Action Concertée Incitative

SÉCURITÉ INFORMATIQUE

Rapport de mi-parcours

I - FICHE D'IDENTITÉ DU PROJET

Nom du Projet : *(maximum 20 caractères)*

SURE-PATHS

Titre du Projet : *(maximum 3 lignes)*

Evaluation stochastique de la fiabilité, de la performabilité et de la sureté de fonctionnement de systèmes, "model checking" probabiliste

Type du Projet :

Projet de recherche	Projet de recherche multi-thématiques	Projet de recherche avec infrastructure	Autre
XXX			

Durée du projet : 3 ans

Description courte du Projet : *(une demi-page maximum)*

L'etude de la fiabilité de systèmes ou de la performabilité (évaluation de performances en présence de fautes réduisant les capacités de service) se heurte tout comme l'évaluation de performances à l'explosion combinatoire de l'espace des états mais aussi à l'évaluation de probabilités d'événements très faibles. L'analyse exhaustive est donc impossible et la simulation standard peu fiable. Le model checking probabiliste reposant sur des calculs de probabilites stationnaires ou transitoires ne dispose donc pas d'outils algorithmiques aussi performants que dans le cadre déterministe. Nos équipes ont proposé de nouvelles techniques qui ont su faire leur preuve dans un contexte assez semblable lié à l'explosion combinatoire des états, le calcul numeriques de probabilités très faibles et l'obtension de garantie : la décomposition modulaire et la représentation tensorielle compacte de l'espace des états et des transitions, les couplages de trajectoires pour déterminer des systèmes plus simples et fournissant une borne ou une garantie, les algorithmes de calcul de bornes, la convergence rapides des méthodes de Monte-Carlo, la simulation parfaite par couplage dans le passé garantissant une mesure exacte, l'emploi d'algorithmes adaptés à ces représentations tensorielles et aux ressources de type GRID, le lien avec des langages ou des formalismes de spécifications ou le model checking probabiliste. Le projet permettra d'étendre ces methodes et prototypes aux problématiques de la sûreté de fonctionnement tant d'un point de vue qualitatif que quantitatif. En effet pour un nombre grandissant d'applications (temps-réel ou embarquées, multimedia), la preuve formelle de correction ou d'absence de fautes ne peut être dissociée de l'évaluation quantitative du système.

Coordinateur du projet :

Nom	Prénom	Laboratoire (sigle éventuel et nom complet)
Plateau	Brigitte	ID, Informatique et Distribution

Action Concertée Incitative

SÉCURITÉ INFORMATIQUE

Rapport de mi-parcours

A2- Équipes ou laboratoires partenaires du Projet [1] :

Identification de l'équipe ou du laboratoire

Équipe ou Laboratoire	ID, Equipe Evaluation de Performances
Adresse	51, Avenue Jean Kuntzmann 38330 Montbonnot

Responsable du projet au sein de l'équipe ou du laboratoire

M. ou Mme. Prénom Nom	Brigitte Plateau
Fonction	Professeur
Téléphone	04 76 61 20 88
Fax	04 76 61 20 99
Mél	Brigitte.Plateau@imag.fr

Autres membres de l'équipe participant au projet

Nom	Prénom	Poste statutaire	% du temps de recherche consacré au projet
Vincent	Jean Marc	MDC UJF	20%
Sulaiman	Eiad	Doctorant	50%
Sbeity	Ihab	Doctorant (bourse demandée)	100%

Identification de l'équipe ou du laboratoire

Équipe ou Laboratoire	projet ARMOR, IRISA-INRIA
Adresse	Campus universitaire de Beaulieu 35042 Rennes cedex

Responsable du projet au sein de l'équipe ou du laboratoire

M. ou Mme. Prénom Nom	Bruno Tuffin
Fonction	Chargé de recherche INRIA
Téléphone	02 99 84 74 94
Fax	02 99 84 25 29
Mél	Bruno.Tuffin@irisa.fr

Autres membres de l'équipe participant au projet

Nom	Prénom	Poste statutaire	% du temps de recherche consacré au projet
Rubino	Gerardo	DR Inria	20%
Sericola	Bruno	CR Inria	20%
El Khadiri	Mohammed	MCF St-Nazaire	50%

Identification de l'équipe ou du laboratoire

[1] Une fiche doit être remplie pour chaque laboratoire ou équipe partenaire.

Équipe ou Laboratoire	PRiSM, Equipe EPRI
Adresse	Université de Versailles Saint Quentin en Yvelines 45, Avenue des Etats-Unis 78035 Versailles Cedex

Responsable du projet au sein de l'équipe ou du laboratoire

M. ou Mme. Prénom Nom	Jean-Michel Fourneau
Fonction	Professeur
Téléphone	01 39 25 40 77
Fax	01 39 25 40 57
Mél	jmf@prism.uvsq.fr

Autres membres de l'équipe participant au projet

Nom	Prénom	Poste statutaire	% du temps de recherche consacré au projet
Pekergin	Nihal	MDC Univ. Paris I	50%
Quessette	Franck	MDC UVSQ	50%
Kloul	Leila	MDC UVSQ	50%
Barth	Dominique	Prof UVSQ	20%
Lecoz	Matthieu	Doctorant MEN	100%
Mokhtari	Amjed	Doctorant MEN	50%

Action Concertée Incitative

SÉCURITÉ INFORMATIQUE

Rapport de mi-parcours

Objectifs et contexte :

