h\,\mu_i\, 1_{\{n_i>0\}} V^k(\mathbf{n}-\mathbf{e}_i) +$$
$$+ [1 - h\,\lambda - \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} - h\,\mu_i\, 1_{\{n_i>0\}}]\, V^k(\mathbf{n}) \Big\} =$$

$$
\begin{aligned}
= \ & [r(\mathbf{n}+\mathbf{e}_i) - r(\mathbf{n})] + h\,\lambda\,[V^k(\mathbf{n}+\mathbf{1}) - V^k(\mathbf{n})] + \\
& + \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} [V^k(\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j) - V^k(\mathbf{n}-\mathbf{e}_j)] + \\
& + h\,\mu_i\, 1_{\{n_i>0\}} [V^k(\mathbf{n}) - V^k(\mathbf{n}-\mathbf{e}_i)] + h\,\mu_i\, 1_{\{n_i=0\}} [V^k(\mathbf{n}) - V^k(\mathbf{n})] + \\
& + [1 - h\,\lambda - \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} - h\,\mu_i 1_{\{n_i>0\}} - h\,\mu_i 1_{\{n_i=0\}}\,]\,[V^k(\mathbf{n}+\mathbf{e}_i) - V^k(\mathbf{n})]\ .
\end{aligned}
$$

Here it is noted that indeed the term with coefficient $1_{\{n_i=0\}}$ is indeed equal to 0. This term however is kept in for clarification of an argument below. First, by substitution of the lower limit $r(\mathbf{n}+\mathbf{e}_i) - r(\mathbf{n}) \geq 0$ in addition to the induction hypothesis $V^k(\mathbf{n}+\mathbf{e}_i)-V^k(\mathbf{n}) \geq 0$ for all i, one directly verifies $V^{k+1}(\mathbf{n}+\mathbf{e}_i)-V^{k+1}(\mathbf{n}) \geq 0$. Next, by substituting the upper limit $r(\mathbf{n}+\mathbf{e}_i) - r(\mathbf{n}) \leq 1$ in addition to the induction hypothesis $V^k(\mathbf{n}+\mathbf{e}_i)-V^k(\mathbf{n}) \leq (n_i +1)\ C$, by noting that all coefficients sum up to 1 (recall that they represent transition probabilities) and by substituting $C \geq 1/h\ (\mu_i - \lambda)$, we obtain:

$$
\begin{aligned}
& V^{k+1}(\mathbf{n}+\mathbf{e}_i) - V^{k+1}(\mathbf{n}) \leq \\
& \leq 1 + h\,\lambda\,[n_i + \lambda]\ C + \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}}\,[n_i + 1]\ C\ + \\
& \quad + h\,\mu_i\, 1_{\{n_i>0\}}\, n_i\ C + 0 + [1 - h\,\lambda - \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} - h\,\mu_i\,]\,[n_i + 1]\ C \ \leq \\
& \quad \leq 1 + h\,\lambda\ C + [h\,\lambda + \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} + h\,\mu_i\,]\,[n_i + 1]\ C - h\,\mu_i\ C\ + \\
& \qquad + [1 - h\,\lambda - \sum_{j\neq i} h\,\mu_j\, 1_{\{n_j>0\}} - h\,\mu_i\,]\,[n_i + 1]\ C \ \leq \\
& \quad \leq [n_i + 1]\ C + [\,1 + h\,\lambda\, C - h\,\mu_i\, C\,] \leq \\
& \quad \leq [n_i + 1]\ C \qquad\qquad (A.5)
\end{aligned}
$$

□

Proof of Result 2

For reduction (21) all the steps performed for the upper bound model remain identical. In particular, let P_L denote the uniformized Markov one-step transition matrices corresponding to the continuous time process matrix Q^L. Lemma 3 can be applied up to relation (A.4). In this case the reduction (P_L matrix) would lead to

$$
(P_L - P\,)V^S(\mathbf{n}) = h\,\lambda\ 1_{\{n_i=U_i \text{ for some } i\}}\ [V^S(\mathbf{n}) - V^S(\mathbf{n}+\mathbf{1})]
$$

