# Actes du 10ème Atelier en Évaluation de Performances

Sara Alouf, Alain Jean-Marie

# Actes du 10ème Atelier en Évaluation de Performances

Sophia Antipolis, 11–13 juin 2014

Sara Alouf et Alain Jean-Marie, éditeurs


ANR MARMOTE




**10ème Atelier en Evaluation de Performances**
**Sophia Antipolis, du 11 au 13 juin 2014**

# Objectif

L'Atelier en Évaluation de Performances est une réunion destinée à faire s'exprimer et se rencontrer les jeunes chercheurs (doctorants et nouveaux docteurs) dans le domaine de la Modélisation et de l'Évaluation de Performances, une discipline consacrée à l'étude et l'optimisation de systèmes dynamiques stochastiques et/ou temporisés apparaissant en Informatique, Télécommunications, Productique et Robotique entre autres.

La présentation informelle de travaux, même en cours, y est encouragée afin de renforcer les interactions entre jeunes chercheurs et préparer des soumissions de nouveaux projets scientifiques. Des exposés de synthèse sur des domaines de recherche d'actualité, donnés par des chercheurs confirmés du domaine renforcent la partie formation de l'atelier.

# Historique

Démarré sous l'impulsion de l'équipe de Raymond Marie en 1991, l'atelier en évaluation de performance a eu lieu à Rennes (1991-93), Grenoble (95), Versailles (96), Paris ENS (01), Reims (03) et Aussois (08). L'objectif est de fédérer une communauté allant des probabilités aux expérimentations en informatique. Elle recouvre donc d'une part les thématiques portant sur la modélisation des systèmes complexes avec les méthodes théoriques récentes, les environnements logiciels d'évaluation de performances jusqu'aux retours d'expérimentations en vraie grandeur.

# Comités

## Comité Scientifique

Le comité scientifique est composé de : Jean-Michel Fourneau, Bruno Gaujal, Marc Lelarge, Jean Mairesse, Laurent Truffet, et Bruno Tuffin

## Comité d'Organisation

Le comité d'organisation est composé de : Sara Alouf, Alain Jean-Marie, Jithin Sreedharan. Il a pu compter sur l'aide précieuse de Valeria Neglia et Laurie Vermeersch.

# Soutiens

L'Atelier a reçu le soutien :
— du Labex UCN@Sophia
— du GDR ASR
— du GDR RO
— du GDR IM
— de l'ANR MARMOTE (projet 12-MONU-0019)
— et d'Inria

# Programme

## Mercredi 11 juin 2014

12h00-12h30 : Arrivée des participants à l'Inria, espace muséal du bâtiment Euler

12h30-14h00 : Repas buffet froid, hall du bâtiment Euler

14h00-15h00 : **Exposé de synthèse 1**

Pierre L'Ecuyer
*Challenges in the Stochastic Modeling of Service Systems : Illustrations with Call Centers* (p. 5)

15h00-16h00 : **Session 1 : Échantillonnage Parfait/Perfect Sampling**

Stéphane Durand
*A perfect sampling algorithm of random walks with forbidden arcs* (p. 7)

Anne Bouillard, Ana Bušíc, Christelle Rovetta
*Simulation parfaite dans un réseau fermé de files d'attente* (p. 9)

16h00-16h30 : Pause café

16h30-18h00 : **Session 2 : Simulation**

Maïder Estécahandy, L. Bordes, S. Collas, C. Paroissin
*Méthodes d'accélération de Monte-Carlo adaptées à des modèles simples en Réseaux de Petri* (p. 11)

Abderaouf Benghalia, Jaouad Boukachour, Dalila Boudebous
*Contribution à la modélisation et à l'évaluation de la performance du transfert de conteneurs maritimes* (p. 13)

Marion Dalle, Jean-Marc Vincent, Florence Perronnin
*Catch me if you can !* (p. 15)

## Jeudi 12 juin 2014

09h00-10h00 : **Session 3 : Analyse de Données/Data Analysis**

Damien Dosimont, Guillaume Huard, Jean-Marc Vincent
*Agrégation temporelle pour l'analyse de traces volumineuses* (p. 17)

Hadrien Hours, Ernst Biersack, Patrick Loiseau
*A causal study of an emulated network* (p. 19)

10h00-10h30 : Pause café/viennoiseries

10h30-11h30 : **Exposé de synthèse 2**

Thomas Bonald
*Réseaux de Whittle et applications* (p. 6)

11h30-12h30 : **Session 4 : Systèmes de Caches/Cache Systems**

Felipe Olmos
*The performance of a LRU cache under dynamic catalog traffic* (p. 21)

Nicaise Choungmo Fofack
*Approximate models for cache analysis with correlated requests* (p. 23)

12h30-13h00 : Trajet en Pullman jusqu'au restaurant

13h00-15h00 : Déjeuner à Bijou Plage (Juan-Les-Pins)

15h00-18h00 : Promenade/découverte de Juan-Les-Pins/tour du Cap d'Antibes

18h00- : Programme libre

## Vendredi 13 juin 2014

09h00-10h00 : **Session 5 : Contrôle des Réseaux/Network Control**

Tatiana Seregina
*Reward-based Incentive Mechanisms for Delay Tolerant Networks* (p. 25)

Maialen Larrañaga, U. Ayesta, I.M. Verloop
*Whittle's index in a multi-class queue with abandonments* (p. 27)

10h00-10h30 : Pause café/viennoiseries

10h30-11h30 : **Exposé de synthèse 3**

Anne Bouillard
*Algorithms for and from network calculus* (p. 6)

11h30-12h00 : **Session 6 : Méthodologie/Methodology**

Luka Stanisic, Arnaud Legrand
*Good Practices for Reproducible Research* (p. 29)

12h00-13h30 : Repas plateau à la cantine de l'Inria

13h30-15h00 : **Session 7 : Modèles Stochastiques/Stochastic Models**

Farah Ait Salaht, Jean-Michel Fourneau, Hind Castel, Nihal Pekergin
*Approche par bornes stochastiques et histogrammes pour l'analyse de performance des réseaux* (p. 31)

Konstantin Avrachenkov, Natalia M. Markovich, Jithin Sreedharan
*On Distribution and dependence of extremes in PageRank-type processes* (p. 33)

Mahmoud El Chamie, Giovanni Neglia, Konstantin Avrachenkov
*Averaging on Dynamic Networks* (p. 35)

15h00-15h30 : **Table ronde : Logiciels dédiés à la modélisation avec les chaînes de Markov**

15h30-16h00 : Conclusions et fin de l'atelier

# Exposés de synthèse

Pierre L'Ecuyer
Université de Montréal, chaire internationale Inria

**Challenges in the Stochastic Modeling of Service Systems : Illustrations with Call Centers**

**Résumé** : Large systems that involve humans (such as health-care systems, call centers, emergency systems, transportation networks, supply chains, communication systems, etc.) are difficult to manage because they are complex and involve significant uncertainty which is itself hard to model in a realistic way. For example, call arrivals in call centers follow stochastic processes whose rates are themselves random and depends significantly on the time of the day, type of day (day of the week, holiday), period of the year, weather, other external events, etc. The arrival processes of different call types may also be dependent. Call durations (service times) have distributions that depend on the call type and on the particular agent who handles the call, and are often time-dependent because the effectiveness of agents depends on their experience, base qualities, motivation, fatigue, etc. Similar complications occur in other systems mentioned above. As a result, valid and reliable stochastic models for these systems are hard to build and maintain. They require continuous learning and adaptation based on incoming data that reflects system evolution.
The main motivation for modeling and simulating these systems is to construct good decision-making policies for their management. In a typical call center with multiple call types and multiple agent types (who can handle subsets of call types), one must decide how many agents to hire and train, for what call types, construct work schedules for these agents that respect union agreements, specify dynamic routing rules for arriving calls and for agents that become available, while meeting certain (stochastic) constraints on the quality of service of the systems (e.g., on the distributions of waiting times and call abandonments), and dothis at the least possible cost. Solving such stochastic optimization problems via simulation, both for long- and medium-term planning (days or months in advance) and for short-term decision making and recourse to face unexpected situations, are challenging tasks.
We will illustrate these types of modeling and optimization problems with concrete examples and data, and will review some recent models and ideas.

Thomas Bonald
Telecom ParisTech et LINCS

**Réseaux de Whittle et applications**

**Résumé** : Les réseaux de Whittle sont un outil puissant et pourtant peu connu de la théorie des files d'attente, généralisant les réseaux de Jackson et les réseaux de Kelly. Ils permettent de représenter certaines dépendances entre les taux de service des différentes files d'attente du réseau tout en gardant une forme explicite de la distribution stationnaire. Cet exposé constituera une introduction à cette classe de réseaux de files d'attente. Nous en donnerons la définition et les principales propriétés, puis montrerons en quoi ils permettent d'obtenir très simplement certains résultats classiques de la théorie des files d'attente, comme l'insensibilité de la discipline de service "processeur partagé". Nous terminerons l'exposé par les applications des réseaux de Whittle aux problèmes d'ingénierie des réseaux de communication et des centres de données.

Anne Bouillard
École normale supérieure

**Algorithms for and from network calculus**

**Résumé** : Network calculus (NC) is a theory based on the (min,plus) algebra and whose aim is to compute worst-case performance bounds in communication networks. This theory abstracts flows circulating in a network and the service offered by the networks elements by functions on which computations are performed. After a review of the basics of the theory, two problems will be addressed. First the problem of computing exact worst-case performance bounds in a feed-forward network, and second, the problem of supervision of a flow that uses some basic concept of NC but with a different aim.

# A perfect sampling algorithm of random walks with forbidden arcs

Stéphane Durand

ENS of Lyon, 15 parvis René Descartes, 69007 Lyon, France
Inria 655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France
stephane.durand@ens-lyon.fr

## 1. Introduction

I will present a result obtained with Bruno Gaujal and Florence Perronin on simulation of grid Markov chain. Random walk are frequently used to model behavior of systems or complex randomized algorithm. one often needs to be able to compute or sample the stationary distribution.
The problem we consider here is sampling the stationary distribution of a random walk on a multidimensional grid of great dimension, when its size prevents from directly computing it.
A known solution is Monte-Carlo method, but it only give a asymptotically perfect approximation. Instead, I will present a method giving a perfect sample in finite time.

## 2. Model

We work on a finite grid $\mathcal{S} := \{1, \dots, N\}^d$, where both the span $N$ and the dimension $d$ are large, to which we add forbidden moves, couple of (point,direction) retrieved from the possibles moves on the grid. We make the hypothesis that the grid stay strongly connected. Hence there exists an unique stationary distribution.

## 3. Perfect sampling

This section is devoted to the construction of a perfect sampling algorithm of a random walk $X(n)$ over $\mathcal{S}$ where certain arcs are forbidden. This is done in several steps

### 3.1. Coupling and Rejection

The random walk over a grid with forbidden arcs is an irreducible, finite, discrete time Markov chain over a finite state space $\mathcal{S}$ denoted $X(n)_{n\in\mathbb{N}}$, with transition matrix $P$. By definition, for any position $\mathbf{a}$ and any direction $\mathbf{m} = \pm\mathbf{e}_i$, $P_{\mathbf{a},\mathbf{a}+\mathbf{m}} = \frac{1}{q_\mathbf{a}}$ where $q_\mathbf{a}$ is the number of possible moves from $\mathbf{a}$.

From $(X(n))_{n\in\mathbb{N}}$, one can construct a continuous time Markov chain $Y(t)_{t\in\mathbb{R}}$ over the same state space. The generator $Q$ of $Y$ is obtained by multiplying each line $\mathbf{a}$ in $P$ by $q_\mathbf{a}$ and defining the diagonal element $Q_{\mathbf{a},\mathbf{a}}$ as $Q_{\mathbf{aa}} = -\sum_\mathbf{b} q_\mathbf{a} P_\mathbf{ab}$. Therefore, the rates from $\mathbf{a}$ to $\mathbf{a}+\mathbf{m}$ are all equal to one : $Q_{\mathbf{a},\mathbf{a}+\mathbf{m}} = 1$.
From $Y(t)$, it is possible to extract a new discrete time Markov chain, $Y(n)_{n\in\mathbb{N}}$ by uniformization. Its transition matrix is $Id + \Lambda^{-1}Q$, where $\Lambda$ (uniformization constant) is any positive real number larger than all $q_\mathbf{a}$'s. Since the total rate out of any state in $Y$ is bounded by $2d$, it can be uniformized by $\Lambda = 2d$.

While it can be difficult to construct a grand coupling for chain $X$, such a construction is easy and natural for the chain $Y$ since the rates are all equal. To couple the walks starting from all states, just pick one direction uniformly among the $2d$ possibilities and make every walk move in that direction. Those for which the move is not possible stay at the same position.
The stationary distribution of the first chain can be obtained from the new one through rejection. We just have to simulate one more step and reject if the new step is the same

**Theorem 1.** *If we can sample $Y$ under its stationary distribution, then using the rejection we obtain samples distributed according to the stationary distribution of $X(n)$.*

### 3.2. Perfect Sampling, coupling from the past

In order to obtain a perfect sample, we need to have a sufficiently long random chain such that the walks starting from every state using this chain end at the same point, which will be the answer of the algorithm.
For that, we need to provide an efficient criterion choosing the move function of the current state and a random word. This was difficult with the first chain, but immediate with the second.
If we go forward, adding moves at the end of the simulation until all trajectories converge, we add a bias caused by the dependence of coupling time on the step.
Instead we need to go backward, adding the new moves at the beginning of the simulation. For every move added, the set of new trajectories is a subset of the previous one, meaning that when the last set is a point, it will stay unchanged by any addition of moves.

**Lemma 1.** *The coupling from the past converges to a point sampled according to the stationary distribution*

### 3.3. intervals

The size of the grid forbid us to compute every trajectories, but we can compute any supersets of the possi-

ble positions. This can increase the coupling time, and could make it infinite in the case of a bad choice of superset (for example, the classical interval). In case of convergence, we know that the point obtained is the same as the one we would obtain with computing trajectories separately.
We use a double superset, made of one interval, behaving as if there were no forbidden move on the grid, and a set of point, created by the forbidden moves. We can show that the expectation of the coupling time is polynomial ($O(N^2 d \log d)$) and thus the number of point stays computable

## 4. Complexity

The goal of this section is to bound the expected time and space cost for random grid, ie a grid with a known number k of random forbidden moves. We suppose the dimension d to be smaller than the span N of the grid.

### 4.1. Rejection

The reject probability is the probability that the next move cannot be taken by the walker who is in a stationary state (of $Y$). In general this probability is bounded by $\frac{1}{2d}$ (for example if the walker has all moves forbidden but one). So that the expected number of rejections is always bounded by 2d. However this bound is very loose. In the case without any forbidden move, the expected number of tries needed is $\frac{N}{N-1}$. The numerical experiments on random graph with a number of forbidden moves smaller or comparable to N show a probability of this order.

### 4.2. Interval Coupling time

We have some results on random or without forbidden moves grid, as approximations.

**Lemma 2** (Coalescence in dimension 1)**.** *For any* $T > 0$, $\mathbb{P}(C > T) \leq \cos^T\left(\frac{\pi}{N+1}\right)\left(1 + O(\frac{1}{N^2})\right)$.

**Lemma 3** (Coalescence in dimension d)**.** *Let us consider a random walk in a grid with no forbidden arcs. Let* $C_d$ *be its coalescence time.*
*(i) The number of simulated step before obtaining the coupled point is at most four time the coalescence time.*
*(ii) The expected coalescence time satisfies* $\mathbb{E}[C_d] = O\left(N^2 d \log d\right)$.

**Lemma 4.** *If forbidden arcs are chosen randomly, uniformly among all arcs in the grid and if* k *is the expected number of forbidden arcs, then the maximal size* $|E|$ *of the set* E*, is bounded in expectation :* $\mathbb{E}[\,|E|\,] \leq \frac{kN}{\pi^2} + O(kd + \frac{kN}{d})$.

With this, we can conclude that most of the time is taken by the convergence of the interval, taking an average of $O(d \log dN^2)$ steps, and the average memory cost of the separated point is of the order of $\frac{kN}{\pi^2}$. A step can be computed in time O(k), so we obtain an algorithm for perfect sampling of grids whose time complexity is $O(kd \log dN^2)$.

# Simulation parfaite dans un réseau fermé de files d'attente

Anne Bouillard, Ana Bušić, Christelle Rovetta *

Inria - Département d'Informatique de l'ENS
23 avenue d'Italie, 75013 Paris, France
christelle.rovetta@inria.fr

## 1. Modèle

Nous présentons une méthode par chaîne bornante pour la simulation parfaite efficace de réseaux fermés de files d'attente. On considère un réseau fermé de $K$ files d'attente ./M/1/C à $M$ clients. Chaque file d'attente $k \in \{1, \dots, K\} := F$ a une capacité $C_k$ et un taux de service $\nu_k$. On note $p_{i,j}$ la probabilité pour un client venant d'être servi dans la file $i$ d'être dirigé dans la file $j$. Une transition peut s'effectuer si la file $i$ n'est pas vide et la file $j$ n'est pas pleine. Le réseau est modélisé par un graphe orienté $G = (F, R)$ avec $R = \{(i,j) \mid p_{i,j} > 0\}$. On le supposera fortement connexe.
On note $\mathcal{S}$ l'espace des états du système. Ainsi $\mathcal{S}$ est l'ensemble des $x = (x_1, x_2, \dots, x_K) \in \mathbb{N}^K$ vérifiant $\sum_{k=1}^{K} x_k = M$ et $\forall k\ 0 \le x_k \le C_k$. Une borne supérieure pour $|\mathcal{S}|$ est donnée par $\binom{K+M-1}{K-1}$. Soit $(i,j) \in R$, on définit $t_{i,j} : \mathcal{S} \to \mathcal{S}$ la fonction qui décrit le service d'un client de la file $i$ vers la file $j$ :

$$t_{i,j}(x) = \begin{cases} x - e_i + e_j & \text{si } x_i > 0 \text{ et } x_j < C_j, \\ x & \text{sinon.} \end{cases}$$

L'évolution de ce système peut être décrite par une chaîne de Markov et être échantillonnée grâce à l'algorithme 1 de simulation parfaite (voir [1]).
La taille exponentielle de l'espace des états et la non-monotonie de la chaîne rendent l'algorithme 1 inutilisable en pratique. On propose dans ce résumé une chaîne bornante sous forme de diagramme afin de pouvoir appliquer la simulation parfaite pour $K$ et $M$ grands. On exploite la contrainte forte du nombre de clients fixé à $M$, les états sont représentés dans le diagramme par des chemins partant de $(0,0)$ et allant à $(K, M)$.

