Proposition de stage de M2

Scalable Translation Validation for High-Performance Computing and Machine Learning

Advisor: Christophe Alias (Inria Lyon, ENS de Lyon)
contact: Christophe.Alias@ inria.fr.

Place: ENS de Lyon

Stipend: ≈ 500 euros/mois

PhD funding available / *Possibilité de poursuite en thèse*

Context

When a program transformation is *not trusted*, we may want to enforce the correctness by checking *automatically* the equivalence of the source program and the target program. In general, this problem is well known to be undecidable. However, on polyhedral programs there exists heuristics [2, 1] which implicitly compute a bisimulation between the source and the target and *may* conclude. However, these heuristics lack of scalability and the equivalence relation is too narrow. We plan to develop *scalable* program equivalence heuristics, able to handle interesting program transformation like *reduction transformations* or *data transformations.* A PhD thesis will start the next year on this topic under the Mlopt ANR PRCI funding.

Objectives

The overall objective of this M2 intership is to investigate translation validation of programs with reductions and compile-time data allocation. The following point may be addressed:

- **Verifying reductions.** Many reduction transformation exist (factorization, semantic tiling, reduction parallelization). How to formalize them in unified way? How to support the composition with loop transformations? How that formalization might be produced by the compiler? Finally, how to check it in a scalable way? The polyhedral model provides a formalization of some of these transformations which enables solver-based checking. A reduction-compliant extension could be investigated.

- **Verifying data allocation.** The same questions arise for compile-time data allocation required by automatic parallelization (array privatisation, array contraction, struct/array permutation, etc) and will be investigated as well. In particular the framework of linear intra-array allocation and affine inter-array allocation could help to find a relevant formulation.

- **Scalability.** If possible, a first direct solver approach will be proposed for simple cases. Then, the scalability will be addressed to handle real-life HPC programs. How to parallelize the whole process? How to reduce the overall complexity? A trace-based solution could also be investigated.
- **Validation.** The first results will be prototyped and validated on HPC benchmarks.

This internship might be followed by a PhD thesis (funded), depending on the performances and the motivation.

Prerequisites

Technical skills required : Notions in compilers, parallelism and program analysis. Experience with C++.

References

[1] Christophe Alias and Denis Barthou. On the recognition of algorithm templates. In *International Workshop on Compiler Optimization meets Compiler Verification (COCV'03)*, 2003.

[2] Denis Barthou, Paul Feautrier, and Xavier Redon. On the Equivalence of Two Systems of Affine Recurrence Equations (Research Note). In *Proceedings of the 8th International Euro-Par Conference on Parallel Processing*, pages 309–313, 2002.