By (18) we can conclude:

$$
\begin{aligned}
0 \leq\ & [V^t(\mathbf{n}+\mathbf{1})-V^t(\mathbf{n})] = \\
& = [V^t(\mathbf{n}+\mathbf{1})-V^t(\mathbf{n}+\mathbf{1}-\mathbf{e}_1)] + [V^t(\mathbf{n}+\mathbf{1}-\mathbf{e}_1)-V^t(\mathbf{n})] \leq \\
& \leq (n_1+1)\,C + [V^t(\mathbf{n}+\mathbf{1}-\mathbf{e}_1)-V^t(\mathbf{n}+\mathbf{1}-\mathbf{e}_1-\mathbf{e}_2)] + \\
& \qquad + [V^t(\mathbf{n}+\mathbf{1}-\mathbf{e}_1-\mathbf{e}_2)-V^t(\mathbf{n})] \leq \\
& \leq (n_1+1)\,C + (n_2+1)\,C + \qquad \leq \\
& \leq \sum_i (n_i+1)\,C = (n+N)\,C
\end{aligned}
\tag{A.6}
$$

so that (19) here becomes:

$$0 \leq G-G^L \leq C \sum_{\mathbf{n}\ \tilde{E}^L} n\ \pi_L(\mathbf{n}) + C\ N\ \pi_L(\tilde{E}^L) = \Delta_L$$

where $\tilde{E}^L = \{\mathbf{n}\quad E^L |\ \mathbf{n} = U_i$ for some i$\}$. This proves (26). □

Proof of formula (27)

To prove (27) first note that clearly $\lambda_L \leq \lambda$. This can be proven either similarly to lemma 2 and a lower estimate or as in (18) of lemma 3 by using r($\mathbf{n}$)=$1_{\{n_i \geq U_i \text{ for some } i\}}$ or by using sample path arguments. By Little's law furthermore we have

$W_L = L_L/\lambda_L \qquad\qquad W = L/\lambda$

Hence, by applying (26) with $G_L = L_L$ and G= L we obtain

$W_L = L_L/\lambda_L \leq L_L/\lambda \leq L/\lambda \quad = W$

and

$W_L = L_L/\lambda_L \geq [L-\Delta_L]/\lambda_L \geq [L-\Delta_L]/\lambda \geq W - \Delta/\lambda$

from which (27) follows.

APPENDIX C

Derivation of formula (28)

In order to derive expression (28) we consider bounds on the probability π_L based on a system which is obtained by the original system by considering batch arrivals and without the fork and join nodes. Let π_2 denote the probability of this batch arrival system on state space E_2 superset of E^L and let P_2 denote the uniformized Markov one-step transition matrices corresponding to the continuous time process matrix Q_2. We prove the following lemma.

Lemma A.1

For any g $\mathcal{M} = \{\ g : E_2 \rightarrow \mathcal{R}\ |\ g(\mathbf{n}+\mathbf{e}_i) - g(\mathbf{n}) \geq 0$ for all i=1,…,N$\}$:

$$\sum_{\mathbf{n}} \pi_L(\mathbf{n})\, g(\mathbf{n}) \leq \sum_{\mathbf{n}} \pi_2(\mathbf{n})\, g(\mathbf{n})$$

In order to prove lemma A.1 we prove some preliminary results.

