intermediate representation adapted to circuit synthesis. In the tools developed by CASH, this
Translating a program into a circuit is done in several steps. First, the front-end generates an
algorithm into a circuit design. This operation is called HLS (High-level synthesis).

new technological locks must be overcome. Among them is the automatic and efficient translation of
reconfigurable circuits (FPGA, Field Programmable Gate Array) [2]. To execute an application on FPGA,
one solution is to implement the main algorithms of a computation in hardware, and map it to
computation units (processor cores, specialized processors). New programming paradigms have to be
found to continue increasing performance in a given energy budget.

Since the end of Dennard scaling, energy efficiency is becoming a major bottleneck for
supercomputers [1]. Most gains in performance now come from the augmentation of the number of
computation units (processor cores, specialized processors). New programming paradigms have to be
found to continue increasing performance in a given energy budget.

One solution is to implement the main algorithms of a computation in hardware, and map it to
reconfigurable circuits (FPGA Field Programmable Gate Array) [2]. To execute an application on FPGA,
new technological locks must be overcome. Among them is the automatic and efficient translation of
an algorithm into a circuit design. This operation is called HLS (High-level synthesis).

Translating a program into a circuit is done in several steps. First, the front-end generates an
intermediate representation adapted to circuit synthesis. In the tools developed by CASH, this
formalism is called “Data-aware Process Network” (DPN) [3] and represents a network of processes that captures the parallelism of an application and the communications between parallel processes. Then, the back-end translates each component of the process network into hardware while ensuring a good reuse of hardware resources. In the end, the circuit can be seen as a very large network of pipelined processes, reading inputs and producing outputs periodically.

The newly created CASH team works on novel approaches for automatic parallelization with a strong focus on hardware compilation for FPGA. CASH focuses on the polyhedral model [4], a sound framework to develop parallelization algorithms from imperative programs.

Principales activités

The overall objective of the thesis is to propose models and compiler algorithms to optimize the throughput of DPN process networks under resource constraints. The first step is to model the latencies in a DPN process network and to instrument DPN process networks to measure dynamically the throughput. A SystemC model will be generated automatically from our DPN compiler. Experiments will be conducted on the PolyBench kernels [5] to identify the bottlenecks, trade-offs and possible optimizations.

The particular structure of DPN (static control, affine array accesses) makes it possible to build on polyhedral scheduling techniques. For example, we expect a combination of loop tiling/affine scheduling (such as [6]) and process splitting to give interesting results. Resource constraints (e.g. number of processes/channels) must be also be modeled and taken into account.

Another direction is to study theoretical throughput bounds on polyhedral kernels. How is the throughput related to the DPN process network structure? What is the maximum throughput attainable among all the process network realizations of a polyhedral kernel? Can we derive statically this throughput bound? We also expect this study to give insights for the throughput optimization itself.

References:


Supervisors:

This thesis will be supervised by Christophe Alias (Inria Researcher, ENS-Lyon) and Matthieu Moy (Assistant professor, HDR, UCBL).

Christophe Alias (http://perso.ens-lyon.fr/christophe.alias)’s research interests includes automatic parallelization (in the polyhedral model), static analysis and high-level synthesis for FPGA circuits. He wrote the DPN compiler which serve as a basis for this PhD thesis.

Matthieu Moy (https://matthieu-moy.fr)’s main research area is hardware simulation (using SystemC) and formal verification (model-checking, abstract interpretation, SMT solving). More recently, he started working on worst-case execution time for software and worst-case traversal time for networks-on-chip, and compilation for critical systems. He joined the LIP laboratory in 2017 and started working on HLS and polyhedral methods.

Keywords:

Compilation, process networks, parallelism, throughput optimization, high-level synthesis.

Compétences

The candidate should have good background in compilation. A good knowledge of parallel programming is required. Prior experience with compiler/static analysis is obviously appreciated. The PhD consists in theoretical aspects and practical ones, hence the candidate should have both a good theoretical background and good programming skills.
Avantages sociaux

- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- Flexible working hours
- Sports facilities

Rémunération


Monthly salary after taxes: around 1596,05€ for 1st and 2nd year. 1678,99€ for 3rd year. (medical insurance included).