Exact analysis of a class of GI/G/1-type performability models[1]

Alma Riska
Member IEEE
Evgenia Smirni
Member IEEE Computer Society
Gianfranco Ciardo
Senior Member IEEE

Keywords Modeling (performability analysis); Markov processes (GI/G/1-type, M/G/1-type, and GI/M/1-type processes; matrix analytic techniques; stochastic complementation).

Summary and Conclusions We present an exact decomposition algorithm for the analysis of Markov chains with a GI/G/1-type repetitive structure. Such processes exhibit *both* M/G/1-type and GI/M/1-type patterns and cannot be solved using existing techniques. Markov chains with a GI/G/1 pattern result when modeling open systems that accept jobs from multiple exogenous sources and are subject to failures and repairs: a single failure can empty the system of jobs, while a single batch arrival can add many jobs to the system. Our method provides exact computation of the stationary probabilities, which can then be used to obtain performance measures such as the average queue length or any of its higher moments, as well as the probability of the system being in various failure states, thus performability measures. We formulate the conditions under which our approach is applicable and illustrate it via the performability analysis of a parallel computer system.

1 Introduction

Acronyms

GI/G/1	single server queue with independent general arrivals and general service
M/G/1	single server queue with Markovian arrivals and general service
GI/M/1	single server queue with independent general arrivals and Markovian service
CTMC	Continuous Time Markov Chain
QBD	Quasi-Birth-Death process
ATM	Asyncronous Transfer Mode

Computer systems have always been prone to failures, making the study of their reliability an important factor in assessing their quality and usefulness. Nowadays, however, such systems are increasingly built to withstand the failure of some of their components without resulting in a total loss of functionality. Such *degradable performance* is highly desirable especially in harsh environments where prompt repairs might not be possible or in life-critical situations where service interruptions can be very costly. The resulting combined study of performance in the presence of faults is called *performability* [14].

[1]This work has been partially supported by the National Science Foundation under grants EIA-9974992, CCR-0098278, ACI-0090221, and ACI-0203971, and by the National Aeronautics and Space Administration under NASA Grant NAG-1-2168.

Studying both performance and reliability aspects in a single model, however, results in a larger and more complex model whose numerical solution is accordingly more difficult. During the past two decades, significant effort has gone into the development of modeling techniques that can address these types of studies [15]. In many cases, M/G/1-type or GI/M/1-type Markov chains have been shown to be appropriate tools. Markov chains that model such processes have an infinite state space with a finite one-dimensional repetitive pattern (we assume continuous-time Markov chains, or CTMCs, but our discussion applies just as well to the discrete-time case).

In this paper, we consider the solution of a class of CTMCs that show a GI/G/1-type repetitive structure, that is, they exhibit *both* M/G/1-type and GI/M/1-type patterns, and cannot be solved by existing techniques [5]. Such chains occur when modeling open systems that accept customers from a variety of exogenous sources (thus the existence of bulk arrivals) and are subject to failures and repairs (thus the system may lose all of its customers when a catastrophic failure occurs, or, if the failure is non-catastrophic, only parts of the system become non-operational and the ability to deliver service is reduced). We stress that our technique applies equally well to situations where the reliability aspects focus on hardware [7], software [10], or both [6, 18, 24].

1.1 Terminology

The state space $\mathcal{S}$ of a GI/G/1-type CTMC can be partitioned into a finite "boundary" set $\mathcal{S}^{(0)} = \{s_1^{(0)}, \ldots, s_m^{(0)}\}$ and a countably infinite sequence of finite "level" sets $\mathcal{S}^{(j)} = \{s_1^{(j)}, \ldots, s_n^{(j)}\}$, for $j \geq 1$. The infinitesimal generator matrix can accordingly be block-partitioned as

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \widehat{\mathbf{F}}^{(4)} & \cdots \\ \widehat{\mathbf{B}}^{(1)} & \mathbf{L}^{(1)} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \cdots \\ \widehat{\mathbf{B}}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \widehat{\mathbf{B}}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \widehat{\mathbf{B}}^{(4)} & \mathbf{B}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1}$$

("L", "F", and "B" stand for "local", 'forward", and "backward", respectively, and a "^" indicates matrices related to $\mathcal{S}^{(0)}$).

Since $\mathit{RowSum}(\mathbf{Q}) = \mathbf{0}$ and only the diagonal of $\mathbf{Q}$ can contain negative entries, the infinite sets of matrices $\{\widehat{\mathbf{F}}^{(j)} : j \geq 1\}$ and $\{\mathbf{F}^{(j)} : j \geq 1\}$ must be summable. For presentation simplicity, we assume that these matrices have finite representations obeying the geometric expressions $\widehat{\mathbf{F}}^{(j)} = \widehat{\mathbf{A}}^{j-1}\widehat{\mathbf{F}}$ and $\mathbf{F}^{(j)} = \mathbf{A}^{j-1}\mathbf{F}$, for $j \geq 1$, where the spectral radii of $\widehat{\mathbf{A}}$ and $\mathbf{A}$ are strictly less than one, to ensure that the infinite sums $\sum_{j=0}^{\infty} \widehat{\mathbf{A}}^j = (\mathbf{I} - \widehat{\mathbf{A}})^{-1}$ and $\sum_{j=0}^{\infty} \mathbf{A}^j = (\mathbf{I} - \mathbf{A})^{-1}$ exist. Since the same local and forward blocks appear from the third row block on, $\mathit{RowSum}(\widehat{\mathbf{B}}^{(j)}) + \mathit{RowSum}(\mathbf{B}^{(j-1)} + \cdots + \mathbf{B}^{(1)})$ must have the same value for all $j \geq 2$. This implies that $\widehat{\mathbf{B}}^{(j)}$ is increasingly smaller, unless of course $\mathbf{B}^{(j)}$ is zero from some j on, and that the infinite set $\{\mathbf{B}^{(j)} : j \geq 1\}$ is also summable, since $\mathit{RowSum}(\mathbf{B}^{(1)} + \cdots + \mathbf{B}^{(j)})$, for any finite j, is bounded by $\mathit{RowSum}(-\mathbf{L} - (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F})$. However, $\mathbf{L}^{(1)}$ can differ from $\mathbf{L}$ in the diagonal, hence $\mathit{RowSum}(\widehat{\mathbf{B}}^{(1)})$ might be different from $\mathit{RowSum}(\widehat{\mathbf{B}}^{(j)}) + \mathit{RowSum}(\mathbf{B}^{(j-1)} + \cdots + \mathbf{B}^{(1)})$, for $j \geq 2$.

We are interested in the computation of the stationary probability vector $\boldsymbol{\pi}$ solution of $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, where $\boldsymbol{\pi}$ can be partitioned into $\boldsymbol{\pi}^{(0)} \in \mathbb{R}^m$ and $\boldsymbol{\pi}^{(j)} \in \mathbb{R}^n$, for $j \geq 1$. Since $\boldsymbol{\pi}$ is infinite, in practice we compute $\boldsymbol{\pi}^{(j)}$ only up to a sufficiently large j, or an aggregate measure of the form $\sum_{j=0}^{\infty} \boldsymbol{\pi}^{(j)} \boldsymbol{\rho}^{(j)T}$, where $\boldsymbol{\rho}^{(j)}$ is a vector expressing the *reward rates* for the states in $\mathcal{S}^{(j)}$.

While GI/G/1-type CTMCs do not have a known solution algorithm in general [5], two special cases do: the GI/M/1-type CTMCs (where $\widehat{\mathbf{F}}^{(j)}$ and $\mathbf{F}^{(j)}$ are zero for $j \geq 2$, that is, only forward jumps to the next level are allowed) and the M/G/1-type CTMCs (the converse: $\widehat{\mathbf{B}}^{(j)}$ and $\mathbf{B}^{(j)}$ are zero for $j \geq 2$, only backward jumps to the previous level are allowed). Neuts proposed a *matrix geometric* solution for the former [16] and a *matrix analythic* solution for the latter [17]. The intersection of these two cases, where $\mathbf{Q}$ is block-tridiagonal, is the class of quasi-birth-death (QBD) processes, which can in principle be solved by either method (the matrix geometric solution is preferred because it is simpler, more widely-known, and at least as efficient). In practice, such CTMCs often arise when observing an open system with a single infinite-waiting-room queue at either the arrival or the service completion times.

The solution of M/G/1 or GI/M/1-type CTMCs requires an auxiliary matrix, $\mathbf{G}$ or $\mathbf{R}$, respectively, typically computed through iterative methods [16, 17]. Several algorithms have been proposed for the efficient computation of $\mathbf{R}$ [8] or $\mathbf{G}$ [12]. See also [9] for a discussion of the fundamental aspects of matrix methods and their use for the solution of QBDs.

1.2 Our contribution

In this work, we propose a decomposition approach that extends both the applicability and the efficiency of matrix geometric and matrix analytic methods through an intelligent *partitioning* of the repetitive portion of the state space into subsets, according to its connectivity. This partitioning allows us to define smaller CTMCs with a solvable structure, obtained from the original CTMC through the use of stochastic complementation, an exact decomposition technique that can be used to study the conditional stationary behavior in individual portions of a larger CTMC. To obtain the solution of the original process, the results of the analysis of these portions are coupled back together [13].

More specifically, the main contribution of our work is a decomposition algorithm that can be used for the performability analysis of complex systems modeled as GI/G/1-type processes. Since such processes can capture both the normal behavior (arrivals and service) and the reliability aspects (failures and repairs) of system operation, they provide the necessary means to model a system's performability. In particular, GI/G/1-type processes are the tool of choice if one needs to analyze the user-perceived performance of a computer system. In such cases, the normal operation of the system corresponds to the M/G/1-type pattern of the GI/G/1-type model, while software or hardware failures correspond to the GI/M/1-type pattern of the GI/G/1-type model. Our method exploits this structure by reflecting a key performance component of the state (the number of customers in the system) in the level j of the sets $\mathcal{S}^{(j)}$, while the remaining component (the status of the service and repair processes, and the inter-arrival and service distributions, if not memoryless) are captured *within* the states of a level set. Only the latter components truly affect the solution complexity of our approach.

