Compilation and Analyses, Software and Hardware
CASH Joint Inria Team proposal

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy

University Claude Bernard Lyon 1 / CNRS / ENS Lyon / Inria (LIP Laboratory)

11 décembre
Outline

Context

Research statement

Some Research Directions

Collaborations & Positioning

Conclusion
Who

- **Christophe Alias (CR Inria):**
  - high-level synthesis, compilation, polyhedral model.

- **Laure Gonnord (MCF Lyon 1):**
  - abstract interpretation, compilation, semantics.

- **Ludovic Henrio (CR CNRS):**
  - programming languages, actors, semantics.

- **Matthieu Moy (MCF Lyon 1):**
  - hardware simulation, many-core, dataflow languages.

- **Paul Iannetta (PhD student):**
  - abstract interpretation, polyhedral model.

- **Julien Braine (PhD student):**
  - program analysis, array properties verification.
High-Performance Computing: Growing Challenges

- Power-efficiency
  - New kind of accelerators (CPU → GPU → FPGA)

- Data movement = bottleneck (memory wall)
  - Optimize communication and computation

- Programming model: efficient hardware/software implementations
  - Express or extract efficient parallelism

- Optimized (software/hardware) compilation for HPC software with data-intensive computations
Our “end-users”

Gas prospector

Application developer

Compute kernel

for \( i := 1 \) to \( N - 2 \)
for \( j := 1 \) to \( N - 2 \)

\[
G_x := (C[i+2,j+1]+C[i+2,j]+C[i+2,j+2]) - (C[i,j+1]+C[i,j]+C[i,j+2])
\]

\[
G_y := (C[i+1,j+2]+C[i,j+2]+C[i+2,j+2]) - (C[i+1,j]+C[i,j]+C[i+2,j])
\]

\[
B[i,j] := \sqrt{(2G_x)^2 + (2G_y)^2}
\]

CASH

Target Machine
Power-efficiency and FPGA

FPGA ≈ dedicated hardware, but reconfigurable

Best power-efficiency without FPGA ≈ 14.6 GFlops/W
(NVIDIA Volta GV100 GPU + IBM Power9)

≈ 2006  •  end of Dennard scaling ⇒ no more free lunch with energy efficiency!

2015  •  Microsoft achieves 40 GFlops/W with 500,000 FPGA

2015  •  Intel acquires Altera

2017  •  Intel begins shipping Xeon Phi with integrated FPGA

2018  •  Dell and Fujitsu use FPGAs in servers (+ Intel FPGA SDK for OpenCL)

⇝ How to program FPGA?

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy
High-Level Synthesis (HLS)

1990’s
- VHDL/Verilog are the only way to produce hardware

2000’s
- Early steps of High-Level Synthesis (HLS):
  - Focus on computation, not communication
  - Marginal raise of abstraction level, semantics unclear

2010
- Better input languages and interfaces. Still not adopted by circuit designers.

2015
- FPGA become a credible building block for HPC. Industry is now pushing HLS technologies!

FPGA + HLS = best of software and hardware?
Outline

Context

Research statement

Some Research Directions

Collaborations & Positioning

Conclusion
Dataflow and Parallelism

Credo: **dataflow** is a good model to handle complex HPC applications:

- All the available parallelism is expressed
- Natural intermediate language for an HPC compiler (compile to/from dataflow program representations)
- Suitable for static analysis of parallel systems (correctness, throughput, etc.)

