Application BASTRI

Fiches Equipes

CAMUS (SR0870KR)

Compilation pour les Architectures MUlti-processeurs et multi-coeurS
CAMUS (SR0411YR) CAMUS

Statut: Décision signée

Responsable : Philippe Clauss

Mots-clés de "A - Thèmes de recherche en Sciences du numérique - 2023" : A1.1.1. Multi-cœurs, pluri-coeurs , A1.1.4. HPC , A2.1.1. Sémantique des langages de programmation , A2.1.6. Programmation concurrente , A2.2.1. Analyse statique , A2.2.4. Architectures parallèles , A2.2.5. Environnements d'exécution , A2.2.6. GPGPU, FPGA... , A2.2.7. Compilation adaptative , A2.4. Méthodes formelles pour vérification, sureté, certification

Mots-clés de "B - Autres sciences et domaines d'application - 2023" : B4.5.1. Informatique "verte" , B6.1.1. Génie logiciel , B6.6. Systèmes embarqués

Domaine : Algorithmique, programmation, logiciels et architectures
Thème : Architecture, langages et compilation

Période : 01/10/2023 -> 30/09/2027
Dates d'évaluation : 19/03/2020

Etablissement(s) de rattachement : U. STRASBOURG
Laboratoire(s) partenaire(s) : ICUBE (UMR7357)

CRI : Centre Inria de l'Université de Lorraine
Localisation : Icube Illkirch-Graffenstaden - IUT Robert Schuman
Code structure Inria : 051089-1

Numéro RNSR : 200920957V
N° de structure Inria: SR0870KR

Présentation

L'équipe CAMUS s'attache à développer, adapter et étendre des techniques de parallélisation et d'optimisation automatiques, ainsi que des méthodes de preuves et de certification, pour l'exploitation efficace des processeurs multi-coeurs existants et à venir.

Ses recherches s'articulent autour de cinq problématiques, toutes intimement liées, pour atteindre les objectifs qui sont la performance, la correction et la productivité. Ces problématiques sont : la parallélisation et l'optimisation statiques de programmes (où tout le parallélisme détecté statiquement est exprimé, ainsi que tout le parallélisme "possible" qui sera éventuellement exploité dynamiquement), le profilage et la modélisation du comportement à l'exécution (des modèles expressifs de représentation du comportement serviront de moteur aux processus de parallélisation dynamique), la parallélisation et l'optimisation dynamiques (ces processus pouvant fonctionner au sein d'une machine virtuelle), la programmation et la compilation objet pour multi-coeurs (où le parallélisme d'objets, exprimé ou détecté, doit résulter en une exécution efficace), et la preuve de transformations de programmes pour multi-coeurs (la correction des nombreuses transformations statiques ou dynamiques devant être maîtrisée).

Axes de recherche

Les objectifs de l'équipe relèvent directement de la mise en adéquation entre le logiciel et la nouvelle voie d'évolution des processeurs à travers les multi-coeurs. Performance, correction et productivité doivent être les effets ressentis par les utilisateurs. Ils seront le fruit de recherches que nous articulons autour des problématiques suivantes :

- Problématique 1 : Parallélisation et optimisation statiques de programmes
- Problématique 2 : Profilage et modélisation du comportement à l'exécution
- Problématique 3 : Parallélisation et optimisation dynamiques, machine virtuelle
- Problématique 4 : Programmation et compilation objet pour multi-coeurs
- Problématique 5 : Preuve de transformations de programmes pour multi-coeurs

Contact

- **Responsable :** Philippe Clauss
- **Tél :** 03.68.85.45.37
- **Secrétariat Tél :** 03.87.54.72.80

En savoir plus

- Site de l'équipe
- Site sur inria.fr
- Derniers Rapports d'Activité : 2015 , 2016 , 2017 , 2018 , 2019 , 2020 , 2021 , 2022 , 2023

Documents sur la structure

- Intranet
- Privés

Décisions

- 13516 (01/03/2019) : création
- 14582 (09/12/2020) : prolongation
- 15180 (13/12/2021) : prolongation
- 15579 (22/08/2022) : prolongation
- 16468 (13/09/2023) : création

Localisation

- **Adresse postale :** Icube Pôle API 300 Boulevard Sébastien Brant 67400 Illkirch-Graffenstade FRANCE
- **Coordonnées GPS :** 48.52569703, 7.737738043

La performance en temps étant notre principal objectif, les applications cibles de nos développements sont caractérisées par des phases de calculs intensifs. De telles applications sont nombreuses dans les domaines du calcul scientifique, de l'optimisation, de la fouille de données et du multimedia.