With regard to degradable behavior, we observe that the *reward structure* assigned to the states of CTMC is very flexible, since it allows to define the reward, or "usefulness", of a state $s_i^{(j)}$ in terms of both its index i, describing the failure status of each component, and of the level j, describing the present workload in the system.

Since general GI/G/1-type models do not have a known exact solution [5], they are usually studied through approximations based on QBD processes or complex eigenvalue methods [4]. The QBD approximation essentially truncates the "arbitrary forward and backward jump" behavior, but the resulting process is orders of magnitude larger than with the approach we propose, thus its solution is much more computationally expensive. Our methodology based on aggregation/decomposition techniques is applicable to an important subset of GI/G/1-type processes and provides instead *exact* solutions. Since we approach the problem in a divide-and-conquer way, we ensure that the subproblems are smaller in size and computationally more efficient to solve than the approximated original problem. Note that we do not consider the identification of the appropriate partitioning of the state space required to apply our method; this is a graph partitioning problem and is the subject of future work.

In the following, we review the necessary background required to introduce our algorithm: matrix analytic methods and stochastic complementation (Sect. 2). Then, we formally present the proposed algorithm (Sect. 3) and we give an example of a system subject to failures and repairs that can be analyzed with our algorithm (Sect. 4).

Notation

$\mathcal{A}$	calligraphic letters indicate sets
$\mathbf{a}$, $\boldsymbol{\alpha}$	lower case boldface Roman or Greek letters indicate row vectors
$\mathbf{A}$	upper case boldface Roman letters indicate matrices
$\mathcal{A}^{(1)}$, $\mathbf{A}_{\mathcal{A}_1}$	superscripts in parentheses or subscripts indicate family of related entities
$\mathbf{a}[1]$, $\mathbf{A}[1,2]$	square brackets indicate vector and matrix elements
$\mathbf{a}[\mathcal{A}]$, $\mathbf{A}[\mathcal{A},\mathcal{B}]$	sets of indices within square brackets indicate subvectors or submatrices
$\mathbf{1}$	row vector of 1's, of the appropriate dimension
$\mathbf{0}$	row vector or a matrix of 0's, of the appropriate dimension
$Norm(\cdot)$	matrix equal to its argument with all nonzero rows normalized to sum to one
$RowSum(\cdot)$	square diagonal matrix whose entry in position (r,r) is the sum of the entries on the r^{th} row of the argument (which can be a rectangular matrix)

2 Background

We briefly review the basic terminology used to describe the processes we consider, as well as previous results on the solution of M/G/1 and GI/M/1-type processes. We also present some results on stochastic complementation needed to combine the results from the study of individual portions of the overall CTMCs.

2.1 M/G/1-type processes

The infinitesimal generator matrix $\mathbf{Q}$ of a CTMC having an M/G/1-type structure, as defined by Neuts [16], is a special case of Eq. 1:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \widehat{\mathbf{F}}^{(4)} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L}^{(1)} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Various recursive algorithms for the stationary solution of such chains exist [5, 12, 17]. Here, we outline one that provides a stable calculation for the values of $\boldsymbol{\pi}^{(j)}$. Using Ramaswami's recursive formula [19] we have:

$$\forall j \geq 1, \quad \boldsymbol{\pi}^{(j)} = -\left(\boldsymbol{\pi}^{(0)}\widehat{\mathbf{S}}^{(j)} + \sum_{l=1}^{j-1} \boldsymbol{\pi}^{(l)}\mathbf{S}^{(j-l)}\right) \mathbf{S}^{(0)^{-1}},$$

where $\widehat{\mathbf{S}}^{(j)} = \sum_{l=j}^{\infty} \widehat{\mathbf{F}}^{(l)}\mathbf{G}^{l-j}$, for $j \geq 1$, $\mathbf{S}^{(j)} = \sum_{l=j}^{\infty} \mathbf{F}^{(l)}\mathbf{G}^{l-l}$, for $j \geq 0$ (letting $\mathbf{F}^{(0)} \equiv \mathbf{L}$), and $\mathbf{G}$ can be computed as the solution of the matrix equation[2] $\mathbf{0} = \mathbf{B} + \mathbf{L}\mathbf{G} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}\mathbf{G}^{j+1}$. Several iterative algorithms for the computation of $\mathbf{G}$ exist [5, 17], a particularly efficient one is based on Toeplitz matrices [12]. See [20] for cases where $\mathbf{G}$ can be explicitly defined and does not require any calculation.

Given the above definition of $\boldsymbol{\pi}^{(j)}$ and the normalization condition, $\boldsymbol{\pi}^{(0)}$ can be calculated as the unique solution of the linear system in m variables

$$\boldsymbol{\pi}^{(0)} \left[\widehat{\mathbf{L}}^{(0)\diamond} - \widehat{\mathbf{S}}^{(1)}\mathbf{S}^{(0)^{-1}}\widehat{\mathbf{B}}^{\diamond} \;\middle|\; \mathbf{1}^T - \sum_{j=1}^{\infty} \widehat{\mathbf{S}}^{(j)} \left(\sum_{j=0}^{\infty} \mathbf{S}^{(j)}\right)^{-1} \mathbf{1}^T \right] = [\mathbf{0} \mid 1],$$

where the symbol "$\diamond$" indicates that we (consistently) discard any one column of the corresponding matrix, since we added a column representing the normalization condition. Once $\boldsymbol{\pi}^{(0)}$ is known, we can iteratively compute $\boldsymbol{\pi}^{(j)}$ for $j = 1, 2, \ldots$, stopping when the accumulated probability mass is close to one.

2.2 GI/M/1-type and QBD processes

The infinitesimal generator of a CTMC of the GI/M/1-type, also defined by Neuts in [16], is another special case of the structure of Eq. 1:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}}^{(1)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F} & \cdots \\ \widehat{\mathbf{B}}^{(4)} & \mathbf{B}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

[2] $\mathbf{G}[k,l]$ represents the probability of first entering $\mathcal{S}^{(j-1)}$ through state $s_l^{(j-1)}$, starting from $\mathcal{S}^{(j)}$ in state $s_k^{(j)}$.

A further special case of the above structure is that of a QBD CTMC, whose infinitesimal generator can be block-partitioned as:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

While the QBD case falls under both the M/G/1 and the GI/M/1-type case, it is most commonly associated with GI/M/1-type matrices because both can be solved using the well-known matrix geometric approach [16]:

$$\forall j \geq 1, \ \boldsymbol{\pi}^{(j)} = \boldsymbol{\pi}^{(1)}\mathbf{R}^{j-1}, \tag{2}$$

where, in the GI/M/1-type case, $\mathbf{R}$ is the solution of the matrix equation

$$\mathbf{F} + \mathbf{RL} + \sum_{j=1}^{\infty} \mathbf{R}^{j+1}\mathbf{B}^{(j)} = \mathbf{0} \tag{3}$$

and can be computed using iterative numerical algorithms[3]. Then, together with the normalization condition $\boldsymbol{\pi}^{(0)}\mathbf{1}^T + \boldsymbol{\pi}^{(1)}\sum_{j=1}^{\infty}\mathbf{R}^{j-1}\mathbf{1}^T = 1$, which we can rewrite as $\boldsymbol{\pi}^{(0)}\mathbf{1}^T + \boldsymbol{\pi}^{(1)}(\mathbf{I} - \mathbf{R})^{-1}\mathbf{1}^T = 1$, we obtain the linear system in $m + n$ variables

$$[\boldsymbol{\pi}^{(0)}|\boldsymbol{\pi}^{(1)}] \left[\begin{array}{c|c|c} \mathbf{1}^T & \widehat{\mathbf{L}}^{(0)\diamond} & \widehat{\mathbf{F}}^{(1)} \\ \hline (\mathbf{I}-\mathbf{R})^{-1}\mathbf{1}^T & \left(\sum_{j=1}^{\infty}\mathbf{R}^{j-1}\widehat{\mathbf{B}}^{(j)\diamond}\right) & \mathbf{L} + \sum_{j=1}^{\infty}\mathbf{R}^{j}\mathbf{B}^{(j)} \end{array}\right] = [1|\mathbf{0}] \tag{4}$$

which yields a unique solution for $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$. For $j \geq 2$, $\boldsymbol{\pi}^{(j)}$ can be obtained numerically from Eq. 2, but many useful performance metrics such as expected system utilization, throughput, or queue length can be computed exactly in explicit form using $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\mathbf{R}$ alone.

For QBD processes, Eq. 3 reduces to the quadratic matrix equation $\mathbf{F} + \mathbf{RL} + \mathbf{R}^2\mathbf{B} = \mathbf{0}$.

2.3 Stochastic complementation

In this section, we briefly outline the concept of stochastic complementation [13] and focus on results needed to derive our method. While [13] introduces the concept of stochastic complementation for discrete-time Markov chains with finite state spaces we define it instead for the infinite case, a straightforward extension, and state the results in terms of CTMCs.

Partition the state space $\mathcal{S}$ of an ergodic CTMC with infinitesimal generator matrix $\mathbf{Q}$ and stationary probability vector $\boldsymbol{\pi}$, satisfying $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, into two disjoint subsets, $\mathcal{A}$ and $\overline{\mathcal{A}}$.

[3] $\mathbf{R}[k,l]$ represents the expected time spent in state $s_l^{(j+1)}$ after a transition out of state $s_k^{(j)}$ and before entering a state in $\mathcal{S}^{(j)}$, measured using $-1/\mathbf{L}[k,k]$, the holding time in state $s_k^{(j)}$, as the time unit.

Definition 1 **[13] (Stochastic complement)** The stochastic complement of $\mathcal{A}$ is

$$\mathbf{A} = \mathbf{Q}[\mathcal{A},\mathcal{A}] + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}](-\mathbf{Q}[\overline{\mathcal{A}},\overline{\mathcal{A}}])^{-1}\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}], \tag{5}$$

where $(-\mathbf{Q}[\overline{\mathcal{A}},\overline{\mathcal{A}}])^{-1}[r,c]$ represents the mean time spent in state $c \in \overline{\mathcal{A}}$, starting from state $r \in \overline{\mathcal{A}}$, before reaching any state in $\mathcal{A}$, and $((-\mathbf{Q}[\overline{\mathcal{A}},\overline{\mathcal{A}}])^{-1}\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}])[r,c']$ represents the probability that, starting from $r \in \overline{\mathcal{A}}$, we enter $\mathcal{A}$ through state c'. □

The stochastic complement $\mathbf{A}$ is the infinitesimal generator of a new CTMC which mimics the original CTMC but "skips over" states in $\overline{\mathcal{A}}$. The following theorem formalizes this concept.

