

TITRE: Mémoire Transactionnelle fortement concurrente

EQUIPE/THEME: Combinatoire-algorithmique

DIRECTEURS: Alessia Milani

COURRIELS: milani@labri.fr

MOTS-CLES: programmation concurrente, structure de données fortement concurrentes, synchronisation, mémoire transactionnelle, algorithmique, complexité

DIRECTEURS HABILITES: Nicolas Hanusse

DESCRIPTION du SUJET:

Problématique générale

La dissipation thermique empêchant la progression des fréquences d'horloge, les architectures multi-cœurs, dans lesquelles plusieurs cœurs (unités de calcul) sont gravés sur une même puce, est actuellement la seule technique permettant d'augmenter la puissance de calcul. La programmation concurrente est indispensable lorsque l'on souhaite bénéficier des architectures multi-cœurs.

Le paradigme de programmation concurrente basé sur les verrous est actuellement le plus répandu, mais demande de l'expertise pour éviter des situations de blocage tout en garantissant une bonne exploitation du parallélisme possible grâce à l'architecture sous-jacente. La complexité provient de la synchronisation de fils d'exécution (threads) concurrents qui doivent accéder des une parties communes de la mémoire partagée. Sans cette synchronisation, les accès concurrents pourraient laisser la mémoire dans un état incohérent et ainsi générer des résultats incorrects ou provoquer dans le pire des cas des défaillances du logiciel.

Le changement de masse des architectures et le fait que la plupart de programmeurs ne sont pas experts en programmation concurrente demande de repenser les abstractions pour la programmation concurrente.

La mémoire transactionnelle proposée par M. Herlihy et Moss [1] est considéré un des paradigmes de programmation concurrente parmi les plus prometteuses. La mémoire transactionnelle doit permettre de traduire de façon quasi-automatique du code séquentiel (qui ne comporte qu'un thread) en code concurrent. En particulier, en utilisant la notion de transaction bien connue dans les bases de données, le programmeur doit spécifier quels sont les blocs de code qui doivent être exécutés comme des transactions (comme une operation indivisible) et la mémoire transactionnelle est censée prendre en charge la synchronisation des accès concurrents en garantissant le progrès du calcul et la cohérence de la mémoire.

Objectifs de la thèse

Depuis la présentation du paradigme de mémoire transactionnelle, plusieurs algorithmes ont été proposés pour l'implémenter. Ces algorithmes ont des propriétés différentes et dans la plupart des cas ont été évalués au moyen de simulations. Aucun algorithme a des performances supérieures aux autres pour tout workload et en générale les performances des algorithmes existants ne sont pas encore suffisantes pour remplacer la programmation à l'aide de verrous.

Le but de cette thèse est de contribuer à établir les fondements théoriques du calcul transactionnel. En particulier, on souhaite fournir des modèles et mesures pour analyser la

complexité et le performance de mémoires transactionnelles et comprendre comment les techniques algorithmiques utilisées ou les propriétés implementées impactent sur la complexité et les performances d'une mémoire transactionnelle.

[1] Maurice Herlihy, J. Eliot B. Moss: Transactional Memory: Architectural Support for Lock-Free Data Structures. ISCA 1993: 289-300