⇝ Dataflow = transverse and fundamental topic of CASH.
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. Compiling and Scheduling Dataflow Programs
4. HLS-specific Dataflow Optimizations
5. Simulation of Hardware
Application domain

- HPC (solvers, stencils) & big data (deep learning)
- Typical applications heavily use linear algebra kernels (matrix & tensor operations)
- Examples applications using FPGA
  - HPC: oil & gas prospecting (ex: Chevron, system running on FPGA)
  - Big Data: Torch scientific computing framework (Facebook, already has an FPGA backend)
Outline

Context

Research statement

Some Research Directions

Collaborations & Positioning

Conclusion
CASH: Compilation and Analysis, Software and Hardware

1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. Compiling and Scheduling Dataflow Programs
4. HLS-specific Dataflow Optimizations
5. Simulation of Hardware

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. **Compiling and Scheduling Dataflow Programs**
4. HLS-specific Dataflow Optimizations
5. Simulation of Hardware
Compiling & Scheduling Dataflow Programs (1/2)

Dataflow program

Parallel Machine

Formal Verification

Parametrization Dev. Interaction

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy
**Locks:**

- Different levels of granularities that do not coexist well.
- What’s the frontier between static and dynamic?
- Many syntax-based optimisations.
Medium-term:

▶ Express compilation/analysis activities for the dataflow model.
▶ Understand the impact of local parallelism optimization on global performance

Long-term:

▶ Unify several kinds of parallelism in a same formal semantic framework.

Experiments with SigmaC

Experience on concurrent programming languages, dataflow synchronization, semantics.
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. **Compiling and Scheduling Dataflow Programs**
4. HLS-specific Dataflow Optimizations
5. Simulation of Hardware

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. Compiling and Scheduling Dataflow Programs
4. **HLS**-specific Dataflow Optimizations
5. Simulation of Hardware
HLS-specific Dataflow Optimizations (1/2)

- Kernel
- Process Network
- FPGA IP
- C-to-Dataflow
- HLS tool
- Abstraction
- Automatic Parallelization
- Dataflow Optimization
- Simulation
- Cost Model

Locations:
- Lack of clear semantics: mixed sequential/dataflow
- HLS optimizations are too conservative
- The polyhedral model targets sequential code (not dataflow)

Christophe Alias, Laure Gonnord, Ludovic Henrio, Matthieu Moy
Locks:

- Lack of clear semantics: mixed sequential/dataflow
- HLS optimizations are too conservative
- The polyhedral model targets sequential code (not dataflow)
HLS-specific Dataflow Optimizations (2/2)

**Short/Medium term:**
- **Dataflow-to-HLS code generator**
  start with the DPN model (used by XtremLogic)
- Factor channels and control
- Dataflow optimization for throughput
  solved for a single process [MICPRO 2012]

**Long term:**
- Models and algorithms for data movement minimization
  [PhD Plesco 2010]
- Parametrization for scaling parallelization
  Parametric tiling [PhD Iooss 2016]
- Hardware synthesis for dynamic control/data
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. Compiling and Scheduling Dataflow Programs
4. **HLS-specific Dataflow Optimizations**
5. Simulation of Hardware
1. Definition of dataflow representations of parallel programs
2. Expressivity and Scalability of Static Analyses
3. Compiling and Scheduling Dataflow Programs
4. HLS-specific Dataflow Optimizations
5. **Simulation of Hardware**
Simulation of Hardware (1/2)

Other simulator

Physical Environment (real or model)

Other System

Not yet implemented

Power/Temperature Model

In parallel!
Simulation of Hardware (1/2)

Other simulator

Physical Environment (real or model)

Locks:
- Heterogeneous simulation (functional, physics, ...)
- Scale up (parallelism)

Power/Temperature Model

In parallel!
Simulation of Hardware (2/2)

Short/Medium-term:
- Work with CEA-LIST and LIP6 on convergence of approaches
  
  ANR Project submitted

- Deal with loose information (intervals instead of individual values for physics)

Long-term:
- Framework for parallel and heterogeneous simulation: simulation backbone and adapters
  
  [PhD Becker 2017]
Outline

Context

Research statement

Some Research Directions

Collaborations & Positioning

Conclusion
Main Collaborations

CEA/CITI (Lionel Morel)  Compilation and scheduling, polyhedral model (coadvising P. Iannetta)

CEA-LIST (Tanguy Sassolas)  Simulation of System-on-a-Chip

Colorado State University (Sanjay Rajopadhye)  Automatic parallelization, polyhedral model