Theorem 1 **[13]** The stochastic complement $\mathbf{A}$ of $\mathcal{A}$ is an infinitesimal generator, and is irreducible if $\mathbf{Q}$ is. If $\boldsymbol{\alpha}$ is its stationary probability vector satisfying $\boldsymbol{\alpha}\mathbf{A} = \mathbf{0}$, then $\boldsymbol{\alpha} = \mathit{Norm}(\boldsymbol{\pi}[\mathcal{A}])$. □

In other words, $\boldsymbol{\alpha}$ gives the individual probability of being in each state of $\mathcal{A}$ for the original CTMC, conditioned on being in $\mathcal{A}$. This implies that the stationary probability distribution $\boldsymbol{\alpha}$ of the stochastic complement differs from the corresponding portion of the stationary distribution of the original CTMC $\boldsymbol{\pi}[\mathcal{A}]$ only by a constant $a = \boldsymbol{\pi}[\mathcal{A}]\mathbf{1}^T$, which represents the probability of being in $\mathcal{A}$ in the original CTMC. The value a is known as the *coupling factor* of the stochastic complement.

Stochastic complementation can be used as a divide-and-conquer strategy: to this end, computing the stationary distribution of each stochastic complement must be easier than computing the stationary distribution of the original CTMC directly. Since the inverse $(-\mathbf{Q}[\overline{\mathcal{A}},\overline{\mathcal{A}}])^{-1}$ is often a full matrix, its computation can be costly. However, there are cases where we can take advantage of the special structure of the CTMC and avoid this computation. To consider these cases, rewrite the definition of stochastic complement in Eq. 5 as

$$\mathbf{A} = \mathbf{Q}[\mathcal{A},\mathcal{A}] + \mathit{RowSum}(\mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}])\,\mathbf{Z} \qquad \text{where} \qquad \mathbf{Z} = \mathit{Norm}(\mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}])\;(-\mathbf{Q}[\overline{\mathcal{A}},\overline{\mathcal{A}}])^{-1}\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]. \tag{6}$$

The r^{th} diagonal element of $\mathit{RowSum}(\mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}])$ represents the rate at which the set $\mathcal{A}$ is left from its r^{th} state to reach any of the states in $\overline{\mathcal{A}}$, while the r^{th} row of $\mathbf{Z}$, which sums to one, specifies how this rate should be redistributed over the states in $\mathcal{A}$ when the process eventually reenters it.

Lemma 1 **(Single entry)** If $\mathcal{A}$ can be entered from $\overline{\mathcal{A}}$ only through a single state $c \in \mathcal{A}$, the matrix $\mathbf{Z}$ defined in Eq. 6 is trivially computable: it is a matrix of zeros except for its c^{th} column, which contains all ones. □

Definition 2 **(Pseudo-stochastic complement)** The pseudo-stochastic complement of $\mathcal{A}$ is

$$\mathbf{A} = \mathbf{Q}[\mathcal{A},\mathcal{A}] + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\,\mathbf{1}^T\,\mathit{Norm}(\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]),$$

where $\boldsymbol{\pi}$ is the stationary distribution for the states of the original CTMC.

Theorem 2 The pseudo-stochastic complement $\mathbf{A}$ of $\mathcal{A}$ is an infinitesimal generator and is irreducible if $\mathbf{Q}$ is. If $\boldsymbol{\alpha}$ is its stationary probability vector satisfying $\boldsymbol{\alpha}\mathbf{A} = \mathbf{0}$, then $\boldsymbol{\alpha} = Norm(\boldsymbol{\pi}[\mathcal{A}])$. **Proof:** See Appendix A. □

Comparing the definitions of stochastic and pseudo-stochastic complements, we see that the former is naturally based on considering all paths starting and ending in $\mathcal{A}$ (given the interpretation of the inverse as an infinite sum over all possible paths), while the latter makes use of the conditional stationary probability vector for $\overline{\mathcal{A}}$. What is interesting, though, is that, even if the two complements are described by different matrices, Theorems 1 and 2 imply that they *have the same stationary probability vector* $\boldsymbol{\alpha}$. The intuition behind this property is given by the stochastic meaning of the matrix $-\mathbf{Q}[\overline{\mathcal{A}}, \overline{\mathcal{A}}]^{-1}$ used in the formulation of the stochastic complement. Its entry in row r and column c represents the expected amount of time spent in state $c \in \overline{\mathcal{A}}$ starting from state $r \in \overline{\mathcal{A}}$, before entering $\mathcal{A}$, a quantity that is of course related to the stationary probability vector $\boldsymbol{\pi}[\overline{\mathcal{A}}]$ used in the formulation of the pseudo-stochastic complement.

3 Our approach

The interactions between states of a CTMC with infinitesimal generator $\mathbf{Q}$ given by Eq.(1) exhibit the patterns for both GI/M/1-type and M/G/1-type processes, thus they are more general than either. Our approach is based on decomposition: informally, we relegate the GI/M/1-type and the M/G/1-type behaviors into different CTMCs which not only must be individually solvable by known methods, but also, just as important, must provide results that are meaningful to the original process. We stress that we can do this even if, in the original CTMC, there are states with arbitrarily long forward and backward transitions to other states, as long as $\mathbf{Q}$ satisfies the conditions outlined in Section 3.1. At a high level, these are our solution steps:

1. We determine a "gate" state g in $\mathcal{S}^{(0)}$, a covering $\mathcal{U}^{(0)}$ (for "upper") and $\mathcal{L}^{(0)}$ (for "lower") of $\mathcal{S}^{(0)}$, with $\mathcal{U}^{(0)} \cup \mathcal{L}^{(0)} = \mathcal{S}^{(0)}$ and $\mathcal{U}^{(0)} \cap \mathcal{L}^{(0)} = \{g\}$, and a partition of each level set $\mathcal{S}^{(j)}$ into two disjoint sets $\mathcal{U}^{(j)}$ and $\mathcal{L}^{(j)}$. These allow us to capture the GI/M/1-type behavior of $\mathbf{Q}$ with an "upper CTMC" having state space $\mathcal{U} = \bigcup_{j=0}^{\infty} \mathcal{U}^{(j)}$, and the M/G/1-type behavior of $\mathbf{Q}$ with a "lower CTMC" having state space $\mathcal{L} = \bigcup_{j=0}^{\infty} \mathcal{L}^{(j)}$.

2. We define the rates for the upper CTMC using stochastic complementation in the special setting of Lemma 1. Then, we solve it using the matrix geometric method [17], obtaining both the matrix $\mathbf{R}$ and the conditional stationary probabilities of the states in $\mathcal{U}$ for the original CTMC.

3. Using the matrix $\mathbf{R}$ just computed, we apply pseudo-stochastic complementation to define the rates for the lower CTMC. Then, we solve it using the matrix analytic method [12, 17].

4. Finally, we "couple" the two solutions and obtain an expression for the stationary probability vector of the original CTMC. Unlike standard applications of stochastic complementation, where the state spaces of the stochastic complements being solved

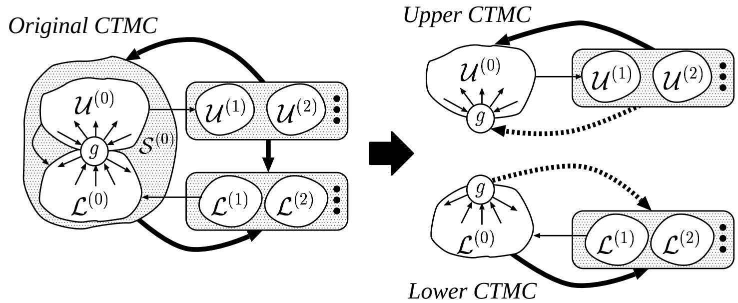

Figure 1: The overall idea of our approach.

are normally disjoint, our upper and lower CTMCs share the state g, and this greatly simplifies the coupling.

The overall idea is illustrated in Fig. 1. We now describe in detail the key steps of our approach.

3.1 Determining the upper and lower sets

The partition of each $\mathcal{S}^{(j)}$, $j \geq 1$, must be consistent with a partition of the set of repetitive state indices $\{1, \ldots, n\}$ into $\mathcal{N}_u$ and $\mathcal{N}_l$, that is: $\mathcal{U}^{(j)} = \left\{ s_i^{(j)} : i \in \mathcal{N}_u \right\}$ and $\mathcal{L}^{(j)} = \left\{ s_i^{(j)} : i \in \mathcal{N}_l \right\}$. Recall that g belongs to both $\mathcal{U}^{(0)}$ and $\mathcal{L}^{(0)}$, hence to both $\mathcal{U}$ and $\mathcal{L}$. We then use the notation $\not{\mathcal{U}}^{(0)}$, $\not{\mathcal{L}}^{(0)}$, $\not{\mathcal{U}}$, and $\not{\mathcal{L}}$ to indicate the analogous sets without g, that is, $\not{\mathcal{U}}^{(0)} = \mathcal{U}^{(0)} \setminus \{g\}$, $\not{\mathcal{L}}^{(0)} = \mathcal{L}^{(0)} \setminus \{g\}$, $\not{\mathcal{U}} = \mathcal{U} \setminus \{g\}$, and $\not{\mathcal{L}} = \mathcal{L} \setminus \{g\}$. The covering of $\mathcal{S}^{(0)}$, or equivalently its partition into $\not{\mathcal{U}}^{(0)}$, $\not{\mathcal{L}}^{(0)}$, and $\{g\}$, is completely determined by the choice of g: $\not{\mathcal{L}}^{(0)}$ is the set of states in $\mathcal{S}^{(0)}$ that can be reached from $\mathcal{L}^{(1)}$ without visiting g, while $\not{\mathcal{U}}^{(0)} = \mathcal{S}^{(0)} \setminus \mathcal{L}^{(0)}$ (there are no constraints on transitions from $\mathcal{U}$ to $\not{\mathcal{L}}^{(0)}$). The definition of these two partitions, hence of the gate g, must satisfy the following conditions (see Fig. 2):

- Forward transitions are allowed from $\mathcal{U}^{(j)}$ to $\mathcal{U}^{(j+1)}$, $j \geq 0$. Backward transitions are allowed from $\mathcal{U}^{(j)}$, $j \geq 1$, to any lower level $\mathcal{U}^{(k)}$, $k < j$, and to $\mathcal{S}^{(0)}$. Note that, in Fig. 2, the set $\mathcal{S}^{(0)}$, which is exactly equal to $\mathcal{U}^{(0)} \cup \mathcal{L}^{(0)}$, is represented separately from them, in gray, simply to stress the fact that arcs reaching $\mathcal{S}^{(0)}$ or leaving from it can reach or leave any state in $\mathcal{U}^{(0)}$ or $\mathcal{L}^{(0)}$.