En évaluation quantitative de la sécurité et de la performabilité, les verrous sont de deux ordres : comment spécifier et obtenir les états et les transitions d'un système complexe dans un formalisme se prêtant aux calculs et ensuite comment effectivement réaliser ces calculs. Le premier problème a été en partie réglé par les représentations tensorielles initiées par nos équipes via le formalisme des RAS (Réseaux d'Automates Stochastiques) et qui ont depuis été généralisées à toutes les approches quantitatives modulaires (superposition de réseaux de Petri, Algèbre de Processus, chaînes de Markov en interaction) par diverses équipes Françaises, Européennes ou Américaines (Haddad en France, Donatelli, Buchholz, et Kemper en Europe, Ciardo et Stewart aux USA). Par contre, les algorithmes de calcul ne pourront jamais résoudre les problèmes de ces tailles. C'est pourquoi nous proposons diverses méthodes de biais (en simulation) ou de bornes (pour les calculs numériques) permettant de travailler sur des modèles stochastiques plus simples (plus petits, ou plus réguliers, ou avec une plus grande fréquence pour une transition remarquable). Ces algorithmes permettent de donner des garanties plutôt que des valeurs exactes et permettent donc un "model checking" stochastique et la vérification de contraintes quantitatives de sureté. Les approches tensorielles ont déjà été appliquées dans le contexte du "model checking" stochastique (APPN par Buchholtz) et les approches de bornes sur les probabilités sont de plus en plus fréquentes dans le contexte des systèmes à grand espace d'états (travaux de Buchholtz sur l'approche de Courtois et les produits tensoriels à Performance 2002, travaux de Donatelli, Moreaux et Haddad sur une approche mixte Courtois et trajectoires, tutoriel de Fourneau à Performance 2002 sur les approches algorithmiques des bornes trajectorielles). En simulation l'échantillonnage préférentiel (*importance sampling*) et la ramification de trajectoires (*importance splitting*) ont été étudiés par les équipes ayant les contributions méthodologiques les plus importantes du domaine (voir les travaux de Nakayama, Nicola, Heidelberger, Shahabuddin, Glasserman et Villen). Les équipes sont complémentaires aussi bien pour les approches numériques (analyse Markovienne de la sureté dans ARMOR, approche tensorielle et bornes numériques dans ID et PRiSM) et de simulation (Monte Carlo dans ARMOR, couplage dans le passé et parallélisme dans PRiSM et ID). Cette collaboration s'appuie aussi sur des colloborations internationales dans les laboratoires avec des spécialistes reconnus du domaine : ID avec Stewart, Ciardo et Donatelli (via CNRS-NSF), et PRiSM avec Stewart (via CNRS-NSF), Hillston, et Dayar pour la modélisation Markovienne, ARMOR avec H. Cancela et K. Trivedi pour la simulation.

Action Concertée Incitative

SÉCURITÉ INFORMATIQUE

Rapport de mi-parcours

Résultats et travaux en cours

1 Loi phase type dans les réseaux d'automates stochastiques, ID

Les Réseaux d'Automates Stochastiques (SAN en anglais) est un formalisme de haut niveau qui permet la modélisation de chaînes de Markov très grandes et très complexes de façon compacte et structurée. Avec le temps continu, le délai de franchissement des transitions (durée de tir des transitions) dans les SANs suit une distibution exponentielle, ce qui permet de modéliser un ensemble important de systèmes ayant des activités concurrentes.
Cependant, l'emploi de distributions plus générales reste toujours souhaitable pour la modélisation de nombreux phénomènes réels, notamment dans le contexte de la fiabilité. Il s'agit, notament, de la famille de distributions phase-type qui peuvent être décrites par un "graphe de services exponentiels". Elles sont constituées d'une succession d'étapes où la durée de service de l'étape numéro i suit une loi exponentielle de moyenne μ_i. Comme cas particulier de cette famille de distributions, nous citons la distribution d'Erlang, hyper-exponentielle, et Cox qui généralise les deux premiers types de distribution. L'utilisation de la loi Phase-Type a un grand intérêt, car elle permet de décrire de durées réelles et plus complexes que celles pouvant être décrites par la loi exponentielle : par exemple, le processus de fabrication d'un produit peut passer par plusieurs étapes de construction et de vérification, avant que le produit ne soit validé. Les taux de ces différentes étapes suivent des lois exponentielles.
D'autre part, les lois Phase-Type ne sont pas des lois "sans mémoire". Lorsqu'une autre transition est franchie, nous devons déterminer quoi faire avec le travail déjà réalisé par une transition Phase-Type en cours. Cela peut se faire en distinguant les deux politiques de mémoire d'une transition Phase-Type :

Recommencer (*Restart*) : le travail est interrompu, et le temps déjà passé est perdu. Le travail est repris lorsque la transition sera à nouveau franchissable (ce qui peut être immédiatement). Les lois exponentielles "sans mémoire" ont, de part la propriété de la loi, ce type de politique. Pour les lois Phase-Type, ce comportement doit être pris en compte dans la modélisation.

