



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

 - Simulation parallèle

 - 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 (ex : *réseaux de files d'attente*), 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, issues d'états initiaux différents, jusqu'à ce qu'elles **couplent**. La distribution de l'état à cet instant-là est la distribution stationnaire.

Technique du « couplage depuis le passé » de Propp-Wilson.

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

Obtenues par C. Rovetta pendant sa première année de thèse

- ▶ un algorithme pour l'échantillonnage exact dans les réseaux de files d'attente *fermés*
annoncé comme **objectif à risque plus élevé** car l'espace des états n'a pas de structure de type « treillis »
- ▶ le toolbox Clones (CLOsed queueing Networks Exact Sampling)

et déjà publiées dans deux conférences avec comité de lecture

Échantillonnage exact / simulation parallèle (WP2)

Réalisations

- ▶ un algorithme d'échantillonnage exact pour les processus de naissance et de mort multidimensionnels avec transitions interdites (« treillis avec des trous »)
- ▶ un calcul de borne sur le temps de couplage pour cet algorithme
- ▶ un algorithme de simulation temps-parallèle d'une trajectoire (élément de base de l'échantillonnage exact) : « Catch me if you can ».
- ▶ Identification d'un goulot d'étranglement dans le simulateur OpenMP qui pénalise la simulation parallèle

Simulation parallèle (fin)

Réalisations

- ▶ 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
- ▶ Simulation Parallèle pour l'analyse de la fiabilité modélisé par des Dynamic Fault Trees (pas dans la proposition initiale)

Théorie (WP4)

Contexte 1 : bornes stochastiques

Les distributions de probabilité peuvent être comparées entre elles. Ces relations sont les **ordres stochastiques**.

Il est parfois possible d'étudier un système aléatoire en en construisant un autre dont les distributions sont plus grandes/petites (dans le sens de cet ordre stochastique) que celles du système initial, et plus faciles à calculer.

Réalisations

- ▶ Calcul de bornes pour des chaînes de Markov quelconques à l'aide de matrices de **petit rang algébrique**

Théorie (suite)

Contexte 2 : solutions à forme produit

Certaines classes de modèles de réseaux aléatoires ont une distribution stationnaire facile à calculer car à « **forme produit** ». Par contre, certaines caractéristiques (constantes de normalisation, paramètres intervenant dans les distributions marginales, ...) restent à calculer de façon efficace

Réalisations

- ▶ Solutions analytiques à forme close pour des réseaux de files d'attente avec signaux
- ▶ Algorithmes adaptés pour les calculs des distributions stationnaires

Applications : WP5.1 (modèle de cloud)

Modèle cloud : principe et défis

La description du **trafic** (arrivées de jobs, durées, ...) est obtenue sous forme de traces. Or :

- ▶ Faire correspondre statistiquement ces traces à des distributions introduit des erreurs
- ▶ Utiliser directement les histogrammes issus des traces conduit à une explosion de la taille des chaînes de Markov

Résultats

Algorithmes d'agrégation d'histogrammes,

- ▶ fournissant les meilleures *bornes stochastiques* possibles
- ▶ appliquées à des histogrammes de trafic considéré comme stationnaire ou stationnaire par morceaux

Analyse d'une file d'attente simple et d'un réseau de files.

Applications : autres

Résultats pour les autres applications du WP5

- ▶ WP5.2 : parallélisation de la simulation d'une cellule biologique
- ▶ WP5.4 (modèle de laser) : algorithmes d'échantillonnage de la distribution stationnaire d'une bande électronique ; méthode de décomposition temporelle
- ▶ WP5.5 (modèles en Économie) : preuve de la convergence d'algorithmes « best response » et « smoothed best response »

Contributions dans des thèmes connexes à MARMOTE

- ▶ Analyse des modèles de files d'attente avec signaux de type *Restart*
- ▶ Contrôle stochastique appliqué aux réseaux d'énergie (« smart grids »)

Logiciel (WP3) : Contexte

Logiciel : constat

Pas de bibliothèque logicielle permettant d'accéder facilement aux résultats des modèles classiques aussi bien qu'aux algorithmes avancés de résolution.

