Performability analysis of fault-tolerant computer systems

Hédi Nabli, Bruno Sericola

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

Performability Analysis of Fault-Tolerant Computer Systems

Hédi Nabli, Bruno Sericola

N˚ 2254

May 1994

PROGRAMME 1

Architectures parallèles,

bases de données,

réseaux et systèmes distribués

Rapport de recherche

ISSN 0249-6399

Performability Analysis of Fault-Tolerant Computer Systems

Hédi Nabli*, Bruno Sericola*

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet Model

Rapport de recherche n° 2254 — May 1994 — 36 pages

Abstract: Performability is a composite measure for the performance and reliability, which may be interpreted as the cumulative performance over a finite mission time. The computation of its distribution allows the user to ensure that the system will achieve a given performance level. The system is assumed to be modeled as a Markov process with finite state space and a reward rate (performance measure) is associated with each state. We propose, in this paper, a new algorithm to compute the performability distribution of fault-tolerant computer systems. The main advantage of this new algorithm is its low polynomial computational complexity. Moreover it deals only with non negative numbers bounded by one. This important property allows us to determine truncation steps and so to improve the execution time of the algorithm.

Key-words: Fault tolerance, repairable systems, Markov processes, performability, performance, reliability, uniformization

(Résumé : tsvp)

This work has been partly supported by the FASST Esprit project

*{Hédi.Nabli} {Bruno.Sericola}@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Analyse de la performabilité des systèmes informatiques tolérant les pannes

Résumé : La performabilité est une mesure composée pour la performance et la fiabilité, qui peut être interprétée comme la performance cumulée sur un temps de mission fini. Le calcul de cette distribution permet à l'utilisateur de s'assurer que son système atteindra un certain niveau de performance. Le système est modélisé par un processus de Markov à espace d'état fini et un taux de récompense (mesure de performance) est associé à chaque état. Nous proposons, dans cet article, un nouvel algorithme pour calculer la distribution de la performabilité des systèmes informatiques tolérant les pannes. L'avantage principal de ce nouvel algorithme est sa complexité de calcul polynômiale faible. De plus, il ne traite que des nombres positifs bornés par 1. Cette propriété importante nous permet de déterminer des seuils de troncature et donc d'améliorer le temps d'exécution de l'algorithme

Mots-clé : Tolérance aux pannes, systèmes réparables, processus de Markov, performabilité, performance, fiabilité, uniformisation

1 Introduction

As recognized in a large number of studies, the quantitative evaluation of fault-tolerant computer systems requires to deal simultaneously with aspects of both performance and reliability. For this purpose, Meyer [1] developed the concept of performability, which may be interpreted as the cumulative performance over a finite mission time. The increasing need in evaluating cumulative measures comes from the fact that in highly available systems, steady state measures can be very poor, even if the mission time is not small. The use of expectations also suffers from similar drawbacks. Considering, for instance, critical applications, it is crucial for the user to ensure that the probability that its system will achieve a given performance level is high enough.

Formally, the system fault-repair behavior is assumed to be modeled by a homogeneous Markov process. Its state space is divided into disjoint subsets, which represents the different configurations of the system. A performance level (reward rate) is associated with each of these configurations. This reward rate quantifies the ability of the system to perform in the corresponding configuration. Performability is then the accumulated reward over the mission time.

The distribution of this random variable has been studied in previous papers. Some of these papers [2], [3], [4], [5], [6], [7] are restricted to the case of acyclic Markov processes which are used to model non repairable systems.

To model the repair of faulty components in repairable systems, cyclic Markov processes are needed. For absorbing Markov processes, Beaudry [8] gave an algorithm to compute the distribution of accumulated reward until system failure when the reward rates are strictly positive and Ciardo et al. [9] extended this method to semi-Markov processes for non-negative reward rates.

For finite mission time and when the reward rates are either 0 or 1, the accumulated reward over the mission time is the interval availability. The distribution of interval availability has been studied in [10] using the uniformization technique. The computational complexity of this method has been improved in [11].

The distribution of accumulated reward over a finite mission (with general reward rates) is more complex to obtain. In [12], Iyer et al. proposed an algorithm to compute recursively the moments of the accumulated reward over the mission time, with a polynomial computational complexity in the number of states. In [13], the distribution of this random variable has been derived using Laplace transform and numerical inversion procedures to get the result in the time domain. De Souza e Silva and Gail [14] proposed a method based on the uniformization technique, however their method exibits a exponential computational complexity in the number of reward rates. Using the same technique, Donatiello and Grassi [15] obtained an algorithm with a polynomial computational complexity. However this algorithm seems to be numerically unstable since the coefficient computed in their recursion can have positive and negative signs and are unbounded which can lead to severe numerical errors and overflow problems. More recently, De Souza e Silva and Gail [16] obtained also an algorithm with a polynomial computational complexity which is linear in a parameter that is smaller than the number of rewards, but their algorithm seems to have the same unstability problem due the use of both positive and negative coefficients. Pattipati et al. [17] obtained the distribution of the accumulated reward for non-homogeneous Markov processes as the solution of a system of linear hyperbolic partial differential equations which is numerically solved by using a discretization approach.

In this paper we develop a new algorithm to compute the distribution of accumulated over a finite mission time. As in [15] or [16], this method used is based on the uniformization technique. The main contribution of this paper is that our algorithm is numerically stable by the fact it deals only with non negative numbers bounded by 1. Moreover the computational complexity is improved by the use of truncation steps and the precision of the result can be given in advance.

The remainder of the paper is organized as follows. In the next section, we introduce the mathematical model and the notation. The third section gives the proposed solution and describes the algorithm and its computational complexity. In the fourth section, a model of a fault-tolerant computer system is presented and solved for a given performability measure. The last section is devoted to some conclusions.

2 Mathematical model

We consider systems that can be modeled by a homogeneous Markov process with a finite state space. A performance level or reward rate is associated with each state of the process. These reward rates are assumed to be time independent as usual, and the random variable of interest is the accumulated reward over a finite mission time.

More formally, let $X = \{X_u, u \geq 0\}$ be a homogeneous Markov process over a finite state space denoted by $E = \{1, \ldots, M\}$. A reward rate $\rho(i)$ is associated with each state i of E. Since two different states may have the same reward rate, we denote by $r_m > r_{m-1} > \ldots > r_0$ the $m+1$ different reward rates ($m < M$). The state space E can be then divided into disjoint subsets $B_m, B_{m-1}, \ldots, B_0$ where B_i, $0 \leq i \leq m$ is composed by the states of E having as reward rate r_i, that is

$$B_i = \{j \in E / \rho(j) = r_i\}.$$

The number of states in subset B_i is denoted by M_i, for $i = 0, \ldots, m$. The process X is given by its infinitesimal generator, denoted by A, in which the ith diagonal entry $A(i,i)$ verifies $A(i,i) = -\sum_{j \neq i} A(i,j)$ and by its initial probability distribution α whose ith entry is denoted by α_i. Moreover, for any subset S of E, we denote by 1_S (resp. 0_S) the column vector of size the number of states in S, with all elements equal to 1 (resp. 0).

Let us denote by Z the uniformized Markov chain [18] over the state space E with respect to the uniformization rate λ and by P its transition probability matrix whose (i,j) entry is denoted by $P(i,j)$. The uniformization rate λ verifies $\lambda \geq \sup(-A(i,i), i \in E)$ and P is related to A by $P = I + A/\lambda$, where I denotes the identity matrix. If $N = \{N_u, u \geq 0\}$ is a Poisson process with parameter λ, independent of the process Z, then it is well-known that the two processes X and $\{Z_{N_u}, u \geq 0\}$ have the same probabilistic behavior if they have the same initial distribution. In the following, to simplify notation, we will consider X as the uniformized process. We decompose P and the initial probability vector α with respect to the partition $\{B_m, B_{m-1}, \ldots, B_0\}$ of E as follows:

$$P = \begin{pmatrix} P_{B_m B_m} & \cdots & P_{B_m B_0} \\ \vdots & & \vdots \\ P_{B_0 B_m} & \cdots & P_{B_0 B_0} \end{pmatrix}, \quad \alpha = (\alpha_{B_m}, \ldots, \alpha_{B_0}),$$

where the submatrix $P_{B_i B_j}$, $0 \leq i, j \leq m$, contains the transition probabilities from states of B_i to states of B_j and subvector α_{B_i}, $0 \leq i \leq m$, contains the initial probabilities corresponding to states of B_i.

The random variable of interest which is denoted by Y_t and represents the accumulated reward over the interval of time $[0,t]$ is defined by

$$Y_t = \int_0^t \rho(X_u) du.$$

Using the decomposition of E with respect to the partition $\{B_m, B_{m-1}, \ldots, B_0\}$, this random variable can also be written as

$$Y_t = \sum_{i=0}^{m} r_i \int_0^t \mathbf{1}_{\{X_u \in B_i\}} du,$$

where

$$\mathbf{1}_{\{c\}} = \begin{cases} 1 & \text{if condition } c \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

The random variable Y_t takes its values in the interval $[r_0 t, r_m t]$ and we wish to calculate $\mathbb{P}\{Y_t > s\}$. The reward rates r_i are arbitrary real numbers, but we can assume without loss of generality that $r_0 = 0$. This can be done by replacing r_i by $r_i - r_0$ and s by $s - r_0 t$. So, in the sequel, we set $r_0 = 0$.

For a fixed value $s \in [0, r_m t[$ and $i \in E$, we define for any integer $n \geq 0$,

$$F_i(s,t,n) = \mathbb{P}\{Y_t > s, N_t = n \mid X_0 = i\}.$$

We denote by $F_{B_l}(s,t,n)$ the column vector of dimension M_l and whose ith entry is equal to $F_i(s,t,n)$ for $i \in B_l$. Consequently, with this notation we have

$$\mathbb{P}\{Y_t > s\} = \sum_{i=1}^{M} \alpha_i \sum_{n=0}^{\infty} F_i(s,t,n) = \sum_{l=0}^{m} \alpha_{B_l} \sum_{n=0}^{\infty} F_{B_l}(s,t,n).$$

The following theorem gives the forward renewal equation, satisfied by the column vectors $F_{B_i}(s,t,n)$, that will be used in the next section to get the distribution of Y_t.