- Forward transitions are allowed from $\mathcal{S}^{(0)}$ to any $\mathcal{L}^{(j)}$ and from any $\mathcal{L}^{(j)}$ toward any $\mathcal{L}^{(k)}$, $k > j$. Backward transitions are allowed from $\mathcal{L}^{(j)}$ to $\mathcal{L}^{(j-1)}$, $j \geq 1$.

- Local transitions (not shown) are allowed within each $\mathcal{U}^{(j)}$ or $\mathcal{L}^{(j)}$, $j \geq 0$.

- Transitions from $\mathcal{U}^{(j)}$ to any $\mathcal{L}^{(k)}$, $k \geq 1$ are allowed.

- Transitions from $\not{\mathcal{U}}^{(0)}$ to $\not{\mathcal{L}}^{(0)}$ are allowed.

- Transitions from g are allowed to any state except those in $\bigcup_{j=2}^{\infty} \mathcal{U}^{(j)}$.

- No other transition is allowed. In particular, there is no direct transitions from $\not{\mathcal{L}}$ to $\mathcal{U}$.

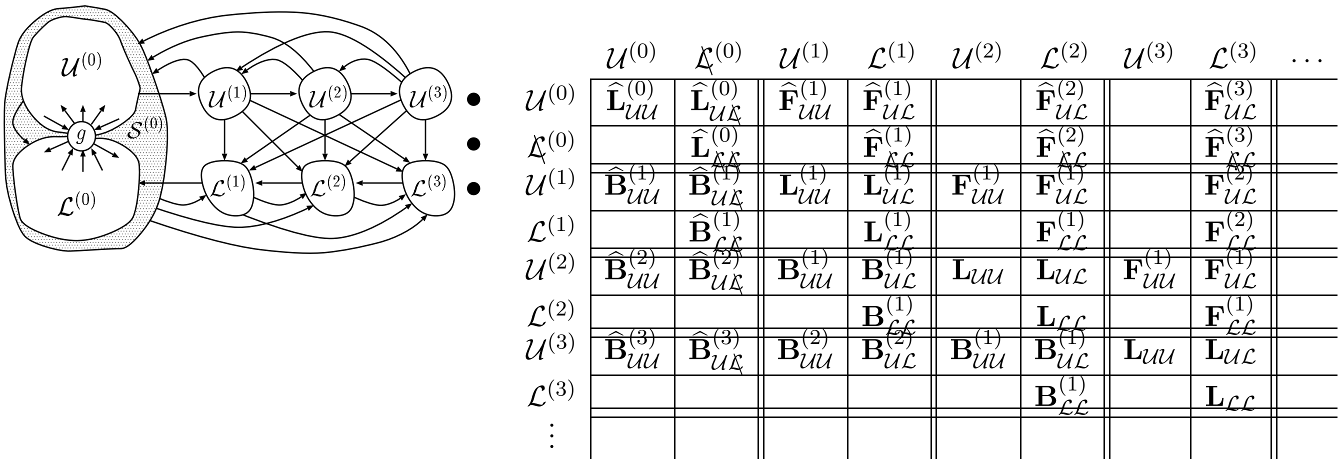

Figure 2: Two-level gated interaction and the corresponding nonzero pattern in matrix $\mathbf{Q}$.

These requirements imply that any path from $\not{\mathcal{L}}$ to $\mathcal{U}$ must visit g. Furthermore, they imply not only that $\mathbf{Q}[\mathcal{U},\mathcal{U}]$ has a GI/M/1-type structure (possibly QBD) and $\mathbf{Q}[\mathcal{L},\mathcal{L}]$ has a M/G/1-type structure (possibly QBD), but also that their structure is preserved even after applying stochastic complementation to them, as discussed in the following sections.

In practice, a suitable gate might exist in $\mathcal{S}^{(1)}$ but not in $\mathcal{S}^{(0)}$; in this case, we simply redefine a new $\mathcal{S}^{(0)}$ as the union of the original sets $\mathcal{S}^{(0)}$ and $\mathcal{S}^{(1)}$ and re-index all levels accordingly. Also, note that this partition is not uniquely defined in general. For example, any infinite subchain having a strictly QBD pattern can be considered as a M/G/1-type or a GI/M/1-type portion, hence its states could be considered either as belonging to either $\mathcal{U}$ or $\mathcal{L}$. In such cases, we will consider them to be in $\mathcal{L}$. If we cannot find a partition satisfying the required conditions, our approach in not applicable, and we must stop. Otherwise, we can proceed with the following steps.

3.2 Defining and solving the upper CTMC

The upper CTMC is obtained as the stochastic complement of $\mathcal{U}$. To compute its rates efficiently through stochastic complementation, the single-entry condition must be satisfied (Lemma 1): this is one the reasons for requiring the existence of the gate g which all paths from $\not{\mathcal{L}}$ to $\mathcal{U}$ must visit (the other being that the application of pseudo-stochastic complementation for the lower CTMC is also simplified). The blocks composing $\mathbf{Q}$ in Eq. 1 can be block-partitioned according to the upper and lower sets, as shown in Fig. 2 on the right (where non-zero blocks are labeled with a shorthand notation).

When we apply stochastic complementation to $\mathcal{U}$, any transition from $\mathcal{U}$ to $\not{\mathcal{L}}$ is simply rerouted to the gate state g. The infinitesimal generator of this new CTMC is then

$$\mathbf{Q}_{\mathcal{U}} = \begin{bmatrix} \widetilde{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}} & \widehat{\mathbf{F}}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \widetilde{\mathbf{B}}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{L}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{F}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{0} & \mathbf{0} & \cdots \\ \widetilde{\mathbf{B}}^{(2)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{L}_{\mathcal{U}\mathcal{U}} & \mathbf{F}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{0} & \cdots \\ \widetilde{\mathbf{B}}^{(3)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(2)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{L}_{\mathcal{U}\mathcal{U}} & \mathbf{F}^{(1)}_{\mathcal{U}\mathcal{U}} & \cdots \\ \widetilde{\mathbf{B}}^{(4)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(3)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(2)}_{\mathcal{U}\mathcal{U}} & \mathbf{B}^{(1)}_{\mathcal{U}\mathcal{U}} & \mathbf{L}_{\mathcal{U}\mathcal{U}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where the block $\widetilde{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}}$ is the same as $\widehat{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}}$ except in column g, which has also the contribution obtained by applying Eq. 6 and Lemma 1,

$$\widetilde{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}} = \widehat{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}} + RowSum(\widehat{\mathbf{L}}^{(0)}_{\mathcal{U}\not\mathcal{L}})\mathbf{1}^T\mathbf{e}_g + RowSum\left(\sum_{l=1}^{\infty}\widehat{\mathbf{F}}^{(l)}_{\mathcal{U}\mathcal{L}}\right)\mathbf{1}^T\mathbf{e}_g$$

($\mathbf{e}_g$ is a row vector of zeros except for a 1 in position g),

$$\widetilde{\mathbf{B}}^{(1)}_{\mathcal{U}\mathcal{U}} = \widehat{\mathbf{B}}^{(1)}_{\mathcal{U}\mathcal{U}} + RowSum(\widehat{\mathbf{B}}^{(1)}_{\mathcal{U}\not\mathcal{L}})\mathbf{1}^T\mathbf{e}_g + RowSum\left(\mathbf{L}^{(1)}_{\mathcal{U}\mathcal{L}} + \sum_{l=1}^{\infty}\mathbf{F}^{(l)}_{\mathcal{U}\mathcal{L}}\right)\mathbf{1}^T\mathbf{e}_g,$$

and, for $j \geq 2$,

$$\widetilde{\mathbf{B}}^{(j)}_{\mathcal{U}\mathcal{U}} = \widehat{\mathbf{B}}^{(j)}_{\mathcal{U}\mathcal{U}} + RowSum(\widehat{\mathbf{B}}^{(j)}_{\mathcal{U}\not\mathcal{L}})\mathbf{1}^T\mathbf{e}_g + RowSum\left(\sum_{l=1}^{j-1}\mathbf{B}^{(l)}_{\mathcal{U}\mathcal{L}} + \mathbf{L}_{\mathcal{U}\mathcal{L}} + \sum_{l=1}^{\infty}\mathbf{F}^{(l)}_{\mathcal{U}\mathcal{L}}\right)\mathbf{1}^T\mathbf{e}_g.$$

The result is a GI/M/1-type process that we solve using the matrix geometric method outlined in Sect. 2.2. We can choose to discard the g^{th} column of $\widetilde{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{U}}$ and $\widetilde{\mathbf{B}}^{(j)}_{\mathcal{U}\mathcal{U}}$ when applying Eq. 4, so the contribution of the stochastic complementation is actually captured by the diagonal of matrices $\mathbf{L}^{(1)}_{\mathcal{U}\mathcal{U}}$ and $\mathbf{L}_{\mathcal{U}\mathcal{U}}$, and no computation is required in practice to set up the linear system of Eq. 4.