## 2. Représentation par un diagramme

### 2.1. Diagramme

On appelle *diagramme complet* un graphe orienté $\mathcal{D} = (N, \mathcal{A})$. Les nœuds sont placés dans une grille à $(K+1)$ colonnes et $(M+1)$ lignes.

- $N = \{(c, \ell),\ | \ c \in \{1, \dots, K-1\},\ | \ \ell \in \{0, \dots, M\}\} \cup \{(0,0)\} \cup \{(K, M)\}$.
- $\mathcal{A} = \{((c-1, \ell), (c, \ell')) \mid 0 \le \ell' - \ell \le C_c\}$.

On note $\Pi(\mathcal{D})$ l'ensemble de tous les chemins de $(0,0)$ à $(K, M)$. Ce sont des chemins de longueur $K$. On définit la fonction $f : \mathcal{S} \to \Pi(\mathcal{D})$ qui à chaque élément $x = (x_1, \dots, x_K) \in \mathcal{S}$ associe un chemin $f(x) = ((0,0), (1, x_1), \dots, (c, \sum_{i=1}^{c} x_i), \dots, (K, M))$. La fonction $f$ est une bijection entre $\mathcal{S}$ et $\Pi(\mathcal{D})$.

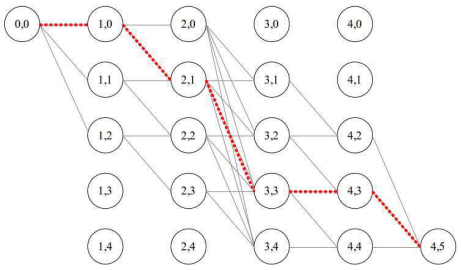


FIGURE 1 – Chemins dans un diagramme complet à 5 files et 4 clients avec $C_1 = C_5 = 2$, $C_2 = C_4 = 1$ et $C_3 = \infty$. Le chemin en pointillés représente l'état $x = (0, 1, 2, 0, 1)$.

On dit que $D = (N, A)$ ou $A \subseteq \mathcal{A}$ est un *diagramme* si $(0,0)$ est le seul nœud ayant un degré entrant égal à 0 et $(K, M)$ est le seul nœud ayant un degré sortant égal à 0. Un diagramme peut être décrit par l'ensemble $\Pi$ de ses chemins, on le note alors $D = (N, A_\Pi)$. À partir d'un diagramme on peut définir un ensemble d'états et inversement :

$$\phi(S) = (N, A_{f(S)}) \qquad \text{et} \qquad \psi(D) = f^{-1}(\Pi(D)).$$

Si $S = \psi(D)$, on dit que $D$ est un *représentant* de $S$. Si $S \subseteq \psi(D)$, on dit que $D$ est un *sur-représentant* de $S$. Le diagramme complet $\mathcal{D}$ est un répresentant de $\mathcal{S}$.

**Algorithm 1:** Simulation parfaite avec $\mathcal{S}$

**Data**: $(U_{-n} = (i_{-n}, j_{-n}))_{n \in \mathbb{N}}$ suite i.i.d.
**Result**: $s \in \mathcal{S}$ suivant la distribution stationnaire
**begin**
  $n \leftarrow 1$;
  $t \leftarrow t_{U_{-1}}$;
  **while** $|t(\mathcal{S})| \neq 1$ **do**
    $n \leftarrow 2n$;
    $t \leftarrow t_{U_{-n}} \circ \cdots \circ t_{U_{-1}}$;
  **end**
  **return** $s_0 = \{t(\mathcal{S})\}$
**end**

*. Auteure correspondante.

### 2.2. Transitions

Soit $(i,j) \in R$. On définit la fonction $T_{i,j}$ par :

$$T_{i,j}(D) = \phi \circ t_{i,j} \circ \psi(D) \text{ pour } D \subset \mathcal{D}.$$

**Proposition 1** *(i) Si* $D$ *est un sur-représentant de* $S$ *alors* $T_{i,j}(D)$ *est un sur-représentant de* $t_{i,j}(S)$.
*(ii) Si* $|\psi(D)| = 1$ *alors* $|\psi \circ T_{i,j}(D)| = 1$.

Pour effectuer une transition $T_{i,j}$ sur un diagramme, on sépare ses arêtes en trois sous-ensembles :

— $\mathcal{V}ide = \{a \in A \mid a \in w \in \Pi(D) \text{ et } f^{-1}(w)_i = 0\}$,
— $\mathcal{P}lein = \{a \in A \mid a \in w \in \Pi(D) \text{ et } f^{-1}(w)_j = C_j\}$,
— $\mathcal{A}ctif = \{a \in A \mid a \in w \in \Pi(D)\ , f^{-1}(w)_i > 0 \text{ et } f(w)_j < C_j\}$.

Les ensembles $\mathcal{V}ide$ et $\mathcal{P}lein$ contiennent les arêtes ne pouvant pas subir la transition $T_{i,j}$. L' ensemble $\mathcal{A}ctif$ a contrario contient les arêtes pouvant subir la transition $T_{i,j}$. On note $\mathcal{A}ctif'$ l'ensemble des arêtes du diagramme $T_{i,j}(D)$ ayant subi la transition $T_{i,j}$. Il est obtenu à partir de $\mathcal{A}ctif$ en abaissant ou en remontant les arêtes des colonnes comprises entre $i-1$ et $j-1$. On obtient ensuite :

$$T_{i,j}(D) = (N, A') \text{ avec } A' = \mathcal{V}ide \cup \mathcal{P}lein \cup \mathcal{A}ctif'.$$

Une transition $T_{i,j}(D)$ s'effectue avec une complexité en temps en $O(KM^2)$.

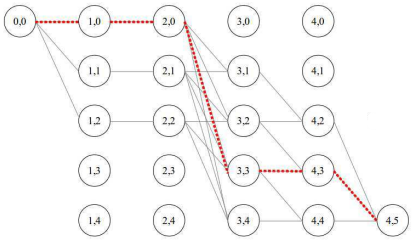


FIGURE 2 – $T_{2,3}(\mathcal{D})$

## 3. Simulation parfaite & diagramme

**Théorème 1** *Il existe une suite finie de transitions* $T = T_{i_\rho,j_\rho} \circ \cdots \circ T_{i_1,j_1}$ *telles que* $|\Pi(T(\mathcal{D}))| = 1$.

Le théorème 1 prouve que l'algorithme 2 se termine. La proposition 1 assure qu'avec une même suite $(U_{-n})$ si $|\Pi(T(\mathcal{D}))| = 1$ alors $w_0 = \{\Pi(T(\mathcal{D}))\} = \{t(S)\}$ ce qui implique que $w_0$ est distribué selon la loi stationnaire.
On appelle *temps de couplage* la valeur de $n$ permettant de terminer l'algorithme de simulation parfaite. La figure 3 donne le temps de couplage moyen pour 200 simulations des deux algorithmes pour un même réseau. L'expérimentation avec l'algorithme 1 ne peut pas en temps raisonnable (moins de douze heures) dépasser $M = 23$.

**Algorithm 2:** Simulation parfaite avec $\mathcal{D}$

**Data**: $(U_{-n} = (i_{-n}, j_{-n}))_{n\in\mathbb{N}}$ suite i.i.d.
**Result**: $w \in \Pi(\mathcal{D})$
**begin**
  $n \leftarrow 1$;
  $T \leftarrow T_{U_{-1}}$;
  **while** $|\Pi(T(\mathcal{D}))| \neq 1$ **do**
    $n \leftarrow 2n$;
    $T \leftarrow T_{U_{-n}} \circ \cdots \circ T_{U_{-1}}$;
  **end**
  **return** $w_0 = \{\Pi(T(\mathcal{D}))\}$
**end**

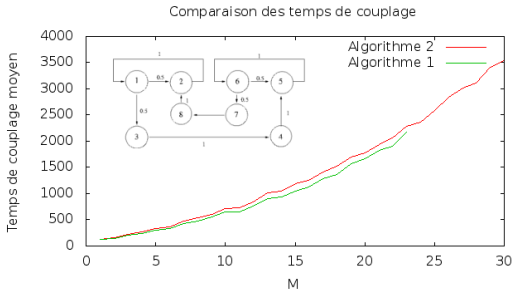


FIGURE 3 – $K = 8$ et $C = \frac{M}{2}$

## 4. Conclusion

La principale contribution est la description d'une nouvelle technique de simulation parfaite pour les réseaux fermés de files d'attente. Notre approche permet d'échantillonner la distribution stationnaire pour des réseaux ayant des files à capacité finie ou infinie, à serveurs uniques ou multiples. La démonstration de bonnes propriétés du diagramme impliquera que les transitions $T_{i,j}(D)$ peuvent s'effectuer en $O(KM)$. D'autre part, les simulations montrent que les temps de couplage sont assez proches. Une avancée majeure serait de trouver une borne théorique donnant le temps de couplage de l'algorithme 2 en fonction de celui de l'algorithme 1.

## Bibliographie

1. Propp, J.G. and Wilson, D.B. – Exact sampling with coupled Markov chains & applications to statistical mechanics. – Random Structures & Algorithms 9 (1996), no. 1-2, 223-252.

# Méthodes d'accélération de Monte-Carlo adaptées à des modèles simples en Réseaux de Petri

M. Estécahandy[†,‡,*], L. Bordes[†,*], S. Collas[‡,*], C. Paroissin[†,*]

[†]LMA, Université de Pau et des Pays de l'Adour, 64000, PAU, France.
[‡]TOTAL, 64000, PAU, France.
[*]maider.estecahandy@total.com, laurent.bordes@univ-pau.fr, stephane.collas@total.com, christian.paroissin@univ-pau.fr

## 1. Introduction

Dans le milieu pétrolier, obtenir des indicateurs de fiabilité précis sur des barrières instrumentées de sécurité (HIPS) est un enjeu important. Étant donné la constante évolution du contexte opératoire de ces équipements (installation sous-marine, changement climatique, politiques de maintenances sophistiquées, ...), l'analyse fiabiliste de ces barrières devient de plus en plus difficile à mettre en œuvre. De fait, les langages de modélisation usuels et les méthodes analytiques et numériques de calculs standards apparaissent de moins en moins adaptés pour prendre en compte la complexité des modèles mathématiques sous-jacents. Une solution alternative efficace est la simulation de Monte-Carlo (MC) combinée aux Réseaux de Petri (RdP) [1]. Néanmoins, obtenir des indicateurs de fiabilité précis sur ces équipements très fiables demeure un sérieux défi. En effet, la simulation d'événements rares nécessite un temps de simulation très long. Afin de répondre à cette problèmatique, des techniques d'accélération de Monte-Carlo ont été développées telles que l'*Importance Sampling* et le *Multilevel Splitting* [2] . Toutefois, ces approches nécessitent d'avoir une connaissance approfondie du modèle mathématique du système et sont fortement dépendantes du choix de paramètres qui sont difficiles à déterminer de manière automatique. Il s'ensuit que ces méthodes sont en pratique difficilement adaptables aux RdP. Une technique moins courante consiste à tronquer à droite les distributions des événements menant le plus directement à l'événement d'intérêt. On se réfère à la Méthode de Conditionnement Temporel [3] (MCT), également appelée *Failure Forcing* [4] dans le cadre markovien. Cette technique ne nécessite pas de connaître les lois intervenant dans le modèle, mais est uniquement définie lorsque l'événement d'intérêt est absorbant. Par conséquent, afin de remédier à ce problème, nous introduisons dans un premier temps une extension de MCT (EMCT) pour des modèles simples représentant des cycles de vie répétés où la défaillance qui mène à l'événement rare est soit directe, soit en conflit avec d'autres types de pannes. Ensuite, nous proposons une procédure informatique, valable uniquement si les composants sont indépendants, et nous la combinons enfin à la EMCT. Par suite, notre but est d'évaluer la *Probability of Failure on Demand* (PFD) de ces HIPS quelle que soit leur structure. Ainsi, nous présentons des résultats numériques qui illustrent le potentiel de ces méthodes.

## 2. Analyse fiabiliste

Dans le domaine de la sûreté de fonctionnement, l'une des principales quantités d'intérêt mesurant les performances d'un HIPS est la PFD. Nous proposons la définition suivante.

**Définition 2.1** *L'indicateur* $\mathrm{PFD}(t)$ *correspond à l'indisponibilité instantanée à l'instant* $t$ *d'un système relatif à la sécurité le rendant incapable d'accomplir correctement sa fonction de sécurité. Cette quantité est définie de la manière suivante pour tout* $t \in \mathbb{R}^+$ *:*

$$\mathrm{PFD}(t) = \mathbb{P}\left(\Phi(X(t)) = 0\right)$$

*où :*

- $X(t) = (X_1(t), X_2(t), \ldots, X_m(t)) \in \{0,1\}^m$ *est l'état des* $m$ *composants du système à l'instant* $t$ *;*
- $\Phi$ *est une fonction de* $\{0;1\}^m$ *dans* $\{0;1\}$*, type fonction de structure, où 0 représente l'ensemble des états indésirés du système et 1 les états complémentaires.*

Nous avons formalisé la PFD pour trois différentes structures de composants.

1. un composant qui tombe en panne selon une loi de probabilié quelconque et est réparé durant des campagnes d'inspections ;
2. un composant qui a $p$ pannes différentes en conflit. Ces pannes apparaissent et sont réparées selon des lois quelconques. La panne d'intérêt est la première panne ;
3. un composant qui possède une panne de type 1 en conflit avec une panne de type 2. La panne d'intérêt est la $1^{\text{ère}}$ panne.

## 3. Extension de MCT (EMCT)

Comme indiqué dans l'introduction, nous proposons une extension de MCT qui s'adapte à des cycles de vie répétés où l'événement d'intérêt est soit direct, soit en conflit avec $p$ autres événements.

**Principe 3.1** *Le principe d'EMCT, pour estimer* $\mathrm{PFD}(t)$*, est de conditionner chaque événement en conflit de durée* $\xi^{(1)}, \ldots, \xi^{(p)}$ *de sorte que la durée de l'événement la plus courte soit inférieure à un délai positif* $\Delta \geqslant t$ *. Nous notons* $\tilde{\xi}^{(k)}$*, pour* $k = 1, \ldots, p$*, les durée conditionnées correspondantes. Leur distribution est donnée, pour tout* $k \in \{1, \ldots, p\}$*, par :*

$$\mathcal{L}(\tilde{\xi}^{(k)}) = \mathcal{L}\left(\xi^{(k)} \mid \min\left(\xi^{(1)}, \ldots, \xi^{(p)}\right) \leq \Delta\right).$$

Par suite, en appliquant le Principe 3.1, nous avons calculé les PFD associées aux systèmes de la Section 2, ce qui nous a permis d'obtenir un facteur de déconditionnement pour chaque structure.

## 4. Méthode de Dissociation

Cette procédure qui peut être associée à la méthode de Monte-Carlo (MC) classique ou combinée à EMCT est décrite de la manière suivante.

**Principe 4.1** *Nous supposons les* $m$ *composants du système indépendants. Au lieu d'observer le comportement conjoint de ces* $m$ *composants au cours de* $n$ *simulations, cette méthode, dite de Dissociation, consiste à observer chacun des* $m$ *comportements individuellement. Ainsi, la première étape revient à simuler* $n$ *fois chaque processus univarié indépendamment. Dans un deuxième temps, nous combinons chacun des comportements individuels entre eux afin d'obtenir le comportement global du système. En d'autres termes, la seconde étape est une phase de post-traitement des* $n$ *simulations, qui permet de croiser tous les cas possibles afin d'obtenir au final* $n^m$ *observations.*

Cette méthode présente un intérêt si nous considérons, par exemple, un système de deux composants réparables avec deux réparateurs disponibles. Cependant, dans le cas où un seul réparateur est à disposition, elle ne pourra pas être appliquée.