Theorem 2.1 *For every $l \in \{0, 1, \ldots, m\}$ and for every $n \geq 1$, we have*

$$\begin{aligned} F_{B_l}(s,t,n) &= \sum_{k=0}^{m} \int_0^{\frac{s}{r_l}} P_{B_l B_k} F_{B_k}(s - r_l u, t-u, n-1) \lambda e^{-\lambda u} du \mathbf{1}_{\{s < r_l t\}} \\ &+ \sum_{k=0}^{m} \int_0^{\frac{r_m t - s}{r_m - r_l}} P_{B_l B_k} F_{B_k}(s - r_l u, t-u, n-1) \lambda e^{-\lambda u} du \mathbf{1}_{\{s \geq r_l t\}} \\ &+ e^{-\lambda t} \frac{\lambda^n}{n!} (t - \frac{s}{r_l})^n 1_{B_l} \mathbf{1}_{\{s < r_l t\}} \end{aligned}$$

where the first and the third terms are equal to 0 when $l = 0$ and the second term is equal to 0 when $l = m$.

Proof. The function $F_i(s,t,n)$ defined previously as

$$F_i(s,t,n) = \mathbb{P}\{Y_t > s, N_t = n \mid X_0 = i\}$$

satisfies the classical forward renewal equation, for $n \geq 1$, that is

$$F_i(s,t,n) = \sum_{j \in E} P(i,j) \int_0^t F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du. \tag{1}$$

When $s - r_i u < 0$, we have

$$F_j(s - r_i u, t-u, n-1) = \mathbb{P}\{N_{t-u} = n-1 \mid X_0 = j\} = e^{-\lambda(t-u)} \frac{(\lambda(t-u))^{n-1}}{(n-1)!}$$

and when $s - r_i u \geq r_m(t-u)$, we have $F_j(s - r_i u, t-u, n-1) = 0$.

It follows that when $s < r_i t$, we get

$$\begin{aligned}
F_i(s,t,n) &= \sum_{j \in E} P(i,j) \int_0^{\frac{s}{r_i}} F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du \\
&\quad + \sum_{j \in E} P(i,j) \int_{\frac{s}{r_i}}^t e^{-\lambda(t-u)} \frac{(\lambda(t-u))^{n-1}}{(n-1)!} \lambda e^{-\lambda u} du \\
&= \sum_{j \in E} P(i,j) \int_0^{\frac{s}{r_i}} F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du \\
&\quad + e^{-\lambda t} \frac{\lambda^n}{n!} (t - \frac{s}{r_i})^n,
\end{aligned}$$

and when $s \geq r_i t$, we get

$$F_i(s,t,n) = \sum_{j \in E} P(i,j) \int_0^{\frac{r_m t - s}{r_m - r_i}} F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du$$

since in this case, we have

$$\frac{r_m t - s}{r_m - r_i} \leq t.$$

Putting now together the two cases, we get

$$\begin{aligned}
F_i(s,t,n) &= \sum_{j \in E} P(i,j) \int_0^{\frac{s}{r_i}} F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du \mathbf{1}_{\{s < r_i t\}} \\
&\quad + \sum_{j \in E} P(i,j) \int_0^{\frac{r_m t - s}{r_m - r_i}} F_j(s - r_i u, t-u, n-1) \lambda e^{-\lambda u} du \mathbf{1}_{\{s \geq r_i t\}} \\
&\quad + e^{-\lambda t} \frac{\lambda^n}{n!} (t - \frac{s}{r_i})^n \mathbf{1}_{\{s < r_i t\}}.
\end{aligned}$$

The theorem follows by rewriting this last relation in vector notation.

□

3 Model solution and algorithmical aspects

In this section, we first give the main result of this paper, which is the distribution of the accumulated reward over the finite mission time $[0,t]$, Y_t. Next, an algorithm to compute this distribution is proposed and analyzed through its computational complexity.

Theorem 3.1

$$\mathbb{P}\{Y_t > s\} = \sum_{n=0}^{\infty} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\sum_{k=0}^{n}\sum_{j=1}^{m}\binom{n}{k} s_j^k (1-s_j)^{n-k} b^{(j)}(n,k)\mathbf{1}_{\{r_{j-1}t \le s < r_j t\}}$$

where $s_j = \dfrac{s - r_{j-1}t}{(r_j - r_{j-1})t}$ *and coefficients* $b^{(j)}(n,k)$ *are given by*

$$b^{(j)}(n,k) = \sum_{l=0}^{m} \alpha_{B_l} b_{B_l}^{(j)}(n,k)$$

and column vectors $b_{B_l}^{(j)}(n,k)$ *are given by the following recursive expressions*

for $j \le l \le m$ **and** $1 \le k \le n$

$$\begin{aligned} b_{B_l}^{(j)}(n,0) &= \begin{cases} 1_{B_l} & \text{for } j = 1 \\ b_{B_l}^{(j-1)}(n,n) & \text{for } j > 1 \end{cases} \\ b_{B_l}^{(j)}(n,k) &= \frac{r_l - r_j}{r_l - r_{j-1}} b_{B_l}^{(j)}(n,k-1) + \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k-1) \end{aligned}$$

for $0 \le l \le j-1$ **and** $0 \le k \le n-1$

$$\begin{aligned} b_{B_l}^{(j)}(n,k) &= \frac{r_{j-1} - r_l}{r_j - r_l} b_{B_l}^{(j)}(n,k+1) + \frac{r_j - r_{j-1}}{r_j - r_l} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \\ b_{B_l}^{(j)}(n,n) &= \begin{cases} b_{B_l}^{(j+1)}(n,0) & \text{for } j < m \\ 0_{B_l} & \text{for } j = m \end{cases} \end{aligned}$$

Proof. (See Appendix A).

□

An important observation here is that for $j \leq l \leq m$ we have

$$0 \leq \frac{r_l - r_j}{r_l - r_{j-1}} = 1 - \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \leq 1,$$

for $0 \leq l \leq j-1$ we have

$$0 \leq \frac{r_{j-1} - r_l}{r_j - r_l} = 1 - \frac{r_j - r_{j-1}}{r_j - r_l} \leq 1,$$

and for $0 \leq l \leq m$ we have

$$\sum_{i=0}^{m} P_{B_l B_i} 1_{B_i} = 1_{B_l}.$$

For every $n \geq 0$, the initial value $b_{B_l}^{(1)}(n,0)$ is equal to 1_{B_l} if $l \geq 1$ and the final value $b_{B_l}^{(m)}(n,n)$ is equal to 0_{B_l} if $l \leq m-1$. We then easily obtain by recurrence that

$$0_{B_l} \leq b_{B_l}^{(j)}(n,k) \leq 1_{B_l}.$$

Moreover, for every $j = 1, \ldots, m$, we have $0 \leq s_j < 1$. These remarks are essential from a computational point of view since the manipulation of non-negative quantities and bounded by 1 allows us to avoid the unstability problems which may appear in the algorithms described in [15] and [16].

Let us now define for every $j = 1, \ldots, m$, a partition of the state space E as

$$U_j = B_m \cup \cdots \cup B_j \text{ and } D_j = B_{j-1} \cup \cdots \cup B_0.$$

For every $j = 1, \ldots, m$, we also define the following column vectors

$$b_{U_j}(n,k) = \left(b_{B_m}^{(j)}(n,k)^{\mathrm{T}}, \ldots, b_{B_j}^{(j)}(n,k)^{\mathrm{T}}\right)^{\mathrm{T}} \text{ and } b_{D_j}(n,k) = \left(b_{B_{j-1}}^{(j)}(n,k)^{\mathrm{T}}, \ldots, b_{B_0}^{(j)}(n,k)^{\mathrm{T}}\right)^{\mathrm{T}}$$

where T denotes the transpose operator.

With this notation, Fig. 1 and Fig. 2 illustrate the sequence of computations (drawn only for $n = 0, 1, 2, 3$) that have to be done in order to evaluate the $b_{B_l}^{(j)}(n,k)$'s. Note that the upper part of the diagonal of each triangle of cells is reported in the upper part of the first column of the next one and the lower part of the first column each triangle of cells is reported in the lower part of the diagonal of the previous triangle of cells.

The study of the recurrence described in Theorem 3.1 leads to the following remarks.

In the case where $j = m$, illustrated in Fig. 2, the triangle of cells can be calculated either in a diagonal by diagonal manner provided that the first cell of a diagonal is known or in a line by line manner.

In the case where $j = 1$, the triangle of cells is computed in a line by line manner but it can be also calculated in a column by column manner provided that the first cell of a column is known.

This is not possible for the other triangles of cells (that is for $j = 2, \ldots, m-1$). These cells can be calculated only in a line by line manner.

Note that $m = 1$, the performability distribution described above is the same as the one given in [11], where the interval availability distribution is computed. It follows that the method presented here is the natural extension of the method in [11].

The way in which the computation of each cell (n, k) is performed is shown in Fig. 3.

We now show that the computation of the last triangle of cells, which corresponds to $j = m$, in a diagonal by diagonal manner is very useful to reduce the complexity in the case where the value of s is near from the value of $r_m t$.