Reprendre (*Resume*) : au contraire de la politique **Recommencer**, lors du franchissement d'une autre transition, on conserve le temps déjà écoulé. La prochaine fois que la transition sera franchissable (qui peut être une continuation), c'est le temps résiduel qui sera utilisé comme délai de franchissement éventuel.
Notre approche consiste à introduire les transitions phase-type directement au modèle SAN, en décrivant aussi leur sémantique dans le modèle lui-même (qui sera appelé modèle PH-SAN). Lors du calcul, ce modèle PH-SAN est transformé en un modèle SAN standard (avec ses transitions locales, fonctionnelles et synchronisantes), en prenant en considération les différentes politiques d'éxécution de transitions Phase-type. Ensuite, les matrices élémentaires du descripteur (matrice de transition de la chaîne de Markov sous-jacente) sont générées à partir de ce nouveau modèle.

2 Bornes stochastiques, Lumpabilité et SAN, ID-PRISM

Malgré les nombreux travaux sur la résolution numérique des chaînes de Markov, (voir par exemple Stewart : Introduction to the Numerical Solution of Markov Chains, ou les actes de la conférence Numerical Solution of Markov chain publiés dans Linear Algebra and its Applications, V 386 (2004)) ce problème reste toujours très difficile dans le cas d'un espace d'états relativement grand.

La théorie des bornes stochastiques (voir Stoyan : Comparison Methods for Queues and Other Stochastic Models) peut être utilisée pour construire une nouvelle chaîne ayant une structure plus facile à résoudre et garantissant une borne inférieure ou supérieure sur les indices de performance.

L'ordre le plus souvent utilisé est l'ordre stochastique fort (noté "st"), permettant d'obtenir des bornes pour toutes les fonctions de récompense croissantes. L'autre avantage de l'ordre "st" est la possibilité de trouver, pour chaque matrice de transition, une borne optimale (voir les travaux de J-M. Vincent). Malheureusement, cet algorithme, assurant l'optimalité de la borne, ne facilite pas la résolution numérique

dans l'état actuel de nos connaissances. Bien au contraire, la nouvelle chaîne peut s'avérer plus complexe à résoudre.

Aussi avons nous développé deux familles d'algorithmes qui permettent de calculer une borne plus simplement. La première famille d'algorithme emploie la lumpabilité alors que la seconde famille repose sur des patrons de matrice permettant une résolution adaptée (plutôt que les méthodes générales).

Toutes ces méthodes reposent sur l'idée suivante : les propriétés de monotonie et de comparabilité, conditions nécéssaires de la comparaison des trajectoires des chaines de Markov, imposent des inégalités sur les lignes et colonnes des matrices.

L'algorithme de Vincent remplace ces inégalités par des égalités. Les autres algorithmes respectent les inégalités mais ajoutent des contraintes supplémentaires d'ordre structurels (pour les patrons) ou numériques (pour la lumpabilité).

La lumpabilité ordinaire ou agrégation forte est associé à une partition des états. On dit que la chaîne est agrégeable au sens fort si et seulement si le processus obtenu en agrégeant les états selon la partition (tous les états d'un même ensemble sont regroupés en un seul point) reste Markovien. La condition d'agrégation impose que les blocs décrivant la transition d'un ensemble i vers un ensemble j sont de somme en ligne constante.

On a précédemment démontré que les contraintes de monotonie et de comparaison sont compatibles avec la lumpabilité ordinaire. En ajoutant un shéma garantissant l'irréductibilité de la chaîne, on obtient l'algorithme LIMSUB.

Cet algorithme suppose que la matrice de la chaîne soit stockée sur disque dans un format adapté (en colonne, en débutant par la dernière colonne et les états étant regroupés par appartenance aux ensembles de la partition). Ce stockage n'est bien sûr pas celui de la génération des états lors de la construction d'un modèle, ce qui oblige à de fastidieuses renumérotations des états, parfois plus longues que l'algorithme de borne. Il est donc naturel de coupler l'algorithme LIMSUB aux Réseaux d'automates stochastiques, ce que nous avons fait dans cette première année du projet. Le formalisme des RAS a plusieurs avantages que nous allons employer :

1. Il permet de stocker facilement en mémoire la chaîne
2. il permet d'accéder facilement aux sommets dans un ordre quelconque
3. le coût d'accès est encore moindre si la génération est dans l'ordre lexicographique
4. Il permet d'uniformiser facilement la chaîne (c'est à dire de travailler sur une chaine en temps discret ayant la même solution que le problème en temps continu).

Les algorithmes classiques de résolution sur les RAS tiennent compte des propriétés 1,3 et 4. Nous allons plutôt ici utiliser les propriétés 1,2 et 4 de manière à générer les états dans l'ordre de la partition associée à la lumpabilité. La propriété d'uniformisation est indispensable puisque les RAS utilisés modélisent des systèmes en temp continu alors que les algorithmes de bornes s'appliquent aux chaînes en temps discret. On peut donc générer une borne en ayant très peu d'objets en mémoire : 2 vecteurs de la taille de l'espace des états, quelques vecteurs de la taille de la partition et la description su RAS par un produit tensoriel. L'algorithme est implanté dans PEPS.

3 Les bornes stochastiques "st" selon un patron, PRISM

Nous avons déjà dit que l'algorithme de Vincent utilise des égalités dans les conditions de la comparaison et de la monotonie "st". En permettant d'utiliser les valeurs plus grandes dans le cas de borne supérieure que celles imposées par ces contraintes, on peut créer et/ou supprimer les transitions dans la matrice bornante imposant ainsi la structure de la borne. L'idée est de trouver une structure de matrice permettant une résolution numérique facile et de faire une preuve générale de borne indépendante du patron. Plus grand au sens "st" signifie déplacer la distribution de probabilité vers des états plus grands ; ce qui signifie simplement sur la matrice de transition déplacer une probabilités vers les états à droite.

