



Courtesy Laurent Chusseau

Évaluation T0+24 MARMOTE

MARkovian MOdeling Tools and
Environments

<https://wiki.inria.fr/MARMOTE/Accueil>

Contenu

Qui sommes-nous ?

Qu'avons-nous réalisé ?

- Résultats théoriques

 - Échantillonnage

 - xxx

 - Théorie

- Résultats des applications

- Résultats logiciels

- Publications

Comment avons-nous travaillé ?

Qu'allons-nous faire ?

- Intensifier l'ingénierie logicielle

- Développer les applications

- Communiquer

1

Présentation

Qui sommes-nous ?

Partenaires Institutionnels et leur « partner heads » :

- ▶ INRIA/MAESTRO : Alain Jean-Marie, coordinateur
- ▶ INRIA/DYOGENE (anc. TREC) : Ana Bušić
- ▶ INRIA/MESCAL : Bruno Gaujal
- ▶ Univ. Versailles-St-Quentin/PriSM : Jean-Michel Fourneau, co-coordonateur
- ▶ Telecom SudParis/SAMOVAR : Hind Castel-Taleb
- ▶ Univ. Paris-Est-Créteil/LACL : Nihal Pekergin
- ▶ Univ. Paris 6/LIP6 : Emmanuel Hyon

Spécialistes de la *modélisation Markovienne* et de ses applications en télécoms, systèmes distribués, fiabilité...

Issus de la même école : vocabulaire commun, travaux communs antérieurs...

Que voulons-nous faire dans MARMOTE

Objectifs du projet :

- ▶ Faire avancer les méthodes de résolution pour chaînes de Markov
 - ▶ échantillonnage exact de la distribution stationnaire (« simulation parfaite »)
 - ▶ simulation Monte Carlo parallèle
 - ▶ solutions à forme close et algorithmes de résolution de flots
 - ▶ simulation numérique : calcul de bornes, méthodes de transformées, ...
- ▶ Faire avancer les outils logiciels
- ▶ Tester le tout dans plusieurs cas d'application

2

Qu'avons-nous réalisé ?

Résultats : échantillonnage exact (WP1)

Échantillonnage exact : principe

Principe : simuler un ensemble de trajectoires jusqu'à ce qu'elles **couplent**. La distribution de l'état à cet instant-là est la distribution stationnaire.

Échantillonnage exact : défis

Deux paramètres influent sur la complexité algorithmique :

- ▶ contrôler le **nombre de trajectoires** à simuler
- ▶ contrôler le **temps de couplage** des trajectoires

Échantillonnage exact (suite)

Réalisations :

- ▶ un algorithme pour l'échantillonnage dans les réseaux de Jackson
- ▶ un calcul de borne sur le temps de couplage pour les processus de naissance et de mort multidimensionnels avec transitions interdites
- ▶ un algorithme de simulation temps-parallèle d'une trajectoire (élément de base de l'échantillonnage exact) : « Catch me if you can ».

Résultats xxx

xxx : principe et défis

bla bla pédagogique

Réalisations

bla bla réalisations

Théorie (WP4)

Réalisations :

- ▶ Solutions analytiques à forme close pour des réseaux de files d'attente avec signaux et algorithmes adaptés pour les calculs des distributions stationnaires
- ▶ Calcul de *bornes stochastiques* sur des distributions (meilleure borne dans le sens \leq_{st}) et applications à l'analyse d'une file soumis à un flux dont on a mesuré une trace.
- ▶ Simulation Parallèle pour l'analyse de la fiabilité modélisé par des Dynamic Fault Trees (pas dans la proposition initiale)
- ▶ Simulation Parallèle pour certains modèles spécifiées par Stochastic Process Algebra par l'approche dite du Parallel Prefix, conditions suffisantes d'application et méthodes de simulation.

