Evaluer la fiabilité de systèmes redondants

J. M. Fourneau

Projet SurePaths entre ID, IRISA et PRiSM

November 19, 2005

Problématique

- Indices stochastiques de fiabilité, de disponibilité ou de performabilité
- Transitoire (horizon fini) : probabilité d'être en panne à la date t sachant l'état initial, durée d'activité entre 0 et t,
- Stationnaire (horizon infini) : probabilité d'un mode dégradé
- Chaîne de Markov (pour faire des calculs explicites)
- Evaluer numériquement ou estimer par simulation la probabilité d'un événement trop rare qui se produit dans un espace d'états trop grand

Solution

- Structurer l'espace
  - Méthodes par composition de sous modèles en isolation et interaction par taux fonctionnels ou synchronisations (réseaux d'automates stochastiques B.Plateau)
  - Description tensorielle de la matrice de la chaîne
- Calculer des garanties: bornes déterministes ou stochastiques
- Améliorer la convergence des estimateurs

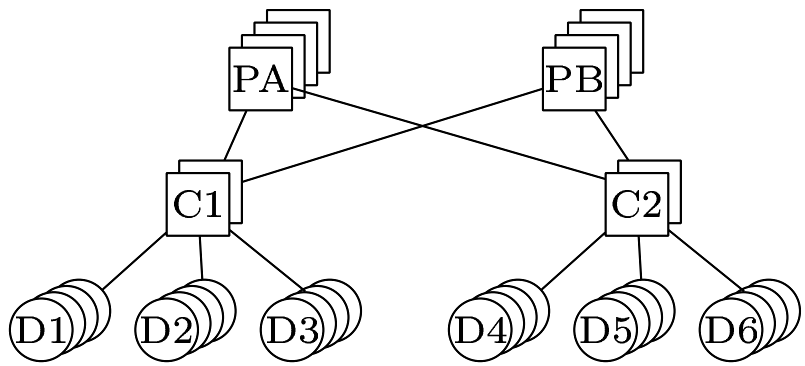
Un cas typique

Des problèmes de disponiblité pour des systèmes multi-composants

- Les composants de même type ne sont pas différentiés.
- Des pannes et des réparations, simples ou multiples, exponentielles ou de lois générales
- une discipline de service pour les réparations (par exemple avec des priorités)
- 1 ou plusieurs réparateurs
- plusieurs modes de pannes (Soft, Hard); la réparation dépend du mode de la panne et de l'état du système
- Les états UP (système opérationnel) : un nombre minimal de composant en marche dans chaque type
- Etats UP : en grand nombre mais en très faible proportion

Description du modèle

Muntz et al. 1989, Carrasco 1999



- Les composants
  - Processeurs : 2 types (PA et PB) non indépendants !
    4 processeurs de chaque type : un seul processeur par type fonctionne, les autres sont là en cas de pannes.
  - Contrôleurs : 2 types ($C1$ et $C2$); 2 contrôleurs de chaque type.
  - Disques : 6 piles de 4 disques. $D1$, $D2$ et $D3$ liées aux contrôleurs $C1$, et les piles $D4$, $D5$ et $D6$ aux contrôleurs $C2$.

- Les états UP :
  - au moins un processeur en activité (PA ou PB)
  - au moins un contrôleur en activité pour chaque type ($C1$ et $C2$)
  - au plus un disque en panne par pile
- Les pannes et les réparations :
  - 2 modes de pannes : Soft et Hard; pour chaque type de composant
  - une seule réparation à la fois, choix uniforme
  - Pannes simples sauf la panne du processeur PA qui peut entrainer la panne du processeur PB
  - seul le processeur en activité peut tomber en panne
  - les taux de panne dépendent du type du composant et du mode

Taille du problème : Matrice creuse

- Etats : $6^2 \times 15^8$, soit un peu plus de 92 milliards.
- Degré moyen de 11 $\rightarrow 10^{12}$ transitions;
- 10 octets par transition (structure creuse) $\rightarrow$ ordre de grandeur 10^{13} octets pour la matrice
- Impossible de générer raisonnablement un tel modèle, même sur disque

Mais on a exactement 1300200 états UP (seulement)

Taille du problème : Représentation tensorielle

- 2 automates à 15 etats pour les processeurs
- 6 automates à 15 états pour les disques
- 2 automates à 6 états pour les contrôleurs
- Représentation très efficace
- Et surtout compatible avec les méthodes numériques itératives ou projectives (Arnoldi, GMRES)
- Mais il faut plusieurs vecteurs des états atteignables en mémoire
- En pratique, quelques millions.d'états
- Un logiciel disponible (PEPS)