Letting $\overline{\boldsymbol{\alpha}} = [\overline{\boldsymbol{\alpha}}^{(0)}, \overline{\boldsymbol{\alpha}}^{(1)}, \ldots]$ be the stationary probability vector of the upper CTMC, we obtain $\mathbf{R}$, $\overline{\boldsymbol{\alpha}}^{(0)}$, and $\overline{\boldsymbol{\alpha}}^{(1)}$ by applying the matrix geometric method. Then, we can obtain any $\overline{\boldsymbol{\alpha}}^{(j)}$, for $j \geq 2$, using the relation $\overline{\boldsymbol{\alpha}}^{(j)} = \overline{\boldsymbol{\alpha}}^{(1)}\mathbf{R}^{(j-1)}$. Of course, $\overline{\boldsymbol{\alpha}}$ also represents the probabilities of the states in $\mathcal{U}$ for the overall CTMC conditioned on being in $\mathcal{U}$. These are needed to formulate the pseudo-stochastic complement of $\mathcal{L}$, in the next section.

3.3 Defining and solving the lower CTMC

To derive the infinitesimal generator $\mathbf{Q}_{\mathcal{L}}$ of the lower CTMC, with state space $\mathcal{L}$, we apply Definition 2. Recall that, now the gate state g belongs to $\mathcal{L}$, hence the complement of $\mathcal{L}$ in $\mathcal{S}$ is $\not\mathcal{U}$. The pseudo-stochastic complement of $\mathcal{L}$ is then $\mathbf{Q}_{\mathcal{L}} = \mathbf{Q}[\mathcal{L},\mathcal{L}] + \mathbf{Q}[\mathcal{L},\not\mathcal{U}]\,\mathbf{1}^T\,Norm(\boldsymbol{\pi}[\not\mathcal{U}]\mathbf{Q}[\not\mathcal{U},\mathcal{L}]) = \mathbf{Q}[\mathcal{L},\mathcal{L}] + \mathbf{Q}[\mathcal{L},\not\mathcal{U}]\,\mathbf{1}^T\,Norm(\overline{\boldsymbol{\alpha}}[\not\mathcal{U}]\mathbf{Q}[\not\mathcal{U},\mathcal{L}])$, where the last equality holds because $\overline{\boldsymbol{\alpha}}[\not\mathcal{U}]$ and $\boldsymbol{\pi}[\not\mathcal{U}]$ differ only by a constant, which is irrelevant due to the normalization. Furthermore, only row g of $\mathbf{Q}[\mathcal{L},\not\mathcal{U}]$ can have nonzero entries, thus $\mathbf{Q}_{\mathcal{L}}$ differs from $\mathbf{Q}[\mathcal{L},\mathcal{L}]$ only in that row. In other words, only transition from g to $\not\mathcal{U}$ must be rerouted to $\mathcal{L}$.

Since the block-partition defined in Fig. 2 places g with the upper set of states, more precisely in $\mathcal{U}^{(0)}$, we need to repartition the first two rows and columns of blocks in Fig. 2, to reflect the fact that now g belongs to $\mathcal{L}^{(0)}$ instead. In other words, we need to define the blocks

$$\begin{array}{lllllllll}
\widehat{\mathbf{B}}^{(1)}_{\mathcal{L}\mathcal{L}} & = & \mathbf{Q}[\mathcal{L}^{(1)},\mathcal{L}^{(0)}] & \widehat{\mathbf{L}}^{(0)}_{\mathcal{L}\mathcal{L}} & = & \mathbf{Q}[\mathcal{L}^{(0)},\mathcal{L}^{(0)}] & \widehat{\mathbf{F}}^{(j)}_{\mathcal{L}\mathcal{L}} & = & \mathbf{Q}[\mathcal{L}^{(0)},\mathcal{L}^{(j)}] \\
\widehat{\mathbf{B}}^{(j)}_{\mathcal{U}\mathcal{L}} & = & \mathbf{Q}[\mathcal{U}^{(j)},\mathcal{L}^{(0)}] & \widehat{\mathbf{L}}^{(0)}_{\not\mathcal{U}\mathcal{L}} & = & \mathbf{Q}[\not\mathcal{U}^{(0)},\mathcal{L}^{(0)}] & \widehat{\mathbf{F}}^{(j)}_{\not\mathcal{U}\mathcal{L}} & = & \mathbf{Q}[\not\mathcal{U}^{(0)},\mathcal{L}^{(j)}]
\end{array}$$

Then, the infinitesimal generator of the lower CTMC with state space $\mathcal{L}$ is

$$\mathbf{Q}_{\mathcal{L}} = \begin{bmatrix} \widetilde{\mathbf{L}}^{(0)}_{\mathcal{L}\mathcal{L}} & \widetilde{\mathbf{F}}^{(1)}_{\mathcal{L}\mathcal{L}} & \widetilde{\mathbf{F}}^{(2)}_{\mathcal{L}\mathcal{L}} & \widetilde{\mathbf{F}}^{(3)}_{\mathcal{L}\mathcal{L}} & \widetilde{\mathbf{F}}^{(4)}_{\mathcal{L}\mathcal{L}} & \cdots \\ \widehat{\mathbf{B}}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{L}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(2)}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(3)}_{\mathcal{L}\mathcal{L}} & \cdots \\ \mathbf{0} & \mathbf{B}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{L}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(2)}_{\mathcal{L}\mathcal{L}} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{L}_{\mathcal{L}\mathcal{L}} & \mathbf{F}^{(1)}_{\mathcal{L}\mathcal{L}} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}^{(1)}_{\mathcal{L}\mathcal{L}} & \mathbf{L}_{\mathcal{L}\mathcal{L}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where $\widetilde{\mathbf{L}}^{(0)}_{\mathcal{L}\mathcal{L}} = \widehat{\mathbf{L}}^{(0)}_{\mathcal{L}\mathcal{L}} + \mathbf{e}_g^T \boldsymbol{\gamma}^{(0)}$ and $\widetilde{\mathbf{F}}^{(j)}_{\mathcal{L}\mathcal{L}} = \widehat{\mathbf{F}}^{(j)}_{\mathcal{L}\mathcal{L}} + \mathbf{e}_g^T \boldsymbol{\gamma}^{(j)}$, for $j \geq 1$, where the "correcting" row vectors $\boldsymbol{\gamma}^{(j)}$, whose computation we show next, represent the rerouted rates of going from g to $\mathcal{L}^{(j)}$ through $\mathcal{U}$, for $j \geq 0$.

The total rate at which the original CTMC leaves state g toward $\mathcal{U}$ once it is in g is

$$u = \mathbf{Q}[g, \mathcal{U}]\, \mathbf{1}^T \;=\; \sum_{i \in \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}} \mathbf{Q}[g, i].$$

To compute $Norm(\overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}])$ we obtain first

$$\begin{aligned} \overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}^{(0)}] &= \overline{\boldsymbol{\alpha}}^{(0)}[\mathcal{U}^{(0)}]\widehat{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{L}} + \sum_{l=1}^{\infty} \overline{\boldsymbol{\alpha}}^{(l)} \widehat{\mathbf{B}}^{(l)}_{\mathcal{U}\mathcal{L}}, \\ \overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}^{(1)}] &= \overline{\boldsymbol{\alpha}}^{(0)}[\mathcal{U}^{(0)}]\widehat{\mathbf{F}}^{(1)}_{\mathcal{U}\mathcal{L}} + \overline{\boldsymbol{\alpha}}^{(1)} \mathbf{L}^{(1)}_{\mathcal{U}\mathcal{L}} + \sum_{l=1}^{\infty} \overline{\boldsymbol{\alpha}}^{(l+1)} \mathbf{B}^{(l)}_{\mathcal{U}\mathcal{L}}, \\ \overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}^{(j)}] &= \overline{\boldsymbol{\alpha}}^{(0)}[\mathcal{U}^{(0)}]\widehat{\mathbf{F}}^{(j)}_{\mathcal{U}\mathcal{L}} + \sum_{l=1}^{j-1} \overline{\boldsymbol{\alpha}}^{(l)} \mathbf{F}^{(j-l)}_{\mathcal{U}\mathcal{L}} + \overline{\boldsymbol{\alpha}}^{(j)} \mathbf{L}_{\mathcal{U}\mathcal{L}} + \sum_{l=j+1}^{\infty} \overline{\boldsymbol{\alpha}}^{(l)} \mathbf{B}^{(l-j)}_{\mathcal{U}\mathcal{L}}\;, \qquad \text{for } j > 1. \end{aligned}$$

Then, the normalization factor $f = \overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}]\mathbf{1}^T$ is given by

$$\overline{\boldsymbol{\alpha}}^{(0)}[\mathcal{U}^{(0)}] \left(\widehat{\mathbf{L}}^{(0)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \sum_{j=1}^{\infty} \widehat{\mathbf{F}}^{(j)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T \right) + \overline{\boldsymbol{\alpha}}^{(1)} \left(\widehat{\mathbf{B}}^{(1)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \mathbf{L}^{(1)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T \right)$$

$$+ \sum_{j=2}^{\infty} \overline{\boldsymbol{\alpha}}^{(j)} \left(\widehat{\mathbf{B}}^{(j)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \sum_{l=1}^{j-1} \mathbf{B}^{(l)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T + \sum_{l=1}^{\infty} \mathbf{F}^{(l)}_{\mathcal{U}\mathcal{L}}\mathbf{1}^T \right)$$

and the correcting vectors $\boldsymbol{\gamma}^{(j)}$, for $j \geq 0$, can be computed as

$$\boldsymbol{\gamma}^{(j)} = u\, Norm(\overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}])[\mathcal{L}^{(j)}] = u\overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}^{(j)}]/f.$$

In practice, we must truncate any infinite summation involving the vectors $\overline{\boldsymbol{\alpha}}^{(l)}$, such as those involved in the computation of $\overline{\boldsymbol{\alpha}}[\mathcal{U}]\mathbf{Q}[\mathcal{U}, \mathcal{L}^{(j)}]$ or of the normalization factor f, to a value of l large enough. Finally, once $\mathbf{Q}_{\mathcal{L}}$ has been built, we can use the M/G/1 algorithm outlined in Sect. 2.1 to solve for its stationary probability vector $\boldsymbol{\alpha}$.