Nous avons proposé un formalisme de patron matriciel décrivant des conditions supplémentaires, liées à la structure de la borne, pour chaque élément de la matrice. Ce patron matriciel est une matrice dont les éléments appartiennent à un alphabet où chaque lettre correspond à un type de condition différent.

Le patron booléen, par exemple, est un patron avec les éléments 0 ou 1 avec la sémantique suivante : si l'élément en la position (i, j) dans le patron a la valeur 1, dans la matrice bornante à cette position on doit avoir une valeur strictement positive (c'est à dire une transition). Si par contre l'élément correspondant dans le patron vaut 0, dans la matrice bornante on doit avoir la valeur 0 (pas de transition). Ce type de patron impose la structure exacte du graphe de transitions de la borne ce qui pour la plupart des applications peut être une condition trop forte. Pour cette raison on a introduit une nouvelle lettre dans l'alphabet, signifiant l'absence de conditions supplémentaires liées à la structure. Il est possible d'avoir les conditions structurelles dépendant de la matrice initiale. Par exemple, la condition suivante : “si à la

position (i, j) dans la matrice initiale il y a un élément non-nul, alors dans la matrice bornante l'élément (i, j) doit être non-nul" permet de garder une transition.

Nous avons proposé un algorithme qui pour une matrice de transition initiale calcule une borne "st" ayant la structure décrite par le patron, ou indique que cela n'est pas possible. Nous avons également montré que cet algorithme renvoie une telle borne pour chaque patron qui est compatible avec la matrice initiale (un patron est dit compatible avec une matrice s'il existe au moins une borne st ayant la structure définie par le patron).

Ce travail représente une généralisation de l'approche algorithmique dans la méthode des bornes stochastiques. Le même algorithme peut être utilisé pour différentes structures de bornes. En effet, pour définir une nouvelle structure de borne il est seulement nécessaire de définir le patron associé. Nous avons proposé des patrons pour certaines structures connues :

- Upper-Hessenberg : c'est à dire un matrice triangulaire supérieure augmentée de la sous diagonale. On peut résoudre par un algorithme d'élimination simple et linéaire.
- Complément stochastique avec bloc D triangulaire supérieure : c'est à dire une matrice décomposable en 4 blocs dont le bloc inférieur droit est triangulaire supérieur et en prenant un bloc supérieur gauche de petite taille. On peut résoudre plus facilement par l'approche de Quessette.
- Single Input Markov Chain : matrice décomposable en blocs tels que pour rentrer dans les états d'un bloc, il est nécessaire de passer par un état d'entrée. L'algorithme de Feinberg et Chiu permet une résolution hiérarchique rapide.

Il est également possible d'imposer grâce à un patron certaines propriétés de la chaîne bornante (par exemple l'irréductibilité).

4 Model checking et encadrement stochastique, PRISM

Le "model checking" stochastique est un moyen de vérification des performances des systèmes probabilistes spécifiés à l'aide des chaînes de Markov (discrètes ou continues) et des logiques temporelles comme PCTL (Probabilistic Computational Tree Logic) dans le cas des chaînes discrètes , CSL (Continuous Stochastic Logic) dans le cas des chaînes continues et PRCTL (Probabilistic Reward Computational Tree Logic) dans les chaînes associées à des récompenses.

Le modèle checking stochastique de sûreté de fonctionnement repose sur la vérification de la validité de formules spécifiant des mesures de récompenses à l'état stationnaire et transitoire. On a donc les mêmes problèmes d'explosion combinatoire des espaces d'état lors des vérifications. Actuellement les "model checkers" probabilistes se contentent de calculer exactement les probabilités du modèle et de vérifier ensuite la contrainte spécifiée par la formule. Il est clair que l'utilisation des concepts des bornes stochastiques simplifie dans certains cas, la vérification. Nous avons proposé, dans le cadre de nos travaux, une approche de vérification de ces formules se basant sur les techniques d'encadrement stochastiques. Cette approche réduit la taille de l'espace des états et la complexité de la résolution numérique. Elle permet la vérification des formules spécifiant des mesures de récompenses désignés par les opérateurs suivants : $\mathcal{I}_I^n(\phi) \mid \mathcal{C}_I^n(\phi) \mid \mathcal{E}_I^n(\phi) \mid \mathcal{E}_I(\phi)$.

- Le premier opérateur $\mathcal{I}_I^n(\phi)$ est vérifié si la récompense à l'instant n est états vérifiant la formule ϕ est dans l'intervalle I.
- Le deuxième opérateur $\mathcal{C}_I^n(\phi)$ est vérifié si la valeur de la récompense cumulée depuis l'instant 0 jusqu'à l'instant n pour les états vérifiant la formule ϕ est dans l'intervalle I.
- L'opérateur $\mathcal{E}_I^n(\phi)$ est vrai si le taux de récompense depuis l'instant 0 à l'instant n pour les états vérifiant la formule ϕ est dans l'intervalle I.
- Et finalement l'opérateur de récompense stationnaire $\mathcal{E}_I(\phi)$ est satisfait si la récompense stationnaire des états vérifiant la formule ϕ est à valeur dans I.