## 5. Exemples numériques

Pour chacune des structures présentées dans la Section 2, nous avons estimé la PFD à l'instant $t = 8759$ de systèmes constitués de 1 à 4 composants identiques en parallèle. Afin d'illustrer les résultats obtenus, nous présentons dans la Table 1 ceux de la structure 2 pour 1 et 2 composants en parallèle [1]. Pour chaque composant, la loi de défaillance de la 1ère panne suit une loi $\mathrm{Exp}(5 \times 10^{-6})$ [2] et de la 2ème une loi $\mathrm{Exp}(1.74 \times 10^{-4})$. Les deux pannes sont réparées selon une loi $\mathrm{Exp}(1.742 \times 10^{-1})$. Pour chacune des méthodes, l'estimateur est obtenu après 1 000 000 de simulations. Lorsque l'événement n'est pas très rare ($\mathrm{PFD}(t) > 10^{-5}$), réaliser 1 000 000 de simulations semble suffisant pour observer que EMCT donne de meilleurs résultats que MC pour un temps de calcul équivalent. Par contre, lorsque la quantité d'intérêt devient très rare ($\mathrm{PFD}(t) \leq 10^{-9}$), ce nombre de simulations ne permet pas d'obtenir d'estimateurs avec MC et EMCT. Cependant, nous pouvons observer que les combinaisons avec la méthode de Dissociation permettent d'améliorer significativement les résultats obtenus quel que soit le degré de rareté de l'événement d'intérêt. Des algorithmes sont en cours de traitement afin de moyenner différentes estimations pour obtenir des estimateurs plus stables.

| **Nb de comp. en //** | 1 | | 2 | |
|---|---|---|---|---|
| | PFD(8759)=$2.9 \times 10^{-5}$ | | PFD(8759)=$8.2 \times 10^{-10}$ | |
| | $\widehat{\mathrm{PFD}}(8759)$<br>$\hat{\sigma}(8759)$ | Durée (sec) | $\widehat{\mathrm{PFD}}(8759)$<br>$\hat{\sigma}(8759)$ | Durée (sec) |
| **MC** | $2.2 \times 10^{-5}$<br>$4.7 \times 10^{-6}$ | 6294 | 0<br>0 | 8489 |
| **EMCT** | $2.5 \times 10^{-5}$<br>$3.7 \times 10^{-6}$ | 6397 | 0<br>0 | 22996 |
| **MC-DISS** | | | $8.1 \times 10^{-10}$<br>$2.2 \times 10^{-10}$ | 15472 |
| **EMCT-DISS** | | | $7.5 \times 10^{-10}$<br>$1.6 \times 10^{-10}$ | 25097 |

TABLE 1 – Analyse comparative de la structure 2

## Bibliographie

1. IEC 62551 ed1.0, "Analysis techniques for dependability - Petri net techniques", International Electrotechnical Commission, Geneva, 2012.
2. B. Tuffin, "Introduction to rare event simulation", Inria Rennes (France), AEP9, 2008.
3. R. Garnier, "Une méthode efficace d'accélération de la simulation des réseaux de Petri stochastiques ", Thesis, Bordeaux 1 university (France), 1998.
4. E.E. Lewis, F. Böhm, "Monte Carlo simulation of Markov unreliability models", Nuclear Engineering and Design, vol. 77, issue 1, pp. 49–62, 1984.

1. Les programmes ont été implémentés avec le logiciel $\mathcal{R}$.

2. Nous avons pris des lois exponentielles uniquement afin de faire des comparaisons avec des valeurs analytiques. $\mathrm{Exp}(\lambda)$ signifie loi exponentielle de taux de défaillance $\lambda$.

# Contribution à la modélisation et à l'évaluation de la performance du transfert de conteneurs maritimes

Abderaouf Benghalia, Jaouad Boukachour, Dalila Boudebous

Université du Havre
IUT du Havre, 5 rue Boris Vian, BP 4006,
76610 Le Havre, France
abderaoufb@yahoo.fr,
jaouad.boukachour@univ-lehavre.fr,
dalila.boudebous@univ-lehavre.fr

Face au flux croissant de conteneurs et aux contraintes de compétitivité de plus en plus rigoureuses, toute plate-forme portuaire doit assurer sa croissance et sa rentabilité, tout en sachant maîtriser ses impacts environnementaux [1]. A cet effet, le Grand Port Maritime du Havre (GPMH) a entrepris la construction d'un terminal multimodal (cf. Figure 1) ayant pour objectif le transfert massifié des conteneurs au sein du périmètre portuaire et également sur l'axe Seine. Le futur terminal multimodal est une plate-forme intermédiaire qui permet d'assurer le transport de conteneurs (collecte-livraison) en utilisant une nouvelle gestion des transferts entre les terminaux par : trains, barges fluviales et route. L'objectif de notre travail est l'étude de l'impact de cette plate-forme sur la performance du port du Havre.
L'évaluation de la performance repose essentiellement sur des méthodes de modélisation et des outils de mesure. Le choix des indicateurs de performance ne doit pas se faire à la légère, mais plutôt d'une manière pertinente suivant une démarche structurée. L'amélioration des indicateurs de performance d'un port est souvent un enjeu très important, notamment


FIGURE 1 – Plan du port du Havre

en raison des coûts associés et de l'impact sur les capacités de manutention de conteneurs [1]. Notre travail concerne la modélisation et la simulation du transfert des conteneurs par navettes ferroviaires entre le futur terminal multimodal et les terminaux maritimes du port du Havre. Il s'agit d'étudier la performance des transferts des conteneurs par rapport au coût, au délai et à la réduction de l'émission de $CO_2$.
Notre contribution fait état, d'une part, de l'application de la méthode ECOGRAI [2] pour l'identification des indicateurs de performance, et d'autre part, de la simulation à événements discrets pour calculer et mesurer ces indicateurs. A cet effet, nous proposons une démarche dénommée « ECOGRAISIM » [3] qui est basée sur la méthode ECOGRAI et la simulation. La méthode ECOGRAI [2] permet de concevoir et de développer les Systèmes d'Indicateurs de Performance (SIP) pour les entreprises industrielles ou de services. Elle permet de guider la conception et l'implantation d'un SIP. Notre démarche ECOGRAISIM consiste à appliquer les quatre premières étapes de la méthode ECOGRAI (cf. Figure 2). La première étape permet d'établir la grille GRAI (Graphe à Résultats et Activités Inter-reliés) [3] pour la modélisation de la structure de pilotage du système et l'identification des centres de décision. GRAI, méthode de modélisation d'entreprises a été crée dans les années 1980[2]. Cette méthode, peut être appliquée dans divers domaines. Elle permet [4] de détecter les points à améliorer et les points forts du système étudié. GRAI est basée sur une grille caractérisée par des lignes décrivant les niveaux décisionnels (stratégique, tactique et opérationnel), des colonnes décrivant les différentes fonctions du système de pilotage et des flèches représentant le type du flux. L'intersection entre les fonctions et les niveaux de décision nous a permis de déterminer les centres de décision. Dans notre cas d'étude, nous nous sommes focalisés sur un centre de décision appelé « optimiser les processus de manutention et de transfert ». Ce centre est en relation avec le centre de décision appelé « dimensionner et ordonnancer les missions en prenant en compte l'émergence des conteneurs à venir » car un dimensionnement et un ordonnancement optimal influencent directement l'optimisation globale. Notre centre de décision « optimiser les processus de manutention et de transfert » est en interaction avec les informations externes afin de prendre en compte les attentes des clients. Dans la deuxième étape, nous définissons les objectifs concernant les centres de décision et nous identifions les variables de décision dans la troisième étape. En effet, L'objectif de notre centre de décision est la minimisation des différentes pénalités concernant les retards, les coûts et

l'émission de $CO_2$. Il s'agit également de la minimisation de l'écart entre la date de livraison de conteneurs prévue et la date de livraison de conteneurs au plus tard. Enfin dans la quatrième étape, on obtient les indicateurs de performance. L'exigence d'établir un lien entre les éléments des triplets [2] « objectif – variable de décision – indicateur de performance » permet de ne retenir que les indicateurs pertinents dans le cadre de la stratégie globale du terminal multimodal [2]. Ils doivent être de type SMART [4] (S : Spécifique ; bien décrit et compréhensible, M : Mesurable, A : Atteignable, R : Raisonnable, T : Temporel ; lié avec une échelle de temps.) ou encore appelés indicateurs de performance intelligents. Une fois la liste des indicateurs est établie, la prochaine étape consiste à modéliser le système afin de simuler son comportement.

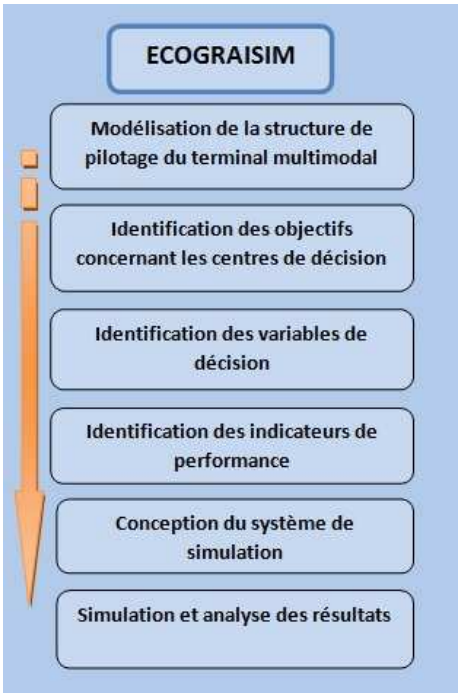


FIGURE 2 – Démarche ECOGRASIM

Au niveau stratégique, sont prises en compte les décisions sur l'équipement à utiliser et son agencement. Au niveau tactique, nous considérons la capacité des équipements et des ressources, l'affectation des voies et aussi les horaires des unités de transport. Enfin, le niveau opérationnel fait référence aux décisions nécessaires pour optimiser les processus de manutention et de transfert des conteneurs.
Nous avons appliqué la méthode ECOGRAI pour déterminer les indicateurs de performance de transfert massifié des conteneurs entre les terminaux maritimes et le futur terminal multimodal du port du Havre. Par ailleurs, un système de simulation à événements discrets a été développé pour étudier le transfert de conteneurs en prenant en compte différentes règles de gestion. Ce simulateur teste et compare différents scénarios de transfert des conteneurs. Il permet aussi de mesurer les indicateurs de performance déterminés par la démarche ECOGRAISIM que nous avons proposée.

**Bibliographie**

1. Benghalia, A., Boukachour, J., Boudebous, D. - Simulation of the passage of containers through le Havre seaport, The 14th International Conference on Harbor,Maritime and Multimodal Logistics Modelling and Simulation, Vienna, Austria, 2012.
2. Lauras, M. - Méthodes de diagnostic et d'évaluation de performance pour la gestion de chaines logistiques, Thèse de doctorat, L'institut National Polytechnique De Toulouse, 2004.
3. Benghalia, A., Boukachour, J., Boudebous, D. - Évaluation de la performance du trafic des conteneurs maritimes, 9th International Conference on Integrated Design and Production, Tlemcen, Algeria, 2013.
4. Ducq, Y., Doumeingts, G. - La méthodologie grai et les techniques internet : le projet gvc (grai virtual consulting), 3e Conférence Francophone de MOdélisation et SIMulation, Conception, Analyse et Gestion des Systèmes Industriels, MOSIM'01, – Troyes, 2001.
5. Beaudry, M. - Indicateurs de performance et tableau de bord. Courses, http ://video.coursgratuits.net/5/p-strategie-indicateurs-de-performance.php

# Catch me if you can !

Marion Dalle, Jean-Marc Vincent, Florence Perronnin *

Univ. Grenoble Alpes
F-38000 Grenoble, France
CEA, LETI, MINATEC Campus
F-38054 Grenoble, France
marion.dalle@cea.fr

## 1. Introduction

L'évaluation des performances de très grands réseaux et systèmes nécessite parfois une simulation très longue de modèles markoviens, que ce soit par exemple pour capturer des séquences d'événements rares, ou bien pour évaluer des taux de couverture, ou encore pour analyser des basculements de comportement dans des modèles "raides".

Pour réduire la durée de la simulation d'une seule trajectoire longue, l'idée est d'utiliser plus efficacement les ressources à disposition, en particulier les ressources de calcul (cœurs), en anticipant les calculs sur des tranches de temps se recouvrant au minimum. L'argument clé est, sous certaines conditions, l'efficacité du couplage de trajectoires qui permet de garantir l'obtention de la trajectoire complète.

L'objectif de ce travail est donc de proposer un algorithme parallèle de simulation de trajectoires longues basé sur une technique de "couplage horizontal" de trajectoires, d'analyser son coût et de donner les premiers résultats d'efficacité.

## 2. Simulation de modèles markoviens

On considère un modèle markovien à temps discret $\{X_t\}_{t\in\mathbb{N}}$ décrit par sa fonction de transition $\Phi : X_t = \Phi(X_{t-1}, e_t)$ où $e_t$ est l'événement se produisant à l'instant $t$ (généré aléatoirement). La simulation classique d'une trajectoire de longueur $T$ consiste, pour chaque instant, à générer aléatoirement l'événement selon les probabilités du modèle puis à mettre à jour l'état selon la fonction de transition (algorithme 1).

**Algorithme 1 :** Algorithme séquentiel

**Données** : $x_0$ (état initial), $T$
**pour** $t$ *de 1 à* $T$ **faire**
  $e_t \leftarrow genere\_evenement()$;
  $x_t \leftarrow \Phi(x_{t-1}, e_t)$;

Une technique de parallélisation consiste à calculer séparément des segments disjoints de la trajectoire, en anticipant l'état de départ dans chacun de ces segments. Cette technique initialement proposée par Nicol et al. [4] ou Fujimoto [3] est la suivante :

La période simulée est partitionnée en $P$ *tranches* de temps de durée fixe $\Delta = T/P$ démarrant à $t_i = i\Delta$ pour $0 \leqslant i \leqslant P-1$ affectées respectivement aux $P$ cœurs. Pour chaque cœur on génère également la séquence des événements et l'état initial à l'instant $t_i$. Plusieurs *phases* s'ensuivent durant lesquelles : chaque cœur $i$ simule la trajectoire du segment $[t_i, t_{i+1}]$ selon l'algorithme 1 en partant de l'état $x_{t_i}$ ; les états de début de tranche sont corrigés par la valeur obtenue en fin de la tranche précédente à la fin de la phase précédente. Les phases sont répétées jusqu'à ce que les états de début de tranche ne soient plus modifiés.

Il est clair que dans le pire des cas cet algorithme simulation parallèle nécessite autant d'étapes de calcul que l'algorithme séquentiel (au pire $P$ phases).

Le temps de calcul peut encore être réduit au prix de la précision en simulant un *ensemble d'états possibles* au lieu d'une trajectoire exacte [2]. L'idée est de remplacer le calcul de l'état par le calcul de bornes sur cet état : ainsi, les conditions aux limites des tranches de temps sont moins strictes (inclusion) ce qui réduit le nombre de phases de l'algorithme. Cela permet d'obtenir rapidement une estimation de la partie de l'espace d'état atteinte par le modèle.

## 3. L'algorithme *Catch me if you can*

L'algorithme que nous proposons vise à adapter la méthode développée par Nicol et al. en conditionnant l'arrêt du calcul de la trajectoire sur une tranche de temps par une condition de couplage.

### 3.1. Couplage de chaînes de Markov

Considérons une séquence d'événements $\{e_1, e_2, \cdots, e_T\}$ qui permet la génération de deux trajectoires $\{X_t^a\}$ et $\{X_t^b\}$ par applications successives de $\Phi$ à partir des états initiaux $x_0^a$ et $x_0^b$. Étant donné la nature du schéma itératif, si les deux trajectoires couplent à l'instant $t$ alors elles seront confondues après $t$. Le premier instant de rencontre des deux trajectoires est appelé instant de couplage, noté $\tau$.

### 3.2. Principe de parallélisation

On suppose que la séquence des événements est fixée et partagée en lecture par tous les cœurs. Elle peut être calculée au préalable ou à la volée en parallèle. À

*. Ce travail a été partiellement financé par l'ANR Marmote

chaque cœur $i$, $0 \leqslant i \leqslant P-1$, on attribue un intervalle de temps de taille variable pour calculer un fragment de trajectoire. C'est à dire une date de départ et un état de départ $x_{t_i}$ correspondant (fixé arbitrairement ou par une heuristique).
Chaque cœur $i$ calcule donc en parallèle la trajectoire issue de $x_{t_i}$ à partir de la date $t_i$. Le calcul est arrêté dès que la trajectoire couple avec une trajectoire déjà calculée par un autre cœur, ce qui permet d'éviter des calculs redondants. C'est pourquoi cet algorithme s'appelle *catch me if you can*. L'algorithme proposé est illustré en Figure 1.

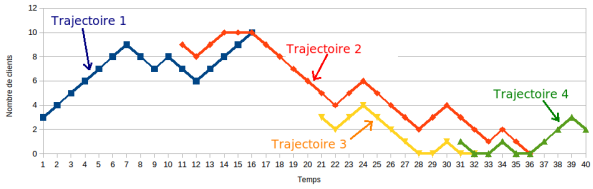


FIGURE 1 – Simulation d'une période de 40 itérations sur 4 cœurs. La durée totale de simulation est 25 (temps de couplage entre trajectoires 2 et 4) au lieu de 39 en séquentiel.

Si le principe de cet algorithme est simple, les difficultés résident dans sa mise en œuvre. En effet, même si l'on travaille en mémoire partagée, le test compare des données associées à des cœurs différents, ce qui correspond à des synchronisations en mémoire qui peuvent dégrader fortement les performances de l'algorithme.
Du point de vue théorique, si on note $N_i$ le nombre d'itérations effectuées par le cœur $i$ avant couplage, la durée totale de simulation (*makespan*) en nombre d'itérations est $N = \max_i N_i$. Pour des $t_{i+1} - t_i$ de l'ordre de $\frac{T}{P}$ on peut montrer que :

$$N \simeq \frac{T}{P} + \alpha,$$

où $\alpha$ correspond au surcoût lié au couplage de trajectoires et peut être majoré stochastiquement par le maximum de $P$ temps de couplage, ce qui croît très lentement en fonction de $P$ et laisse espérer au moins une efficacité asymptotique.

## 4. Premières analyses

L'un des critères de performance de l'algorithme est le nombre d'itérations (calculs de la fonction $\Phi$) nécessaires à l'obtention de la trajectoire complète.