3.4 Combining the results from the upper and lower CTMCs

Once we computed the conditional probability vectors $\overline{\boldsymbol{\alpha}}$, of the states in $\mathcal{U}$, and $\boldsymbol{\alpha}$, of the states in $\mathcal{L}$, we can compute the overall probability distribution $\boldsymbol{\pi}$ of original CTMC.

Since g belongs to both $\mathcal{U}$ and $\mathcal{L}$, we can scale $\overline{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ so that the probability of g in both has the same value; then the entries for *any* state in $\mathcal{U}$ or $\mathcal{L}$ will be correctly scaled in relation to each other. For example, we can scale $\overline{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ so that the entry for g assumes value 1 in both, i.e., divide $\overline{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ by $\overline{\boldsymbol{\alpha}}[g]$ and $\boldsymbol{\alpha}[g]$, respectively. Then, the sum of the scaled entries corresponding to $\not{\mathcal{U}}$ is $(1-\overline{\boldsymbol{\alpha}}[g])/\overline{\boldsymbol{\alpha}}[g]$, while the sum of the scaled entries corresponding to $\not{\mathcal{L}}$ is $(1-\boldsymbol{\alpha}[g])/\boldsymbol{\alpha}[g]$. Thus, the overall sum of the scaled entries corresponding to $\not{\mathcal{U}}$, g, and $\not{\mathcal{L}}$ is

$$s = \frac{1-\overline{\boldsymbol{\alpha}}[g]}{\overline{\boldsymbol{\alpha}}[g]} + 1 + \frac{1-\boldsymbol{\alpha}[g]}{\boldsymbol{\alpha}[g]} = \frac{\overline{\boldsymbol{\alpha}}[g]+\boldsymbol{\alpha}[g]-\overline{\boldsymbol{\alpha}}[g]\boldsymbol{\alpha}[g]}{\overline{\boldsymbol{\alpha}}[g]\boldsymbol{\alpha}[g]},$$

and we can express the entries of the stationary probability vector of the original CTMC as

$$\boldsymbol{\pi}[i] = \begin{cases} \overline{\boldsymbol{\alpha}}[i]/(\overline{\boldsymbol{\alpha}}[g]s) & \text{if } i \in \not{\mathcal{U}} \\ \boldsymbol{\alpha}[i]/(\boldsymbol{\alpha}[g]s) & \text{if } i \in \not{\mathcal{L}} \\ 1/s & \text{if } i = g \end{cases} .$$

Of course, in practice, we compute only a finite portion of $\overline{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$, hence of $\boldsymbol{\pi}$.

3.5 Multiple upper or lower classes

In our exposition so far, we restricted ourselves to CTMCs where we can identify one upper and one lower set of states. However, it might be possible to further decompose the state space of certain models to fully exploit their beneficial structural characteristics.

For CTMCs with a strictly GI/M/1-type or M/G/1-type structure, this opportunity has been observed before. For example, if the repeating states $\mathcal{S}^{(j)}$ of a GI/M/1-type CTMC can be partitioned into two subsets $\mathcal{S}_a^{(j)}$ and $\mathcal{S}_b^{(j)}$ such that any path from $\mathcal{S}_b^{(j)}$ to $\mathcal{S}_a^{(j)}$ must visit $\mathcal{S}^{(0)}$, the probabilistic interpretation of $\mathbf{R}$ implies that its submatrix corresponding to rows of "b states" and columns of "a states" is zero [15]. Analogously, the probabilistic interpretation of $\mathbf{G}$ suggest that a similar property exists for the $\mathbf{G}$ matrix in the case of M/G/1-type processes.

Other types of simplifications are possible as well. For example, in [2, 3], we discuss the solution of special QBD and M/G/1-type CTMCs where the structure of the CTMC has another special property: all transitions from $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j-1)}$ are directed to a single return state $s_n^{(j-1)}$. This ensures that all the columns of matrix $\mathbf{G}$ except for column n are zero, and allows for a very memory- and time-efficient computation of the expected measures of interest.

Here, we consider a special case of GI/G/1-type structure which can be seen as an extension of the one that can be managed by the algorithm we just introduced. Consider for example the structure shown in Fig. 3 on the left, where there are two sets of upper states, $\mathcal{U}_a$ and $\mathcal{U}_b$, and two sets of lower states, $\mathcal{L}_c$ and $\mathcal{L}_d$. This requires the identification of four gate states g_a, g_b, g_c, and g_d. The sets $\mathcal{U}_a^{(0)}$ and $\mathcal{U}_b^{(0)}$ contain any state that can be reached from g_a or g_b on a path to $\mathcal{U}_a^{(1)}$ or $\mathcal{U}_b^{(1)}$, respectively. Analogously, $\mathcal{L}_c^{(0)}$ and $\mathcal{L}_d^{(0)}$ contain any

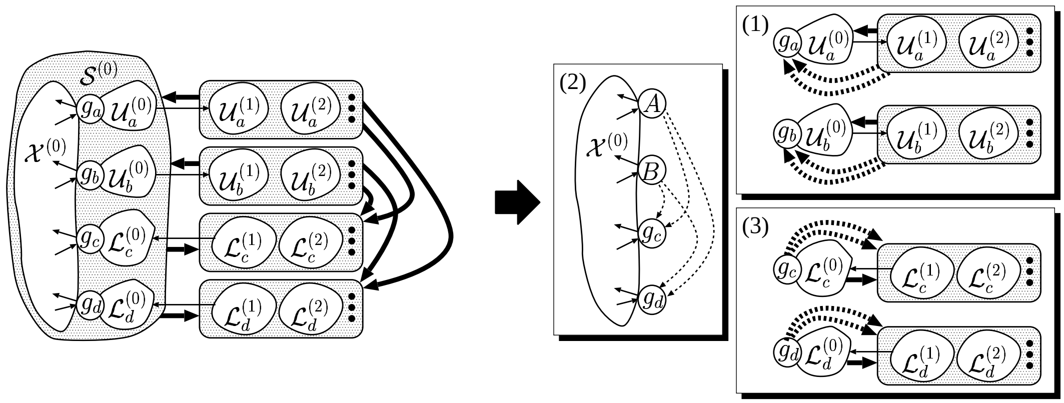

Figure 3: Multiple upper and lower sets.

state that $\mathcal{L}_c^{(1)}$ or $\mathcal{L}_d^{(1)}$ can reach on a path to g_c or g_d, respectively. These four sets must be disjoint, except for the gate states themselves, since some or all of g_a, g_b, g_c, and g_d can coincide. $\mathcal{X}^{(0)}$ contains any remaining state: $\mathcal{X}^{(0)} = \mathcal{S}^{(0)} \setminus (\mathcal{U}_a^{(0)} \cup \mathcal{U}_b^{(0)} \cup \mathcal{L}_c^{(0)} \cup \mathcal{L}_d^{(0)})$. To extend our approach to this situation, we define:

1. Two upper GI/M/1-type CTMCs obtained using stochastic complementation on the sets $\mathcal{U}_a$ and $\mathcal{U}_b$, respectively.

2. One finite CTMC with state space $\mathcal{X}^{(0)} \cup \{A, B, g_c, g_d\}$ obtained using stochastic complementation on the set $\mathcal{U}_a \cup \mathcal{U}_b \cup \mathcal{X}^{(0)}$, then aggregating the set $\mathcal{U}_a$ into the single macro-state A and the set $\mathcal{U}_b$ into the single macro-state B.

3. Two lower M/G/1-type CTMCs obtained using pseudo-stochastic complementation on the sets $\mathcal{L}_c$ and $\mathcal{L}_d$, respectively.

The order of the solution, however, is critical. First, we must solve the two upper CTMCs shown in portion (1) of the figure, to compute the conditional stationary probability vectors $\overline{\alpha}_a$ and $\overline{\alpha}_b$ of the states in $\mathcal{U}_a$ and $\mathcal{U}_b$, respectively. These probabilities can then be used to compute the rates leaving the macro-states A and B in the CTMC with state space $\mathcal{X}^{(0)} \cup \{A, B, g_c, g_d\}$: for example the rate from A to g_c is $\overline{\alpha}_a \cdot \mathbf{Q}[\mathcal{U}_a, \mathcal{L}_c] \cdot \mathbf{1}^T$. After computing the stationary probability vector $\boldsymbol{\beta}$ for this finite CTMC, we can say that $[\boldsymbol{\beta}[A]\overline{\alpha}_a | \boldsymbol{\beta}[B]\overline{\alpha}_b]$ is proportional to the corresponding portion $\boldsymbol{\pi}[\mathcal{U}_a \cup \mathcal{U}_b]$ of the stationary probability vector for the original CTMC; this is enough to compute the rates from g_c and g_d to $\mathcal{L}_c$ and $\mathcal{L}_d$, respectively, using pseudo-stochastic complementation, for the CTMCs shown in portion (3) of the figure, since only states in $\mathcal{U}_a$ and $\mathcal{U}_b$ can reach states in $\mathcal{L}_c$ and $\mathcal{L}_d$. Once these last two CTMCs are also solved, the solutions from the five CTMCs can be combined using the probability of the four gate states g_a, g_b, g_c, and g_d as reference points for the required scaling (while the finite CTMC does not explicitly have states g_a and g_b, their probabilities according to this CTMC can be obtained as $\boldsymbol{\beta}[A]\overline{\alpha}_a[g_a]$ and $\boldsymbol{\beta}[B]\overline{\alpha}_b[g_b]$).

Note that, unlike the special structures mentioned for the GI/M/1-type and the M/G/1-type CTMCs, which simply increase the efficiency of standard solution approaches, *the structure of Fig. 3 represents a true extension* which we can now solve.. This is because our approach with a single upper and a single lower set of states can be used to solve such structure only if g_a coincides with g_b and g_c coincides with g_d (in which case the role of the single gate g could then be played indifferently by $g_a \equiv g_b$, provided $\mathcal{X}^{(0)}$ is considered part of the lower states, or by $g_c \equiv g_d$, provided $\mathcal{X}^{(0)}$ is considered part of the upper states). Of course, when our approach with a single upper and a single lower set can be applied, identifying further partitions of the upper set, the lower set, or both, is still nevertheless important, but simply because it allows a more efficient solution.