Oslo, Uppsala, Darmstadt  Semantics & typing of concurrent languages

Verimag/PACSS (David Monniaux)  Proving correction of programs with arrays (coadvising J. Braine)

STMicroelectronics  Simulation of hardware

Xtremlogic startup  High-level synthesis
Positioning

- CASH = Only compilation-centered team in Lyon
- France: compilation (CORSE, ...), analysis (ANTIQUE, ...), HLS (CAIRN, ...). Particularities of CASH:
  - Emphasis on static aspects
  - Static analysis for compilation
- International:
  - HPC: High-level languages (PELAB, Linköping; programming languages, Uppsala; ...)
  - HLS: VAST, California; System group, London; ...
  - Static analysis: Automatic verification, Oxford; ...
  - Dataflow: Compaan, Netherland; ...
  - Simulation: Rolf Drechsler, Bremen; ...

(Details on positioning in the long document)
Outline

Context

Research statement

Some Research Directions

Collaborations & Positioning

Conclusion
Summary

▶ Ever-growing level of **parallelism** for software implementations:
  ▶ Strong interest for reconfigurable circuits (FPGA) and high-level synthesis (**HLS**) in HPC
  ▶ Need for new programming models and techniques to **analyze** and **optimize** programs for parallel architectures (many-core, GPU, ...)

▶ **Synergies:**
  ▶ Abstract interpretation ↔ compilation ↔ dataflow
  ▶ Compilation ↔ Hardware (FPGA)
  ▶ Theory ↔ Practice

▶ Industrial partnerships: STMicroelectronics (simulation), Kalray (many-core), XtremLogic (HLS)

▶ Fertile context: LIP + Inria + “Fédération Informatique de Lyon”: HPC and theory (AriC/Avalon/Plume/Roma)
Summary

- Ever-growing level of **parallelism** for software implementations:
  - Strong interest for reconfigurable circuits (FPGA) and high-level synthesis (**HLS**) in HPC
  - Need for new programming models and techniques to **analyze** and **optimize** programs for parallel architectures (many-core, GPU, ...)

- **Synergies:**
  - Abstract interpretation ↔ compilation ↔ dataflow
  - Compilation ↔ Hardware (FPGA)
  - Theory ↔ Practice

- **Industrial partnerships:** STMicroelectronics (simulation), Kalray (many-core), XtremLogic (HLS)

- **Fertile context:** LIP + Inria + “Fédération Informatique de Lyon”: HPC and theory (AriC/Avalon/Plume/Roma)

Thanks! Questions?
Outline

Details on Positioning
Related teams in Lyon

- **Within LIP:**
  - **Avalon**: same application domain (HPC). Avalon targets application-level programming models, we target compute kernels.
  - **AriC**: arithmetic operators, float to fix point transformation: could be integrated into an HLS flow.
  - **Plume**: dataflow semantics, abstract interpretation, parallel languages semantics and verification
  - **Roma**: scheduling and resource allocation for I/O, throughput and energy, I/O models for FPGA

- **CITI:**
  - **SOCRATE**: programming models for software defined radio, simulation of SoCs

- **LIRIS:**
  - **Beagle** (modeling, simulations): potential case-studies
Inria teams in Grenoble

- **CORSE**: Static vs Dynamic compilation
- **CTRL-A & SPADES**: formal methods, components.
- **DATAMOVE**: data management for HPC.
- **CONVECS**: languages for concurrent systems.
Other Inria teams

- Compilation, scheduling, HLS:
  - **CAIRN**: HLS for FPGA & polyhedral model
  - **CAMUS**: Compilation, parallelism, polyhedral model (static + dynamic)
  - **PACAP**: Dynamic compilation and scheduling, embedded systems
  - **PARKAS**: Compilation of dataflow programs for embedded systems, deterministic parallelism

- Abstract Interpretation:
  - **ANTIQUÉ**: Abstract interpretation, data-structures, verification.
  - **CELTIQUE**: Abstract interpretation, decision procedures and interactive proofs