Given a tolerance error ε specified by the user, we define integer N as

$$N = \min \left\{ n \in \mathbb{N} \;\middle|\; \sum_{j=0}^{n} e^{-\lambda t} \frac{(\lambda t)^j}{j!} \geq 1 - \frac{\varepsilon}{2} \right\}. \tag{2}$$

The distribution of Y_t given in Theorem 3.1 can then be written as

$$\mathbb{P}\{Y_t > s\} = \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \sum_{j=1}^{m} \binom{n}{k} s_j^k (1-s_j)^{n-k} b^{(j)}(n,k) \mathbf{1}_{\{r_{j-1}t \leq s < r_j t\}} + e(N)$$

where $e(N)$ verifies

$$\begin{aligned} e(N) &= \sum_{n=N+1}^{+\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \sum_{j=1}^{m} \binom{n}{k} s_j^k (1-s_j)^{n-k} b^{(j)}(n,k) \mathbf{1}_{\{r_{j-1}t \leq s < r_j t\}} \\ &\leq \sum_{n=N+1}^{+\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \end{aligned}$$

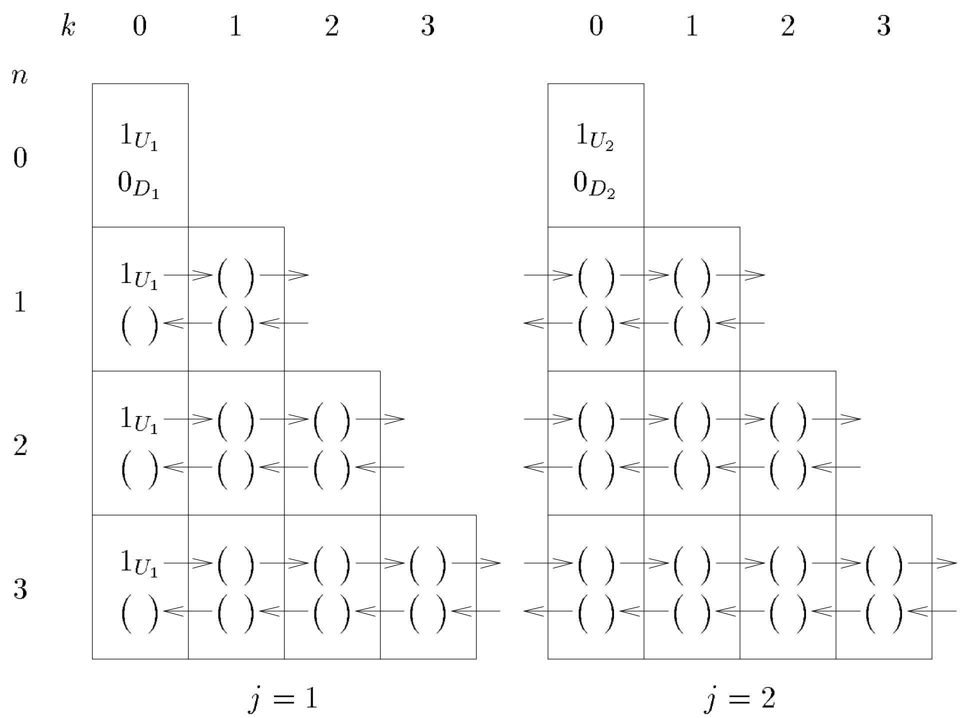

Figure 1: In cell (n,k) the vectors $b_{U_j}(n,k)$ and $b_{D_j}(n,k)$.

$$
\begin{aligned}
&= 1 - \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \\
&\leq \varepsilon/2
\end{aligned}
$$

Another truncation can be performed when the value of s is such that $r_{m-1}t < s < r_m t$. In this case, we have

$$
\begin{aligned}
\mathbb{P}\{Y_t > s\} &= \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} s_m^k (1-s_m)^{n-k} b^{(m)}(n,k) + e(N) \\
&= \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} s_m^{n-k} (1-s_m)^{k} b^{(m)}(n,n-k) + e(N)
\end{aligned}
$$

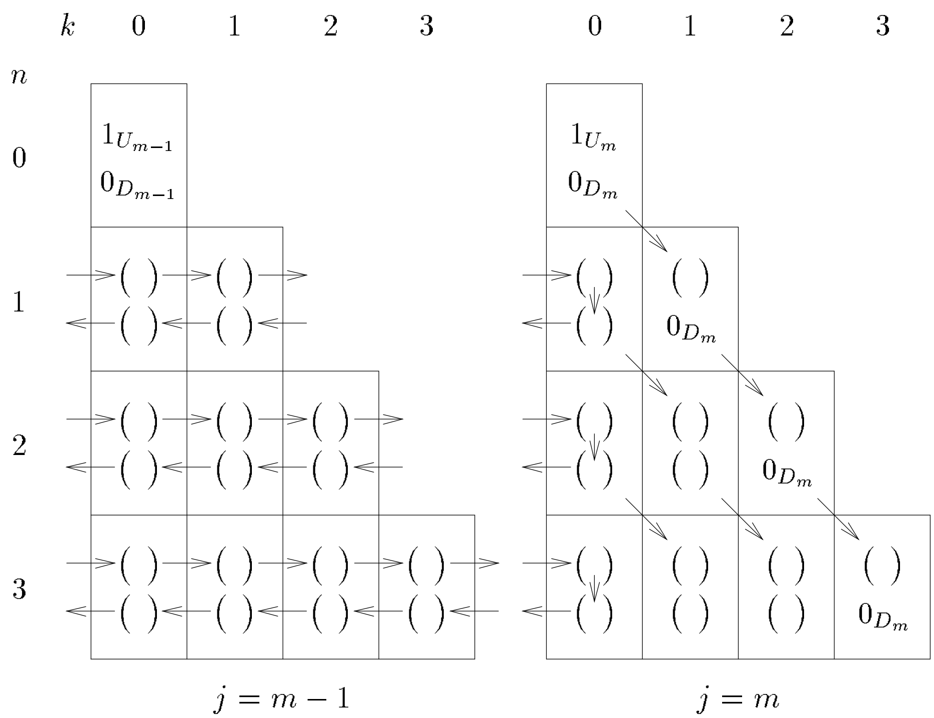

Figure 2: In cell (n,k) the vectors $b_{U_j}(n,k)$ and $b_{D_j}(n,k)$.

$$
\begin{aligned}
&= \sum_{k=0}^{N}\sum_{n=k}^{N} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\binom{n}{k} s_m^{n-k}(1-s_m)^k b^{(m)}(n,n-k) + e(N) \\
&= \sum_{k=0}^{C}\sum_{n=k}^{N} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\binom{n}{k} s_m^{n-k}(1-s_m)^k b^{(m)}(n,n-k) + e_1(N,C) + e(N)
\end{aligned}
$$

where $e_1(N,C)$ verifies

$$
\begin{aligned}
e_1(N,C) &= \sum_{k=C+1}^{N}\sum_{n=k}^{N} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\binom{n}{k} s_m^{n-k}(1-s_m)^k b^{(m)}(n,n-k) \\
&\le \sum_{k=C+1}^{N}\sum_{n=k}^{N} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\binom{n}{k} s_m^{n-k}(1-s_m)^k
\end{aligned}
$$

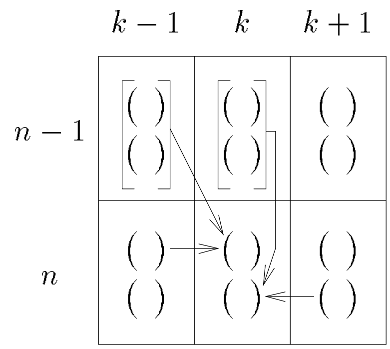

Figure 3: Computation of cell (n,k).

$$
\begin{aligned}
&= \sum_{k=C+1}^{N} \sum_{n=k}^{N} e^{-\lambda t s_m} \frac{(\lambda t s_m)^{n-k}}{(n-k)!} e^{-\lambda t(1-s_m)} \frac{(\lambda t(1-s_m))^k}{k!} \\
&= \sum_{k=C+1}^{N} e^{-\lambda t(1-s_m)} \frac{(\lambda t(1-s_m))^k}{k!} \sum_{n=k}^{N} e^{-\lambda t s_m} \frac{(\lambda t s_m)^{n-k}}{(n-k)!} \\
&\le \sum_{k=C+1}^{N} e^{-\lambda t(1-s_m)} \frac{(\lambda t(1-s_m))^k}{k!} \\
&\le 1 - \sum_{k=0}^{C} e^{-\lambda t(1-s_m)} \frac{(\lambda t(1-s_m))^k}{k!}
\end{aligned}
$$

so, the truncation step C is chosen such that

$$
C = \min\left\{ c \in \mathbb{N} \;\middle|\; \sum_{h=0}^{c} e^{-\lambda t(1-s_m)} \frac{((\lambda t(1-s_m))^h}{h!} \ge 1 - \frac{\varepsilon}{2} \right\}. \tag{3}
$$

In practise the value of s must be very close to $r_m t$ since the requirement is generally that the random variable Y_t is close to its maximum value $r_m t$ with a probability close to 1. Since we have

$$
1 - s_m = \frac{r_m t - s}{(r_m - r_{m-1})t},
$$

the value of C will be small with respect to the value of N, when s is near from $r_m t$. The global computational scheme using the truncation step C is shown in Fig. 4, where only the gray part has to be computed.

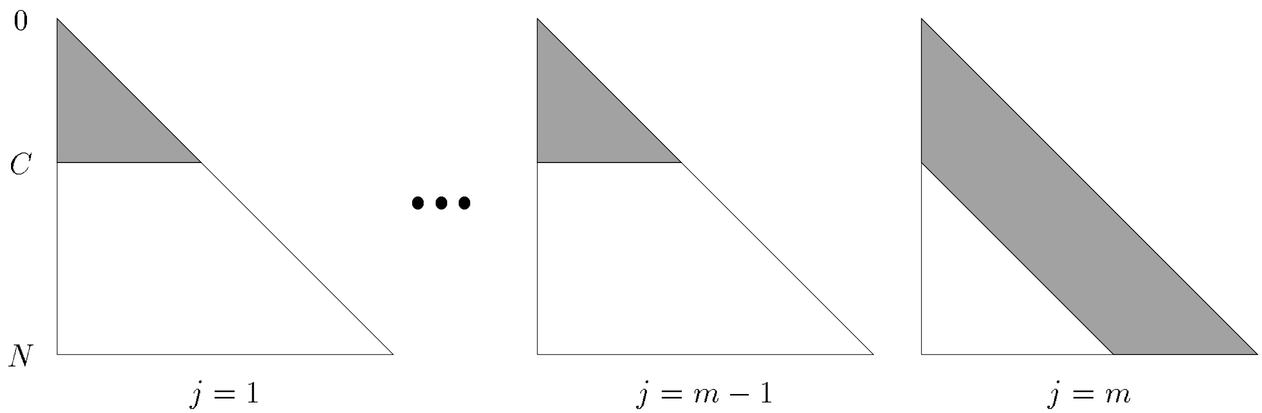

Figure 4: In gray, the computed area.