Ainsi la vérification de ces formules nécessite le calcul des probabilités stationnaires et transitoires à un instant bien déterminé ou à une séquence d'instants succesifs.

Illustrons ceci sur un exemple. On considère un système de file d'attente Geo/D/1/B avec une gestion d'accès de type RED (Random Early Detection). On modélisé par une chaîne de Markov discrète où on ne présente que le nombre de paquets en attente. Ceci est un modèle approché car le mécanisme RED repose sur une destruction probabiliste des paquets entrant avec un seuil qui dépent de la moyenne mobile de la file et non pas de sa valeur instantanée. Néanmoins, on ne représente ici que la valeur actuelle de la taille de la file et on suppose que le taux de rejet du paquet dépend de cette taille. Lorsque la taille de la file dépasse la moitié du buffer, RED commence à rejeter aléatoirement des paquets lors de leur admission. Pour évaluer les pertes des paquets dans la chaîne, on désigne comme fonction de récompense le nombre moyen de paquets perdus par slot. A chaque état est associé sa valeur de récompense et des formules (appelées propositions atomiques) spécifiant l'état.

On associe aux états de la chaîne où il y a des pertes de paquets la formule $\phi = rejet$ et à l'état où le buffer est plein la formule $\phi = plein \wedge rejet$. ϕ peut aussi s'exprimer suivant les logiques temporelles

PCTL ou CSL, exemple $\phi = \diamond^k plein$ spécifiant les états pour lesquels on atteint l'état du système plein dans au plus k étapes. S'intéréssant à mesurer des récompenses pour les états qui vérifient la formule ϕ, le "model checking" possède des opérateurs qui expriment les mesures de récompense du système et en particulier dans les états qui vérifient ϕ. Dans le cas de la file exemple, pour évaluer le taux de rejet des paquets depuis l'instant 0 jusqu'à l'instant n et voir s'il dépasse ou pas un certain seuil r, il suffit de vérifier l'opérateur $\mathcal{E}_I^n(rejet)$ avec $I = [0, r]$ dans le "model checker".

En utilisant des récompense qui sont croissantes, on peut employer des bornes "st" sur la chaine pour obtenir une borne sup ou inf des récompenses et il est possible d'éviter de calculer les distributions de la chaîne d'origine. La vérification des formules s'effectue en comparant les récompenses des bornes inférieures et supérieures avec les seuils de l'intervalle $I = [a, b]$. Soit min et max les bornes inf et sup calculées, on a quatre cas en général :

1. min et max sont dans I. On peut donc conclure "Oui" immédiatement
2. min est supérieur à b. On peut donc conclure "Non" immédiatement.
3. max est inférieur à a. On peut donc conclure "Non" immédiatement.
4. min est inférieur à a ou max est supérieur à b, on ne peut alors rien conlure avec cette borne. Il faut améliorer la borne ou dans le pire des cas, faire le calcul exact.

La stratégie de constructon des partitions pour l'algorithme LIMSUB repose d'abord sur la division de l'espace d'états en deux sous-ensembles : le premier, S_{yes}, contient les états qui satisfont la formule ϕ et le deuxième, S_{no}, contient les états qui ne la satisfont pas. On réduit avec cette décomposition le travail uniquement sur l'espace d'état S_{yes}. De plus, on agrège les états de S_{yes} qui ont des valeurs de récompense proches pour obtenir un espace d'état plus petit et on calcule les bornes (supérieures et inférieures) sur les chaînes bornantes de taille inférieure à celle des chaînes de Markov d'origine.

5 Calcul de mesures transitoires sur de grands espaces d'états, IRISA-ID

La sûreté de fonctionnement d'un système informatique, est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre. La vie d'un système est perçue par ses utilisateurs comme une alternance entre deux états du service délivré par rapport à l'accomplissement de la fonction du système. Ces deux états du service sont le service correct, où le service délivré accomplit la fonction du système, et le service incorrect, où le service délivré n'accomplit pas la fonction du système. Une défaillance est alors une transition de service correct à service incorrect et une transition de service incorrect à service correct est une restauration. On représente généralement l'évolution du système par une chaîne de Markov $\{X_t\}$ évoluant en temps continu sur un espace d'états E fini. On se donne alors une partition de l'espace d'états E en deux sous-ensembles : l'ensemble U des états opérationnels qui représentent les états du système correspondant à la délivrance du service correct et l'ensemble D des états non opérationnels qui représentent les états du système correspondant à la délivrance du service incorrect. On peut ainsi voir l'évolution du système à travers une suite alternée de périodes opérationnelles où le service délivré est correct et de périodes non opérationnelles où le service délivré est incorrect. Les mesures de la sûreté de fonctionnement s'expriment alors en fonction du processus $\{X_t\}$ de la façon suivante.