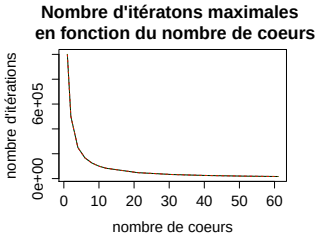


FIGURE 2 – Modèle de file M/M/1/50 charge $\rho = 80\%$, période de simulation $T = 100000$. 300 répétitions d'expériences par point. Intervalles de confiance inférieurs à 1%

L'exemple de la figure 2 illustre l'accélération obtenue en parallélisant le calcul de la trajectoire. Dans ce modèle le temps moyen de couplage est grossièrement borné par une fonction linéaire en la capacité de la file [1] (ici de l'ordre de 250 itérations), ce qui est négligeable devant $\frac{T}{P}$. Par contre si l'on augmente le nombre de cœurs le temps de couplage devient non-négligeable. Cet effet sera accentué par le coût lié aux contraintes de synchronisations entre les différents cœurs. Dans notre cas de simulation un nombre de cœurs de l'ordre de 10 donnerait un bon compromis entre accélération et utilisation de ressources.

## Bibliographie

1. Jantien Dopper, Bruno Gaujal, and Jean-Marc Vincent. Bounds for the Coupling Time in Queueing Networks Perfect Simulation. In *Numerical Solutions for Markov Chain (NSMC06)*, pages 117–136, Charleston, June 2006.
2. Jean-Michel Fourneau and Franck Quessette. Monotonicity and efficient computation of bounds with time parallel simulation. In *Computer Performance Engineering*, volume 6977 of *LNCS*, pages 57–71. Springer Berlin Heidelberg, 2011.
3. Richard M. Fujimoto. *Parallel and Distributed Simulation Systems*. Wiley-Interscience, 2000.
4. David Nicol, Albert Greenberg, and Boris Lubachevsky. Massively parallel algorithms for trace-driven cache simulations. *IEEE Trans. Parallel Distrib. Syst.*, 5(8) :849–859, 1994.

# Agrégation temporelle pour l'analyse de traces volumineuses

Damien Dosimont[3,1,2], Guillaume Huard[1,3,2], Jean-Marc Vincent[1,3,2]

[1] U. Grenoble Alpes, LIG, 38000 Grenoble, France
[2] CNRS, LIG, F-38000 Grenoble, France
[3] Inria
prénom.nom@imag.fr

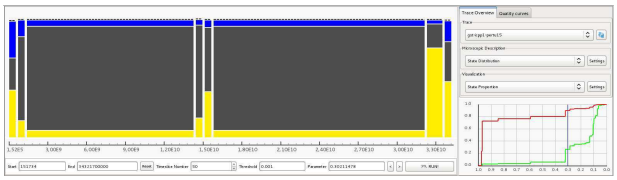

FIGURE 1 – L'outil Ocelotl montrant les différentes phases d'un décodage vidéo avec GStreamer. Une perturbation aux alentours de $1,5 \cdot 10^{10}$ ns est mise en évidence. Les courbes représentant la perte d'information et la réduction de complexité forment l'interface de choix du compromis d'agrégation.

## 1. Visualisation de traces volumineuses

L'analyse de la trace d'exécution d'une application est difficile quand la quantité d'événements qu'elle contient est importante. Les principales limites sont dues à la surface d'écran disponible, en particulier lors de l'utilisation de techniques représentant les ressources et le temps. Le diagramme de Gantt, employé par les analystes pour comprendre les relations de causalité et la structure de l'application, en est un exemple.

Différentes approches tentent de résoudre ces problèmes liés au passage à l'échelle de l'analyse. L'agrégation visuelle agrège les objets graphiques trop petits pour être affichés [1] ou adapte le rendu aux pixels disponibles [2]. Toutefois, ces approches ne représentent pas d'information utile à l'analyste dans le premier cas et sont instables en cas de redimensionnement dans le second.

L'agrégation d'information s'attaque au problème en amont, en diminuant la complexité des données à afficher. Viva [6] représente les ressources sous forme de treemap et propose une agrégation temporelle de leurs valeurs, mais le temps n'est pas représenté explicitement. Jumpshot [4] fournit, quant à lui, différent niveaux d'abstraction temporels de la trace mais ne permet pas d'agréger les ressources.

Dans le but de fournir une vue d'ensemble temporelle d'une trace, ce que ne fournissent pas les techniques actuelles, nous proposons une nouvelle technique, implémentée dans l'outil Ocelotl (Figure 1). Cette technique permet une analyse temporelle macroscopique qui n'est pas gênée par l'affichage d'un grand nombre de ressources. Elle représente le déroulement de l'application au cours du temps en agrégeant les parties de la trace où le comportement des ressources est homogène. Cette agrégation est modulée dynamiquement par l'utilisateur qui choisit un compromis entre la complexité et la perte d'information.

## 2. Représentation macroscopique de la trace

Notre contribution se décompose en deux points. Nous avons d'abord généralisé un algorithme d'agrégation temporelle [3], destiné à l'analyse macroscopique des grands systèmes. Un système est d'abord décrit sous la forme d'un modèle microscopique, associé à des métriques permettant d'interpréter son comportement. Le processus d'agrégation permet ensuite de représenter ce système sous la forme d'une partition du modèle microscopique. Cette partition est obtenue grâce à un compromis, dont l'utilisateur a le contrôle, entre la complexité (liée au nombre d'agrégats représentés) et la quantité d'information perdue en agrégeant. L'algorithme aura donc tendance à agréger les valeurs les plus proches en priorité afin de minimiser cette perte d'information pour une complexité donnée. Dans notre cas, nous n'autorisons l'agrégation qu'entre zones temporelles contigües. Notre première contribution a été de définir le modèle microscopique représentant le comportement de l'application, en découpant la trace en tranches de temps fixes pour lesquelles sont calculées la distribution des états des ressources. Nous avons aussi dû étendre l'algorithme initial à des quantités multidimensionnelles afin de pouvoir agréger le modèle microscopique que nous proposons. La représentation de la trace est faite avec une simple ligne qui montre le découpage en une succession temporelle d'agrégats. Chacun de ces agrégats représente un comportement particulier dans la trace. En outre, nous représentons pour chaque agrégat la proportion des différents états qu'il contient.

Notre seconde contribution est l'outil Ocelotl dans lequel la technique est implémentée. En tirant parti de l'infrastructure d'analyse de traces FrameSoC [5], dans laquelle il vient se greffer, Ocelotl peut afficher des traces de plusieurs Go et de dizaines de millions d'événements dans des délais raisonnables. A titre d'exemple, une trace de 2.3 Go et 15 millions

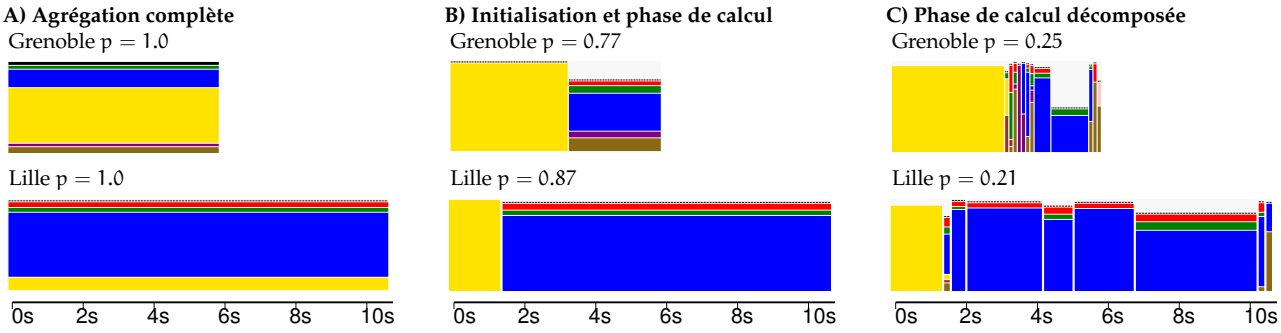


FIGURE 2 – Analyse comparative du benchmark NAS LU.A sur les sites de Grenoble et de Lille. Trois compromis distincts sont représentés. Les couleurs représentent la proportion des différents états MPI pour chaque agrégat ((`MPI_Recv` en bleu, `MPI_Wait` en rouge et `MPI_Send` en vert).

d'événements demande 5 mn de prétraitement sur un PC à 4 cœurs Intel Core i7-2760QM 2.4 GHz et 8 Go de DDRAM. Cela est possible grâce à l'utilisation de bases de données relationnelles pour stocker les traces, ce qui permet des requêtes optimisées afin d'éviter de saturer la mémoire. Ocelotl fournit des interactions pour zoomer, dézoomer et changer dynamiquement le niveau de détail de l'agrégation avec des temps de réaction immédiats. Il permet aussi de commuter vers un diagramme de Gantt sur une zone temporelle particulière, afin d'obtenir plus de détails.

## 3. Analyse de différents types de systèmes

Notre technique a été validée par l'analyse de plusieurs cas d'études. Notre premier cas est la lecture d'une vidéo avec GStreamer, perturbée par un stress CPU artificiel. La figure 1 montre la détection de cette perturbation dans une vidéo de 34 s.
La figure 2 représente l'exécution d'une application MPI (NAS Benchmark) sur deux plate-formes de calcul équivalentes en terme de nombre de ressources (Grid'5000, sites de Grenoble, 160 cœurs et de Lille, 152 cœurs). L'intérêt est de faire apparaître l'influence de la plate-forme sur le déroulement de l'application (phases de calcul de tailles différentes, proportions et séquences des états MPI qui varient, en particulier lors des communications).

## 4. Perspectives

La technique d'agrégation temporelle que nous avons implémentée dans Ocelotl permet de représenter une grande quantité d'information sous la forme d'une description macroscopique simple dans des délais adaptés à une analyse interactive.
Nous aimerions, cependant, pouvoir représenter conjointement les ressources, en particulier dans les cas hétérogènes, afin de déterminer leur influence sur les performances de l'application. Nous sommes ainsi focalisés actuellement sur une extension de ces travaux à l'agrégation simultanée de l'espace des ressources et du temps.

## Bibliographie

1. Jacques Chassin de Kergommeaux. Pajé, an Interactive Visualization Tool for Tuning Multi-Threaded Parallel Applications. *Parallel Computing*, 26(10) :1253–1274, August 2000.
2. Andreas Knüpfer, Holger Brunst, Jens Doleschal, Matthias Jurenz, Matthias Lieber, Holger Mickler, Matthias S. Müller, and Wolfgang E. Nagel. The Vampir Performance Analysis Tool-Set. In *Tools for High Performance Computing*, pages 139–155. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
3. Robin Lamarche-Perrin, Yves Demazeau, and Jean-Marc Vincent. The Best-partitions Problem : How to Build Meaningful Aggregations. In *Proceedings of the 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'13)*, Atlanta, USA, 2013.
4. Ewing Lusk and Anthony Chan. Early experiments with the OpenMP/MPI hybrid programming model. In Rudolf Eigenmann and Bronis R. de Supinski, editors, *OpenMP in a New Era of Parallelism*, volume 5004 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2008. IWOMP, 2008.
5. Generoso Pagano, Damien Dosimont, Guillaume Huard, Vania Marangozova-Martin, and Jean-Marc Vincent. Trace Management and Analysis for Embedded Systems. In *Proceedings of the IEEE7th International Symposium on Embedded Multicore SoCs (MCSoC-13)*, Tokyo, Japan, sep 2013.
6. Lucas Mello Schnorr, Arnaud Legrand, and Jean-Marc Vincent. Detection and analysis of resource usage anomalies in large distributed systems through multi-scale visualization. *Concurrency and Computation : Practice and Experience*, 24(15) :1792–1816, 2012.

# A causal study of an emulated network

Hadrien Hours, Ernst Biersack, Patrick Loiseau

EURECOM, Campus SophiaTech
Route des Chappes, 06410 Biot, France
firstname.lastname@eurecom.fr

## 1. Introduction

With the diversity of applications and technologies present in communication networks, the study of their performance has become a complex issue. The large number of parameters to take into account to model communication network performance increases the risk of spurious associations between the explanatory variables. Therefore studies based on correlation become a challenging approach. On the other hand, causal models, and their representation as Directed Acyclic Graphs (DAGs), offer simple and attractive models. Using simple graphical criteria, one can predict interventions on the system by using data collected through passive measurements. In this paper, we present an example of a causal study of communication network performance. We place ourselves in an emulated environment, where our predictions can be verified, and we show the benefits of the approach.

## 2. Causal study of an emulated network

For this paper we study the performance of TCP observing FTP traffic. We focus on the network performance, represented by the TCP throughput. To be able to verify our predictions we emulate a network using the Mininet software [1]. The experiment consists in one client that downloads files from a FTP server, both machines are connected to different routers (**R1** and **R2** respectively). In order to create a more realistic scenario we add two other machines, one connected to **R1** and the other to **R2**, creating cross traffic. The FTP traffic is recorded at the server side using *Tcpdump* tool and the dataset that we obtain is presented in Table 1.

Using the PC algorithm [4] with the Hilbert Schmidt Independence Criterion (HSIC) [5], we obtain, from the data summarized in Table 1, the causal model represented Figure 1. While TCP is a feedback based protocol, it is important to notice that we are studying the causal relationships between the parameters defining our system. In this case we obtain a causal graph, represented as a DAG, where the causes of the throughput are disclosed and its direct causes are represented as its parents. In this model we observe the receiver window (*rwin*), delay (*rtt*), and loss (*retrscore*), the empirical causes of the throughput, that are, indeed, found as its direct parents. Due to space constraints we cannot describe all the properties of this model and focus on prediction.

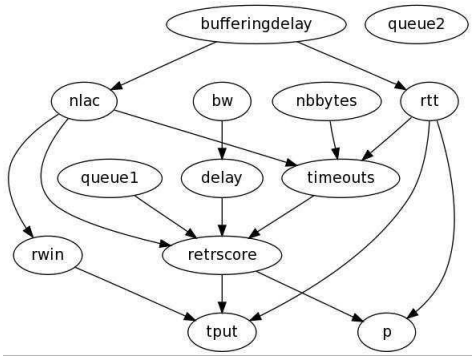


FIGURE 1 – Causal model of emulated FTP traffic

We want to estimate the distribution of the *throughput* after an intervention on the *retrscore*, where its value is set to 1%.

To estimate the total effect, on a parameter $Y$, of an intervention on a parameter $X$, we use the *Back-door Criterion* [3] defined as follows.

**Definition 1** ***(Back-door criterion)*** *A (set of) variable(s) $Z$ is said to satisfy the back-door criterion relative to an ordered pair of variables ($X_i$,$X_j$) in a DAG $G$ if : (i) no node in $Z$ is a descendant of $X_i$ ; and (ii) $Z$ blocks every path between $X_i$ and $X_j$ that contains an arrow into $X_i$.*

Using this definition, we can predict the effect of an intervention as follows.

**Theorem 1** ***(Back-door adjustment)*** *If a set of variables $Z$ satisfies the back-door criterion relative to $(X, Y)$, then the causal effect of $X$ on $Y$ is identifiable and given by the formula :*

$$P(y \mid do(X = x)) = \sum_z P(Y = y \mid X = x, Z = z)P(Z = z).$$

where the $do()$ operator represents the intervention of setting the parameter $X$ to the value $x$

We estimate the probability density functions (pdfs) by first estimating the marginals using normal kernels and then modeling the multivariate and conditional pdfs using T-Copulae [2].

To predict the effect of an intervention on the *retrscore* parameter, using the causal model presented in Figure 1, we apply the Back-door adjustment formula (Theorem 1) with the blocking set $Z = \{nlac, rtt\}$.

| Parameter | Definition | Min | Max | Avg | CoV |
|---|---|---|---|---|---|
| *bw* | minimum bandwidth (Mbps) | 1 | 25 | 7.1 | 0.91 |
| *delay* | propagation delay (ms) | 30 | 180 | 86 | 0.48 |
| *queue1* | size of **R1** buffer (pkts) | 10 | 400 | 98 | 1.1 |
| *queue2* | size of **R2** buffer (pkts) | 10 | 400 | 100 | 0.99 |
| *nlac* | Narrow Link Available Capacity (Kbps) | 12 | 3.07e3 | 630 | 5 |
| *rwin* | **C1** advertised receiver window (KB) | 74 | 2.23 | 290 | 0.65 |
| *bufferingdelay* | part of the RTT due to queuing delay (ms) | 1 | 6.76e3 | 120 | 2.4 |
| *rtt* | Round Trip Time (ms) | 84 | 6.91e3 | 3.1e2 | 0.99 |
| *timeouts* | number of timeouts (units) | 0 | 682 | 79 | 1.5 |
| *retrscore* | fraction of retransmitted packets (no unit) | 0 | 0.61 | 3.7e-3 | 5.1 |
| *p* | fraction of loss events (no unit) | 0 | 0.64 | 3.8e-3 | 8.4 |
| *nbbytes* | number of bytes sent by the server (MB) | 6 | 150 | 110 | 0.21 |
| *tput* | throughput (Kbps) | 6 | 1.10e3 | 280 | 0.81 |

TABLE 1 – Summary of Mininet network emulation experiments dataset

To verify our prediction we set up a new experiment where we drop 1% of the packets at the router **R1** and compare the throughput we obtain with the one that was predicted using the equation from Theorem 1. Figure 2 presents the probability density functions corresponding to the throughput prior to intervention (in dash-dot line), the post-intervention throughput estimated with the Back-door criterion (dashed line) and the throughput observed when we manually modify the loss (solid line). While not perfectly matching, the graph-based prediction (dashed line) and the real density after manual intervention (solid line) show similar shapes. In particular, both densities have a single mode around the same value (~ 100 kbps). The fact that the experimental throughput values after intervention (solid line) show less variance can be explained by the small number of samples (20) used to make the estimation. Also our method to estimate the post-intervention throughput (dashed line) uses normal kernels and a T-copula which tend to widen and smooth the estimated post-intervention density.