Garantie Stochastique

- Calculer une borne sur la probabilité à l'aide d'un modèle plus simple obtenu automatiquement, la borne fournissant la garantie:
  - Dérivation automatique de matrices de transition
  - repose sur la comparaison de variables aléatoires, de processus et de trajectoires
  - Monotonie des opérateurs (conservation de l'ordre entre distribution)
- Ordre strong sur V.A. avec espace des états totalement ordonné.
- On sait maintenant faire à partir des matrices de transition en forme creuse ou en forme tensorielle
- et pour calculer des distributions stationnaires et transitoires

Exemple Ordre Strong et Monotonie

- Ordre Strong sur une VA discrete:
  $p <_{st} q$ ssi $\sum_{j=k}^{n} p_j \leq \sum_{j=k}^{n} q_j$ pour tout k
- Exemple: $\alpha = (0.1, 0.3, 0.4, 0.2)$ et $\beta = (0.1, 0.1, 0.5, 0.3)$.
- On a $\alpha <_{st} \beta$ car:

$$\left[\begin{array}{lcl} 0.2 & \leq & 0.3 \\ 0.2 + 0.4 & \leq & 0.3 + 0.5 \\ 0.2 + 0.4 + 0.3 & \leq & 0.3 + 0.5 + 0.1 \end{array}\right.$$

- Associé aux récompenses croissantes
- Monotonie de la matrice de transition: si les lignes sont croissantes.

Réduire la taille

- Repose sur l'agrégation (ou lumpabilité) et sur une partition en macro-états
- Une chaine est fortement agrégeable si la propriété de Markov reste vraie au niveau des macro-états
- Test algorithmique connu et simple (somme par bloc constante).
- Idée : Construire une borne agrégable.

Exemple

- Constuire une borne agrégée sur deux états de:

$$P = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.4 & 0.4 & 0.0 & 0.1 \\ 0.2 & 0.1 & 0.5 & 0.2 & 0.0 \\ 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.2 & 0.2 & 0.1 & 0.5 \end{bmatrix}$$

- 2 macros états: $(1,2)$ et $(3,4,5)$.
- 2 étapes : borner et agréger

$$\left[\begin{array}{cc|ccc} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.4 & 0.3 & 0.1 & 0.1 \\ \hline 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0 & 0.1 & 0.7 & 0.1 \\ 0 & 0.1 & 0.1 & 0.3 & 0.5 \end{array}\right] \quad \left[\begin{array}{cc|ccc} 0.5 & 0. & 0.3 & 0.2 & 0.0 \\ 0.1 & 0.4 & 0.3 & 0.1 & 0.1 \\ \hline 0.1 & 0. & 0.7 & 0.1 & 0.1 \\ 0.1 & 0 & 0.1 & 0.7 & 0.1 \\ 0 & 0.1 & 0.1 & 0.3 & 0.5 \end{array}\right] \quad \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix}$$

Améliorer la qualité

- Mieux choisir les agrégats
- Minimiser les contraintes de la monotonie et de l'ordre
- Composition de sous-systèmes : un ordre naturel partiel
- Actuellement l'ordre total est nécéssaire mais il impose trop de contraintes non naturelles
- Comment se limiter à un ordre partiel naturel.
- Développer d'autres techniques pour réduire la complexité.

Plusieurs algorithmes

- Selon la structure de départ: matrice creuse ou tenseur
- Selon l'ordre sur les V.A. (Strong ou icx) et sur les états (total ou partiel)
- Selon la méthode retenue pour réduire la complexité.
- Intégré dans certains cas à PEPS, ou à XBORNE

Simulation

- Simulation à événements rares
- Simulation parfaite (couplage dans le passé)
  - Règle le problème de la preuve de convergence
  - Utilise facilement du parallélisme massif (Grille)
  - Repose sur des concepts de monotonie stochastique identiques à ceux employés pour la garantie stochastique
  - Un logiciel disponible (PSI).

En guise de conclusion

- Méthodes de composition et de garantie stochastique
- Pour améliorer les techniques existantes (par exemple SAN et mesures transitoires pour la fiabilité)
- Ou développer de nouvelles approches
- Plus largement, méthodes utiles en évaluation de performances ou pour le modèle checking stochastique
- Mêmes problèmes de taille d'espace et d'événements rares
- Récompenses plus complexes (par exemple selon les trajectoires)