Simulation d'ordonnancement temps réel avec prise en compte de l'impact des caches

Maxime Chéramy[1], Pierre-Emmanuel Hladik[1] and Anne-Marie Déplanche[2]

[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[2]Université de Nantes, IRCCyN UMR CNRS 6597, ECN,
1 rue de la Noe, BP92101, F-44321 Nantes cedex 3, France

Résumé

Nous nous intéressons à l'évaluation et la comparaison des politiques d'ordonnancement multiprocesseur en prenant en compte l'architecture matérielle.
Nous visons à faciliter ce travail via un outil de simulation intégrant certains aspects pouvant avoir un impact sur les performances réelles, à savoir :

- les caches,
- la durée de prise de décision de l'ordonnanceur,
- les changements de contexte.

Introduction

Depuis les années 90, de nombreux travaux ont été menés sur l'**ordonnancement temps réel multiprocesseur**. Les premiers algorithmes théoriquement optimaux se sont révélés inutilisables en pratique de part leur nombre important de préemptions et migrations. À partir de ces observations, de nouvelles politiques ont été proposées afin de les réduire.
Cependant, les auteurs des algorithmes fournissent des propriétés théoriques sur ceux-ci qui **ne permettent pas de comparer facilement les politiques** d'ordonnancement. Ceci est d'autant plus vrai dans le cas multiprocesseur, où **les architectures apportent de nouvelles difficultés** avec des caches partagés, de nouveaux bus, etc.

La comparaison des politiques en tenant compte des architectures réelles se fait généralement sur cible réelle [1]. Cependant, ce travail est long, difficile et **ne permet pas de conduire facilement de larges expérimentations**.
Afin de contourner ces problèmes, nous proposons une simulation à « grain intermédiaire ». **Notre contribution est une approche complémentaire aux analyses théoriques et expérimentales**.

Données d'entrée

Le **modèle de Liu et Layland est étendu** par des informations statistiques permettant de caractériser le **comportement mémoire d'une tâche** :

- Nombre d'instructions ($instr$),
- Cycles Par Instructions ($base\_cpi$) sans les pénalités d'accès mémoire,
- Taux d'accès mémoire (mix),
- Profil d'accès mémoire (SDP).

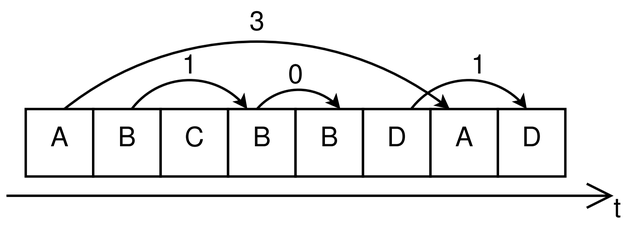

FIGURE 1: Séquence d'accès mémoire. A, B, C et D sont des lignes de cache et les nombres indiquent les « stack distances ».

Un cache est défini par sa taille, son associativité et la pénalité temporelle d'un défaut de cache. Les **caches** sont inclusifs et peuvent être **partagés**.
Pour le moment, nous nous limitons à des **caches de données** avec une politique de remplacement **LRU**.

Modèles

La **durée d'exécution des travaux est calculée dynamiquement** pendant la simulation. Elle est égale à :

$$c_\tau = instr_\tau \cdot cpi_\tau \qquad (1)$$

Le calcul du CPI prend en compte les pénalités liées aux caches :

$$cpi_\tau = base\_cpi_\tau + mix_\tau \cdot P_\tau \qquad (2)$$

avec

$$P_\tau = pm_0 + \sum_{x=1}^{X} mr_{\tau,Lx} \cdot pm_{Lx} \qquad (3)$$

Le calcul du taux de défaut de cache peut se faire à l'aide de modèles déjà existants et qui permettent de traiter les préemptions et le partage de caches [2]. Ces modèles se basent sur le SDP, mix et CPI.

SimSo

Les modèles présentés sont intégrés dans un simulateur d'ordonnancement [3].

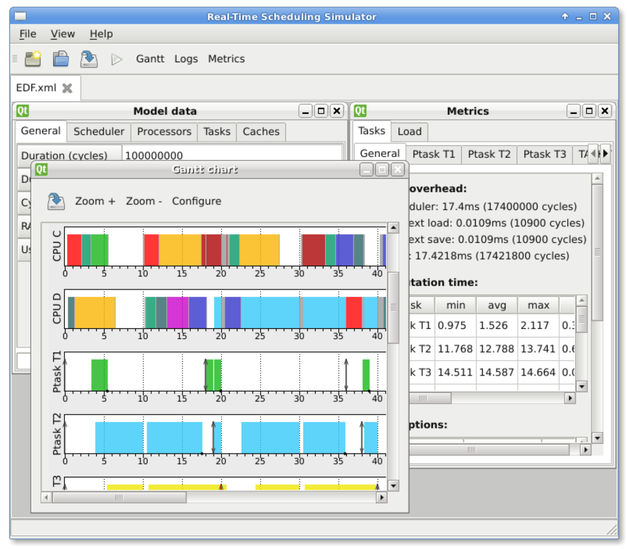

FIGURE 2: Interface graphique de SimSo

Conclusion

Notre objectif est la **comparaison des politiques en prenant en compte certains surcoûts liés à l'ordonnancement**. Pour cela, nous avons choisi une approche intermédiaire entre les approches théoriques et expérimentations réelles.
Notre démarche ne vise pas à améliorer directement l'ordonnançabilité mais elle devrait nous permettre de **mieux comprendre et identifier les sources de surcoûts**. Les réduire peut avoir une incidence sur la consommation énergétique ou sur la réactivité du système par exemple.

Références

[1] J. M. Calandrino *et al.*, "Litmus^{RT} : A testbed for empirically comparing real-time multiprocessor schedulers," in *Proc. of the 27th IEEE International RTSS*, 2006.

[2] D. Chandra *et al.*, "Predicting inter-thread cache contention on a chip multi-processor architecture," in *Proc. of the 11th International Symposium on HPCA*, 2005.

[3] M. Chéramy *et al.*, "Simulation of real-time multiprocessor scheduling with overheads," in *Proc. of 3th International Conference on SIMULTECH*, 2013.