La fiabilité, qui mesure de la délivrance continue d'un service, est une fonction notée $R(t)$ définie pour $t \in R^+$ par

$$R(t) = \Pr\{X_s \in U,\ \forall s \in [0, t[\}.$$

La disponibilité ponctuelle, qui est la probabilité d'avoir un service correct à un instant donné, est une fonction notée $PAV(t)$ définie pour $t \in R^+$ par

$$PAV(t) = \Pr\{X_t \in U\}.$$

La disponibilité sur l'intervalle $[0, t)$ est une variable aléatoire, qui mesure la fraction de temps pendant lequel le service est correct sur un intervalle de temps donné. Elle est notée $IAV(t)$ et définie pour $t \in R^+$ par

$$IAV(t) = \frac{1}{t} \int_0^t 1_{\{X_s \in U\}} ds.$$

Nous avons décidé de nous intéresser dans un premier temps au calcul de la fiabilité et au calcul de la disponibilié ponctuelle. Nous avons développé, en langage C, les algorithmes correspondant à ces 2 mesures et nous étudions actuellement la façon de les intégrer dans le logiciel PEPS de manière à permettre la spécification de modèles de grands systèmes. Cette intégration nécessite une attention particulière puisqu'il s'agit de transcrire les opérations matricielles classiques utilisées dans les programmes C en

termes d'opérations de l'algèbre tensorielle utilisées dans le logiciel PEPS. Lorsque cette intégration aura eu lieu, suivra une phase de tests où l'on vérifiera que les résultats produits par PEPS et ceux produits par les programme C sont bien identiques. Lors de cette phase de tests, on s'intéressera aussi de près aux différents temps d'exécution qui détermineront les tailles des espaces d'états permettant d'être traités pour le calcul de ces mesures.

Des résultats récents sur le calcul de la disponibilité ponctuelle ont permis la mise au point de techniques permettant de diminuer le temps de calcul par détection du régime stationnaire. On procèdera alors de la même façon pour voir si ces techniques peuvent s'implanter efficcement dans le logiciel PEPS.

Enfin, si le temps le permet, nous essaierons de compléter le calcul de ces mesures par le calul de la moyenne et de la distribution de la disponibilité sur intervalle.

6 Simulations efficaces, IRISA-ID

6.1 Simulation d'événements rares et simulation parfaite, ID

L'évaluation de la probabilité d'événements rares constitue l'une des difficultés majeures dans le calcul de la disponibilité asymptotique de grands systèmes. Celui-ci est modélisé par une chaîne de Markov multidimensionnelle, la taille de l'espace d'état explose en fonction de la dimension de la chaîne. L'espace d'état est partitionné en 2 : l'ensemble des états de fonctionnement normal $\mathcal{A}$ et son complémentaire $\mathcal{A}^c$. L'objectif est alors d'estimer la probabilité stationnaire de la chaîne d'être dans l'ensemble $\mathcal{A}^c$. Dans les cas pratiques la probabilité stationnaire de $\mathcal{A}^c$ est très faible. Par conséquent, les méthodes traditionnelles de simulation ne sont pas efficaces car le temps de stabilisation de la chaîne est très long (dépendance de l'état initial) et les échantillons générés sont fortement corrélés.

Une alternative consiste alors à effectuer des simulations dites *parfaites*, qui, avec un surcoût lié au suivi de plusieurs trajectoires simultanément, échantillonnent directement selon la loi stationnaire de la chaîne. Cette méthode, initiée par Propp & Wilson, simule différentes trajectoire de la chaîne en inversant le temps. Elle est d'autant plus efficace si les fonctions de transition de la chaîne sont monotones.

Dans un premier temps nous représentons la chaîne de Markov par un schéma itératif dirigé par des événements

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

En utilisant des propriétés de monotonie de la fonction $\Phi(., e)$ pour tout événement e, nous construisons un noyau de simulation parfaite permettant l'échantillonnage de la chaîne. De plus, les ensembles $\mathcal{A}^c$ étant croissants il est possible d'interrompre la simulation avant le couplage global des trajectoires analysées en parallèle et donc d'accélérer la simulation. Les principaux résultats obtenus sont :

Modélisation Les systèmes à base de réseaux markoviens de files d'attente possèdent des propriétés de monotonie. En particulier les politiques de routage dans les systèmes à capacité finie (rejet, blocage, débordement,..) sont monotones.

Discrétisation La représentation du système par des événements dirigés par des processus de Poisson indépendants permet une uniformisation du processus (passage en temps discret) préservant les propriétés de monotonie.

Accélération L'estimation de la probabilité d'être dans un ensemble croissant a été accélérée par des fonctions d'arrêt adaptées.

Les résultats obtenus ont été implantés dans un logiciel de simulation Ψ^2. A partir d'une description du système sous forme d'un ensemble d'événements et d'une fonction de "reward" monotone (appartenance à $\mathcal{A}^c$), il fournit un échantillon distribué selon la probabilité stationnaire d'être dans $\mathcal{A}^c$.

Cet environnement a été testé sur des exemples caractéristiques : rejet dans les systèmes à routage par débordement avec l'estimation de la probabilité de saturation ; analyse du blocage dans des lignes de production ; saturation dans des réseaux d'interconnection...

Par exemple pour un réseau d'interconnection de type delta comportant 32 files de capacité 30, la taille de l'espace d'état est de $31^{32} \simeq 5.10^{47}$, le temps de génération d'une valeur d'un échantillon permettant l'estimation de la loi marginale d'une file au dernier étage est de l'ordre de $100\mu s$ (sur un PC linux 1.2GHz, 512Mo). L'utilisation d'une telle méthode permet alors, en force brute, de simuler des échantillos significatifs pour l'estimation de probabilités faibles (de l'ordre de 10^{-6} en 3 heures sur un PC standard).

Ce travail doit être approfondi sur plusieurs points :

Méthodologie Cette approche doit être combinée à d'autres techniques, en particulier les méthodes de réduction de variance (variables antithétiques, Importance sampling,...)