4 Application: parallel scheduling in the presence of failures

We now employ our method to solve a system that can be modeled as a GI/G/1-type CTMC. A popular way to allocate processors among competing applications in a parallel system environment is by space-sharing: processors are partitioned into disjoint sets and each application executes in isolation on one of these sets. Space-sharing can be done in a static, adaptive, or dynamic way. If a job requires a fixed number of processors for execution, this requires a static space-sharing policy [22]. Adaptive space-sharing policies [1] have been proposed for jobs that can configure themselves to the number of processors allocated by the scheduler at the beginning of their execution. Dynamic space-sharing policies [11] have been proposed for jobs that are flexible enough to allow the scheduler to reduce or increase the number of processors allocated to them as they run, in response to environment changes. Because of their flexibility, dynamic policies can offer optimal performance; however, they are also the most difficult to implement because they reallocate resources while applications are executing.

Modeling the behavior of scheduling policies in parallel systems often results in CTMCs with matrix geometric form [23]. In these models, neither failures nor arrivals from multiple exogenous sources are considered. To illustrate the applicability of our methodology, we present instead a CTMC that models the behavior of a scheduling policy in a cluster environment subject to software and hardware failures. Our system is a cluster composed of two sub-clusters connected via a high speed medium (e.g., Gigabit Ethernet), while the nodes in each sub-cluster are connected via a lower speed switch (e.g., an ATM switch). For simplicity, we present a small system composed of two sub-clusters only, where a limited number of possible partitions is allowed, but our methodology readily applies to larger systems with multiple partitions. The arrival process is Poisson with parameter λ but each arrival may be a bulk of arbitrary size governed by a geometric distribution with parameter ρ. For clarity's sake, we only draw arcs corresponding to bulks of size one and two only, labeled with the rates $\lambda_1 = \lambda\rho$ and $\lambda_2 = \lambda(1-\rho)\rho$, respectively.

The system employs a space-sharing policy that allows up to two parallel jobs to run simultaneously. A parallel job may execute across the whole cluster (i.e., on both sub-clusters) or occupy only one sub-cluster: the service time is exponentially distributed with rate μ_2 in the former case, or μ_1 in the latter. The policy is as follows. Upon an arrival

- while there are no jobs in the system:
  - if the arrival is of a single job, that is, the bulk size equals one, the whole cluster is assigned to that job,

  - otherwise, if multiple jobs arrive simultaneously, that is, the bulk size is greater than one, two jobs are scheduled, one on each sub-cluster, while the remaining jobs in the bulk, if any, are queued;

- while there are already jobs in the system and one job running using the whole cluster:

  - if the bulk size equals one, the job is simply queued,
  - otherwise, if the bulk size is greater than one, the arriving jobs are enqueued, the current job is stopped and restarted on a single sub-cluster, and one of the queued jobs is also started, on the other sub-cluster;

- while there are jobs in the system already and two jobs are running, each on one sub-cluster:

  - we simply enqueue arriving jobs, regardless of the bulk size.

Upon a completion (departure)

- of a job that was using the entire system:

  - if there is only one job waiting, it is assigned the entire system,
  - if there are multiple jobs waiting, two of them are assigned a sub-cluster each;

- of a job that was using only one sub-cluster:

  - one of the jobs in the queue, if any, is assigned the sub-cluster just released (if there are no jobs waiting, the other job running, if any, is *not* reassigned to the entire system).

The rationale behind these decisions is that, under bursty conditions, we would like to reach our goal of having smaller partitions quickly, even at the cost of killing a running job and rescheduling it on a smaller partition [21].

We consider the performance of our scheduling policy under the following failure scenarios. Each of the two sub-clusters can experience a local hardware failure independently of each other; in this case, only the affected sub-cluster must be brought down for repair, and the system can still accept arrivals. In addition, when a parallel job is assigned the entire cluster, it makes use of software whose execution can cause the entire system to crash. After such a failure, each sub-cluster must be brought down for reboot; consequently, the system's queue is flushed and no arrivals are accepted until repairs are completed. We assume that all event durations are exponentially distributed. The rate for a hardware failure is f_h, for a software failure is f_s, for a repair after a hardware failure is r_h, and for a repair (reboot) after a software failure is r_s.

Fig. 4 depicts a CTMC modeling the behavior of our system and Table 1 describes the meaning of the system states. The infinitesimal generator of this CTMC has a GI/G/1 structure consistent with the requirements of our solution algorithm, since we can immediately identify state *0* as the gate state, states *kW* plus *0S* as the upper states, and states *0H*, *kHH*, *0F*, *kHF* and *kFF* as the lower states. Note that the model assumes that hardware failures do not occur during reboot; if this were not true, we could add a transition from *0S*

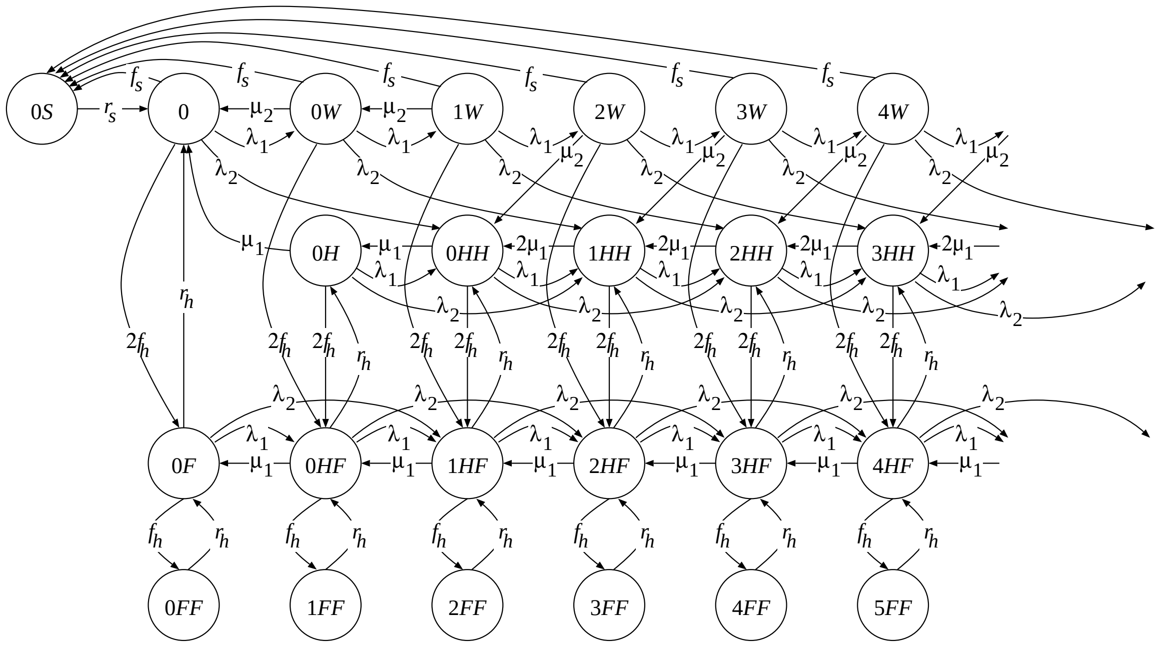

Figure 4: The CTMC for our performability study.

State	Description
0	Empty system, no failures
0S	Empty system, rebooting after a software failure
kW	$k \geq 0$ jobs in the queue, one job executing on the whole system
0H	0 jobs in the queue, one sub-cluster idle, the other running a job
kHH	$k \geq 0$ jobs in the queue, each sub-cluster running a job
0F	0 jobs in the queue, one sub-cluster idle, the other failed
kHF	$k \geq 0$ jobs in the queue, one sub-cluster running a job, the other failed
kFF	$k \geq 0$ jobs in the queue, both sub-cluster failed

Table 1: Meaning of the CTMC states.

to *0F* with rate $2f_h$, and the structure required by our approach would still be present, since there can be transitions from $\mathcal{U}^{(0)}$ (i.e., $\{0S\}$) to any state in $\mathcal{L}$.

The numerical values used for our model parameters are: $f_h = 10^{-6}..10^{-3}$, $f_s = 10 \cdot f_h$, $r_h = 10^{-3}$ or 10^{-1}, $r_s = 10{\cdot}r_h$, $\mu_1 = 3.0$ or 6.0, $\mu_2 = 1.8{\cdot}\mu_1$, $\lambda = 10^{-5}..5$, and $\rho = 0.8$. Fig. 5(a) and 5(b) show the long-term availability of the system, that is, its ability of accepting arrivals, computed as the stationary probability of being in any of the states except *0S* and *kFF*, for $k \geq 0$, for a given choice of r_h and two choices of μ_1, and for a given choice of μ_1 and two choices of r_h, respectively, as a function of f_h and λ (note that the arrival rate λ does affect the availability, since it affects the probability of the system being in the *kW* states, hence of software failures). Fig. 5(c) and 5(d) are analogous, except that they focus on the probability on not being unavailable due to a software failure, that is, they plot the complement of the probability of being in state *0S*. Fig. 5(e) and 5(f) focus instead on the system power, defined as the ratio of system throughput over average job response time, also as a function of f_h and λ and for various choices of r_h and μ_1. It is apparent that there is a significant correlation between the above parameters and the workload that the