Applications

Résultats :

- WP 5.1 (Modèle Cloud) Méthodologie combinant traces, mesures et bornes pour ...
- WP 5.2 Simulation parallèle pour une cellule
- WP 5.3 ...
- WP 5.4 Algorithmes d'échantillonnage de la distribution stationnaire d'une bande électronique ; méthode de décomposition temporelle

Autres contributions

Des contributions dans des thèmes connexes à MARMOTE

- ▶ Analyse des méthodes de Restart par des résultats à forme produit

Logiciel : Contexte

Logiciel : principe

Réaliser un environnement logiciel **ouvert** reprenant les logiciels existants, capable d'accueillir les contributions de différents groupes de recherche, pour qu'il puisse se développer de façon **contributive**

Logiciel : défis

- ▶ Diversité de logiciels : monolithique multifonctionnel/éclaté, langages de programmation divers, ...
- ▶ Diversité d'utilisateurs : développeur de méthodes (interface « bas niveau »), développeur d'applications (interface « haut niveau »)

Logiciel : état de l'art

Au démarrage de MARMOTE, plusieurs logiciels d'analyse Markovienne développés par les membres :

- ▶ Psi, Psi3 (Inria/MESCAL)
- ▶ Xborne (UVSQ/PRiSM)
- ▶ ERS (Inria/MAESTRO)

Plan de travail en parallèle :

- ▶ Développer un logiciel-chapeau qui reprendra toutes ces fonctionnalités
 - ▶ d'abord en **encapsulant** les applications existantes
 - ▶ ensuite en **incorporant** leur code
- ▶ Poursuivre le développement indépendant de Psi3 et Xborne

Logiciel : Xborne

Réalisations :

- ▶ Logiciel d'analyse d'une file soumis à un flux mesuré par une trace (bientôt disponible, sous matlab)
- ▶ Logiciel de résolution de réseaux de files d'attente modélisant des restart (en liaison avec Free Univ. Berlin)

Logiciel : Psi3

- ▶ Ajout dans Psi3 d'un noyau de simulation parallèle « multithreadée » .
- ▶ Extension de la bibliothèque de modèles dans Psi3 : Réseaux de Jackson avec buffers infinis, marches aléatoires dans un réseau avec transitions interdites.

Logiciel : MARMOTE

Réalisations :

- ▶ Définition de l'architecture du logiciel MARMOTE
- ▶ Études de cas autour des *scientific workflow systems*
- ▶ Implémentation des objets de base à partir du logiciel ERS
- ▶ Interfaçage avec le logiciel Xborne (UVSQ/PRIISM)

Logiciel : MARMOTE (suite)

Idée générale :

- ▶ Hiérarchie de modèles Markoviens
On-Off **is a** Homogeneous Birth-Death **is a** General Birth-Death **is a** Continuous-Time Markov Chain
- ▶ Familles de métriques
Distribution : stationnaire, transitoire, de temps d'atteinte, etc.
- ▶ Pour chaque modèle et chaque métrique, des méthodes de résolution (calcul exact, approché, bornes) adaptées.
⇒ les rendre accessibles par une API

Logiciel MARMOTE (fin) : exemple de code

Comparaison de calculs de la distribution stationnaire

```
// specific methods for F81a
Felsenstein81* c1 = new Felsenstein81(...);
Distribution* d1 = c1->stationaryDistribution();
Distribution* d2 = c1->simulateChain(...)->getDistribution();
// generic methods for MCs
MarkovChain* c2 = static_cast<MarkovChain*>(c1);
Distribution* d3 = c2->stationaryDistribution_GaussSeidel();
Distribution* d4 = c2->stationaryDistribution_PowerMethod();
Distribution* d5 = c2->stationaryDistribution_Xborne_LowBound();
Distribution* d6 = c2->replicateSamples_Psi3(...);
Distribution* d7 = c2->simulateChain(...)->getDistribution();
Distribution* d8 = c2->simulateChain2(...)->getDistribution();
// comparison
cout << "Distance L1(d1,d2) = " << d1->distanceL1(d2) << endl;
```

a. Felsenstein 81 est un modèle utilisé en bio-informatique

Publis

Quantitativement :

- ▶ 2 articles dans des revues internationales (*Queueing Systems*, *Stochastic Models*)
- ▶ ~15 dans des conférences internationales avec comité de lecture dont PERFORMANCE, MASCOTS, QEST, EPEW, VALUETOOLS

Qualitativement :

- ▶ L'article de F. Ait Salaht, H. Castel, J.-M. Fourneau et N. Pekergin, «A bounding histogram approach for network performance analysis», a obtenu un des trois best paper award de la conférence IEEE HPCC 2013, Chine.