In order to compute the distribution of Y_t, the main effort is in the computation of the vectors $b_{U_j}(n,k)$ and $b_{D_j}(n,k)$. If d denotes the connectivity degree of matrix P, that is the maximum number of nonzero entries in each row, the computation of the two vectors $b_{U_j}(n,k)$ and $b_{D_j}(n,k)$, that is of one cell of a triangle, is $O(dM)$. The number of cells that have to be computed (see Fig. 4) is equal to

$$m\frac{(C+1)(C+2)}{2} + (N-C)(C+1),$$

where we set $C = N$ when the value of s is such that $s \leq r_{m-1}t$. The total computational effort required is then $O(dM[C(N-C) + mC^2/2])$. Concerning the storage requirements, it is easy to see from Fig. 1, Fig. 2 and Fig. 4 that the storage complexity is $O([(m-1)C + N]M)$.

The total computational effort required by the use of the method in [15] is $O(dMmN^2/2)$ and the storage requirements is $O(mMN)$ (which is the same as our algorithm when $C = N$).

The total computational effort required by the use of the method in [16] is stated to be $O(dM\theta N^2)$, where θ is an integer smaller than $m/2$ and equal or near to 1 in most cases. The storage requirements is $O(MN)$.

Thus the algorithm proposed in this paper compares favorably with the methods of [15] and [16] when we wish to evaluate the upper tail of the distribution of Y_t, which is the case for many performability models. Another improvement in our algorithm leads in its numerical stability since it deals only with positive number bounded by 1. Moreover, if we evaluate the lower tail of the distribution of Y_t (that is $\mathbb{P}\{Y_t > s\}$, for small values of s), the truncation step

C can be replaced by another truncation step C' as shown in Appendix B; the corresponding complexity is simply obtained by replacing C by C'.

Some values of the truncation steps N and C are given in the next section.

4 Application to a fault-tolerant computer system

In order to illustrate the paper, this section presents a model of an architecture of a fault tolerant shared memory multiprocessor which has been proposed in [19].

It consists of n CPU, a bus, and of a Recoverable Shared Memory (RSM). Each CPU is composed with two processors in active redundancy and their outputs are compared in order to detect failures. Each processor accesses the shared memory through a private cache which contains the data the most recently used by the processor. This architecture has been designed to require specialized hardware only for the RSM, so standard processors, caches and cache coherence protocols can be used. The backward recovery protocol used in this architecture to tolerate some processor failures is implemented by the RSM. The basic mechanism in the RSM to provide backward recovery is to maintain two copies for each memory location: a current copy accessed by the CPU's and a recovery copy corresponding to the previous recovery point. When a recovery point is established, the current copy is flushed on the recovery copy, so that they both contain the same data. Subsequent updates to a location are made to only one of the two copies. The other copy keeps the data that was in the location at the last recovery point instant.

We assume here that the bus and the RSM are perfectly reliable. When a fault occurs in a processor, it can be identified as a transient or a permanent fault, for example by running diagnostic checks on the faulty CPU. In the case of a transient fault, the faulty CPU will be still used in the system but in the case of a permanent fault, the faulty CPU will not be used further. The failure rate of each CPU is denoted by β, and a fault is assumed to be transient with probability d and permanent with probability $1 - d$. After each occurence of a fault, the backward recovery protocol is executed and its duration is assumed to be exponentially distributed with rate μ. Moreover, it is assumed that with probability c the backward recovery protocol reconfigures the system correctly, and that it fails with probability $1 - c$. This factor c is usually called the coverage factor of the system.

The Markov process so generated is shown in Fig. 5, when the number of CPU is $n = 3$.

The state i, $1 \le i \le n$, corresponds to the state of the system in which i CPU are operationnal. In this state, the rate at which a fault occurs is $i\beta$. The state d_i, $1 \le i \le n$, corresponds to the state of the system in which the backward recovery protocol try to recon-

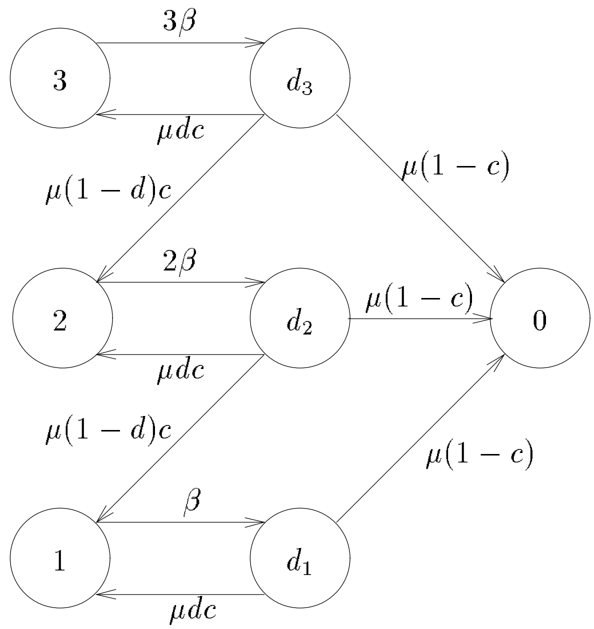

Figure 5: The Markov process for 3 CPU.

figure the system with i CPU if the fault was transient and with $i-1$ CPU, when $i>1$, if the fault was permanent. State 0 denotes the down state of the system.

The cost of the fault tolerance in this architecture is mainly due to the establishment of recovery points. This cost has been evaluated to 30% of the power of the system, in the worst case. If we assume that a standard architecture with n operational processors, which does not tolerate any fault, has a power equal to n, the reward rate associated to state i, $1 \le i \le n$, in our model is chosen to be $r_i = 0.7i$. The reward rates associated to the other states are equal to 0. With this reward structure associated to our model the performability distribution $\mathbb{P}\{Y_t/t > r\}$ represents the probability that the power of the architecture during $[0,t]$ averaged over time is greater than r; the value of r being in the interval $[0, r_n[$.

The values of the parameters are $c = 0.95$, $d = 0.9$ and $\mu = 1$ per second, that is, the mean execution of the backward recovery protocol is equal to 1 second. With these numerical values, Fig. 6 and Fig. 7 show the probability that the power of the system is greater than 99.99% ($\delta = 0.9999$) of its maximum power for a one day mission time as a function of the number n of CPU's and for different values of the failure rate β.

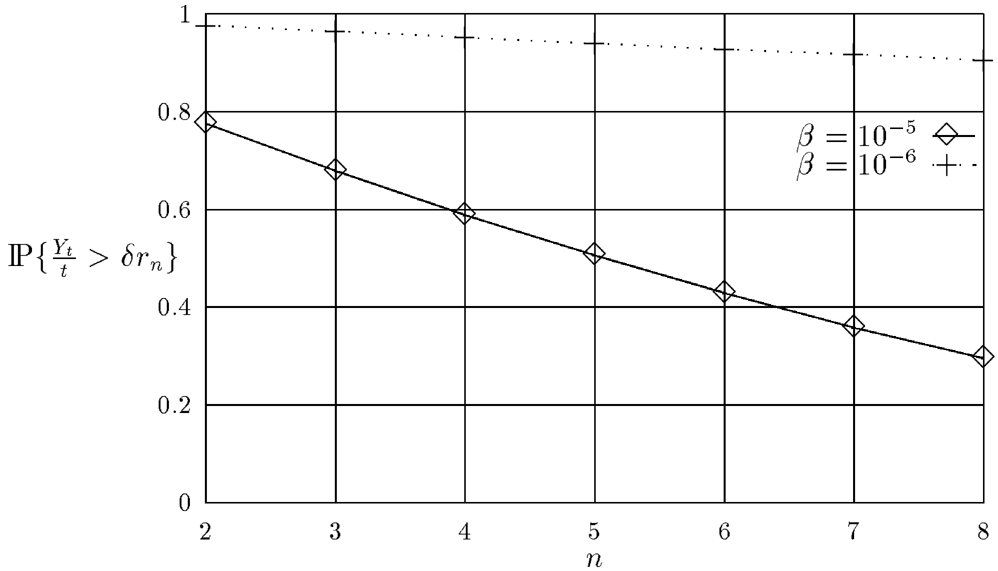

Figure 6: A one day mission time

Note that when $\beta = 10^{-5}$ the probability of achieving more than 99.99% of the maximum power of the system is smaller than 0.8 for every n. To get such a probability greater than 0.95, the number of CPU's must be at most equal to 4 if $\beta = 10^{-6}$.

For smaller values of the failure rate β (see Fig. 7, to achieve more than 99.99% of the maximum power of the system with a probability greater than 0.997, the only possibility is to have a system with 2 CPU's if $\beta = 10^{-7}$ and the number of CPU's can be equal to 8 when $\beta = 10^{-8}$.

Finally, for all these computations the value of the truncation step N is $N = 87701$ and the values of truncation step C increase from $C = 38$ (for $n = 2$) to $C = 109$ (for $n = 8$). These small values of C with respect to N show the improvement in computational cost of our algorithm with respect to the ones developed in [15] and [16].