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 (« bas niveau »), développeur d'applications (« haut niveau »)

Logiciel : état de l'art

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

- ▶ Psi, Psi3 (Inria/MESCAL)
échantillonnage exact, ...
- ▶ Xborne (UVSQ/PRiSM)
construction et analyse de chaînes de Markov
- ▶ ERS (Inria/MAESTRO)
construction et analyse de chaînes de Markov

Logiciel : 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 : Psi3

Réalisations

- ▶ 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/PRiSM)

Logiciel : MARMOTE (suite)

Organisation générale

- ▶ Hiérarchie de modèles Markoviens

On-Off **is a** Homogeneous Birth-Death **is a** General Birth-Death **is a** Quasi-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

Logiciel : autres modules

Réalisations

Nouveaux modules

- ▶ 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)
- ▶ Toolbox Clones (voir supra)

destinés à être intégrés à MARMOTE

Publications

Quantitativement :

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

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 UVSQ puis recrutement comme Post-Doctorante (TSP/SAMOVAR)
- ▶ Participation de Pierre Coucheney (MCF, UVSQ/PRiSM ; T0+12)
- ▶ Recrutement de Christelle Rovetta, Doctorante (Inria/DYOGENE ; T0+11)
- ▶ Recrutement d'Issam Rabhi, Ingénieur WP3 (Inria/MAESTRO ; T0+11)
- ▶ Recrutement de Marion Dalle et Arnaud Panayotis, stagiaires (Inria/MESCAL) et deux autres stagiaires hors MARMOTE

⇒ 16% des ETP à T0+24

⇒ 30% supplémentaires déjà engagés sur 2015

Comment (suite)

Nous avons organisé des groupes de travail :

- ▶ Groupes de travail pré-existants : contrôle optimal stochastique (LIP6, Inria/MAESTRO), ...
- ▶ WP2/WP5.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/WP5.5 : Groupe de travail sur le Contrôle Stochastique avec financement du GDR RO (Recherche Opérationnelle) ; Paris, Inria/MAESTRO, Inria/MESCAL

Comment (suite)

Groupes de travail, suite :

- ▶ WP5.4 : Mise en place d'un groupe de travail sur le modèle de laser ; Inria/MAESTRO
- ▶ WP5.5 : Mise en place d'un groupe de travail sur des modèles de « développement durable » en Économie ; LIP6, Inria/MAESTRO, EconomiX

Réunions plénières : janvier 2013, Octobre 2014, Juillet 2014.

Comment (fin)

Nous avons réajusté le programme de travail

- ▶ WP2/WP5.3 : étude des Dynamic Fault Trees
- ▶ WP5.2 : Suite au départ des contacts biologie, changement de cible vers la modélisation de réactions (bio)chimiques

5

Qu'allons-nous faire ?

Planning futur : accélérer le développement logiciel

Recrutements :

- ▶ Ingénieur développement acté (PSI3, Inria/MESCAL)
- ▶ Post-Doctorant algorithmicien en cours (UVSQ/PRiSM)
- ▶ Ingénieur parallélisme en cours (UVSQ/PRiSM)
- ▶ Post-Doctorant/Ingénieur « stochastic model checking » en cours (UPEC/LACL)

Planning futur : logiciel

Replanification de l'ingénierie :

- ▶ concaténer les deux ingénieurs WP3 INRIA/MAESTRO,
- ▶ mettre en place un GdT entre les ingénieurs de développement
- ▶ donner des séances de formation pour les contributeurs et utilisateurs de MARMOTE
- ▶ sortir une version $0.\beta$ dans les premières semaines de 2015
- ▶ déposer le logiciel
- ▶ développer les connexions avec le workflow management : plan d'expérience, visualisation de données, ...

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
- ▶ Poursuivre l'application « réseaux » (WP5.1) : développer des bornes pour le dimensionnement d'un cloud, et valider par des tests sur des plateformes réelles
 - ⇒ le post-doc a été recruté au 1/11/14
- ▶ Réorienter WP5.2 vers la modélisation de réactions (bio)chimiques :
 - ⇒ recrutement d'un doctorant financé par le Labex CHARMMMAT

Planning futur : applications (suite)

- ▶ Développer de nouveaux « use case » à la frontière de MARMOTE :
 - ▶ modèle de gestion de l'énergie dans le cloud
 - ▶ modèles de gestion de l'énergie en télécoms : utilisation de Xborne et méthodes numériques pour les « Quasi-Birth-Death Processes »
 - ▶ modèle de contrôle de consommation d'électricité dans une *smart grid* : méthodes numériques pour Processus de Décision Markoviens
- ⇒ une façon d'intéresser d'autres collègues à MARMOTE

Planning futur : communication

Commencer à diffuser le projet :

- ▶ Cette année en juin : communication bien reçue à l'Atelier en Évaluation de Performances, Sophia-Antipolis

⇒ à poursuivre

Communiquer dans les conférences nationales : ex : ROADEF, Algotel, MOSIM, MSR, Journées des GDR...

- ▶ Communiquer à l'international (théorie et applications) : soumettre à SIGMETRICS, PERFORMANCE, MASCOTS, QUEST, EPEW, CDC, VALUETOOLS, ...
- ▶ Publier en revue (temps de cycle plus long)