4

Comment l'avons-nous
fait ?

Comment

Nous avons recruté :

- ▶ Participation de Farah Ait Salaht, doctorante DIGITEO
- ▶ Participation de Pierre Coucheney (MCF, UVSQ/PRiSM)
- ▶ Recrutement de Christelle Rovetta, Doctorante (Inria/DYOGENE)
- ▶ Recrutement d'Issam Rabhi, Ingénieur WP3 (Inria/MAESTRO)
- ▶ Recrutement de Farah Ait Salaht, Post-Doctorante (TSP/SAMOVAR)
- ▶ Recrutement de Marion Dalle et Arnaud Panayotis, stagiaires (Inria/MESCAL) et deux autres stagiaires hors financement MARMOTE

⇒ 16% des ETP à T0+24.

Comment (suite)

Nous avons organisé des groupes de travail :

- ▶ Groupes de travail existants : xxx, yyy, zzz
- ▶ WP2/WP 5.x (réseaux, biologie, stochastic model checking) :
Groupe de travail UPEC/LACL, UVSQ/PRiSM,
TSP/SAMOVAR
- ▶ WP1/WP4 : Groupe de lecture hebdomadaire
Inria/DYOGENE ; LIP6, TSP/SAMOVAR
- ▶ WP4/WP 5.5 : groupe de travail sur le Contrôle Stochastique
avec financement du GDR RO (Recherche Opérationnelle) ;
Paris, Inria/MAESTRO, Inria/MESCAL
- ▶ WP 5.4 : Mise en place d'un groupe de travail sur le modèle
de laser ; Inria/MAESTRO
- ▶ WP 5.5 : Mise en place d'un groupe de travail sur des
modèles de « développement durable » en Économie ; LIP6,
Inria/MAESTRO, EconomiX

et des réunions plénières : janvier 2013, Octobre 2014, Juillet 2014.

Comment (fin)

Nous avons réajusté le programme de travail

- ▶ WP 5.2 : Suite au départ des contacts biologie, changement de cible vers la modélisation de réactions (bio)chimiques

4

Qu'allons-nous faire ?

Planning futur : logiciel

Accélérer sur le développement logiciel :

- ▶ Recrutement Ingénieur développement en cours (PSI3, Inria/MESCAL)
- ▶ Recrutement Post-Doctorant algorithmicien (UVSQ/PRiSM)
- ▶ Recrutement Ingénieur parallélisme en cours (UVSQ/PRiSM)
- ▶ Replanification de l'ingénierie :
 - ▶ concaténer les deux ingénieurs WP3 INRIA/MAESTRO,
 - ▶ mettre en place un GdT avec les ingénieurs INRIA/MESCAL et UVSQ/PRiSM
 - ▶ sortir une version 0. β dans les premières semaines de 2015.

Planning futur : applications

Développer les applications :

- ▶ Développer l'application WP5.4 (Laser) dans le logiciel MARMOTE
- ▶ Développer les méthodes du contrôle Markovien dans le logiciel MARMOTE
- ▶ Démarrer l'application « réseaux » (WP 5.1) : le post-doc a été recruté au 1/11/14
- ▶ Démarrer xxx
- ▶ Démarrer yyy
- ▶ Nouveau « use case » : modèle de gestion de l'énergie en télécoms

Planning futur : communication

Commencer à diffuser le projet :

- ▶ Communiquer dans les conférences nationales : ROADEF 2015, ...
Cette année en juin : communication bien reçue à l'Atelier en Évaluation de Performances, Sophia-Antipolis
- ▶ Communiquer à l'international : soumettre à VALUETOOLS, QEST, EPEW, SIGMETRICS, PERFORMANCE, MASCOTS, ...