5 Conclusions

The proposed method for evaluating the accumulated reward distribution for fault-tolerant computer systems is based on the uniformization technique and leads to a new algorithm with

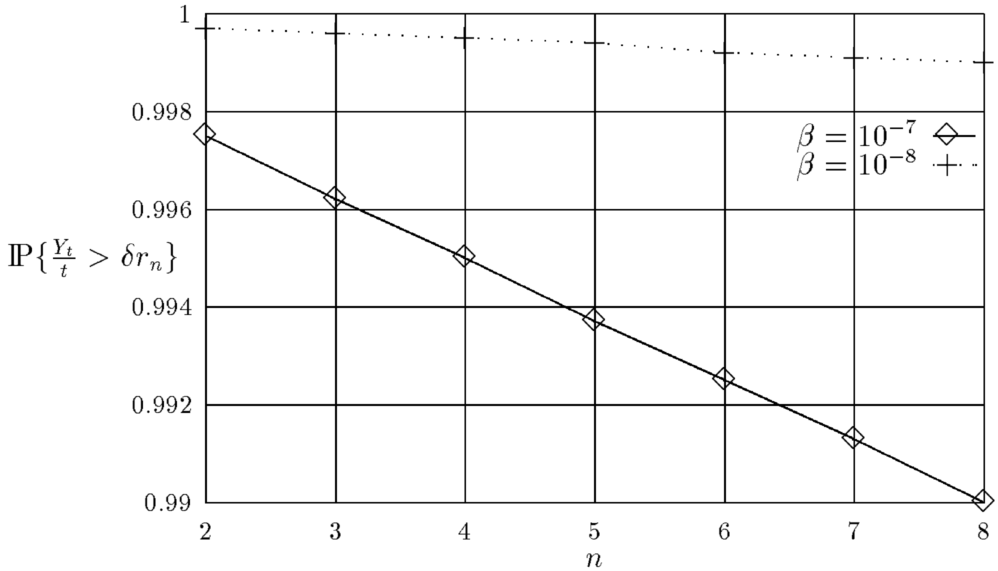

Figure 7: A one day mission time

a low polynomial computational complexity. The number of operations is linear in the number of states of the system and linear in the number of rewards. Its main advantage with respect to existing algorithms, is that it is linear in the truncation step N and quadratic in the truncation step C which is very small in comparison to N, when the upper tail performability distribution is considered. Another advantage of this algorithm is that it deals only with positive numbers bounded by 1, improving so its stability.

Appendix A

To prove Theorem 3.1, we first need two technical lemmas which can be easily proved by recurrence. Let us define for every $l, j \in \{1, \ldots, m\}$ the numbers

$$v_{l,j-1} = \frac{s - r_{j-1}t}{r_l - r_{j-1}} \qquad \text{and} \qquad V_{n,k}(j) = \frac{(v_{j,j-1})^k (v_{j-1,j})^{n-k}}{k!(n-k)!} \mathbf{1}_{\{r_{j-1}t \le s < r_j t\}},$$

and the functions

$$f_{l,j}(u) \quad = \quad \frac{r_j(t-u) - (s - r_l u)}{r_j - r_{j-1}}$$

$$
\begin{aligned}
&= \frac{r_j t - s}{r_j - r_{j-1}} + \frac{r_l - r_j}{r_j - r_{j-1}} u \\
&= v_{j-1,j} + \frac{r_l - r_j}{r_j - r_{j-1}} u,
\end{aligned}
$$

and

$$
\begin{aligned}
g_{l,j}(u) &= \frac{(s - r_l u) - r_{j-1}(t-u)}{r_j - r_{j-1}} \\
&= \frac{s - r_{j-1} t}{r_j - r_{j-1}} - \frac{r_l - r_{j-1}}{r_j - r_{j-1}} u \\
&= v_{j,j-1} - \frac{r_l - r_{j-1}}{r_j - r_{j-1}} u,
\end{aligned}
$$

where $t \geq 0$ and $s \in [0, r_m t[$. Note that $g_{l,j}(u) = t - u - f_{l,j}(u)$ and $v_{l,j-1} = t - v_{j-1,l}$.

Lemma 5.1 *For every $l, j \in \{1, 2, \ldots, m\}$, for every $n \geq 1$ and $0 \leq k \leq n-1$, we have*

- *if $j \leq l$*

$$
\begin{aligned}
&\int_0^{\frac{s}{r_l}} \frac{[g_{l,j}(u)]^k [(f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\ f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s < r_l t\}} \\
&= \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \sum_{p=k+1}^{n} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{p-k-1} V_{n,p}(j) \\
&+ \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{n-1-k} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_j t \leq s < r_l t\}}
\end{aligned}
$$

- *if $j > l$*

$$
\int_0^{\frac{s}{r_l}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\ f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s < r_l t\}} = 0
$$

Proof. Let us first consider the indicator function in the integral. If $j \leq l$, we have

1) when $j=l$, $f_{l,j}(u)=\dfrac{r_l t-s}{r_l-r_{l-1}}$ and $g_{l,j}(u)=\dfrac{s-r_{l-1}t}{r_l-r_{l-1}}-u=v_{l,l-1}-u$. So,

$$\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}=\mathbf{1}_{\{s<r_l t, u\leq v_{l,l-1}\}},$$

and since $(u\geq 0$ and $u\leq v_{l,l-1})\Longrightarrow v_{l,l-1}\geq 0\Longrightarrow s\geq r_{l-1}t$, we get

$$\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}=\mathbf{1}_{\{0\leq u\leq v_{l,l-1}\}}\mathbf{1}_{\{r_{l-1}t\leq s<r_l t\}}.$$

2) when $j<l$, we get in the same way,

$$\begin{aligned}
&\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}\\
=\;&\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\\
=\;&\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\mathbf{1}_{\{s<r_l t\}}\text{ , since } v_{l,j}<v_{l,j-1}\Longleftrightarrow s<r_l t.
\end{aligned}$$

Now $(u\geq 0$ and $u\leq v_{l,j-1})\Longrightarrow v_{l,j-1}\geq 0\Longrightarrow s\geq r_{j-1}t$. So, we get

$$\begin{aligned}
&\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}\\
=\;&\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\mathbf{1}_{\{r_{j-1}t\leq s<r_l t\}}\\
=\;&\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\mathbf{1}_{\{r_{j-1}t\leq s<r_l t\}}\left(\mathbf{1}_{\{v_{l,j}<0\}}+\mathbf{1}_{\{v_{l,j}\geq 0\}}\right)\\
=\;&\mathbf{1}_{\{0\leq u\leq v_{l,j-1}\}}\mathbf{1}_{\{r_{j-1}t\leq s<r_j t\}}+\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\mathbf{1}_{\{r_j t\leq s<r_l t\}}.
\end{aligned}$$

These two cases ($j=l$ and $j<l$) can be grouped in only one by choosing as a convention that $\mathbf{1}_{\{v_{l,j}<u\leq v_{l,j-1}\}}\mathbf{1}_{\{r_j t\leq s<r_l t\}}=0$ when $j=l$. Using this convention, we get

$$\begin{aligned}
&\int_0^{\frac{s}{r_l}}\frac{[g_{l,j}(u)]^k[f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!}\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}du\mathbf{1}_{\{s<r_l t\}}\\
=\;&\int_0^{v_{l,j-1}}\frac{[g_{l,j}(u)]^k[f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!}du\mathbf{1}_{\{r_{j-1}t\leq s<r_j t\}}\\
+\;&\int_{v_{l,j}}^{v_{l,j-1}}\frac{[g_{l,j}(u)]^k[f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!}du\mathbf{1}_{\{r_j t\leq s<r_l t\}}.
\end{aligned}$$

Let $J_{n-1,k}$ (resp. $L_{n-1,k}$) be the first term (resp. the second term) of the right hand side of this relation. Integrating by parts, we easily get for $k < n-1$

$$J_{n-1,k} = \frac{r_j - r_{j-1}}{r_l - r_{j-1}} V_{n,k+1}(j) + \frac{r_l - r_j}{r_l - r_{j-1}} J_{n-1,k+1}$$

and

$$L_{n-1,k} = \frac{r_l - r_j}{r_l - r_{j-1}} L_{n-1,k+1}.$$

The initial conditions of these terms are given by

$$J_{n-1,n-1} = \frac{r_j - r_{j-1}}{r_l - r_{j-1}} V_{n,n}(j),$$

and

$$L_{n-1,n-1} = \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_j t \le s < r_l t\}}.$$

It follows that

$$J_{n-1,k} = \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \sum_{p=k+1}^{n} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{p-k-1} V_{n,p}(j),$$

and

$$L_{n-1,k} = \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{n-1-k} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_j t \le s < r_l t\}},$$

which completes the proof in the case where $j \le l$.

Let us now consider the case $j > l$.

