

Compilation de modèles Lustre multipériodiques vers du code parallèle

Timothy Bourke et Marc Pouzet

18 décembre 2017

L'équipe PARKAS collabore avec Airbus et Safran sur un problème scientifiquement et industriel majeur : compiler un modèle SCADE pour une architecture multicœurs. L'objectif est de générer du code efficace, traçable, et offrant le même niveau de confiance sur son déterminisme et sa correction sémantique que dans le cas séquentiel. Le problème n'est pas simple, car le comportement temporel des architectures multicœurs modernes est très difficile à établir. Il existe cependant de nouvelles architectures dont le comportement temporel est un plus prévisible ; c'est le cas de l'architecture MPPA développée par Kalray. Le sujet est encore très ouvert et aucune solution ne s'est encore imposée.

Il y a deux questions scientifiques à aborder ici : (a) modéliser au niveau du programme source des systèmes multipériodiques, c'est-à-dire permettant d'exprimer des calculs rapides et lents, avec l'intuition que les calculs lents peuvent être réalisés en parallèle et se synchroniser moins souvent ; (b) revoir la compilation pour exploiter ces périodes et générer, à partir d'une estimation des temps de calcul des opérations du programme source, du code séquentiel et parallèle d'efficacité garantie.

Nous travaillons actuellement sur le sujet, en collaboration avec Airbus, Safran et le groupe Core d'ANSYS/Esterel-Technologies (qui développe le compilateur KCG de SCADE). Nous disposons d'une machine MPPA de Kalray pour réaliser des expériences.

La théorie des systèmes n-synchrones [1]¹, qui introduit des horloges périodiques dans un langage synchrone de manière conservatrice peut être reprise et adaptée pour répondre à la première question. Elle permet de définir des transformations du code source qui respectent la sémantique du source et améliorent l'efficacité du code séquentiel. Pour la deuxième question, on pourra s'appuyer sur l'analyse des dépendances du programme, à partir de l'article [2]² pour décomposer un nœud Lustre en tâches atomiques. Dans le stage, il faudra définir avec soin les étapes de compilation, les prouver correctes (à la main ou à l'aide d'un assistant de preuve). Le travail pourra être réalisé dans le cadre du compilateur Vélus ou sous la forme d'un prototype, en OCaml.

Ce stage s'adresse à un étudiant ayant un goût prononcé pour le parallélisme et la compilation.

1. <https://www.di.ens.fr/~pouzet/talks/pop106.pdf>
2. <http://www.di.ens.fr/~pouzet/bib/jdaes10.pdf>

Références

- [1] A. Cohen, M. Duranton, C. Eisenbeis, C. Pagetti, F. Plateau, and M. Pouzet. N-synchronous Kahn networks : a relaxed model of synchrony for real-time systems. In *Proceedings of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2006)*, pages 180–193, Charleston, SC, USA, Jan. 2006. ACM, ACM Press.
- [2] M. Pouzet and P. Raymond. Modular static scheduling of synchronous data-flow networks : An efficient symbolic representation. In *Proceedings of the 9th ACM International Conference on Embedded Software (EMSOFT 2009)*, pages 215–224, Grenoble, France, Oct. 2009. ACM Press.