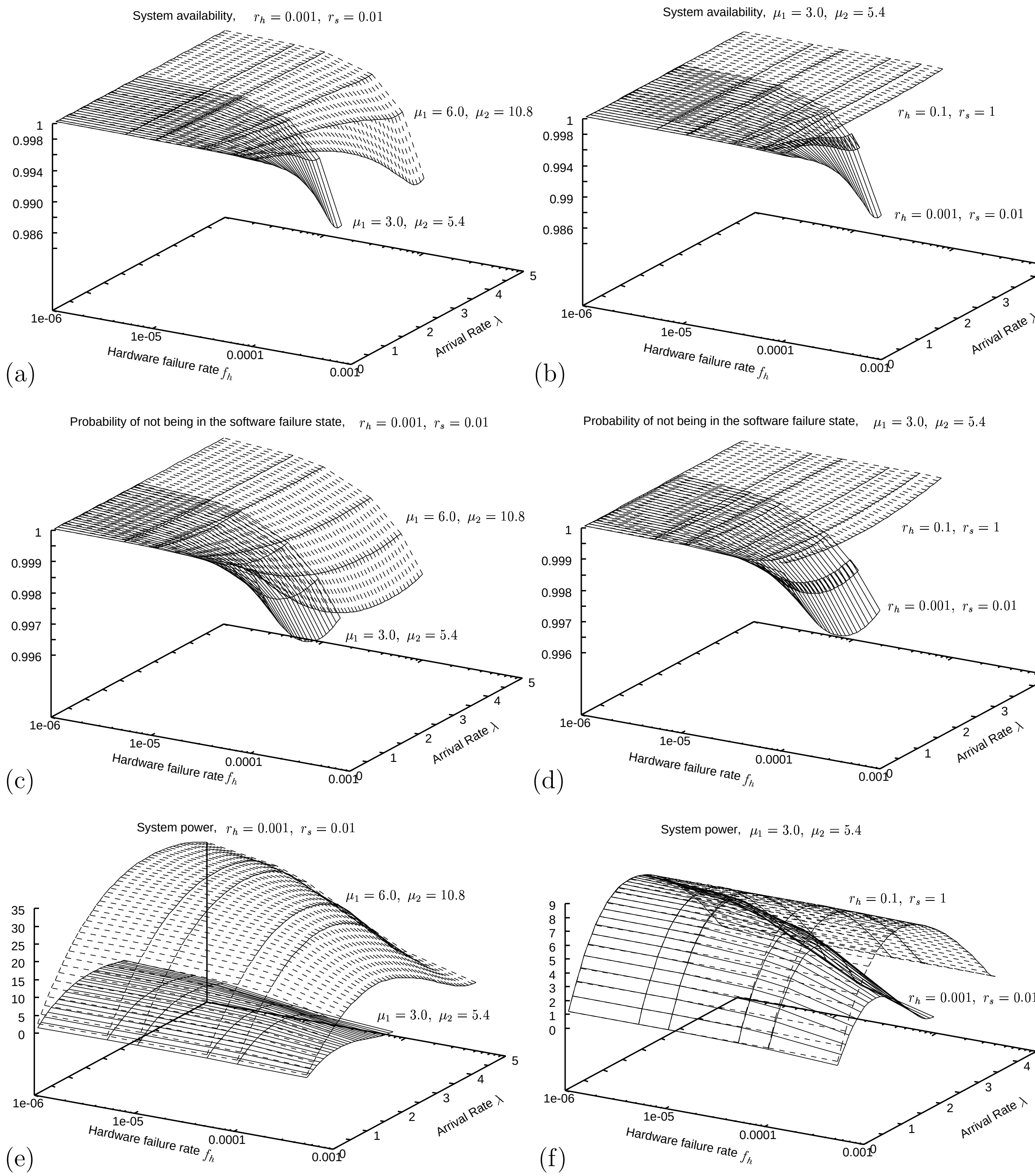

Figure 5: Performance results.

system can handle. Note that, in all plots, the missing data points correspond to parameter combinations for which the system becomes unstable.

References

[1] S.-H. Chiang, R. K. Mansharamani, and M. K. Vernon. Use of application characteristics and limited preemption for run-to-completion parallel processor scheduling policies. In *Proc. 1994 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 33–44, May 1994.

[2] G. Ciardo, A. Riska, and E. Smirni. An aggregation-based solution method for M/G/1-type processes. In B. Plateau, W. J. Stewart, and M. Silva, editors, *Numerical Solution of Markov Chains*, pages 21–40. Prensas Universitarias de Zaragoza, Zaragoza, Spain, Sept. 1999.

[3] G. Ciardo and E. Smirni. ETAQA: An efficient technique for the analysis of QBD-processes by aggregation. *Perf. Eval.*, 36-37:71–93, 1999.

[4] H. R. Gail, S. L. Hantler, and B. A. Taylor. Use of characteristic roots for solving infinite state markov chains. In W. K. Grassman, editor, *Computational Probability*, pages 205–255, Boston, MA, 2000. Kluwer Academic Publishers.

[5] W. K. Grassmann and D. A. Stanford. Matrix analytic methods. In W. K. Grassmann, editor, *Computational Probability*, pages 153–204. Kluwer Academic Publishers, Boston, MA, 2000.

[6] K. Kanoun and M. Ortalo-Borrel. Fault-tolerant system dependability - explicit modeling of hardware and software component-interactions. *IEEE Transactions on Reliability*, 49(4):363–376, Dec. 2000.

[7] S. W. Kwak and K. B. K. Task scheduling strategies for reliable tmr controllers using task grouping and assignment. *IEEE Transactions on Reliability*, 49(4):355–362, Dec. 2000.

[8] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains, and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 231–265. Springer-Verlag, 1993.

[9] G. Latouche and V. Ramaswami. *Introduction to Matrix Geometric Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia PA, 1999.

[10] L. Ledoux. Availability modeling of modular software. *IEEE Transactions on Reliability*, 48(2):159–168, June 1999.

[11] C. McCann, R. Vaswani, and J. Zahorjan. A dynamic processor allocation policy for multiprogrammed shared memory multiprocessors. *ACM Trans. Comp. Syst.*, 11(2):146–178, 1993.

[12] B. Meini. Solving M/G/1 type Markov chains: Recent advances and applications. *Stochastic Models*, 14(1 & 2):479–496, 1998.

[13] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–271, June 1989.

[14] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Trans. Comp.*, C-29(8):720–731, Aug. 1980.

[15] R. Nelson. *Probability, Stochastic Processes, and Queueing Theory*. Springer-Verlag, 1995.

[16] M. F. Neuts. *Matrix-geometric Solutions in Stochastic Models*. Johns Hopkins University Press, Baltimore, MD, 1981.

[17] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker, Inc., New York, NY, 1989.

[18] I. Onyuksel and S. H. Hosseini. Amdahl's law: a generalization under processor failures. *IEEE Transactions on Reliability*, 44(3):455–462, Sept. 1995.

[19] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Comm. Stat. – Stochastic Models*, 4:183–263, 1988.

[20] V. Ramaswami and G. Latouche. A general class of Markov processes with explicit matrix-geometric solutions. *OR Spektrum*, 8:209–218, Aug. 1986.

[21] E. Rosti, E. Smirni, G. Serazzi, L. W. Dowdy, and K. C. Sevcik. On processor saving scheduling policies for multiprocessor systems. *IEEE Trans. Comp.*, 47(2):178–189, Feb. 1998.

[22] E. Smirni, E. Rosti, L. W. Dowdy, and G. Serazzi. A methodology for the evaluation of multiprocessor non-preemptive allocation policies. *Journal of Systems Architecture*, 44:703–721, 1998.

[23] M. S. Squillante, F. Wang, and M. Papaefthymiou. Stochastic analysis of gang scheduling in parallel and distributed systems. *Perf. Eval.*, 27/28:273–296, 1996.

[24] S. R. Welke, B. W. Johnson, and J. H. Aylor. Reliability modeling of hardware/software systems. *IEEE Transactions on Reliability*, 44(3):413–418, Sept. 1995.

A Proof of Theorem 2

It is easy to see that $\mathbf{A}$ is an infinitesimal generator: its off-diagonal entries are non-negative by construction, and its rows sum to zero because:

$$\begin{aligned}
\mathbf{A}\mathbf{1}^T &= \left(\mathbf{Q}[\mathcal{A},\mathcal{A}] + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T Norm(\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}])\right)\mathbf{1}^T \\
&= \mathbf{Q}[\mathcal{A},\mathcal{A}]\mathbf{1}^T + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T Norm(\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}])\mathbf{1}^T \\
&= \mathbf{Q}[\mathcal{A},\mathcal{A}]\mathbf{1}^T + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T \\
&= \mathbf{Q}[\mathcal{A},\mathcal{S}]\mathbf{1}^T \;=\; \mathbf{0}^T.
\end{aligned}$$

The irreducibility of the pseudo stochastic complement of $\mathcal{A}$ follows directly from its probabilistic interpretation and the fact that the original process is irreducible. We now need to show that $\boldsymbol{\alpha} = \boldsymbol{\pi}[\mathcal{A}]/(\boldsymbol{\pi}[\mathcal{A}]\mathbf{1}^T)$ satisfies $\boldsymbol{\alpha}\mathbf{A} = \mathbf{0}$. Starting from the definition of pseudo stochastic complement, we have:

$$\begin{aligned}
\boldsymbol{\alpha}\mathbf{A} &= \frac{\boldsymbol{\pi}[\mathcal{A}]}{\boldsymbol{\pi}[\mathcal{A}]\mathbf{1}^T}\left(\mathbf{Q}[\mathcal{A},\mathcal{A}] + \mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T Norm(\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}])\right) \\
&= \frac{1}{\boldsymbol{\pi}[\mathcal{A}]\mathbf{1}^T}\left(\boldsymbol{\pi}[\mathcal{A}]\mathbf{Q}[\mathcal{A},\mathcal{A}] + \boldsymbol{\pi}[\mathcal{A}]\mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T \frac{\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]}{\boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]\mathbf{1}^T}\right) \\
&= \frac{1}{\boldsymbol{\pi}[\mathcal{A}]\mathbf{1}^T}\left(\boldsymbol{\pi}[\mathcal{A}]\mathbf{Q}[\mathcal{A},\mathcal{A}] + \boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]\right) \;=\; \mathbf{0},
\end{aligned}$$

where the last two steps are obtained considering that $\boldsymbol{\pi}[\mathcal{A}]\mathbf{Q}[\mathcal{A},\overline{\mathcal{A}}]\mathbf{1}^T = \boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}]\mathbf{1}^T$ (they represent the flow from $\mathcal{A}$ to $\overline{\mathcal{A}}$ and from $\overline{\mathcal{A}}$ to $\mathcal{A}$ in steady state, respectively) and that $\boldsymbol{\pi}[\mathcal{A}]\mathbf{Q}[\mathcal{A},\mathcal{A}] + \boldsymbol{\pi}[\overline{\mathcal{A}}]\mathbf{Q}[\overline{\mathcal{A}},\mathcal{A}] = \mathbf{0}$ (since $\boldsymbol{\pi}$ is the stationary probability vector). $\square$