3) when $j = l+1$, $f_{l,j}(u) = v_{l,l+1} - u$ and $g_{l,j}(u) = v_{l+1,l}$. So,

$$\mathbf{1}_{\{g_{l,j}(u) \ge 0,\ f_{l,j}(u) > 0\}} = \mathbf{1}_{\{r_l t \le s, u < v_{l,l+1}\}},$$

and since $(u \ge 0$ and $u < v_{l,l+1}) \Longrightarrow v_{l,l+1} > 0 \Longrightarrow s < r_{l+1} t$, we get

$$\mathbf{1}_{\{g_{l,j}(u) \ge 0,\ f_{l,j}(u) > 0\}} = \mathbf{1}_{\{0 \le u < v_{l,l+1}\}} \mathbf{1}_{\{r_l t \le s < r_{l+1} t\}}.$$

4) when $j > l+1$, we get in the same way,

$$\begin{aligned}
&\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}} \\
&= \mathbf{1}_{\{v_{l,j-1}\leq u<v_{l,j}\}} \\
&= \mathbf{1}_{\{v_{l,j-1}\leq u<v_{l,j}\}}\mathbf{1}_{\{s\geq r_l t\}}\ \text{, since } v_{l,j-1}\geq v_{l,j} \Longleftrightarrow s\geq r_l t.
\end{aligned}$$

Now, since $(u \geq 0 \text{ and } u < v_{l,j}) \Longrightarrow v_{l,j} > 0 \Longrightarrow s < r_j t$, we get

$$\begin{aligned}
&\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}} \\
&= \mathbf{1}_{\{v_{l,j-1}\leq u<v_{l,j}\}}\mathbf{1}_{\{r_l t\leq s<r_j t\}} \\
&= \mathbf{1}_{\{v_{l,j-1}\leq u<v_{l,j}\}}\mathbf{1}_{\{r_l t\leq s<r_j t\}}\left(\mathbf{1}_{\{v_{l,j-1}\leq 0\}}+\mathbf{1}_{\{v_{l,j-1}>0\}}\right) \\
&= \mathbf{1}_{\{0\leq u<v_{l,j}\}}\mathbf{1}_{\{r_{j-1} t\leq s<r_j t\}}+\mathbf{1}_{\{v_{l,j-1}\leq u<v_{l,j}\}}\mathbf{1}_{\{r_l t\leq s<r_{j-1} t\}}.
\end{aligned}$$

Now, since $\mathbf{1}_{\{s\geq r_l t\}}\mathbf{1}_{\{s<r_l t\}} = 0$, we have $\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}\mathbf{1}_{\{s<r_l t\}} = 0$ and the second part of the proof is completed.

□

Lemma 5.2 *For every $l, j \in \{1, 2, \ldots, m\}$, for every $n \geq 1$ and $0 \leq k \leq n-1$, we have*

- *if $j > l$*

$$\begin{aligned}
&\int_0^{v_{l,m}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!}\mathbf{1}_{\{g_{l,j}(u)\geq 0,\ f_{l,j}(u)>0\}}du\,\mathbf{1}_{\{s\geq r_l t\}} \\
&= \frac{r_j - r_{j-1}}{r_j - r_l}\sum_{p=0}^{k}\left(\frac{r_{j-1}-r_l}{r_j-r_l}\right)^{k-p} V_{n,p}(j) \\
&+ \frac{r_j - r_{j-1}}{r_{j-1} - r_l}\left(\frac{r_j-r_l}{r_{j-1}-r_l}\right)^{n-1-k}\frac{(v_{j,l})^n}{n!}\mathbf{1}_{\{r_l t\leq s<r_{j-1} t\}}
\end{aligned}$$

where the second term of the right hand side is equal to 0 when $j = l+1$.

- *if* $j \le l$

$$\int_0^{v_{l,m}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u)\ge 0,\ f_{l,j}(u)>0\}} du \mathbf{1}_{\{s \ge r_l t\}} = 0.$$

Proof. From the previous lemma (cases 1) and 2)), we have that when $j \le l$, $\mathbf{1}_{\{g_{l,j}(u)\ge 0,\ f_{l,j}(u)>0\}} \mathbf{1}_{\{s \ge r_l t\}} = 0$, which proves the second part of this lemma. When $j > l$ (cases 3) and 4) of previous lemma), we have

$$\begin{aligned} & \int_0^{v_{l,m}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u)\ge 0,\ f_{l,j}(u)>0\}} du \mathbf{1}_{\{s \ge r_l t\}} \\ = & \int_0^{v_{l,j}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} du \mathbf{1}_{\{r_{j-1} t \le s < r_j t\}} \\ + & \int_{v_{l,j-1}}^{v_{l,j}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} du \mathbf{1}_{\{r_l t \le s < r_{j-1} t\}}. \end{aligned}$$

Let $J_{n-1,k}$ (resp. $L_{n-1,k}$) be the first term (resp. the second term) of the right hand side of this relation. Integrating by parts, we easily get

$$J_{n-1,k} = \frac{r_j - r_{j-1}}{r_j - r_l} V_{n,k}(j) + \frac{r_{j-1} - r_l}{r_j - r_l} J_{n-1,k-1} \qquad \text{for } k > 0,$$

and

$$L_{n-1,k} = \frac{r_j - r_l}{r_{j-1} - r_l} L_{n-1,k+1} \qquad \text{for } k < n-1.$$

The initial conditions of these terms are given by

$$J_{n-1,0} = \frac{r_j - r_{j-1}}{r_j - r_l} V_{n,0}(j),$$

and

$$L_{n-1,n-1} = \frac{r_j - r_{j-1}}{r_{j-1} - r_l} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_l t \le s < r_{j-1} t\}}.$$

It follows that

$$J_{n-1,k} = \frac{r_j - r_{j-1}}{r_j - r_l} \sum_{p=0}^{k} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{k-p} V_{n,p}(j),$$

and

$$L_{n-1,k} = \frac{r_j - r_{j-1}}{r_{j-1} - r_l} \left(\frac{r_j - r_l}{r_{j-1} - r_l} \right)^{n-1-k} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_l t \leq s < r_{j-1} t\}},$$

which completes the proof in the case where $j > l$.

□

In what follows, using the notation introduced, we show that the solution of the forward renewal equation of Theorem 2.1 can be written as

$$F_{B_l}(s,t,n) = \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{m} b_{B_l}^{(j)}(n,k) V_{n,k}(j), \tag{4}$$

where the column vectors $b_{B_l}^{(j)}(n,k)$ are recursively given in Theorem 3.1.

By convention,

$$\text{a sum } \sum_{i=x}^{y} (\ldots) \text{ will be equal to 0 if } x > y. \tag{5}$$

For $n = 0$, that is 0 transitions during $[0,t]$, we must have by definition of Y_t

$$\begin{aligned} F_{B_l}(s,t,0) &= e^{-\lambda t} 1_{B_l} \mathbf{1}_{\{s < r_l t\}} \\ &= e^{-\lambda t} 1_{B_l} \sum_{j=1}^{l} \mathbf{1}_{\{r_{j-1} t \leq s < r_j t\}}. \end{aligned}$$

So, by identification, with relation (4), we have

$$b_{B_l}^{(j)}(0,0) = \begin{cases} 1_{B_l} & \text{if } 1 \leq j \leq l \\ 0_{B_l} & \text{if } l+1 \leq j \leq m \end{cases}$$

Now, for $n \geq 1$, by replacing expression (4) in Theorem 2.1, we get using the notation introduced above

$$F_{B_l}(s,t,n)$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{m} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \int_0^{\frac{s}{r_l}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\ f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s < r_l t\}}$$

$$+\lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{m} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \int_0^{v_{l,m}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\ f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s \geq r_l t\}}$$

$$+\lambda^n e^{-\lambda t} \frac{(t - \frac{s}{r_l})^n}{(n)!} 1_{B_l} \mathbf{1}_{\{s < r_l t\}}.$$

Let us denote respectively by γ_1, γ_2 and γ_3 the three terms of the right hand side of this relation, so that

$$F_{B_l}(s,t,n) = \gamma_1 + \gamma_2 + \gamma_3.$$

Expansion of term γ_1

Using lemma 5.1, we obtain

$$\gamma_1$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{m} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \int_0^{\frac{s}{r_l}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\ f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s < r_l t\}}$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{l} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \sum_{p=k+1}^{n} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{p-k-1} V_{n,p}(j)$$

$$+\lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{l-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{n-1-k} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_j t \leq s < r_l t\}}.$$

Interchanging in the first term the summation over index p and the summation over index k, we obtain

$$\gamma_1 = \lambda^n e^{-\lambda t} \sum_{p=1}^{n} \sum_{j=1}^{l} \sum_{k=0}^{p-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{p-k-1} V_{n,p}(j)$$

$$+\lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{l-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{n-1-k} \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_j t \le s < r_l t\}}.$$

For a further identification, it is more convenient to interchange indices p and k in both terms of γ_1 and to replace index j by a new index q in the second term of the right hand side of γ_1. We then get

$$\gamma_1 = \lambda^n e^{-\lambda t} \sum_{k=1}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{k-p-1} V_{n,k}(j)$$

$$+\lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{q=1}^{l-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_q}{r_l - r_{q-1}} \right)^{n-1-p} \frac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_q t \le s < r_l t\}}.$$

With the convention (5), we can write $\displaystyle\sum_{k=1}^{n} \sum_{p=0}^{k-1} (\ldots) = \sum_{k=0}^{n} \sum_{p=0}^{k-1} (\ldots)$. So,

$$\gamma_1 = \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{k-p-1} V_{n,k}(j)$$

$$+\lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{q=1}^{l-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_q}{r_l - r_{q-1}} \right)^{n-1-p} \frac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_q t \le s < r_l t\}}.$$

In order to identify γ_1 with relation (4), we need to write product $\dfrac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_q t \le s < r_l t\}}$ as a function of $V_{n,k}(j)$. This can be done as follows. For every j, $v_{q,l}$ can be written as

$$v_{q,l} = z_j v_{j,j-1} + z_{j-1} v_{j-1,j},$$

where $z_j = \dfrac{r_l - r_j}{r_l - r_q}$ and we also have

$$\mathbf{1}_{\{r_q t \leq s < r_l t\}} = \sum_{j=q+1}^{l} \mathbf{1}_{\{r_{j-1} t \leq s < r_j t\}}.$$

We then have

$$\begin{aligned}
& \frac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_q t \leq s < r_l t\}} \\
= & \sum_{j=q+1}^{l} \frac{(z_j v_{j,j-1} + z_{j-1} v_{j-1,j})^n}{n!} \mathbf{1}_{\{r_{j-1} t \leq s < r_j t\}} \\
= & \sum_{k=0}^{n} \sum_{j=q+1}^{l} z_j^k z_{j-1}^{n-k} \frac{(v_{j,j-1})^k (v_{j-1,j})^{n-k}}{k!(n-k)!} \mathbf{1}_{\{r_{j-1} t \leq s < r_j t\}} \\
= & \sum_{k=0}^{n} \sum_{j=q+1}^{l} \left(\frac{r_l - r_j}{r_l - r_q} \right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k} V_{n,k}(j).
\end{aligned}$$

Finally, the term γ_1 is equal to :

$$\begin{aligned}
\gamma_1 = & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{k-p-1} V_{n,k}(j) \\
+ & \lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{q=1}^{l-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_q}{r_l - r_{q-1}} \right)^{n-1-p} \\
& \times \sum_{k=0}^{n} \sum_{j=q+1}^{l} \left(\frac{r_l - r_j}{r_l - r_q} \right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k} V_{n,k}(j) \\
= & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{k-p-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
+ & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=2}^{l} \sum_{q=1}^{j-1} \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_j}{r_l - r_q} \right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k}
\end{aligned}$$

$$\times \sum_{p=0}^{n-1} \left(\frac{r_l - r_q}{r_l - r_{q-1}} \right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j)$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}} \right)^{k-p-1} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j)$$

$$+ \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{q=1}^{j-1} \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_j}{r_l - r_q} \right)^{k} \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k}$$

$$\times \sum_{p=0}^{n-1} \left(\frac{r_l - r_q}{r_l - r_{q-1}} \right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j),$$

where the second equality is obtained by interchanging the summation over index q and the summation over index j and the third equality is obtained by the use of convention (5).