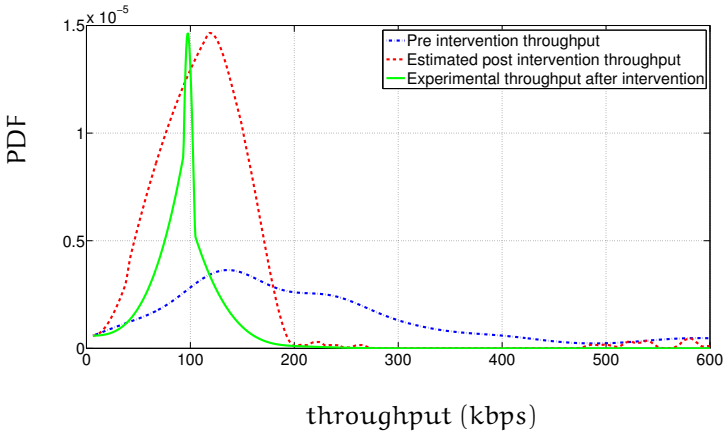


FIGURE 2 – Throughput distribution after an intervention on the retransmission score

## 3. Concluding remarks

Due to space constraints, many aspects of this study could not be presented. The study of the causal model we obtain discloses many properties of the system that would need further explanations and would lead to future studies. However it should be noticed, from the Back-door adjustment equation (Theorem 1) that the values present in our original dataset dictate the range of predictions that can be made. We are now working on adopting parametric modeling of the probabilities of the parameters that would allow to overcome this limitation.

## References

1. N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *CoNEXT '12*, pages 253–264.
2. P. Jaworski, F. Durante, W.K. Härdle, and T. Rychlik. *Copula Theory and Its Applications*. Lecture Notes in Statistics. Springer, 2010.
3. J. Pearl. *Causality : Models, Reasoning and Inference*. Cambridge University Press, 2009.
4. P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, Vol. 9 :62–72, 1991.
5. K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. *CoRR*, abs/1202.3775, 2012.

# The performance of a LRU cache under dynamic catalog traffic

Felipe Olmos[1], Bruno Kauffmann[2], Alain Simonian[2], Yannick Carlinet[2]

Orange Labs and CMAP École Polytechnique
[1]luisfelipe.olmosmarchant@orange.com
[2]firstname.lastname@orange.com

**Abstract**

We propose a simple traffic model featuring a dynamic catalog to construct a theoretical estimation of the hit ratio for a LRU cache offered such a traffic regime. We validate the accuracy of our theoretical estimates by computing the empirical hit ratio for real request sequences coming from traces of the Orange network.

## 1. Introduction

Caching performance evaluation is a relevant topic today in the context of Content Delivery Networks.

Most traffic models used to estimate the performance of a cache server are based on a fixed document catalog. As an example, the Independent Reference Model (IRM) further assumes that all requests are i.i.d., which allows to calculate theoretical estimates of the hit ratio, defined as the number of cache hits over number of requests.

However, in many applications, content is dynamic: It appears at a certain instant, its popularity varies over time and can eventually disappear. In this paper, we aim at building a tractable model that captures this feature (see [3] for details).

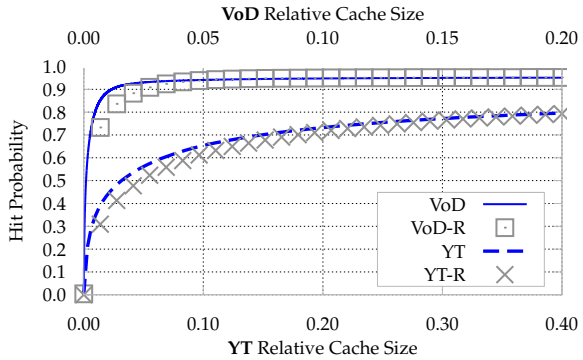


Figure 1: *Hit ratio of the original request sequence versus the result of semi-experiment (3).*

## 2. Semi-experimental driven modeling

We base our analysis on two datasets. The first dataset, called **YT**, comes from three months of YouTube traffic of the Orange Network in Tunisia. The second, called **VoD**, collects over three years requests from the Orange VoD service in France.

To discover the relevant features of our request sequences, we apply the semi-experimental methodology [2]. These semi-experiments aim to break specific structures of the request process by means of random shuffling; we then compare the resulting hit ratio, computed via simulation, with that of the original sequence and determine the importance of the broken structure. They consist in: 1) shuffling all requests, thus breaking all correlations; 2) shifting each document request sequence randomly, thus breaking the correlation between first requests; 3) fixing the first and last request of each document and shuffle all requests in between, thus breaking correlations within this sequence.

The results of semi-experiment (1) showed that the hit ratio curve differs considerably from that of the original sequence, thus confirming that IRM does not capture all the features of our datasets. In the case of (2) and (3), the curves do not differ considerably. This means that catalog arrivals and individual document request sequences can be modeled as homogeneous Poisson processes. For conciseness, we show in Figure 1 only the results of semi-experiment (3).

Taking into account the previous insights, we propose a two layered model. In the first layer, we consider a Poisson process $\Gamma$ of rate $\gamma$, hereafter called the *catalog arrival process*. This process models the publications of documents in the catalog.

The second layer consists in the *document request processes*. Specifically, when a document d arrives to the catalog at time $a_d$, we model its request process as an homogeneous Poisson process of rate $\lambda_d$ on the inter-

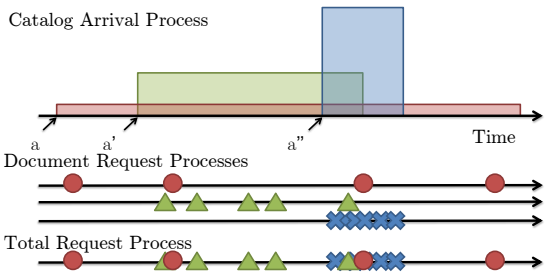


Figure 2: ***Top:*** *The boxes represent the lifespan and popularity.* ***Bottom:*** *Document request processes. Their superposition is the total request process.*

val $[a_d, a_d + \tau_d]$, where the random variable $\tau_d$ denotes the document lifespan. We further assume that the distribution of the pair $(\lambda_d, \tau_d)$ is stationary.
We consolidate the two layers into the *total request process*, which is the superposition of the document request processes. Figure 2 shows a schematic view of our model.

## 3. Hit ratio estimates and model validation

In order to estimate the hit ratio, we introduce the auxiliary process $X_t$, that counts the number of different requested objects up to time $t$. We then observe that for a LRU cache of size $C$, if an object is requested at time zero and later at time $\Theta > 0$, the latter request will be a hit if and only if $X_\Theta < C$. Equivalently, the request at time $\Theta$ will be a hit if and only if $\Theta < T_C$, where $T_C$ is the first passage time of process $X$ at level $C$.
We then invoke the so-called "Che approximation" [1], in which we assume that the random variable $T_C$ is concentrated and thus can be approximated by a constant $t_C$.
To calculate the constant $t_C$, we notice that, by definition of $T_C$, we have the equality $X_{T_C} = C$. Thus, we impose that the constant $t_C$ must satisfy this equality in mean, that is, $\mathbb{E}[X_{t_C}] = C$.
Process $X$ is a non-homogeneous Poisson process. Define its mean function by $\Xi(t) = \mathbb{E}[X_t]$; as $\Xi$ is an increasing function, $t_C$ is then given by the formula $t_C = \Xi^{-1}(C)$. We make $\Xi$ explicit in the following proposition.

**Proposition 1** *The mean of the process $X$ is given by*

$$\Xi(t) = \gamma\,\mathbb{E}\left[2t + \left(1 - e^{-\lambda t}\right)\left(\tau - t - \frac{2}{\lambda}\right)\mathbb{1}_{\{\tau \geq t\}}\right] + \gamma\,\mathbb{E}\left[2\tau + \left(1 - e^{-\lambda \tau}\right)\left(t - \tau - \frac{2}{\lambda}\right)\mathbb{1}_{\{\tau < t\}}\right]$$

*where $(\lambda, \tau)$ is distributed as any $(\lambda_d, \tau_d)$.*

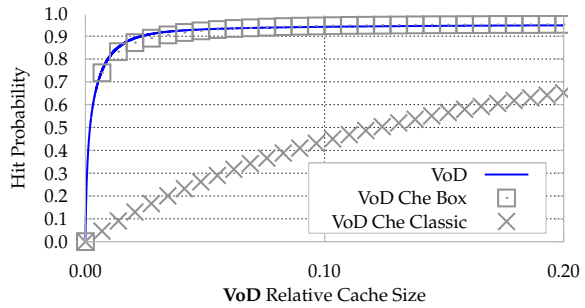

Figure 3: *Fitting for the Che estimation*

Finally, we write the expected number of hits for a document $d$ in terms of its associated pair $(\lambda_d, \tau_d)$ and $t_C$:

**Proposition 2** *Under the Che approximation, given pair $(\lambda_d, \tau_d)$, the expected number of hits $\overline{H}_d$ equals*

$$\overline{H}_d = \begin{cases} \lambda_d \tau_d - 1 + e^{-\lambda_d \tau_d} \\ \\ (\lambda_d \tau_d - 1)(1 - e^{-\lambda_d t_C}) + \lambda_d t_C e^{-\lambda_d t_C} \end{cases}$$

*if $\tau_d < t_C$ and $\tau_d \geq t_C$, respectively.*

For each document, unbiased estimates for $\lambda$ and $\tau$ can be easily calculated; call these estimators $\widehat{\lambda}$ and $\widehat{\tau}$, respectively. Such estimators are valid only for documents with a number of requests $n \geq 2$; we cannot neglect, however, the documents that have only one request, since they compose a considerable part of the data. Due the fact that they add only misses, we can still incorporate them into the hit ratio as follows:

$$\mathrm{HR} = \frac{\mathbb{E}[H_d]}{\mathbb{E}[n_d]} = \frac{\mathbb{E}[H_d \,|\, n_d \geq 2]}{\mathbb{E}[n_d \,|\, n_d \geq 2] + \dfrac{\mathbb{P}(n_d = 1)}{\mathbb{P}(n_d \geq 2)}}.$$

Using the previous estimates, we compute $t_C$ and $\mathbb{E}[H_d \,|\, n_d \geq 2]$ by plugging estimates $\widehat{\tau}$ and $\widehat{\lambda}$ into the formulas obtained in Proposition 1 and 2 and take averages.
We observe the result of this estimation on the VoD dataset in Figure 3, where we compare it to the actual hit ratio of the request sequence, obtained via simulation, and the estimation obtained via the "classic" Che approximation under the IRM setting. We observe that the hit ratio is noticeably underestimated in the latter case, whereas our model fits well the real hit ratio.

## References

1. Hao Che, Ye Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *Selected Areas in Communications, IEEE Journal on*, 20(7), 2002.
2. N. Hohn, D. Veitch, and P. Abry. Cluster processes, a natural language for network traffic. *IEEE Transactions on Signal Processing, special issue "Signal Processing in Networking"*, 51(8), Aug. 2003.
3. Felipe Olmos, Bruno Kauffmann, Alain Simonian, and Yannick Carlinet. Catalog dynamics: Impact of content publishing and perishing on the performance of a LRU cache. In *26th Internation Teletraffic Congress*. IEEE Communications Society, 2014. To appear.

# Approximate models for cache analysis with correlated requests

Nicaise E. Choungmo Fofack

Orange Labs
38/40 rue du Général Leclerc, 92130
Issy-Les-Moulineaux, France
nicaise.choungmofofack@orange.com

## 1. Introduction

Content distribution and in-network caching have emerged as key features of several network architectures to accommodate the current content usage patterns (Video on Demand, User-Generated Contents, Web browsing, the recent Dynamic Page Caching concept of Akamaï, etc.) while reducing congestion and improving access speed as networks increase in size. In order to guarantee the latter performance, network designers and engineers need tools to quickly gain insights on the behaviour of the multi-cache systems that arise from their content-oriented architectures when deploying caches across the networks.

The analysis of these cache networks is significantly challenging due to their irregular topologies, the heterogeneity of their nodes (different replacement policies and capacities), and the statistical correlation of requests. Few modelling attempts have been proposed to solve this problem on general and heterogeneous Time-To-Live (TTL)-based cache networks where requests are described by Markov-renewal processes [1] or Markov-arrival processes (MAPs) [2]. However, the latter derivations suffer from a large complexity which limits their practical interest.

In this work, we aim at presenting an analytical methodology to approximate the performance metrics of heterogeneous networks where $(i)$ caches are running the Least Recently Used (LRU), First-In First-Out (FIFO), or Random (RND) replacement algorithms, $(ii)$ all requests are routed as a feed-forward (or hierarchical) network, $(iii)$ streams of requests are described by simple MAPs [3] as briefly explained in Section 2.1.

## 2. Single cache approximation (SCA)

We consider a cache system that can accommodate $C$ files requested from a catalog of size $K$ ($K > C$). We assume that the cache is connected to a server where all files are permanently stored such that requests arrived first on the cache and the missed ones are forwarded to the origin server. We also assume a *zero delay* cache-server system i.e. request processing/forwarding and file downloading times are negligible in comparison to the inter-request times [1].
In this section we present a simplified traffic model which accounts for request correlations, then we recall the TTL-based model of LRU and FIFO/RND caches, and we provide the approximate metrics of interest (hit and occupancy ratios).

### 2.1. Workload model

Let us denote by $\{\mathcal{R}_k, k = 1, \ldots, K\}$ the general point process describing the sequence of requests for file $k$ on the cache. Here, we aim at providing a simple and accurate approximation of $\mathcal{R}_k$ based on a *minimal available information* that engineers could easily measure or estimate at edge nodes of the network. As inputs of our process model, we choose the *request rate*, the *squared coefficient of variation* (scv), the *skewness* factor, and the *lag-1* (and possibly *lag-2*) autocorrolation. These quantities are general enough to capture the main statistical parameters (per-file popularity, temporal locality, and correlation coefficients) of the arrival process $\mathcal{R}_k$ with a MAP (Cf. the *MAP-Match* procedure in [3]). Therefore, all request streams in this paper will be approximated by the switched MAP$(\mathbf{D}_0, \mathbf{D}_1)$ described in [3, Sect.4].

### 2.2. Cache model

We recall a result from [1] that the characteristic time (i.e. the maximum inter-request time of a given file that yields a cache hit) converges in distribution (under some conditions that hold in practice) to deterministic and exponentially distributed random variables as $K \uparrow \infty$ for LRU/FIFO and RND caches respectively under general stationary request processes. Hence LRU, FIFO and RND caches may be analysed through their corresponding TTL-based models. In this work, we consider that TTLs are i.i.d with an Erlang distribution $E_r(m, \mu)$ (or a PH-distribution PH$(\alpha, \mathbf{T})$ with $m$ transient states). For RND and LRU/FIFO caches we set $m = 1$ and $m \approx 10 \gg 1$ (sufficiently large to approximate a constant) respectively.

### 2.3. Performance metrics

In this section, we focus on the calculation of the file hit $\{H_k\}_{k=1}^K$ (resp. occupancy $\{O_k\}_{k=1}^K$) probabilities

defined as the stationary probability that a request for file $k$ yields a cache hit (resp. the stationary probability that file $k$ is in the cache at any time). To do so, we should find the rate $\mu$ of the TTL distribution. This is done by solving the following equation $\sum_{k=1}^{K} O_k = C$. Based on the exact results derived in [1, 2] we are able to derive the closed-form expressions of $H_k$ (see [2, Lemma 5]) and $O_k$ (see [1, Prop. 2.9]) needed in the latter equation.

## 3. Networks with feed-forward routing

In this section, we generalize previous results to cache networks where requests flow in the same direction (i.e. a cache does not query a cache from which it receives missed requests) by describing the main network traffic transformations in cache network deployments where requests are correlated and routed on a unique feed-forward network.

### 3.1. Miss process characterization

The SCA extends easily to tandem or **linear cache networks**. Indeed, one need to characterize the miss process of each cache, then apply the *MAPMatch* procedure on the miss streams, and calculate the metrics of interest as done in Section 2. If request processes are MAPs, so are miss processes of PH-distributed TTL models [2, Thms 3 & 4]. However, the complexity of the exact miss process characterization [2] is significantly reduced with our workload model.

### 3.2. Multiple sources

A cache may receive requests from several (not necessarily independent) sources such as caches or users. This is the common case for **tree networks**. The exact characterization of the aggregation of $N$ MAPs requires a strong independence assumption among request streams being superposed, an important calculation to evaluate the Kronecker sum, and a huge memory to store the resulting MAP (limitations of the exact method in [2]). In this work, we propose to calculate the parameters needed for MAP matching [3] as weighted sum of corresponding parameters of MAP components (e.g the resulting scv is $c_v^2 = \sum_{n=1}^{N} \frac{\lambda_n}{\Lambda} c_n^2$ where $\lambda_n$ and $c_n^2$ are the rate and the scv of the $n$-th MAP being aggregated, $\Lambda = \sum_{n=1}^{N} \lambda_n$ is the total request rate). Using the MAP matched of the overall process and the SCA, we obtain the performance metrics.

### 3.3. Several destinations

In many situations, a cache may select the next hop among two or more caches to forward its missed requests. The latter task may be performed based on some available informations (e.g. fault tolerance, multiple shortest paths, load balancing, etc.). We denote by $r_j$ the probability that the outgoing-link $j$ is chosen for request forwarding such that $\sum_j r_j = 1$ (e.g. $r_j$ may be the fraction of time the outgoing-link $j$ is up while others are down). If the cache miss processes are MAPs, so are the request processes on each link $j$ (see [2, Lemma 4]). Finally, the MAPMatch procedure [3] is applied on each $r_j$-thinned request process.