Modélisation Actuellement seuls certains types d'événements monotones ont été implantés, les disciplines de routages telles que "Join the shortest queue", décomposition de clients, les arrivées

groupées, certain types de clients négatifs présentent également des propriétés de monotonie et doivent donc être implémentées et testées. De plus d'autres événements tels que la fusion de clients , les départs groupés,... sont "anti-monotones". Cette propriété doit être explorée plus en détail.

Expérimentation Les exemples utilisés attaquent des problèmes basés sur les réseaux de files d'attente. Peut-on généraliser cette approche à d'autres formalismes tels que les réseaux d'automates stochastiques ? Intuitivement, la réponse devrait être positive, il faudrait alors combiner les approches de bornes stochastiques (cf sections précédentes) avec la simulation parfaite.

6.2 Simulation d'événements rares par les méthodes de Monte Carlo, IRISA

La simulation de type Monte Carlo est le seul outil d'analyse lorsque les hypothèses faites sur le modèle ne sont pas suffisamment strictes ou lorsque l'espace d'états est trop grand pour être traité par les méthodes précédentes. La simulation standard, c'est à dire mimant directement le comportement du système, s'avère cependant totalement inefficace lorsqu'il s'agit d'étudier des événements rares ; des techniques dites d'accélération, consistant à réduire la variance des estimateurs ou à diminuer le temps de simulation, sont alors nécessaires.

Nous travaillons sur les estimateurs d'événements rares en général, et étudions leur robustesse lorsque les événements deviennent de plus en plus rares. Ceci est mathématiquement caractérisé par l'introduction d'un paramètre ε tel que, lorsque $\varepsilon \to 0$, la probabilité γ de l'événement considéré vérifie $\gamma \to 0$. En pratique, ε peut représenter par exemple le taux de défaillance maximal d'un composant (pour les modèles dynamiques) ou la fiabilité d'un composant (pour les modèles statiques). Dans les modèles de performance, $\varepsilon = 1/B$ où B est la taille d'un tampon lorsqu'on cherche la probabilité de perte. Dans la litérature, les propriétés de robustess habituellement étudiées sont l'erreur relative bornée, qui consiste à vérifier si la taille relative de l'intervalle de confiance (théorique) obtenu reste majorée lorsque $\varepsilon \to 0$, c'est à dire lorsque l'événement devient de plus en plus rare. Une autre propriété (plus faible) est celle d'optimalité asymptotique pour les estimateurs utilisant l'échatillonnage préférentiel. Au cours de travaux précédents, nous avions mis en évidence une autre propriété, appelée *approximation normale bornée*, qui certifie que le niveau de confiance (c'est à dire la probabilité que la valeur cherchée soit dans l'intervalle considéré) de l'estimation reste valide lorsque $\varepsilon \to 0$ pour une taille d'échantillon fixée.

Nos travaux consistent à généraliser ces notions de robustesse asymptotique des estimateurs ; nous avons ainsi pu montrer qu'il existe des estimateurs d'événements rares efficaces pour lesquels les propriétés de la litérature que nous venons de rappeler ne sont pas vérifiées, illustré sur un problème de fiabilité d'un réseau de communication. Ce problème vient du fait que les propriétés n'intègrent pas une composante importante d'un estimateur : le temps de simulation par réplication. Ceci nous a conduit à définir les notions d'efficacité relative bornée et d'approximaton normale bornée généralisée (comme généralisations de l'erreur relative bornée et de l'approximation normale bornée respectivement), qui étudient respectivement la robustesse de la taille relative et du niveau de confiance de l'intervalle de confiance lorsque $\varepsilon \to$, mais pour un temps de simulation donné au lieu d'un nombre de réplications donné. Ces propriétés sont illustrées être en fait celles qu'un estimateur doit vérifier. De même, nous nous intéressons à la fonction de couverture des estimateurs, la notion d'approximation normale bornée, basée sur le théorème de Berry-Esseen qui borne la distance entre la loi empirique et la loi normale, ne donnant qu'une condition *suffisante* pour obtenir une bonne couverture, mais a priori non nécessaire. La fonction de couverture permet d'étudier empriquement la qualité de l'intervalle.

6.3 Méthode quasi-Monte Carlo pour la simulation des chaînes de Markov

Comme décrit dans les sections précédentes, les chaînes de Markov constituent un outil privilégié pour la modélisation et l'analyse de la sûreté de fonctionnement et l'évaluation des performances des systèmes informatiques et de télécommunication.

Nous avons conçu une nouvelle méthode de simulation basée sur les techniques quasi-Monte Carlo pour accélérer l'évaluation de ces chaînes. Les méthodes quasi-Monte Carlo forment un analogue déterminsite de Monte Carlo où les nombres aléatoires uniformes sont remplacés par des nombres ne mimant plus le hasard mais ayant la propriété de se répartir très rapidement uniformément sur l'espace considéré. Ces méthodes sont cependant connues rencontrer deux difficultés majeures : l'erreur d'estimation est difficile à estimer en pratique et leur efficacité par rapport aux méthodes Monte Carlo décroit avec la dimension du problème considéré (qui est linéaire en le nombre d'étapes de la chaîne de Markov considérée). Le premier problème est remédié en perturbant légèrement et aléatoirement les suites considérées sans leur faire perdre leur propriété de bonne répartition, puis en appliquant le théorème de la limite centrale sur ces différentes randomisations. Le deuxième problème, plus directement lié à la structure de l'analyse des chaînes de Markov, est résolu en appliquant un algortihme simple. Seulement une suite de dimension 1 peut être considérée. les "réplications" de la chaîne de Markov à étudier sont simulées en parallèle étape par étape en utilisant cette suite au lieu des nombres pseudo-aléatoires. Après chaque étape, les chaînes

doivent être réordonnées selon une relation d'ordre total (qui doit par hypothèse exister sur l'espace d'états), avant de recommencer la simulation pour l'étape suivante...

