2018-00468 - Efficient, Parallel, Discrete-Event Simulation for Data-aware Process Networks using Polyhedral Methods - Post-Doctorant Inria Grenoble Research center

Contract type: Public service fixed-term contract  
Level of qualifications required: PhD or equivalent  
Fonction: Post-Doctoral Research Visit

About the research centre or Inria department

Grenoble Rhône-Alpes Research Center groups together a few less than 800 people in 35 research teams and 9 research support departments.

Staff is localized on 5 campuses in Grenoble and Lyon, in close collaboration with labs, research and higher education institutions in Grenoble and Lyon, but also with the economic players in these areas.

Present in the fields of software, high-performance computing, Internet of things, image and data, but also simulation in oceanography and biology, it participates at the best level of international scientific achievements and collaborations in both Europe and the rest of the world.

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 subgroup 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.

Assignment

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 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 would be to run the generated process network through the back-end and execute the result on an FPGA. However, the back-end and synthesis are time-consuming operations and running on an FPGA provides only limited debugging tools. The other option is to simulate the process network before the back-end. We currently use a minimal simulator based on POSIX threads, using one thread per process. This solution is operational but slow due to the number of context-switches required.

A new simulator will be developed during spring 2018. This new simulator will use the principles of discrete-event simulation. We plan to use the SystemC simulator for this. SystemC is the standard tool for high-level circuit synthesis. It has an efficient scheduler using a cooperative scheduling policy for which context-switches are efficient.

Main activities

Objectives of the post-doc:

We expect a significant gain in terms of performance from the SystemC-based simulator, but on the other hand, a basic implementation in SystemC cannot exploit the parallelism of the host machine (the simulator will be sequential). Several approaches have been proposed to run a SystemC simulation in parallel, each of them being specific to a coding style. A generic parallelization approach would miss a lot of optimization opportunities: our process networks have good properties for parallelization (lot of FIFO-based communication, static control, massive parallelism), and they are generated automatically using polyhedral methods. We can imagine a lot of possible optimizations, that are to be explored during the post-doc:

* Partition the simulator, using one SystemC instance per partition and running partitions in parallel, following e.g. the approach of Denis Becker's Ph.D [3].
* Automatically generate a partitioning that minimizes inter-partition communications and balances the load evenly between partitions.
* Exploit FIFO-based communication to optimize the communication and synchronization between partitions. For example, it is possible for different partitions to execute different simulated instants in parallel (in a sequential simulation, this is called "temporal decoupling" and has already be shown to work very well with FIFO [4]).

References:


[3] Denis Becker. Parallel SystemC/TLM Simulation of Hardware Components described for High-Level Synthesis

Fast and Accurate TLM Simulations using Temporal Decoupling for FIFO-based Communications Design, Automation and Test in Europe (DATE), 2013

Skills

The candidate should have good background in compilation (knowledge of polyhedral methods would obviously be appreciated), and should be familiar with parallel programming. Skills in discrete-event simulation in general and/or SystemC in particular are also appreciated. A good knowledge of C++ is necessary.

Benefits package

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

This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST). Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

Recruitment Policy:

As part of its diversity policy, all Inria positions are accessible to people with disabilities.

Warning: you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.
Remuneration

Gross salary: 2650 Euros per month