HAL
open science

Une heuristique efficace pour l'ordonnancement périodique de tâches avec contraintes de stockage

Karine Deschinkel, Sid Touati

To cite this version:

Karine Deschinkel, Sid Touati. Une heuristique efficace pour l'ordonnancement périodique de tâches avec contraintes de stockage. 10e Congrès annuel de la société française de Recherche Opérationnelle et d'Aide à la Décision - ROADEF 2009, INRIA Nancy Grand Est en collaboration avec le LORIA, Feb 2009, Nancy, France. hal-00647768

HAL Id: hal-00647768

https://inria.hal.science/hal-00647768

Submitted on 2 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Une heuristique efficace pour l'ordonnancement périodique de tâches avec contraintes de stockage

K. Deschinkel, and S.A.A Touati

PRiSM, Université de Versailles Saint-Quentin en Yvelines, 45 avenue des Etats-Unis, 78035 Versailles
Karine.Deschinkel@prism.uvsq.fr
Sid.Touati@prism.uvsq.fr

1 Description du problème

Cet article traite du problème d'optimisation du besoin en stockage dans les graphes de tâches périodiques. En pratique, notre problème tend à minimiser le besoin en registres dans les programmes embarqués, où les instructions d'une boucle sont représentées par un graphe de dépendances de données cyclique (GDD). Dans cet article, nous supposons une exécution parallèle des instructions sans aucun modèle de ressources - la parallélisme étant borné par les contraintes de stockage uniquement. Notre but est d'analyser le compromis entre le besoin en registres et le parallélisme dans un problème d'ordonnancement périodique de tâche.
Les techniques existantes d'allocation périodique de registres appliquent généralement un ordonnancement périodique d'instructions sous contraintes de ressources, tout en étant sensible aux contraintes de registres. Plusieurs travaux proposent des algorithmes d'ordonnancement d'instructions d'une boucle (sous contraintes de ressources et de période) tel que le code résultant ne crée pas plus de R variables simultanément en vie. Or, en pratique, de tels algorithmes n'engendrent pas toujours des codes rapides, car les contraintes de registres sont plus critiques que les contraintes de performances : ceci car les accès mémoire coûtent plus de 100 unités de temps (cycles processeur) alors que les latences des instructions sont inférieures à 10 cycles. Dans cet article, nous garantissons les contraintes des registres avant la phase d'ordonnancement d'instructions sous contraintes de ressources : nous analysons et modifions le GDD afin de garantir que tout ordonnancement d'instructions sous contraintes de ressources ne crée pas plus de R variables simultanément en vie. Notre modification du GDD essaie de ne pas altérer l'extraction du parallélisme si possible en prenant comme métrique la valeur du circuit critique du graphe. Cette idée a déjà été abordée dans [1].

2 Une heuristique en deux phases

Le problème d'ordonnancement périodique de tâches (instructions) cycliques avec minimisation du nombre de registres, dans sa forme la plus générale, peut se formuler sous forme d'un programme linéaire en nombre entiers où les variables de décision sont les dates de début de chacune des tâches, des variables binaires indiquant une réutilisation ou non de registres entre deux tâches (non nécessairement distinctes), et les distances de réutilisation. Ce problème étant NP-complet [2], une résolution exacte s'avère trop coûteuse en pratique. Ici, nous avons exploité le fait que le problème en question fait apparaître un problème sous-jacent d'affectation pour lequel nous connaissons un algorithme en temps polynomial (méthode Hongroise) pour proposer une heuristique appropriée appelée SIRALINA [3]. Cette heuristique, en deux phases, nécessite dans un premier temps la résolution d'un programme linéaire entier avec une matrice des contraintes totallement unimodulaire, puis la résolution d'un problème d'affectation linéaire.

3 Résultats

Nous avons testé SIRALINA sur un ensemble de benchmarks, issus de la communauté d'optimisation de codes, et correspondants à des codes de calcul haute performance pour l'embarqué (LAO, MEDIABENCH, SPEC2000-CINT, SPEC200-CFP, SPEC2006). Nos expériences montrent que SIRALINA produit des solutions très satisfaisantes en très peu de temps. Par conséquent, cette méthode est en train d'être incluse dans un compilateur pour l'embarqué chez STMicroelectronics.

4 Perspectives

Nous devons étudier la manière de prendre en compte dans notre modélisation des spécificités du problème liées à l'architecture matérielle ou à la réalisabilité de l'ordonnancement sous contraintes de ressources. Par exemple, l'utilisation de banc de registres rotatifs force l'existence d'un seul circuit hamiltonien dans le graphe de réutilisation. Le graphe de réutilisation est ordonnançable sous contraintes de ressources à condition que tout circuit dans ce graphe ait une distance positive. Le traitement de ces différentes spécificités constitue des pistes de travail supplémentaires.

Références

1. S-A-A Touati et C. Eisenbeis : Early Periodic Register Allocation on ILP Processors. Parallel Processing Letters, 14(2), Juin 2004. World Scientific.
2. S-A-A Touati : Register Pressure in Instruction Level Parallelisme. Thèse, Université de Versailles, France, Juin 2002. ftp.inria.fr/INRIA/Projects/a3/touati/thesis.
3. K. Deschinkel et S-A-A Touati : Efficient Method for Periodic Task Scheduling with Storage Requirement Minimisation, Lecture Notes in Computer Science (COCOA 2008), Canada, Août 2008.