Expansion of term γ_2

Using lemma 5.2, we obtain

$$\gamma_2$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=1}^{m} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \int_0^{\frac{s}{r_l}} \frac{[g_{l,j}(u)]^k [f_{l,j}(u)]^{n-1-k}}{k!(n-1-k)!} \mathbf{1}_{\{g_{l,j}(u) \geq 0,\, f_{l,j}(u) > 0\}} du \mathbf{1}_{\{s \geq r_l t\}}$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=0}^{k} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{k-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) V_{n,p}(j)$$

$$+ \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+2}^{m} \frac{r_j - r_{j-1}}{r_{j-1} - r_l} \left(\frac{r_j - r_l}{r_{j-1} - r_l} \right)^{n-1-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n,k) \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_l t \leq s < r_{j-1} t\}}$$

$$= \lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{j=l+1}^{m} \sum_{k=p}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{k-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) V_{n,p}(j)$$

$$+ \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+2}^{m} \frac{r_j - r_{j-1}}{r_{j-1} - r_l} \left(\frac{r_j - r_l}{r_{j-1} - r_l} \right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n,k) \frac{(v_{j,l})^n}{n!} \mathbf{1}_{\{r_l t \leq s < r_{j-1} t\}}.$$

Following the same approach used for the expansion of term γ_1, we get

$$\begin{aligned}
\gamma_2 &= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
&+ \lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{q=l+2}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_q - r_l}{r_{q-1} - r_l} \right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n,p) \frac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_l t \le s < r_{q-1} t\}}.
\end{aligned}$$

As previously, the product $\dfrac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_l t \le s < r_{q-1} t\}}$ can be written as

$$\begin{aligned}
& \frac{(v_{q,l})^n}{n!} \mathbf{1}_{\{r_l t \le s < r_{q-1} t\}} \\
&= \sum_{j=l+1}^{q-1} \frac{(z_j v_{j,j-1} + z_{j-1} v_{j-1,j})^n}{n!} \mathbf{1}_{\{r_{j-1} t \le s < r_j t\}} \\
&= \sum_{k=0}^{n} \sum_{j=l+1}^{q-1} z_j^k z_{j-1}^{n-k} \frac{(v_{j,j-1})^k (v_{j-1,j})^{n-k}}{k!(n-k)!} \mathbf{1}_{\{r_{j-1} t \le s < r_j t\}} \\
&= \sum_{k=0}^{n} \sum_{j=l+1}^{q-1} \left(\frac{r_l - r_j}{r_l - r_q} \right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k} V_{n,k}(j).
\end{aligned}$$

Finally, the term γ_2 is equal to :

$$\begin{aligned}
\gamma_2 &= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
&+ \lambda^n e^{-\lambda t} \sum_{p=0}^{n-1} \sum_{q=l+2}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_q - r_l}{r_{q-1} - r_l} \right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n,p) \\
&\qquad \times \sum_{k=0}^{n} \sum_{j=l+1}^{q-1} \left(\frac{r_l - r_j}{r_l - r_q} \right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q} \right)^{n-k} V_{n,k}(j) \\
&= \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l} \right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j)
\end{aligned}$$

$$
\begin{aligned}
&+\ \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=l+1}^{m-1} \sum_{q=j+1}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_j - r_l}{r_q - r_l}\right)^k \left(\frac{r_{j-1} - r_l}{r_q - r_l}\right)^{n-k} \\
&\qquad\qquad \times \sum_{p=0}^{n-1} \left(\frac{r_q - r_l}{r_{q-1} - r_l}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j) \\
&=\ \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l}\right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
&+\ \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=l+1}^{m} \sum_{q=j+1}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_j - r_l}{r_q - r_l}\right)^k \left(\frac{r_{j-1} - r_l}{r_q - r_l}\right)^{n-k} \\
&\qquad\qquad \times \sum_{p=0}^{n-1} \left(\frac{r_q - r_l}{r_{q-1} - r_l}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j),
\end{aligned}
$$

where the second equality is obtained by interchanging the summation over index q and the summation over index j and the third equality is obtained by the use of convention (5).

Expansion of term γ_3

Note that as previously we can write, for every j

$$t - \frac{s}{r_l} = \frac{r_l - r_j}{r_l} v_{j,j-1} + \frac{r_l - r_{j-1}}{r_l} v_{j-1,j},$$

and

$$\mathbf{1}_{\{s<r_l t\}} = \sum_{j=1}^{l} \mathbf{1}_{\{r_{j-1}t \le s < r_j t\}}.$$

It follows that

$$
\begin{aligned}
\gamma_3 &= \lambda^n e^{-\lambda t} 1_{B_l} \frac{(t - \frac{s}{r_l})^n}{n!} \mathbf{1}_{\{s<r_l t\}} \\
&= \lambda^n e^{-\lambda t} 1_{B_l} \sum_{j=1}^{l} \frac{\left(\frac{r_l - r_j}{r_l} v_{j,j-1}) + \frac{r_l - r_{j-1}}{r_l} v_{j-1,j}\right)^n}{n!} \mathbf{1}_{\{r_{j-1}t \le s < r_j t\}}
\end{aligned}
$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \left(\frac{r_l - r_j}{r_l}\right)^k \left(\frac{r_l - r_{j-1}}{r_l}\right)^{n-k} 1_{B_l} \frac{(v_{j,j-1})^k (v_{j-1,j})^{n-k}}{k!(n-k)!} \mathbf{1}_{\{r_{j-1}t \le s < r_j t\}}$$

$$= \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \left(\frac{r_l - r_j}{r_l}\right)^k \left(\frac{r_l - r_{j-1}}{r_l}\right)^{n-k} 1_{B_l} V_{n,k}(j).$$

Expression of $F_{B_l}(s,t,n) = \gamma_1 + \gamma_2 + \gamma_3$

By adding up the three terms γ_1, γ_2, and γ_3, $F_{B_l}(s,t,n)$ becomes be equal to :

$$\begin{aligned}
& F_{B_l}(s,t,n) \\
= \; & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{p=0}^{k-1} \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}}\right)^{k-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
+ \; & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \sum_{q=1}^{j-1} \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_j}{r_l - r_q}\right)^k \left(\frac{r_l - r_{j-1}}{r_l - r_q}\right)^{n-k} \\
& \qquad \times \sum_{p=0}^{n-1} \left(\frac{r_l - r_q}{r_l - r_{q-1}}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j) \\
+ \; & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=1}^{l} \left(\frac{r_l - r_j}{r_l}\right)^k \left(\frac{r_l - r_{j-1}}{r_l}\right)^{n-k} 1_{B_l} V_{n,k}(j) \\
+ \; & \lambda^n e^{-\lambda t} \sum_{k=0}^{n-1} \sum_{j=l+1}^{m} \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l}\right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p) V_{n,k}(j) \\
+ \; & \lambda^n e^{-\lambda t} \sum_{k=0}^{n} \sum_{j=l+1}^{m} \sum_{q=j+1}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_j - r_l}{r_q - r_l}\right)^k \left(\frac{r_{j-1} - r_l}{r_q - r_l}\right)^{n-k} \\
& \qquad \times \sum_{p=0}^{n-1} \left(\frac{r_q - r_l}{r_{q-1} - r_l}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p) V_{n,k}(j).
\end{aligned}$$

If we now identify this relation with relation (4), we get the following expression for the coefficients $b_{B_i}^{(j)}(n,k)$ for every $k \in \{0, 1, \ldots, n\}$:

- for $j = 1, \ldots, l$

$$b_{B_l}^{(j)}(n,k) = \sum_{p=0}^{k-1} \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \left(\frac{r_l - r_j}{r_l - r_{j-1}}\right)^{k-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p)$$

$$+ \sum_{q=1}^{j-1} \frac{r_q - r_{q-1}}{r_l - r_{q-1}} \left(\frac{r_l - r_j}{r_l - r_q}\right)^{k} \left(\frac{r_l - r_{j-1}}{r_l - r_q}\right)^{n-k} \sum_{p=0}^{n-1} \left(\frac{r_l - r_q}{r_l - r_{q-1}}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p)$$

$$+ \left(\frac{r_l - r_j}{r_l}\right)^{k} \left(\frac{r_l - r_{j-1}}{r_l}\right)^{n-k} 1_{B_l}$$

- for $j = l+1, \ldots, m$

$$b_{B_l}^{(j)}(n,k) = \sum_{p=k}^{n-1} \frac{r_j - r_{j-1}}{r_j - r_l} \left(\frac{r_{j-1} - r_l}{r_j - r_l}\right)^{p-k} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,p)$$

$$+ \sum_{q=j+1}^{m} \frac{r_q - r_{q-1}}{r_{q-1} - r_l} \left(\frac{r_j - r_l}{r_q - r_l}\right)^{k} \left(\frac{r_{j-1} - r_l}{r_q - r_l}\right)^{n-k} \sum_{p=0}^{n-1} \left(\frac{r_q - r_l}{r_{q-1} - r_l}\right)^{n-1-p} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(q)}(n-1,p).$$

