HABILITATION À DIRIGER DES RECHERCHES
présentée devant
l'École Normale Supérieure de Lyon

Spécialité : Informatique

Par :
Christophe ALIAS

Contributions to Program Optimization and High-Level Synthesis

---

Devant le jury composé de :

Corinne Ancourt, maître de recherche, MINES ParisTech, France	Rapporteur
Isabelle Guérin-Lassous, professor, University of Lyon 1	Examinateur
Sebastian Hack, professor, University of Saarland, Germany	Rapporteur
Michelle Strout, professor, University of Arizona, USA	Examinateur
Jürgen Teich, professor, University FAU of Erlangen-Nürnberg, Germany	Rapporteur

Summary

Since the end of Dennard scaling, power efficiency is the limiting factor for large-scale computing. Hardware accelerators such as reconfigurable circuits (FPGA, CGRA) or Graphics Processing Units (GPUs) were introduced to improve the performance under a limited energy budget, resulting into complex heterogeneous platforms. This document presents a synthetic description of my research activities since 10 years on compilers for high-performance computing and high-level synthesis of circuits (HLS) for FPGA accelerators. Specifically, my contributions covers both theoretical and practical aspects of automatic parallelization and HLS in a general theoretical framework called the *polyhedral model*.

A first chapter describes our contributions to *loop tiling*, a key program transformation for automatic parallelization which splits the computation atomic blocks called *tiles*. We rephrase loop tiling in the polyhedral model to enable any polyhedral tile shape whose size depends on a single parameter (monoparametric tiling), and we present a tiling transformation for programs with *reductions* – accumulations w.r.t. an associative/commutative operator. Our results open the way for *semantic program transformations*; program transformations which does not preserve the computation but still lead to an equivalent program.

A second chapter describes our contributions to *algorithm recognition*. A compiler optimization will never replace a good algorithm, hence the idea to recognize algorithm instances in a program and to substitute them by a call to a performance library. In our PhD thesis, we have addressed the recognition of templates – functions with first-order variables – into programs and its application to program optimization. We propose a complementary algorithm recognition framework which leverages our monoparametric tiling and our reduction tiling transformations. This automates *semantic tiling*, a new semantic program transformation which increases the grain of operators (scalar → matrix).

A third chapter presents our contributions to the *synthesis of communications with an off-chip memory* in the context of high-level circuit synthesis (HLS). We propose an execution model based on loop tiling, a pipelined architecture and a source-level compilation algorithm which, connected to the C2H HLS tool from Altera, ends up to a FPGA configuration with minimized data transfers. Our compilation algorithm is optimal – the data are loaded as late as possible and stored as soon as possible with a maximal reuse.

A fourth chapter presents our contributions to design a *unified polyhedral compilation model for high-level circuit synthesis*. We present the Data-aware Process Networks (DPN), a dataflow intermediate representation which leverages the ideas developed in chapter 3 to explicit the data transfers with an off-chip memory. We propose an algorithm to compile a DPN from a sequential program, and we present our contribution to the synthesis of DPN to a circuit. In particular, we present our algorithms to compile the control, the channels and the synchronizations of a DPN. These results are used in the production compiler of the Xtremlogic start-up.

Résumé

Depuis la fin du *Dennard scaling*, l'efficacité énergétique est le facteur limitant pour le calcul haute performance. Les accélérateurs matériels comme les circuits reconfigurables (FPGA, CGRA) ou les accélérateurs graphiques (GPUs) ont été introduits pour améliorer les performances sous un budget énergétique limité, menant à des plateformes hétérogènes complexes. Ce document présente une description synthétique de mes recherches depuis 10 ans sur les compilateurs et la synthèse haut-niveau pour les circuits FPGA (High-Level Synthesis, HLS) pour le calcul haute-performance. Specifiquement, mes contributions couvrent les aspects théoriques et pratiques de la parallélisation automatique et la HLS dans le cadre général du *modèle polyédrique*.

Un premier chapitre décrit mes contributions au *tuilage de boucles*, une transformation fondamentale pour la parallélisation automatique, qui découpe le calcul en sous-calculs atomiques appelés *tuiles*. Nous reformulons le tuilage de boucles dans le modèle polyédrique pour permettre n'importe tuile polytopique dont la taille dépend d'un facteur homothétique (tuilage monoparamétrique), et nous décrivons une transformation de tuilage pour des programmes avec des *réductions* – une accumulation selon un opérateur associative et commutatif. Nos résultats ouvrent la voie à des *transformations de programme sémantiques*; qui ne préservent pas le calcul, mais mènent à un programme équivaent.

Un second chapitre décrit mes contributions à la *reconnaissance d'algorithmes*. Une optimisation de compilateur ne remplacera jamais un bon algorithme, d'où l'idée de reconnaître les instances d'un algorithme dans un programme et de les substituer par un appel vers une bibliothèque haute-performance, chaque fois que c'est possible et utile. Dans notre thèse, nous avons traité la reconnaissance de *templates* – des fonctions avec des variables d'ordre 1 – dans un programme et son application à l'optimisation de programes. Nous proposons une approche complémentaire qui s'appuie sur notre tuilage monoparamétrique complété par une transforamtion pour tuiler les réductions. Ceci automatise le *tuilage sémantique*, une nouvelle transformation sémantique qui augmente le grain des opérateurs (scalaire → matrice).

Un troisième chapitre présente mes contributions à la *synthèse des communications avec une mémoire off-chip* dans le contexte de la synthèse de circuits haut-niveau. Nous proposons un modèle d'exécution basé sur le tuilage de boucles, une architecture pipelinée et un algorithme de compilation source-à-source qui, connecté à l'outil de HLS C2H d'Altera, produit une configuration de circuit FPGA qui réalise un volume minimal de transferts de données. Notre algorithme est optimal – les données sont chargées le plus tard possible et stockées le plus tôt possible, avec une réutilisation maximale et sans redondances.

Enfin, un quatrième chapitre présente mes contributions pour construire un *modèle de compilation polyédrique unifié pour la synthèse de circuits haut-niveau*. Nous présentons les réseaux de processus DPN (Data-aware Process Networks), une représentation intermédiaire dataflow qui s'appuie sur les idées développées au chapitre 3 pour expliciter les transferts de données entre le circuit et la mémoire *off-chip*. Nous proposons une suite d'algorithmes pour compiler un DPN à partir d'un programme séquentiel et nous présentons nos contributions à la synthèse d'un DPN en circuit. En particulier, nous présentons nos algorithmes pour compiler le contrôle, les canaux et les synchronisations d'un DPN. Ces résultats sont utilisés dans le compilateur de production de la start-up Xtrem-Logic.