## 4. Conclusion

In this paper, we proposed a methodology that can be used to quickly estimate the performance metrics of single TTL-based (e.g. LRU, FIFO, RND) cache when request streams are approximated by special Markov Arrival Processes. We have also explained how our modelling attempt extends heterogeneous cache networks where all requests are routed on a unique/same feed-forward network (e.g. linear, tree, polytree) built on top of the network topology. Ongoing work are devoted to provide a detailed description of the latter extensions, to perform an extensive evaluation of our models, and to investigate the case of general and heterogeneous caches networks with arbitrary routing.

## References

1. Choungmo Fofack (Nicaise). – On models for performance analysis of a core cache network and power save of a wireless access network. –Ph.D. thesis, Univ. of Nice Sophia Antipolis, `http://tel.archives-ouvertes.fr/tel-00968894`, Feb. 2014.
2. Berger (D. S.), Gland (Philipp), Singla (Sahil) and Ciucu (Florin). – Exact Analysis of TTL Cache Networks : The Case of Caching Policies driven by Stopping Times. – Preprint ArXiv (CoRR, abs/1402.5987), `http://arxiv.org/abs/1402.5987`, 2014.
3. Horváth (Gabor). – Matching marginal moments and lag autocorrelations with MAPs. – In Proc. ValueTools'13, Torino, Italy, Dec. 2013.

# Reward-based Incentive Mechanisms for Delay Tolerant Networks

Seregina Tatiana

CNRS, Univ de Toulouse, INSA, LAAS,
7 avenue du colonel Roche,
F-31400 Toulouse, France
seregina@laas.fr

## 1. Introduction

Delay tolerant networking (DTN) [1] is an architecture that promotes data transmission even in absence of end-to-end connectivity at a given time. In DTNs, forwarding of a message is performed through mobile nodes in an opportunistic manner.

To cope with intermittent connectivity and to increase delivery rate, DTN routing protocols commonly appeal to multi-copy forwarding, that is the message is replicated to relay nodes and it is delivered if one of the copies reaches the destination.

The two-hop routing scheme [2], which we consider in this paper, allows the source to forward copies of the message to any idle node in contact, and a relay that receives the message from the source can transmit it only to the destination.

However, in DTN applications, when mobile devices are controlled by selfish owners, DTN nodes can be unwilling to spend their own resources serving as relays or can try to gain benefits from cooperation.

Some form of virtual currency is proposed by credit-based incentive mechanisms [3] to encourage communication among different nodes in DTNs. The studies in this work are centred on such type of incentives, namely addressing the reward-based incentive scheme introduced in [4] for two-hop routing. Each node that accepts the message has a certain probability to reach the destination, and the incentive scheme proposes to reward the relay that is the first one to deliver. The success probability and expected reward estimated by a relay depend on certain informations that the source announces to a relay in contact.

This paper investigates the impact of the announced information on the expected reward the source has to pay, and the aim is to find a reward policy for the source that is optimal in costs.

The results are obtained in two cases. First, for two relays and a decreasing density function of inter-contact times between relays and the source (destination), it is shown that the optimal reward policy corresponds with adapting the information the source can give to the relays. This adaptive strategy is shown to be of threshold type. Second, for an arbitrary number of relays and exponentially distributed inter-contact times, it is proved that the expected reward the source has to pay remains the same within fixed setting of informing the relays. The source can, however, reduce the expected reward, by adapting the announced information according to the meeting times with the relays.

## 2. System Model and Problem Description

The network considered consists of one fixed source node, one fixed destination node and a finite number of relays. The relays move according to a mobility model with i.i.d. inter-contact times between a relay and the source (resp. destination) corresponding to density function $f_s$ (resp. $f_d$).

The source generates a message at time 0 and attempts to forward it to the destination through the relay nodes by replicating the message, under the two-hop routing scheme.

Forwarding of the message imposes a certain set of costs for a relay node :

1. reception cost, $C_r$, is a fixed energy cost for receiving the message from the source,
2. storage cost, $C_s$, is the cost incurred per unit time a relay carries the message,
3. delivery cost, $C_d$, is the fixed cost for transmitting the message to the destination

As an incentive for cooperation, the source proposes to a mobile node a reward that promises to compensate costs that the node expects to incur if it accepts the message. The source is assumed to be not informed whether the message is delivered by any relay, and continues to offer message copies to relays when encountering. The expected costs and the expected reward estimated by a relay depend on the information the source proposes to the relays.

A mobile node encountering the source knows the time when the transmission process started. In addition, the node may be given the history of forwarding process according one of three settings that the source can use :

1. *full information*, when each mobile node is informed about the number of already existing copies and the corresponding contact times,
2. *partial information*, when each mobile node knows only the number of existing copies,
3. *no information*, when the relays know only their own meeting times with the source.

The reward a relay gets is a random variable depending on the stochastic process describing contacts with the destination. Moreover, the information the relay is given by the source may distinctly affect its estimation of the expected reward.
Thus, within each of three settings and based on the node mobility model, the source needs to determine an amount of reward it should propose to a mobile node when encountering it, and then deploy an optimal policy of informing the nodes in order to reduce the expected reward it should pay for message delivery.

### 2.1. Two Relays, Decreasing Function of Inter-Contact Times

Consider a case of two mobile nodes. Assume a decreasing density function of inter-contact times between the relays and the source (destination).
For fixed $s_1$ and $s_2$, denote by $\mathbf{s}$ the vector $(s_1, s_2)$.

#### 2.1.1. Expected Cost for a Relay

Define $V_i^{(k)}(\mathbf{s})$ as the net cost for relay $i = 1, 2$ under the setting $k \in \{F, P, N\}$, and let $R_i^{(k)}(\mathbf{s})$ be the reward asked by this relay to the source under this setting. Then, the expected cost for relay $i$ takes the form,

$$\mathbb{E}[V_i^{(k)}(\mathbf{s})] = C_r + C_s\mathbb{E}[\tilde{T}_d] + [C_d - R_i^{(k)}(\mathbf{s})]p_i^{(k)}(\mathbf{s}),$$

with $\tilde{T}_d$ being the residual inter-contact time for a relay to meet the destination, and $p_i^{(k)}(\mathbf{s})$ denotes success probability estimated by relay $i$ under the setting $k$ when it meets the source.

#### 2.1.2. Expected Reward Paid by the Source

The expected reward paid by the source for given contact times $\mathbf{s} = (s_1, s_2)$ and under setting $k$, is as follows

$$R_S^{(k)}(\mathbf{s}) = R_1^{(k)}(1 - p_2(\mathbf{s})) + R_2^{(k)}p_2(\mathbf{s}),$$

where $p_2(\mathbf{s})$ is the probability of success of the second relay given $s_1$ and $s_2$.

#### 2.1.3. Adaptive Strategy

It is always beneficial for the source to give information to the first relay independently of $s_1$.

**Proposition 1** $\quad R_1^{(F)}(\mathbf{s}) = R_1^{(P)}(\mathbf{s}) \le R_1^{(N)}(\mathbf{s}).$

The next proposition says that between the choice of informing a relay that is the second one and not giving this information, it is better for the source not to give this information.

**Proposition 2**

$$R_2^{(N)}(\mathbf{s}) \le R_2^{(P)}(\mathbf{s}).$$

Comparing the settings of full information with that of no information, the following proposition shows that for the source the adaptive strategy is of the threshold type.
Define the difference of the success probabilities as a function of $s_1$ and $s_2$,

$$g(s_1, s_2) = p_2^{(N)}(s_1, s_2) - p_2^{(F)}(s_1, s_2), \qquad (1)$$

then for the source, it will be better to give information when $g(s_1, s_2) < 0$.

**Theorem 1** *There exists $0 \le \theta_1 < \infty$ such that*

1. *if $0 \le s_1 < \theta_1$, then $g(s_1, s_1 + v) \ge 0, \forall v \ge 0$;*
2. *if $\theta_1 < s_1 < \infty$, then*
   *(a)* $g(s_1, s_2) < 0, \forall s_2 \in [s_1, s_1 + \omega(s_1))$,
   *(b)* $g(s_1, s_2) > 0, \forall s_2 \in (s_1 + \omega(s_1), \infty)$,

*where $\theta_1$ is a solution of the equation $g(s_1, s_1) = 0$ and $\omega(s_1)$ is a solution of $g(s_1, s_1 + v)$ with respect to $v$ when $g(s_1, s_1) < 0$.*

### 2.2. N Relays, Exponential Inter-Contact Times

Assume the inter-contact time between a relay and the source (resp. destination) is exponentially distributed with rate $\lambda$ (resp. $\mu$).

**Proposition 3** *For each setting $k \in \{F, P, N\}$ in the model with $N$ relays, $\overline{R}^{(k)} = C_1 + NC_2$, where $C_1 = C_d$ and $C_2 = C_r + C_s/\mu$.*

The average reward that the source has to pay is, thus, the same if the source adheres to a fixed informing mode and depends only on the number of involved mobile nodes. This reward can be, however, reduced by adapting the announced information according to meeting times with the nodes.

### References

1. K. Fall – A Delay-Tolerant Network Architecture for Challenged Internets. – In : SIGCOMM, pp. 27–34, 2003.
2. M. Grossglauser and D. Tse – Mobility Increases the Capacity of Ad Hoc Wireless Networks. – IEEE/ACM Trans. Networking, vol. 10, no. 4, pp. 477–486, 2002.
3. Y. Zhang, W. Lou, W. Liu and Y. Fang – A secure incentive protocol for mobile ad hoc networks. – Wirel. Netw., vol. 13, no. 5, pp. 569–582, 2007.
4. O. Brun, R. El-Azouzi, B. Prabhu and T. Seregina – Modeling Rewards and Incentive Mechanisms for Delay Tolerant Networks. – accepted for WiOPT-2014, 2014.

# Whittle's index in a multi-class queue with abandonments

M. Larrañaga[1,2,3], U. Ayesta[2], I.M. Verloop[1,3]

[1]IRIT, CNRS, 31071 Toulouse, France.
[2]LAAS, CNRS, F-31400 Toulouse, France.
[3]Univ. de Toulouse, INP, LAAS, F-314000 Toulouse, France.

## 1. Introduction

Abandonment or reneging takes place when customers, unsatisfied of their long waiting time, decide to voluntarily leave the system. It has a huge impact in various real life applications such as the Internet or call centers, where customers may abandon while waiting in the queue. In this paper we study the phenomena of abandonments in a single server queue with multiple classes of users with the objective of finding resource allocation strategies. In the presence of abandonments and/or convex holding cost, a characterization of the optimal control is out of reach. We therefore develop approximations to tackle the problem, in particular, we study Whittle's index.

Index Rules have enjoyed a great popularity, since a complex control problem whose solution might, a priori, depend on the entire state space turns out to have a strikingly simple structure. In a seminal work, Whittle introduced the so-called Restless Multi Armed Bandit Problems (RMABP), see [2]. In a RMABP all bandits (in our study a bandit is a class of customers) in the system incur a cost, the scheduler selects one bandit to be made active, but all bandits might evolve over time according to a stochastic kernel that depends on whether the bandit was active or *frozen*. The objective is to determine the control policy that, based on the entire state-space description, selects the bandit with the objective of optimizing the average performance criterion. Whittle introduced an approximate control policy of index-type, which is nowadays referred as Whittle's index.

The multi-class single server queue with abandonments can be modeled as a RMABP problem and Whittle's index can be derived.

The reminder of the paper is structured as follows : in Section 2 the model under study is described, and the objective of the study is presented, in Section 3 the Whittle index is derived and in Section 4 the performance of this index policy is measured. For additional results and proofs we refer to [1].

## 2. Model Description

We consider a multi-class single-server queue with $K$ classes of customers. Class-$k$ customers arrive according to a Poisson process with rate $\lambda_k$ and have an exponentially distributed service requirement with mean $1/\mu_k$, $k = 1, \dots, K$. We denote by $\rho_k := \lambda_k/\mu_k$ the traffic load of class $k$, and by $\rho := \sum_{k=1}^{K} \rho_k$ the total load to the system. We model abandonments of customers in the following way : any class-$k$ customer in the system that has not completed service, abandons after an exponentially distributed amount of time with mean $1/\theta_k$, $k = 1, \dots, K$, with $\theta_k > 0$.

The server has capacity 1 and can serve at most one customer at a time, where the service can be preemptive. At each moment in time, a policy $\varphi$ decides which class is served. Because of the Markov property, we can focus on policies that only base their decisions on the current number of customers present in the various classes. For a given policy $\varphi$, $N_k^{\varphi}(t)$ denotes the number of class-$k$ customers in the system at time $t$, (hence, including the one in service), and $\vec{N}^{\varphi}(t) = (N_1^{\varphi}(t), \dots, N_K^{\varphi}(t))$. Let $S_k^{\varphi}(\vec{N}^{\varphi}(t)) \in \{0, 1\}$ represent the service capacity devoted to class-$k$ customers at time $t$ under policy $\varphi$. The constraint on the service amount devoted to each class is $S_k^{\varphi}(\vec{n}) = 0$ if $n_k = 0$ and $\sum_{k=1}^{K} S_k^{\varphi}(\vec{n}) \leq 1$. The above describes a birth and death process.

Let $C_k(n_k, a)$ denote the cost per unit of time when there are $n_k$ class-$k$ customers in the system when class $k$ is not served (if $a = 0$), or when class $k$ is served (if $a = 1$).

We further introduce a cost $d_k$ for every class-$k$ customer that abandoned the system when not being served. We then want to find the optimal scheduling policy $\varphi$ under the average-cost criteria, that is,

$$\limsup_{T\to\infty} \sum_{k=1}^{K} \frac{1}{T} \mathbb{E}\left( \int_0^T \tilde{C}_k(\vec{N}^{\varphi}(t), S_k^{\varphi}(\vec{N}^{\varphi}(t)))\, dt \right), \tag{1}$$

where $\tilde{C}_k(n_k, a) := C_k(n_k, a) + d_k\theta_k n_k$ in state $n_k$ and under action $a$.

## 3. Whittle's index

The approach by Whittle is based on relaxing the original problem, allowing one bandit being made active on average, that is,

$$\limsup_{T\to\infty} \frac{1}{T} \int_0^T \sum_{k=1}^{K} S_k^{\varphi}(\vec{N}^{\varphi}(t))dt \leq 1. \tag{2}$$

This allows to decompose the original control problem into individual problems for each bandit. Whittle's index can then be interpreted as the Lagrange

multiplier of the constraint such that a given state joins the passive set.
The objective is now to determine the policy that solves (1) under Constraint (2). This can be solved by considering the unconstrained control problem

$$\limsup_{T\to\infty} \sum_{k=1}^{K} \frac{1}{T}\mathbb{E}\left(\int_0^T \tilde{C}_k(\vec{N}^{\varphi}(t), S_k^{\varphi}(\vec{N}^{\varphi}(t))) - W(1 - S_k^{\varphi}(\vec{N}^{\varphi}(t)))\, dt\right), \quad (3)$$

where $W$ is the Lagrange multiplier and $\tilde{C}(\cdot,\cdot)$ is assumed to be convex and increasing. We observe that the multiplier $W$ can be interpreted as a subsidy for passivity.
In summary, the relaxed optimization problem can be written as $K$ independent one-dimensional Markov decision problems, namely :

$$\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\left(\int_0^T \tilde{C}(N^{\varphi}(t), S^{\varphi}(N^{\varphi}(t))) - W(1 - S^{\varphi}(N^{\varphi}(t)))\, dt\right). \quad (4)$$

Under an indexability property (established in [1] for the present model), one can define class-based indices (the so-called Whittle's index) such that the solution to (1) is obtained by activating in every moment those classes whose current index is larger than the Lagrange multiplier.
We further prove that the optimal policy is monotone, it is fully characterized by a threshold $n$ such that the passive action is prescribed for states $m \leq n$, and the active for states $m > n$. We will then denote this policy as $\varphi = n$. We can now state the main result.

**Proposition 1** *The Whittle index for problem (4) is*

$$W(n) = \frac{\mathbb{E}(\tilde{C}(N^n, S^n(N^n))) - \mathbb{E}(\tilde{C}(N^{n-1}, S^{n-1}(N^{n-1})))}{\sum_{m=0}^{n} \pi^n(m) - \sum_{m=0}^{n-1} \pi^{n-1}(m)} \quad (5)$$

*with $\pi^n(m)$ the steady-state probability under threshold policy $n$.*

The heuristic for the original $K$-dimensional problem known as Whittle's index policy, prescribes to serve the class $k$ having currently the highest non-negative Whittle's index $W_k(n_k)$.
We now present Whittle's index for linear cost.

**Proposition 2** *Assume linear holding cost $C_k(n_k, a) = c_k n_k$. Then, Whittle's index for class $k$, with $\tilde{c}_k = c_k + d_k\theta_k$, is*

$$W_k(n_k) = \frac{\tilde{c}_k \mu_k}{\theta_k}, \text{ for all } n_k. \quad (6)$$

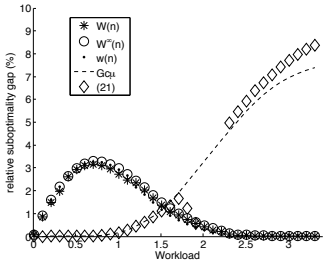


FIGURE 1 – Performance of Whittle's index policy.

Whittle's index for a more general setting (with convex holding cost) as well as limiting properties are analyzed in [1]. This study allows to recover indices that have been obtained in the literature. Observe that (6) is independent on the number of class-k customers, which is not the case for convex holding costs.