Lemma A.2

For any g M : $P_L^k(\mathbf{n})\, g(\mathbf{n}) \;\le\; P_2^k(\mathbf{n})\, g(\mathbf{n})$

Proof. By induction in k. For k=0 it holds as $P_L^0(\mathbf{n})\, g(\mathbf{n}) \;=\; P_2^0(\mathbf{n})\, g(\mathbf{n}) \;=\; g(\mathbf{n})$. Assume it holds for k=t. Then:

$$(P_2^{t+1}\, g - P_L^{t+1}\, g)(\mathbf{n}) = (P_2 - P_L)(P_2^t\, g)(\mathbf{n}) + P_L[(P_2^t - P_L^t)\, g\,](\mathbf{n}) \,.$$

Further for any f M we have:

$$(P_2 - P_L)\, f(\mathbf{n}) = \lambda\, 1_{\{n_i = U_i \;\text{ for some } i \ge 2\}}(f(\mathbf{n}+\mathbf{1}) - f(\mathbf{n})) \ge 0$$

The proof is thus completed by induction to t and lemma A.3 below. □

Lemma A.3

For any f M : $P_2^t\, f$ M (A.7)

Proof. Let f M . Then:

$$\begin{aligned} P_2\, f(\mathbf{n}+\mathbf{e}_i) - P_2\, f(\mathbf{n}) = {} & \lambda\, h\, [f(\mathbf{n}+\mathbf{1}+\mathbf{e}_i) - f(\mathbf{n}+\mathbf{1})] + \\ & + \sum_{j \ne i} \mu_j\, h\, [f(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_i) - f(\mathbf{n}-\mathbf{e}_j)] + \\ & + \mu_i\, h\, 1_{\{n_i > 0\}}[f(\mathbf{n}) - f(\mathbf{n}-\mathbf{e}_i)] + \\ & + \mu_i\, h\, 1_{\{n_i = 0\}}[f(\mathbf{n}) - f(\mathbf{n})] + \\ & + [1 - \lambda\, h - h \sum_j \mu_j\, 1_{\{n_j > 0\}} - \mu_i\, h]\, [f(\mathbf{n}+\mathbf{e}_i) - f(\mathbf{n})] \ge 0 \end{aligned}$$

Here the latter inequality follows by using that f M . Hence we have shown (A.7) for t=1 by

$$P_2\, f \quad M \;\text{ for any f }\; M \,. \tag{A.8}$$

Now for t>1 we proceed by induction as follows: suppose that (A.7) holds for t=k. Then for t=k+1 we have: $P_2^{t+1}\, f = P_2\, (P_2^t\, f)$ M as per induction hypothesis and (A.8). The induction completes the proof. □

Corollary A.1

With $\rho_1 = \lambda / \mu_1 < 1$:

$$\sum_{n_1 \ge U_1} \pi_L(\mathbf{n}) \;\le\; \sum_{n_1 \ge U_1} \pi_2(\mathbf{n}) \;\le\; \rho_1^{n_1} \tag{A.9}$$

$$\sum_{n_1 \ge U_1} n_1\, \pi_L(\mathbf{n}) \;\le\; \sum_{n_1 \ge U_1} n_1\, \pi_2(\mathbf{n}) \;\le\; \{U_1\, (\rho_1^{U_1} - \rho_1^{U_1+1}) + \rho_1^{U_1+1}\} \;/\; (1 - \rho_1) \tag{A.10}$$

Proof. First by taking **n**=(0,…,0) and $k \to \infty$ in lemma A.1 we get

$$\sum_{\mathbf{n}} \pi_L(\mathbf{n})\, g(\mathbf{n}) \;\le\; \sum_{\mathbf{n}} \pi_2(\mathbf{n})\, g(\mathbf{n}) \qquad \text{for any g } \; M \tag{A.11}$$

Now take

$$\begin{aligned} g(\mathbf{n}) &= 1_{\{n_1 \geq U_1\}} && \text{for (A.9)} \\ g(\mathbf{n}) &= n_1\, 1_{\{n_1 \geq U_1\}} && \text{for (A.10)} \end{aligned} \qquad \text{(A.12)}$$

to prove the first inequalities. To prove the second inequalities in (A.9) and (A.10) note that the summation in (A.11) is over all states **n** while the function g as per (A.12) only concerns component n_i. We are thus calculating the expected values of the function g for just the first queue. This is clearly equal to just that of an isolated M/M/1 queue as there is no other dependence between the queues than by a common arrival. □