Le développement d'applications efficaces et correctes pour les processeurs multi-coeurs nécessite des interventions à toutes les étapes, de la conception initiale de l'application à son exécution finale.

En amont, tout le parallélisme potentiel de l'application doit être exhibé. Il s'agit alors d'utiliser des approches d'analyse et de transformations statiques (problématique 1) qui résultent en un code intermédiaire *multi-parallèle* informant la machine virtuelle d'exécution de tout le parallélisme qu'il est possible d'exploiter physiquement. Mais le compilateur n'a que peu de connaissance sur l'environnement d'exécution. Il connaît bien sûr le jeu d'instructions, peut connaître le nombre de coeurs, mais il ne sait pas quelles seront réellement les ressources disponibles à chaque instant de l'exécution (mémoire, nombre de coeurs).

C'est pourquoi un mécanisme de type "machine virtuelle" devra adapter l'application aux ressources (problématique 3). De plus, le compilateur ne pourra exploiter qu'une partie du parallélisme de l'application. En effet, certaines informations du programme (valeurs de variables, adresses mémoire accédées, etc.) n'étant accessibles qu'à l'exécution, une autre partie du parallélisme devra être généré à la volée, lors de l'exécution, là aussi par un mécanisme dynamique.

Cette extraction de parallélisme à la volée passe par la construction de modèles de comportement spéculatifs (problématique 2), ces modèles servant ensuite à la génération de code parallèle spéculatif (problématique 3). Parmi les objectifs de la modélisation du comportement, on peut ajouter le *monitoring*, ou profilage, du comportement d'une version du programme. En effet, la complexité des architectures actuelles et à venir ne permet plus de garantir a priori un comportement optimal d'une version d'un programme. Un processus de monitoring permettra de sélectionner la meilleure parallélisation à la volée.

L'adoption de plus en plus répandue d'approches et de langages de type "objet" met en exergue le besoin de développer des outils propres pour la programmation des architectures multi-coeurs. Le formalisme objets et méthodes implique des schémas d'exécution particuliers, qui, au niveau du code binaire final, se traduisent par des schémas élémentaires très éloignés. La maîtrise du comportement à l'exécution est donc encore bien plus difficile. L'analyse et l'optimisation, qu'elles soient statiques ou dynamiques, doivent prendre en compte dès le départ cette distorsion entre spécification "objet" et code binaire final : comment se traduit la parallélisation d'objets ou de méthodes (problématique 4).

Notre projet repose sur la conception d'une chaîne de production d'une exécution efficace de l'application sur l'architecture multi-coeurs. Chacun des maillons de cette chaîne doit être vérifié formellement afin de garantir aussi bien la correction que l'efficacité. Plus précisément, il faut vérifier que le compilateur produit du code intermédaire correct et garantir que la machine virtuelle réalise bien l'exécution parallèle équivalente sémantiquement au programme source : toutes les transformations subies par l'application, soit statiquement par le compilateur, soit dynamiquement par la machine virtuelle, doivent préserver la sémantique initiale. Elles devront être prouvées et certifiées (problématique 5).

Relations industrielles et internationales

Relation internationales :

- Réseau d'Excellence Européen HiPEAC
- Universidad Politécnica de Madrid, Spain
- Barcelona Supercomputing Center, Spain
- Washington State University, USA
- Université de Batna, Algérie

Relations industrielles :

- Reservoir Labs, New York, USA
- STMicroelectronics, Grenoble-Crolles, France
- Kalray, Paris, France
- Intel, San Diego, CA, USA