Nous avons pu prouver la convergence de la méthode (quand le nombre de réplications augmente). Un ordre de convergence *dans le pire cas* est prouvé être le même que celui de Monte Carlo en moyenne. Dans certains cas particuliers, des convergences plus rapides sont aussi obtenues.

Numériquement, l'ordre d'amélioration obtenu par rapport à Monte carlo est toujours ntable, et peut s'avérer spectaculaire, dépassant les 60000 pour certaines illustrations sur les files d'attente.

Action Concertée Incitative

SÉCURITÉ INFORMATIQUE

Rapport de mi-parcours

Publications

Bornes stochastiques

1. *Algorithms for an irreducible ans lumpable strong stochastic bound*, J.M. Fourneau, M. Lecoz, F. Quessette, Numerical Solution of Markov Chains, 2003, USA.
2. *An open tool to compute stochastic bounds on steady-state distributions and rewards*, Jean Michel Fourneau, Mathiew Le Coz, Nihal Pekergin and Franck Quessette, IEEE MASCOTS 03, USA, Tools session.
3. *A proof of st-comparison for polynomials of a stochastic matrix and how we can improve the accuracy of st-bounds*, T. Dayar, J-M. Fourneau, N. Pekergin, J-M. Vincent, HET-NET, Bradford, UK, july 2004.
4. *Computing closed-form stochastic bounds on transient distributions of Markov chains*, M. Benmammoun, N. Pekergin, Workshop Modelling and Performance Evaluation for Quality of Service in Next Generation Internet, in IEEE SAINT 2005 Conference, Italy, 2005.
5. *A Matrix Pattern Compliant Strong Stochastic Bound*, A. Busic, J.M. Fourneau, Workshop Modelling and Performance Evaluation for Quality of Service in Next Generation Internet, in IEEE SAINT 2005 Conference, Italy, 2005.

Combinaison des approches de simulation

1. *On Numerical Problems in Simulations of Highly Reliable Markovian Systems*, B. Tuffin, Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST), University of Twente, Enschede, the Netherlands September 2004 .
2. *Comparison of Quasi-Monte Carlo-Based Methods for the Simulation of Markov Chains*, L'ecot, C. and Tuffin, B., to appear in Monte Carlo Methods and Applications Journal, 2004.
3. *Pathset based conditioning for transient simulation of highly dependable systems*, H. Cancela, G. Rubino and M. Urquhart, 5th International Conference on Monte Carlo and quasi-Monte Carlo Methods, Juan les Pins, France, juin 2004.
4. *Prefect simulation of queueing networks with blocking and rejection*, J.M Vincent, Workshop Modelling and Performance Evaluation for Quality of Service in Next Generation Internet, in IEEE SAINT 2005 Conference, Italy, 2005.
5. *Randomization of Quasi-Monte Carlo Methods for Error Estimation : Survey and Normal Approximation*, B. Tuffin, Monte Carlo Methods and Applications, Vol.10, Num.3-4, pages 617- 628, 2004.
6. *Quasi-Monte Carlo simulation of Markov chains with randomized copies of a two-dimensional highly-uniform point set*, P. L'Ecuyer, C. L'ecot, B. Tuffin, Monte Carlo and quasi-Monte Carlo Methods, Springer-Verlag, 2005
7. *Coverage Function of Randomized Quasi-Monte Carlo Methods*, Bruno Tuffin, INFORMS Applied Probability Conference, Ottawa, July 2005.
8. *A New Randomized Quasi-Monte Carlo Approach for Markov Chains*, Pierre L'Ecuyer, Christian Lécot, and Bruno Tuffin, INFORMS Applied Probability Conference, Ottawa, July 2005.
9. *Bounded Relative Efficiency in Rare Event Simulation*, H. Cancela, G. Rubino and B. Tuffin, In Proceedings of SAINT 2005 workshops, IEEE CS Press, Trento, January 2005.

Formulation modulaires des modèles

1. *Memory efficient kronecker algorithms with applications to the Modelling of parallel systems*, A. Benoit B. Plateau and W. Stewart, Journal of Future Generation of Computer Systems, Elsevier, june 2004.

2. *Agregation of Stochastic Automata with replicas*, A. Benoit and L. Brenner and P. Fernandes and B. Plateau, Journal of Linear Algebra and its Applications , v 386, pages 111-136, july 2004.
3. *On the benefits of using functionnal transitions and Kronecker algebra*, A. Benoit and P. Fernandes and B. Plateau and W. Stewart, Journal of Performance Evaluation, vol PEVA1119, April 2004.

Model checking probabiliste

1. *Improving Stochastic Model Checking with Stochastic Bounds*, J.M. Fourneau, N. Pekergin, S. Younes Workshop Modelling and Performance Evaluation for Quality of Service in Next Generation Internet, in IEEE SAINT 2005 Conference, Italy, 2005.