2018-00436 - Formal Verification of Process Networks as Compiler Intermediate Representation - ROMA - PHD Campaign - Campagne Doctorants Grenoble Rhône-Alpes

Contract type: Public service fixed-term contract
Level of qualifications required: Graduate degree or equivalent
Fonction: PhD Position

About the research centre or Inria department
Inria, the French national institute for research in computer science and control, is dedicated to fundamental and applied research in information and communication science and technology (ICST). Inria has a workforce of 3,800 people working throughout its eight research centers established in seven regions of France.

Grenoble is the capital city of the French Alps. Combining the urban lifestyle of southern France with a unique mountain setting, it is ideally situated for outdoor activities. The Grenoble area is today an important center of industry and science (second largest in France). Dedicated to an ambitious policy in the arts, the city is host to numerous cultural institutions. With 60,000 students (including 6,000 foreign students), Grenoble is the third largest student area in France.

Context
The CASH (Compilation and Analysis, Software and Hardware) group works on compilation techniques for high-performance computing. We are currently a team at the LIP laboratory (Lyon), and a sub-group of the ROMA team at Inria.

The overall objective of the CASH team is to take advantage of the characteristics of the specific hardware (generic hardware, hardware accelerators or FPGA) to compile energy efficient software and hardware. The long-term objective is to provide solutions for the end-user developers to use at their best the huge opportunities of these emerging platforms. The research directions of the team are:

* Dataflow models for HPC applications: We target representations that are expressive enough to express all kinds of parallelism and allow further optimizations.
* Compiler algorithms and tools for irregular applications: The extensions of these intermediate representations to enable complex control flow and complex data structures, and the design of associated analysis for optimized code generation for multicore processors and accelerators.
* Compiler Algorithms, Simulation and Tools for Reconfigurable Circuits: The application of the two preceding activities on High Level Synthesis, with additional resource constraints.
* Simulation of Systems on a Chip: A parallel and scalable simulation of Systems-on-Chips, which, combined with the preceding activity, will result in a complete workflow for circuit design.

Contact: http://www.ens-lyon.fr/LIP/CASH/
Matthieu Moy <Matthieu.Moy@univ-lyon1.fr>
https://matthieu-moy.fr/

Assignment
Scientific context:
In the beginning of the 2000’s, the clock frequency of computation units reached its limits. Energy-efficiency is becoming a major bottleneck for supercomputers [1]. Increasing the clock frequencies implies a loss of energy efficiency that is no longer acceptable. 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) 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 (http://www.ens-lyon.fr/LIP/CASH/) team works on novel approaches to extract parallelism from an imperative program to an intermediate representation. To evaluate the quality and correctness of the generated process network, one option is to execute them, either by simulating them or by running them through the back-end and execute the result on an FPGA. However, this would not provide strong correctness guarantees, just like testing software doesn't prove its correctness.

**Main activities**

Objectives of the Thesis:

The overall objective of the thesis is to explore the possibilities of formal verification of DPN (Data-aware Process Networks). The main technique we envision is model-checking, i.e. exploration of the state-space.

The first step is to connect to existing verification tools. A good candidate is the CADP toolchain (http://cadp.inria.fr/), developed at Inria by the CONVECS team. This connection can take the form of a compiler that reads DPNs and produces the input format for CADP (LOTOS, or some lower-level formalism like AUT or BCG).

Once a basic connection is done, we can start verifying properties on the generated automata, like absence of deadlocks, temporal logic properties, or equivalence with a reference automaton.

The particular structure of DPN (static control, lot of FIFO-based communication) should make model-checking easier. For example, we expect partial-order reduction techniques to be very efficient on DPN.

The next steps of the Ph.D will therefore be to apply reduction techniques to make the verification scale to large DPN. Many techniques are already implemented in model-checking tools like CADP, and new techniques may be proposed as part of the Ph.D.

Another direction to explore is the feedback we can get from verification, in particular when the verification fails. Typical model-checkers provide a counter-example (a trace violating the property) in this case, but we can use the information provided by the model-checker to get a better diagnosis (e.g. notice that the verification failed because of an incorrect channel dimensioning, or because the execution order is wrong) and ideally suggest a fix.

**Skills**

Expected skills:

The candidate should have good background in compilation. A good knowledge of parallel programming, both from the practical point of view and from the formal point of view (semantics, state-machine, ...) is required. Prior experience with model-checking is obviously appreciated. The Ph.D consists in theoretical aspects and practical ones, hence the candidate should have both a good theoretical background and good programming skills.

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, polyhedral compilation and high-level synthesis for FPGA circuits. He wrote a process-network compiler that he transfered to the Xtreemlogic startup.

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, model-checking, formal methods, parallelism, high-level synthesis.

Benefits package
- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- Flexible working hours

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