## 4. Numerical Results

In this section we numerically analyze the performance of Whittle's index policy. In Figure 1 we plot the relative sub optimality gap of the $W(n)$ index as well as other indices obtained in [1]. The optimal policy is computed using a Value Iteration algorithm. We carried out extensive simulations and were able to conclude that Whittle's Index policy is optimal as the workload increases, whereas indices that have been computed for multi-class systems without abandonments behave poorly.

## References

1. M. Larrañaga, U.Ayesta, I.M. Verloop. – Index policies for multi-class queues with convex holding cost and abandonments. – ACM Sigmetrics 2014.
2. P. Whittle – Restless bandits : Activity allocation in a changing world. – Journal of Applied Probability, 25 :287-298, 1988.

# Good Practices for Reproducible Research

Luka Stanisic and Arnaud Legrand

MESCAL/LIG/Univ. of Grenoble, Grenoble, France
firstname.lastname@imag.fr

## 1. Introduction

In the last decades, computers, operating systems and software running on it have reached such a level of complexity that it has become very difficult (not to say impossible) to control them perfectly and to know every detail about their configuration and operation mode. As a consequence, it becomes less and less reasonable to consider computer systems as deterministic, especially when measuring execution times or performance of large distributed computing systems. These systems have become so complex that it is almost impossible to fully understand their behavior, making the full reproduction of measurements extremely difficult. To some extent, although these systems have been designed and built by humans, studying modern computers has become very similar to studying a natural phenomenon. Just like most other scientific domain, many of the conclusions are based on experiment results and their analysis. Nevertheless, in many computer science articles, the experimental protocol is rarely detailed enough to allow others to reproduce the study and possibly build upon it. This is all the more surprising as reproducibility and falsifiability are yet the basis of modern science as defined by Popper.

In the recent years, the more and more frequent discovery of frauds or mistakes in published results has shed the light on the importance of reproducible research in computational sciences. Many tools partially addressing these issues have thus been proposed in different fields (biology, image processing, etc.) and although there is an urgent need for changing practices in computer science, how we should proceed is not yet clear. We have recently written an article [3] that contains a completely replicable analysis of the large set of experiments. We took care of doing these experiments in a clean, coherent, well-structured and reproducible way that allows anyone to inspect how they were performed.

**In this presentation, we propose to explain and demonstrate the basics of our workflow and the technology we used.**

## 2. Context

The reproducible research [3], of which we will illustrate the internals, was done in the context of High-Performance Computing (HPC). Modern HPC infrastructures are typically made of hybrid machines with several CPUs and GPUs. To efficiently exploit all these heterogeneous resources, programmers have to explicitly allocate computation-intensive kernels and manage data transfers between the different processing units. It is common to rely on task-based runtimes that provide the right abstraction to efficiently perform such tedious management and optimizations. Designing and configuring such runtime systems is itself a challenging problem for which we believe the systematic use of modeling and simulation can provide an answer. To support our claim, we have ported StarPU [1], a dynamic task-based scheduling runtime for hybrid architectures, on top of the SimGrid simulator [2] and we have shown that this combination allowed to provide accurate performance predictions for dense linear algebra kernels.

To validate our approach and our models, we had to perform measurements on several, sometimes not dedicated, machines, and work on complex beta code that often needed to be modified to fit our needs. In this context, even the smallest misunderstanding or inaccuracy about a parameter at small scale often results in a completely different behavior at the macroscopic level. It was thus crucial to carefully collect all the useful metadata, to use a well-planed experiment design and a coherent analysis, all in order to be able to easily reproduce the experiment results.

## 3. From Experiments to Article: Reproducibility vs. Replicability

We can distinguish between two main trends in reproducible research. The first one aims at completely automatizing the whole process (experiments, analysis and the final article), allowing others to replicate it. This approach is particularly popular in computational sciences. However, maintaining such process functional in time and portable across platforms is quite burdensome. Additionally, it imposes restrictions on what can be done, it is difficult to extend and hence not always usable on a daily basis. Furthermore, focusing on automatic replication hides information on why and how things were done. The second approach is much lighter and relies on literate programming of code and analysis scripts (i.e. documenting why and how to use the code) and on the systematic use of recipes (i.e., how to build environments). These two approaches apply to both experiments and analysis part of the research.

### 3.1. Reproducible Experimentation

When measuring execution time of a complex application on a modern computer, two runs in a row with the same setup rarely provide the exact same timing in nanoseconds. Such experiments are de facto not *replicable*, but only *reproducible* provided that enough details about the configuration are available. We have thus taken a particular care to ensure that all our experiment results are reproducible and will describe in our presentation the most important ones.

Before each measurement, we **automatically log information** about the machine, such as the current CPU frequency and governor, the memory hierarchy, the versions of Linux and gcc, etc. We also **systematically recompile** software before using it to conduct experiments, and keep all the configuration and compilation outputs. Additionally, we enforce that all changes to the source code are committed in the **revision control system** and we keep track of the hash of the Git/SVN version of the source code. This enables **provenance tracking**, i.e, to know which results were obtained with which code, so they can be later easily investigated, compared or reproduced. Finally, we save the measurement results (makespan, GFlop/s rate) together with execution traces. During the development period, workflow and data format went through several changes and adjustments and we used a **laboratory notebook** to keep track about all modifications.

### 3.2. Replicable Analysis and Article

Unlike experimentation, analysis is only concerned with the final output (figures, tables, numerical summaries) and not with the time it takes to compute it. Therefore, for a given experimental data set, results of the analysis should always remain the same, which makes them by essence *replicable*. Reproducible research tools and technologies are still in their infancy and there are many different alternatives (Rstudio/knitr, ActivePapers, Elsevier executable papers, etc.). We tried several of them and we have finally chosen to use **org-mode**, more precisely org-babel, that allows to **combine multiple processing and analysis codes** as well as their **output results** inside a plain text file. Compared to other alternatives, we feel that org-mode is mature, flexible and lightweight enough for both a daily usage and to allow to produce high quality reproducible articles.

The article we have written **combines within a single plain text file the body of the article along with all the analysis**, all of which are later exported into standard pdf document. It has a hierarchical structure, with different types of code, including Shell (to manipulate data files), R (for plotting figures) and LaTeX (to finely control formatting details). We made **all raw data and traces publicly available** on figshare [4]. To replicate the article, all these data should be downloaded and unpacked. Then all useful information is extracted into csv files by parsing and filtering raw data files that contain many additional metadata. Finally, R is used to load, process and plot data, ensuring that all figures are consistent throughout the whole article. This way, **our experimental and analysis results can be inspected, referenced, or even reused** in any other research project.

## 4. Conclusion

The approach we propose to present was very effective for doing reproducible research in our context. We strongly believe that it can be easily applied to many other situations and that presenting such methodology can be beneficial to the audience.

## References

1. Cédric Augonnet, Samuel Thibault, Raymond Namyst, and Pierre-André Wacrenier. StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. *Concurrency and Computation: Practice and Experience, Special Issue: Euro-Par 2009*, 23:187–198, February 2011.
2. Henri Casanova, Arnaud Legrand, and Martin Quinson. SimGrid: a Generic Framework for Large-Scale Distributed Experiments. In *proceedings of the 10th IEEE International Conference on Computer Modeling and Simulation (UKSim)*, April 2008.
3. Luka Stanisic, Samuel Thibault, Arnaud Legrand, Brice Videau, and Jean-François Méhaut. Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures. In *Proceedings of the 20th Euro-Par Conference*. Springer-Verlag, August 2014.
4. Companion of the StarPU+SimGrid article. Hosted on Figshare, 2014. Online version of this article with access to the experimental data and scripts (in the org source); http://dx.doi.org/10.6084/m9.figshare.928095.

# Approche par bornes stochastiques et histogrammes pour l'analyse de performance des réseaux

Farah Ait Salaht *

Laboratoire PRiSM
Université de Versailles St-Quentin de Yvelines
45, avenue des États-Unis
78035 Versailles
Farah.Ait-Salaht@prism.uvsq.fr

## 1. Introduction

Les mesures de performance des réseaux deviennent de plus en plus nombreuses et beaucoup de traces sont disponibles pour tester des hypothèses sur le trafic. Par contre, ces traces sont trop volumineuses pour servir directement à l'analyse de performances. De plus, la modélisation du trafic est généralement impossible à effectuer de façon suffisamment précise et l'adéquation avec une loi de probabilité connue n'est pas assez réaliste.

Reposant sur une description par histogramme des traces, nous proposons une nouvelle approche fondée sur la comparaison stochastique pour réduire la taille des distributions empiriques. Nous construisons grâce à un algorithme optimal, que nous avons prouvé [1], deux distributions discrètes plus petites (au sens de leurs supports) et qui sont des bornes stochastiques inférieures et supérieures de la distribution initiale. La théorie de la comparaison stochastique est alors utilisée pour obtenir des bornes de toute mesure de performance qui s'écrit comme une fonction croissante sur la distribution de probabilité (longueur de la file, pertes, etc.). Nous n'avons donc plus besoin du caractère exponentiel des interarrivées et des services qui sont souvent utilisés pour l'analyse des performances de réseaux. Par contre, l'hypothèse de stationnarité du trafic est ici nécessaire.

L'approche reposant sur des histogrammes pour l'évaluation de performance n'est pas nouvelle. La plupart des travaux récents sur le sujet traitent des processus HBSP (Histogram Based Stochastic Process) proposés par Hernández-Orallo et ses co-auteurs [2]. L'idée de base est de réduire la taille des histogrammes par une agrégation de taille constante et de les injecter dans des files FIFO simples pour calculer numériquement la distribution stationnaire de la longueur des files. Par contre, l'utilisation des bornes stochastiques dans ce contexte nous semble originale. D'autant plus qu'elles garantissent l'optimalité des mesures bornantes.

*. Ce travail a été réalisé avec J.M. Fourneau, H. Castel et N. Pekergin.

## 2. Borne stochastique

On considère un espace d'état $\mathcal{G} = \{1, 2, \ldots, n\}$ muni d'un ordre total noté $\leq$. Soient X et Y deux variables aléatoires ayant $\mathcal{G}$ comme support et de distributions de probabilités $\boldsymbol{d}2$ et $\boldsymbol{d}1$ ($\boldsymbol{d}2[i]=\mathtt{Prob}(X = i)$, et $\boldsymbol{d}1[i]=\mathtt{Prob}(Y = i)$, pour $i = 1, 2, \ldots, n$).

**Definition 1 (Ordre Stochastique)** *:*

— *Définition générale :* $X \leq_{st} Y \iff \mathbb{E}f(X) \leq \mathbb{E}f(Y)$ *pour toute fonction croissante* $f : \mathcal{G} \to \mathbb{R}^+$ *pour peu que les espérances existent.*

— *Définition utilisant les distributions :*

$$X \leq_{st} Y \Leftrightarrow \forall i,\ 1 \leq i \leq n,\ \sum_{k=i}^{n} \mathbf{d}2[k] \leq \sum_{k=i}^{n} \mathbf{d}1[k].$$

*On utilisera de façon équivalente* $X \leq_{st} Y$ *et* $\mathbf{d}2 \leq_{st} \mathbf{d}1$.

L'ordre $\leq_{st}$ nous intéresse particulièrement, car il permet de borner les récompenses croissantes, cette propriété est souvent vraie pour les récompenses utilisées en évaluation de performances : moyenne, moment d'ordre quelconque, etc.

## 3. Méthode de réduction de la taille d'une distribution en la bornant [1]

Nous cherchons à réduire la taille des distributions que nous allons employer pour modéliser le trafic et la capacité (débit) de service dans les éléments de réseau. Nous fournissons un encadrement au sens stochastique des résultats. Ces encadrements nous permettent de vérifier que les contraintes de Qualité de Service sont satisfaites.

Soit une distribution $\boldsymbol{d}$, définie par un histogramme à N classes (états), on construit deux distributions bornantes $\boldsymbol{d}1$ et $\boldsymbol{d}2$ qui sont définies par des histogrammes à $n < N$ classes. De plus, $\boldsymbol{d}1$ et $\boldsymbol{d}2$ sont les meilleures approximations de $\boldsymbol{d}$ au sens d'une fonction de récompense $\boldsymbol{r}$ fournie par le modélisateur. Nous avons présenté dans [1] un algorithme reposant sur la programmation dynamique pour calculer de telles distributions bornantes optimales. Pour être plus formel, pour une distribution $\boldsymbol{d}$ définie sur $\mathcal{H}$ ($|\mathcal{H}| = N$), nous calculons deux distributions $\boldsymbol{d}1$ et $\boldsymbol{d}2$ définies respectivement sur $\mathcal{H}^u$, $\mathcal{H}^l$ ($|\mathcal{H}^u| = n$, $|\mathcal{H}^l| = n$) telles que :

1. $\boldsymbol{d}2 \leq_{st} \boldsymbol{d} \leq_{st} \boldsymbol{d}1$,

2. $\sum_{i\in\mathcal{H}} \boldsymbol{r}[i]\boldsymbol{d}[i] - \sum_{i\in\mathcal{H}^l} \boldsymbol{r}[i]\boldsymbol{d}2[i]$ est minimale dans l'ensemble des distributions sur $n$ états stochastiquement inférieure à $\boldsymbol{d}$,
3. $\sum_{i\in\mathcal{H}^u} \boldsymbol{r}[i]\boldsymbol{d}1[i] - \sum_{i\in\mathcal{H}} \boldsymbol{r}[i]\boldsymbol{d}[i]$ est minimale dans l'ensemble des distributions sur $n$ états stochastiquement supérieure à $\boldsymbol{d}$,

Il est important de noter que les deux supports des distributions bornantes sont issus de $\mathcal{H}$ mais ne sont pas en général égaux. C'est l'algorithme qui fixe le support et la distribution. Les distributions $\boldsymbol{d}1$ et $\boldsymbol{d}2$ représentent les bornes optimales sur $n$ états pour la récompense positive croissante $\boldsymbol{r}$.

## 4. Modèle d'une file FIFO

Soit $A(k)$ une variable aléatoire représentant le trafic entrant dans la file pendant le $k$-ième intervalle temporel (slot). Nous notons respectivement par $Q(k)$ et $D(k)$ la longueur du tampon et le nombre de sorties pendant ce même slot. Soit $B$ la capacité du tampon et $S$ la capacité de service. On se place dans un modèle où les arrivées ont lieu avant les services. Les services se terminent en fin d'intervalle temporel. Un paquet reste donc au minimum 1 slot dans la file. L'équation de récurrence sur la longueur du tampon dans ce modèle est bien connue [3] :

$$Q(k) = \mathtt{min}(B,\ (Q(k-1) + A(k) - S)^+), \qquad (1)$$

où $(X)^+ = \mathtt{max}(X, 0)$. On obtient également la distribution de sortie grâce à cette récurrence, pour tout $k$ :

$$D(k) = \mathtt{min}(S,\ Q(k-1) + A(k)). \qquad (2)$$

L'équation 1 (resp. l'équation 2) définit une chaîne de Markov en temps discret si les arrivées sont indépendantes et le processus $A(k)$ est stationnaire.

### 4.1. Bornes sur la file liées aux bornes sur les histogrammes de trafic

Pour une file à capacité finie $B$, nous injectons les trafics bornants (inférieurs et supérieurs) et nous établissons un résultat de comparaison stochastique. Supposons que le trafic d'entrée soit $\tilde{A}(k)$, un processus d'arrivée qui borne supérieurement ou inférieurement le trafic réel, on note $\tilde{Q}(k)$ l'état de la chaîne de Markov associée à ce processus d'arrivée. On note $\tilde{D}(k)$ le nombre de sorties sous les mêmes hypothèses. Le théorème 4.3.9 de [4] permet d'établir les propriétés suivantes :

**Proposition 1 (Bornes supérieures)** *Si* $A(k) \leq_{st} \tilde{A}(k), \forall k \geq 0$ *alors* $Q(k) \leq_{st} \tilde{Q}(k), \forall k \geq 0$

**Proposition 2 (Bornes inférieures)** *Si* $\tilde{A}(k) \leq_{st} A(k)$ *pour tout* $k \geq 0$, *alors* $\tilde{Q}(k) \leq_{st} Q(k)$ *pour tout* $k \geq 0$.

D'après l'équation 2, nous avons également des bornes sur le processus de sortie.

**Proposition 3** *Si* $A(k) \leq_{st} \tilde{A}(k)$ *pour tout* $k \geq 0$, *alors* $D(k) \leq_{st} \tilde{D}(k)$, $\forall k \geq 0$. *De même, si* $\tilde{A}(k) \leq_{st} A(k)$ $\forall k \geq 0$, *alors,* $\tilde{D}(k) \leq_{st} D(k)$, $\forall k \geq 0$.

Et puisque ces propositions sont vraies pour toutes les dates $k$, elles sont également vraies pour les versions stationnaires qui existent lorsque les chaînes sont ergodiques.

## 5. Conclusion

Notre méthode de bornes stochastiques offre un compromis intéressant entre la précision et la vitesse de calcul. En effet, selon le choix de la réduction apporté à la taille des distributions, nous pouvons déterminer des bornes pertinentes et un encadrement très fin du résultat exact en des temps relativement courts. Nous montrons également que l'élément du réseau est stochastiquement monotone ce qui nous permet d'assurer que les distributions bornantes calculées représentent bien des bornes de la distribution exacte. Ainsi, en appliquant nos algorithmes de bornes sur l'histogramme représentant la trace de trafic en entrée, nous définissons des histogrammes bornants sur les différentes mesures de performance : les probabilités de blocage, l'espérance de la longueur du tampon, etc.