It's easy now to verify that this expression of coefficients $b_{B_l}^{(j)}(n,k)$ leads to the recurrences mentionned in the main theorem 3.1 which are :

for $j \leq l \leq m$ **and** $1 \leq k \leq n$

$$\begin{aligned} b_{B_l}^{(j)}(n,k) &= \frac{r_l - r_j}{r_l - r_{j-1}} b_{B_l}^{(j)}(n,k-1) + \frac{r_j - r_{j-1}}{r_l - r_{j-1}} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k-1) \\ b_{B_l}^{(j)}(n,0) &= \begin{cases} 1_{B_l} & \text{for } j = 1 \\ b_{B_l}^{(j-1)}(n,n) & \text{for } j > 1 \end{cases} \end{aligned}$$

for $0 \leq l \leq j-1$ **and** $0 \leq k \leq n-1$

$$\begin{aligned} b_{B_l}^{(j)}(n,k) &= \frac{r_{j-1} - r_l}{r_j - r_l} b_{B_l}^{(j)}(n,k+1) + \frac{r_j - r_{j-1}}{r_j - r_l} \sum_{i=0}^{m} P_{B_l B_i} b_{B_i}^{(j)}(n-1,k) \\ b_{B_l}^{(j)}(n,n) &= \begin{cases} b_{B_l}^{(j+1)}(n,0) & \text{for } j < m \\ 0_{B_l} & \text{for } j = m \end{cases} \end{aligned}$$

So the proof of the theorem 3.1 is completed.

□

Appendix B

Suppose that we wish to calculate the performability distribution, $\mathbb{P}\{Y_t > s\}$ for small values of s such that $0 < s < r_1 t$. Note that in the first triangle of cells which corresponds to the case $j = 1$, each column k can be computed provided that column $k-1$ has been computed and cell (k, k) has been also computed. This can be seen easily on the recurrence of Theorem 3.1. It follows that a truncation step C' can be evaluated easily in the case where $0 < s < r_1 t$.

From Theorem 3.1, we have in this case

$$\mathbb{P}\{Y_t > s\} = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k)$$

where $s_1 = \dfrac{s}{r_1 t}$. For a given value of the error tolerance ε, we have as done before

$$\mathbb{P}\{Y_t > s\} = \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k) + e(N)$$

where $e(N)$ is given by (2). Now a truncation step C' can be as follows

$$\begin{aligned}
\mathbb{P}\{Y_t > s\} &= \sum_{n=0}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k) + e(N) \\
&= \sum_{k=0}^{N} \sum_{n=k}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k) + e(N) \\
&= \sum_{k=0}^{C'} \sum_{n=k}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k) + e_1(N, C') + e(N)
\end{aligned}$$

where $e_1(N, C')$ verifies

$$e_1(N, C') = \sum_{k=C'+1}^{N} \sum_{n=k}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \binom{n}{k} s_1^k (1-s_1)^{n-k} b^{(1)}(n,k)$$

$$
\begin{aligned}
&\leq \sum_{k=C'+1}^{N} \sum_{n=k}^{N} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \binom{n}{k} s_1^k (1-s_1)^{n-k} \\
&= \sum_{k=C'+1}^{N} \sum_{n=k}^{N} e^{-\lambda t s_1} \frac{(\lambda t s_1)^k}{k!} e^{-\lambda t(1-s_1)} \frac{(\lambda t(1-s_1))^{n-k}}{(n-k)!} \\
&= \sum_{k=C'+1}^{N} e^{-\lambda t s_1} \frac{(\lambda t s_1)^k}{k!} \sum_{n=k}^{N} e^{-\lambda t(1-s_1)} \frac{(\lambda t(1-s_1))^{n-k}}{(n-k)!} \\
&\leq \sum_{k=C'+1}^{N} e^{-\lambda t s_1} \frac{(\lambda t s_1)^k}{k!} \\
&\leq 1 - \sum_{k=0}^{C'} e^{-\lambda t s_1} \frac{(\lambda t s_1)^k}{k!}
\end{aligned}
$$

so, the truncation step C' is chosen such that

$$C' = \min\left\{ c \in \mathbb{N} \;\middle|\; \sum_{h=0}^{c} e^{-\lambda t s_1} \frac{(\lambda t s_1)^h}{h!} \geq 1 - \frac{\varepsilon}{2} \right\}.$$

The cells that have to computed in this case are shown in Fig. 8.

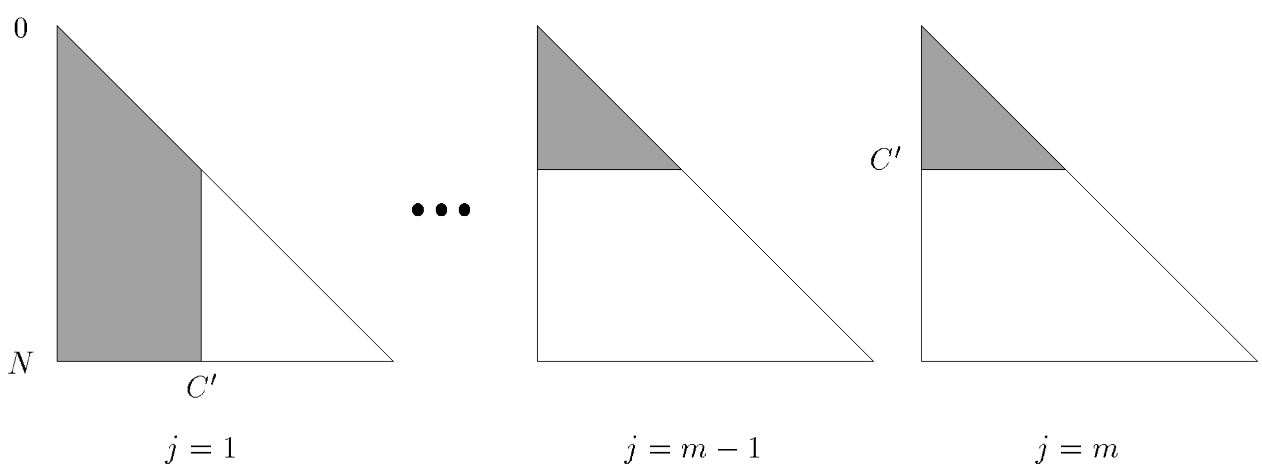

Figure 8: In gray, the computed area.

References

[1] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Trans. Computers*, C-29:720–731, August 1980.

[2] J. F. Meyer. Closed-form solutions for performability. *IEEE Trans. Computers*, C-31:648–657, July 1982.

[3] D. Furchtgott and J. F. Meyer. A performability solution method for degradable nonrepairable systems. *IEEE Trans. Computers*, C-33:550–554, July 1984.

[4] A. Goyal and A. N. Tantawi. Evaluation of performability for degradable computer systems. *IEEE Trans. Computers*, C-36:738–744, June 1987.

[5] L. Donatiello and B. R. Iyer. Analysis of a composite performance measure for for fault-tolerant systems. *J. ACM*, 34:179–189, January 1987.

[6] B. Ciciani and V. Grassi. Performability evaluation of fault-tolerant satellite systems. *IEEE Trans. Communications*, COM-35:403–409, April 1987.

[7] V. Grassi, L. Donatiello, and G. Iazeolla. Performability evaluation of multi-component fault-tolerant systems. *IEEE Trans. Reliability*, 37(2):216–222, June 1988.

[8] M. D. Beaudry. Performance-related reliability measures for computing systems. *IEEE Trans. Computers*, C-27:540–547, June 1978.

[9] G. Ciardo, R. Marie, B. Sericola, and K. S. Trivedi. Performability analysis using semi-Markov reward processes. *IEEE Trans. Computers*, C-39:1251–1264, October 1990.

[10] E. de Souza e Silva and H. R. Gail. Calculating cumulative operational time distributions of repairable computer systems. *IEEE Trans. Computers*, C-35:322–332, April 1986.

[11] G. Rubino and B. Sericola. Interval availability distibution computation. *Proceedings IEEE 23-th Fault-Tolerant Computing Symposium, (Toulouse, France)*, 49–55, June 1993.

[12] B. R. Iyer, L. Donatiello, and P. Heidelberger. Analysis of performability for stochastic models of fault-tolerant systems. *IEEE Trans. Computers*, C-35:902–907, October 1986.

[13] R. M. Smith, K. S. Trivedi, and A. V. Ramesh. Performability analysis: measures, an algorithm, and a case study. *IEEE Trans. Computers*, C-37:406–417, April 1988.

[14] E. de Souza e Silva and H. R. Gail. Calculating availability and performability measures of repairable computer systems using randomization. *J. ACM*, 36:171–193, January 1989.

[15] L. Donatiello and V. Grassi. On evaluating the cumulative performance distribution of fault-tolerant computer systems. *IEEE Trans. Computers*, 40:1301–1307, November 1991.

[16] E. de Souza e Silva and H. R. Gail. *Calculating transient distributions of cumulative reward.* Technical Report CDS-930033, UCLA, University of California, Los Angles, USA, September 1993.

[17] K. R. Pattipati, Y. Li, and H. A. P. Blom. A unified framework for the preformability evaluation of fault-tolerant computer systems. *IEEE Trans. Computers*, 42:312–326, March 1993.

[18] S. M. Ross. *Stochastic Processes.* John Wiley and Sons, 1983.

[19] M. Banâtre, A .Gefflaut, P. Joubert, P. Lee, and C. Morin. *An Architecture for Tolerating Processor Failures in Shared - Memory Multiprocessors.* Technical Report 1965, INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France, March 1993.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399