## Bibliographie

1. F. Aït-Salaht, J. Cohen, H. Castel-Taleb, J.M. Fourneau et N. Pekergin. Accuracy vs. complexity : the stochastic bound approach. In 11th International Workshop on Discrete Event Systems, pages 343-348, 2012.
2. E. Hernández-Orallo and J. Vila-Carbó. Network queue and loss analysis using histogram based traffic models. Computer Communications, 33(2) :190-201, 2010.
3. L. Kleinrock. Queueing Systems, volume I : Theory. Wiley Interscience, 1975.
4. A. Müller and D. Stoyan. Comparison Methods for Stochastic Models and Risks. Wiley, New York, NY, 2002.

# On distribution and dependence of extremes in PageRank-type processes

Konstantin Avrachenkov*, Natalia M. Markovich**, Jithin K. Sreedharan*

* INRIA, Sophia Antipolis, France
k.avratchenkov@inria.fr,
jithin.sreedharan@inria.fr
**Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
markovic@ipu.rssi.ru

## 1. Introduction

Social networks contain clusters of nodes centered at high-degree nodes and surrounded by low-degree nodes. Such a cluster structure of the networks is caused by the dependence (social relationships and interests) between nodes and possibly by heavy-tailed distributions of the node degrees. We consider the degree sequences generated by PageRank type sampling processes and investigate clusters of exceedances of such sequences over large degrees. It is important to investigate its stochastic nature since it allows us to disseminate advertisement or collect opinions more effectively within the clusters.

The dependence structure of sampled degrees exceeding sufficiently high thresholds is measured using a parameter called extremal index, $\theta$. It is defined as follows.

**Definition 1** *[2] The stationary sequence $\{X_n\}_{n\geq 1}$, with $M_n = \max\{X_1, \ldots, X_n\}$ and $F$ as the marginal distribution function, is said to have the extremal index $\theta \in [0,1]$ if for each $0 < \tau < \infty$ there is a sequence of real numbers (thresholds) $u_n = u_n(\tau)$ such that*

$$\begin{aligned} \lim_{n\to\infty} n(1-F(u_n)) &= \tau \text{ and} \\ \lim_{n\to\infty} P\{M_n \leq u_n\} &= e^{-\tau\theta} \end{aligned} \tag{1}$$

*hold.*

Some of the interpretations of $\theta$ are [2] :

- Relation to the mean cluster size : A cluster is considered to be formed by the exceedances in a block of size $r_n$ ($r_n = o(n)$) in $n$ with cluster size $\xi = \sum_{i=1}^{r_n} 1(X_i > u_n)$ when there is at least one exceedance within $r_n$. The point process of exceedances which counts the number of exceedances, normalized by $n$, of $\{X_i\}_{i=1}^n$ over threshold $\{u_n\}$ converge weakly to a compound Poisson process (CP) with rate $\theta\tau$ under condition (1) and a mixing condition, and the points of exceedances in CP correspond to the clusters . Then $\theta = (E\xi)^{-1}$.
- We also have $P\{M_n \leq x\} = F^{n\theta}(x) + o(1), n \to \infty$. Hence $\theta$ allows us to evaluate a limit distribution of the maximum of the node degree and it is also helpful to find quantiles of the maxima. These quantiles show the large degrees in the network which arise with a certain probability.

The main contributions in this work are as follows. We study the extremal and clustering properties due to degree correlations in large graphs. Since the network under consideration is known only through Application Programming Interfaces (API), different graph exploring or sampling algorithms are employed to get the samples. The considered algorithms are Random Walk based that are widely discussed in the literature (see [1] and the references therein). In order to facilitate a painless future study of correlations and clusters of degrees in large networks, we propose to abstract the cluster statistics to a single and handy parameter, $\theta$. We derive analytical expressions of $\theta$ for different sampling techniques. Finally different estimators of $\theta$ are tried out and several other applications are proposed.

## 2. Model and Algorithms

We consider networks represented by an undirected graph $G$ with $N$ vertices and $M$ edges. In accordance with the data from most of the real networks, we assume the network is not known completely (with $N$ and $M$ unknown) and also assume correlation in degrees between neighbor nodes. The dependence structure in the graph is described by the joint degree-degree probability density function $f(d_1, d_2)$ (see e.g., [3]). The probability that a randomly chosen edge has the end vertices with degrees $d_1 \leq d \leq d_1 + \Delta(d_1)$ and $d_2 \leq d \leq d_2 + \Delta(d_2)$ is $(2 - \delta_{d_1 d_2}) f(d_1, d_2)\Delta(d_1)\Delta(d_2)$. Here $\delta_{d_1 d_2} = 1$ if $d_1 = d_2$, zero otherwise. The degree distribution $f_d(d_1)$ can be calculated from the marginal of $f(d_1, d_2)$ as

$$f(d_1) = \sum_{d_2} f(d_1, d_2) = \frac{d_1}{E[D]} f_d(d_1),$$

where $E[D]$ denotes the mean node degree. Then $E[D] = \left[\int\int (f(d_1, d_2)/d_1)\, d(d_1) d(d_2)\right]^{-1}$. Most of the results in this paper are derived assuming continuous probability distributions for $f(d_1, d_2)$ and $f_d(d_1)$ due to the ease in calculating $\theta$ for continuous case. In particular, for analytical tractability and with the sup-

port of empirical evidences, we assume the bivariate Pareto model for the joint degree-degree tail function.

### 2.1. Description of random walks

The different graph exploration algorithms considered in the paper are Random Walk based and transition kernels are defined for degree state space unlike in previous works where they were defined and well studied for vertex set (see [1] and the references therein). We use $f_{\mathscr{X}}$ to represent the probability density function under the algorithm $\mathscr{X}$.

#### 2.1.1. Standard random walk (RW)

In a standard random walk, the next node to visit is chosen uniformly among the neighbours of the current node. Using "mean field" arguments, the joint density function of the standard random walk is derived as $f_{RW}(d_{t+1}, d_t) = f(d_{t+1}, d_t)$.

#### 2.1.2. PageRank (PR)

PageRank is a modification of the random walk which with a fixed probability $1 - c$ samples random node with uniform distribution and with a probability $c$, it follows the standard Random walk transition. Its evolution can be described as

$$f_{PR}(d_{t+1}|d_t) = c f_{RW}(d_{t+1}|d_t) + (1-c) f_d(d_{t+1}).$$

Unfortunately, according to our knowledge, there is no closed form expression for the stationary distribution of PageRank and it is difficult to come up with an easy to handle expression for joint distribution. Therefore, along with other advantages, we consider another modification of the standard random walk.

#### 2.1.3. Random walk with jumps (RWJ)

This algorithm follows a standard Random walk edge with probability $d_t/(d_t+\alpha)$ and jumps to an arbitrary node uniformly with probability $\alpha/(d_t+\alpha)$, where $d_t$ is the degree of current node and $\alpha \in [0, \infty]$ is a design parameter ([1]). This modification makes the underlying Markov Chain time reversible, significantly reduces mixing time, improves estimation error and leads to a closed form expression for stationary distribution. The joint density for the random walk with jumps is derived as

$$f_{JP}(d_{t+1}, d_t) = \frac{E[D] f(d_{t+1}, d_t) + \alpha f_d(d_{t+1}) f_d(d_t)}{E[D] + \alpha}.$$

### 2.2. Calculation of $\theta$

The extremal index of the random walk based sampling algorithms can be calculated by means of copula density as

$$\theta = \lim_{x \to 1} \frac{x - C(x,x)}{1-x} = C'(1,1) - 1,$$

where $C(u,u)$ is the Copula function ($[0,1]^2 \to [0,1]$) and $C'$ is its derivative [4]. With the help of Sklar's theorem, $C(u,u) = P_{\mathscr{X}}(D_1 \le F^{-1}_{\mathscr{X}}(u), D_2 \le F^{-1}_{\mathscr{X}}(u))$ where $\mathscr{X}$ can be RW, PR or RWJ with $F^{-1}_{\mathscr{X}}()$ is the inverse of the stationary distribution function of the corresponding random walk.

### 2.3. Results

The extremal index is calculated analytically for the considered algorithms. Closed form expressions for RW and RWJ are obtained and a lower bound of $\theta$ for PR is derived.

As for numerical experiments, a random graph is generated as follows : $f(d_1, d_2)$ is taken as a bivariate Pareto distribution with suitable parameters and the degree distribution is calculated accordingly from (2). Then an uncorrelated graph is generated using the configuration model with this degree distribution. Finally the Metropolis algorithm is applied to rearrange the edges in order to get the desired joint degree-degree distribution.

The mean field argument for the transition kernel of the RW is checked with the generated graph and found to yield a reasonable match. $\theta$ is estimated for the different algorithms using the rank estimator of Copula function. Finally, for the application of cluster classification, we derive the number of nodes to be sampled to achieve a particular mean number of clusters.

### References

1. K. Avrachenkov, B. Ribeiro, and D. Towsley. Improving random walk estimation accuracy with uniform restarts. In *Lecture Notes in Computer Science*, v.6516, pages 98–109, 2010.
2. J. Beirlant, Y. Goegebeur, J.Teugels, and J. Segers. *Statistics of Extremes : Theory and Applications.* Wiley, Chichester, West Sussex, 2004.
3. M. Boguna, R. Pastor-Satorras, and A. Vespignani. Epidemic spreading in complex networks with degree correlations. *Statistical Mechanics of Complex Networks. Lecture Notes in Physica* v.625, pages 127–147, 2003.
4. A. Ferreira and H. Ferreira. Extremal functions, extremal index and Markov chains. Technical report, Notas e comunicações CEAUL, 12 2007.

# Averaging on Dynamic Networks

Mahmoud El Chamie, Giovanni Neglia, Konstantin Avrachenkov

INRIA Sophia Antipolis
2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

## 1. Introduction

In dynamic networks, the topology of the nodes in a network or links connecting these nodes changes with time. This can be due to mobility, link failure, or nodes failure. We are interested in this paper to study consensus on dynamic networks. Most of the work on consensus in dynamic network settings consider fixed number of nodes that are trying to reach agreement in the presence of either mobility or non-robust links (so only the links are dynamic) [1]. Average consensus on these networks is modeled following a random adjacency matrix $A^{n\times n}(k)$ where $n$ is the number of nodes and is fixed while the elements of this matrix are random (being for example i.i.d. at every iteration $k$). The study of consensus on that model is reduced to studying the convergence of the backward product of random matrices. Some papers give sufficient conditions on the weight matrices at every time iteration that guarantee convergence, others use coefficient of ergodicity as a tool to show the convergence of their system to consensus [2]. However, little study has been made on networks with dynamic number of nodes. In the latter case, the dimensions of the adjacency (and weight) matrices can be unbounded, and thus the traditional tools for studying the consensus are not applicable. We refer in this report to this type of networks : nodes arrive and leave as in a queuing system. We would like to study the effect of averaging in these networks, and to see if the nodes could actually reach consensus.

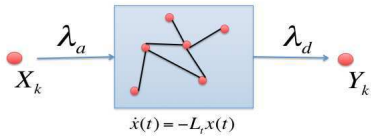


FIGURE 1 – The network model.

## 2. Model

Networks with dynamic nodes are characterized by nodes' arrival and nodes' departure (see Fig. 1). Each node arrives with a random value $X_k$ (where $k$ is the number of arrival and each node is labeled by its arrival number). We suppose that there is initially a node having a label 0 in the system that does not leave (it arrives at time $t = 0$ and does not depart). Let $X_0, X_1, X_2, \ldots$ be pairwise independent random variables following some distribution. Let $A_k$ be the arrival time of node $k$ and $D_k$ its departure time. At any instance $t$, nodes within the system are connected to each other by some network topology. Let $V(t)$ be the set of all nodes in the system at time instant $t$, i.e., $V(t) = \{k \mid A_k \leq t < D_k\}$. Let $Y_k(t)$ be the value of node $k$ that changes with time depending on its history of connection with other nodes, for example it should be clear that $Y_k(t)$ has a constant value before arrival ($Y_k(t) = X_k$ for $t < A_k$) and after departure ($Y_k(t) = Y_k(D_k)$ for $t \geq D_k$) but this value changes during the time it spends in the system because of interaction with other nodes. The nodes perform a continuous time averaging :

$$\dot{\mathbf{x}}(t) = -L_t \mathbf{x}(t), \tag{1}$$

where $\mathbf{x}(t)$ is the state vector of the nodes present in the system at time $t$, and $x_i(t) = Y_{\gamma(i)}(t)$ where $\gamma(i)$ is the arrival number of the $i$-th oldest node in the system (notice that $x_1(t) = Y_0(t)$ is always true for the node labeled 0 because it is the oldest node in system and does not depart). $L_t$ is the Laplacian of the graph at time $t$, and $\dot{\mathbf{x}}(t) = \frac{\partial \mathbf{x}(t)}{\partial t}$. We call this model a *consensus queue* due to its similarity to queuing systems. If the inter-arrival and inter-leave times are exponentially distributed random variables with a FIFO discipline (the nodes first to come are the first to leave except for node 0), then the system is $M/M/1$ consensus queue with arrival rate $\lambda_a$ and departure rate $\lambda_d$.

The Laplacian of the graph $L_t$ is used to give a general model for different graph topologies. However, this work is just preliminary and we will give some simplifications in the next section : 1) we only consider two graph topologies, the complete graph and the tree, 2) the averaging is faster than the dynamics of the queue (i.e., equation (1) converges before the arrival or the departure of a new node).

The model described here is interesting because it can be applied to different and diverse applications. Queuing consensus can be for example a model for human interactions and their behavior. Consider a system where people arrive at an open market and products' prices are not fixed. Each customer has

an initial estimation of the price of the product that varies depending on the interaction process with other customers in the system. You can also consider this model as a representative of a wireless sensor network where sensors monitor some environmental measurement (as temperature or pressure) where nodes can fail according to a Poisson process and new nodes are added to the network. We are interested then by the average in the system, mainly by :

$$Z(t) = \frac{1}{N(t)} \sum_{i=1}^{N(t)} x_i(t), \tag{2}$$

where $N(t) = |V(t)|$ is the number of nodes that are present in the system at time $t$ and $x_i(t)$ are their estimates. We note that $Z(t)$ is not a continuous process in general because with every arrival (or departure) with an estimate $X_k$ (or $Y_k$) different from $Z(t)$, the process jumps to a different value.

## 3. Simple Network Topologies

There are two sources of randomness in this model, the first one is the input estimate $X$ that follows some distribution, and the other one is the queuing system with random arrivals and departures. In the following sections, we will characterize the average in the system and the output process by considering several simplifications :

1. Complete Graph : the network is a full mesh network (all nodes in the system are connected to each others) and once a node enters a system, all nodes will have the average of nodes presented (instantaneous averaging), i.e., $x_i(t) = Z(t)$ for all $i$ in the system.
2. Directed Tree : nodes arriving can only connect to one node chosen uniformly at random in the system, and their estimate changes only once till they leave the network depending on the chosen node's estimate and it's distance from the root.

### 3.1. Complete Graph

Let $Z_k$ be the value of $Z(t)$ just after the $k$-th arrival and before the $k+1$-th arrival. Then $Z_k$ can be written as a weighted average of the nodes in the network, i.e., $Z_k = \sum_{i=0}^{k} w_i X_i$, where $w_i$ is the weight given to the value of node $i$ and it is a random variable depending on the stochastic arrivals and departures. It is important to study these weights to see how the system preserves the history of old values. To do this we take two extreme cases. The first case of no departures (if $\lambda_d = 0$), then we have

$$w_i = \frac{1}{k+1} \text{ for } i = 0, \ldots, k. \tag{3}$$

The second case of very fast departures ($\lambda_d >> \lambda_a$), then we suppose each node that arrives, averages with node 0 and then leaves the system, so

$$w_i = \begin{cases} (\frac{1}{2})^{k+1-i} & \text{for } i = 1, \ldots, k, \\ (\frac{1}{2})^k & \text{for } i = 0. \end{cases} \tag{4}$$

The results are interesting as they show that if the system's departure rate is fast, then the weights for old values decrease exponentially, but if the departure is slow then the weights for old values decrease linearly in $k$. For future work, we would like to characterize the decrease in the average weight of the history as function of the performance parameter of the queue (as function of $\rho = \lambda_a/\lambda_d$).

### 3.2. Directed Tree

For this topology, we assume that there are no departures from the system. Each node arrives, connects uniformly at random to one node (we call it its parent) in the network. Let $L$ be the distance from a newly connected node to the root (node 0). We suppose that each node $i$ arrives at level $L$, connects to its parent $j$, and then averages as follows :

$$Y_i(t) = \frac{1}{L} X_i + \frac{L-1}{L} Y_j(t) \text{ for } t > A_i.$$

Notice that $Y_i(t) = \frac{1}{L} \sum_{s \in P} X_s$ is a constant for $t > A_i$ where $P$ is the set of nodes on the path from the root to $i$. Since $L$ converges to $\log n$ in probability as $n \to \infty$ [3], we conclude that if $X_1, X_2, \ldots$ are i.i.d random variables of mean $\mu$, then the value $Y_i(t)$ for $t > A_i$ converges in probability to $\mu$ as $i \to \infty$.

## References

1. Olfati-Saber, R. ; Murray, R.M., "Consensus problems in networks of agents with switching topology and time-delays", *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520-1533, Sept. 2004.
2. Tahbaz-Salehi, A. ; Jadbabaie, A., "A Necessary and Sufficient Condition for Consensus Over Random Networks" , *IEEE Trans. on Automatic Control*, vol. 53, no. 3, pp. 791-795, April 2008.
3. L. Devroye, "Applications of the theory of records in the study of random trees", *Acta Informatica*, vol. 26, no. 1-2, pp. 123-130, 1988.