IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Paris

Activity Report 2015

Project-Team PARKAS

Parallélisme de Kahn Synchrone

IN COLLABORATION WITH: Département d'Informatique de l'Ecole Normale Supérieure

RESEARCH CENTER
Paris - Rocquencourt

THEME
Embedded and Real-time Systems

Table of contents

Project-Team PARKAS

Creation of the Team: 2011 April 01, updated into Project-Team: 2012 January 01

Keywords:

Computer Science and Digital Science:

1.1.3. - Memory models
2.1.1. - Semantics of programming languages
2.1.3. - Functional programming
2.1.6. - Concurrent programming
2.1.8. - Synchronous languages
2.2.2. - Memory models
2.2.3. - Run-time systems
2.2.4. - Parallel architectures
2.2.5. - GPGPU, FPGA, etc.
2.2.6. - Adaptive compilation
2.3.1. - Embedded systems
2.3.2. - Cyber-physical systems
2.4.3. - Proofs
3.1.3. - Distributed data
3.1.8. - Big data (production, storage, transfer)
6.2.1. - Numerical analysis of PDE and ODE
6.2.7. - High performance computing

Other Research Topics and Application Domains:

5.2.1. - Road vehicles
5.2.2. - Railway
5.2.3. - Aviation
6.4. - Internet of things
6.6. - Embedded systems
9.2.1. - Music, sound
9.4.1. - Computer science
9.4.2. - Mathematics

1. Members

Research Scientists

Albert Cohen [Inria, Senior Researcher, HdR]
Francesco Zappa Nardelli [Inria, Researcher, HdR]
Timothy Bourke [Inria, Starting Research position]

Faculty Members

Marc Pouzet [Team leader, UPMC, Professor, HdR]
Jean Vuillemin [ENS, Professor, HdR]

Engineers

Jean-Baptiste Bréjon [Inria, Engineer, from Oct 2015]

Basile Clément [Inria, until Apr 2015]
Kirill Kononenko [Inria, Apr 2015 to Dec 2015]
Michael Kruse [Inria, Postdoctoral Researcher]
Mircea Namolaru [Inria, Engineer]
Laurent Morin [Inria, External collaborator from Rennes]
Zhen Zhang [Inria, Postdoctoral Researcher]

PhD Students

Riyadh Baghdadi [UPMC and Inria, until Sep 2015]
Guillaume Baudart [ENS Paris]
Andi Drebes [Inria, until Feb 2015]
Adrien Guatto [ENS Paris]
Nhat Minh Lê [ENS Paris and Inria]
Ivan Llopard [UPMC and CEA LETI, until Nov 2015]
Robin Morisset [ENS Paris and Inria, granted by Google]
Jie Zhao [ENS Paris and Chinese scholarship from NDSC, from April 2015]
Ulysse Beaugnon [ENS Paris, from September 2015]
Chandan Reddy Gopal [ENS Paris and Inria, from September 2015]

Administrative Assistant

Anna Bednarik [Inria]

Others

Virgile Andreani [ENS Paris, Master Intern (MPRI), from Mar 2015 - Sep 2015]
Tommaso Borghesi [Inria, Intern from U. Perugia, from Dec 2015]
Abhishek Jain [Inria, Intern from IIT Delhi, until Jan 2015]
Valentin Lorentz [Inria, Master I, Intern (Ecole normale supérieure de Lyon), from April 2015 to July 2015]
Chaitanya Malaviya [Inria, Intern, from Jun 2015 until Aug 2015]
Leo Testard [Inria, Master Intern (UPMC), from June 2015 to Jul 2015]
Sven Verdoolaege [Independent contractor, Polly Labs grant of ARM]

2. Overall Objectives

2.1. Overall Objectives

The goal of the project is the design, semantics and compilation of languages for the implementation of provably safe and efficient computing systems. We are driven by the ideal of a unique source code used both to *program* and *simulate* a wide variety of systems, including (1) embedded real-time controllers (e.g., fly-by-wire, engine control); (2) computationally intensive applications (e.g., video); (3) the simulation of (a possibly huge number of) embedded systems in close interaction (e.g., simulation of electrical or sensor networks, train tracking). All these applications share the need for formally defined languages used both for simulation and the generation of target code. For that purpose, we design languages and experiment with compilers that transform mathematical specifications of systems into target code, that may execute on parallel (multi-core) architectures.

Our research team draws inspiration and focus from the simplicity and complementarity of the data-flow model of Kahn process networks, synchronous concurrency, and the expression of the two in functional languages. To reach our goal, we plan to leverage a large body of formal principles: language design, semantics, type theory, concurrency models (including recent works on the formalisation of relaxed memory models), synchronous circuits and algorithms (code generation, optimization, polyhedral compilation).

3. Research Program

3.1. Presentation and originality of the PARKAS team

Our project is founded on our expertise in three complementary domains: (1) synchronous functional programming and its extensions to deal with features such as communication with bounded buffers and dynamic process creation; (2) mathematical models for synchronous circuits; (3) compilation techniques for synchronous languages and optimizing/parallelizing compilers.

A strong point of the team is its experience and investment in the development of languages and compilers. Members of the team also have direct collaborations for several years with major industrial companies in the field and several of our results are integrated in successful products. Our main results are briefly summarized below.

3.1.1. Synchronous functional programming

In [33], Paul Caspi and Marc Pouzet introduced *synchronous Kahn networks* as those Kahn networks that can be statically scheduled and executed with bounded buffers. This was the origin of the language LUCID SYNCHRONE, [1] [2] an ML extension of the synchronous language LUSTRE with higher-order features, dedicated type systems (clock calculus as a type system [33], [44], initialization analysis [45] and causality analysis [46]). The language integrates original features that are not found in other synchronous languages: such as combinations of data flow, control flow, hierarchical automata and signals [43], [42], and modular code generation [34], [31].

In 2000, Marc Pouzet started to collaborate with the SCADE team of Esterel-Technologies on the design of a new version of SCADE. [3] Several features of LUCID SYNCHRONE are now integrated into SCADE 6, which has been distributed since 2008, including the programming constructs `merge`, `reset`, the clock calculus and the type system. Several results have been developed jointly with Jean-Louis Colaço and Bruno Pagano from Esterel-Technologies, such as ways of combining data-flow and hierarchical automata, and techniques for their compilation, initialization analysis, etc.

Dassault-Systèmes (Grenoble R&D center, part of Delmia-automation) developed the language LCM, a variant of LUCID SYNCHRONE that is used for the simulation of factories. LCM follows closely the principles and programming constructs of LUCID SYNCHRONE (higher-order, type inference, mix of data-flow and hierarchical automata). The team in Grenoble is integrating this development into a new compiler for the language Modelica. [4]

In parallel, the goal of REACTIVEML[5] was to integrate a synchronous concurrency model into an existing ML language, with no restrictions on expressiveness, so as to program a large class of reactive systems, including efficient simulations of millions of communicating processes (e.g., sensor networks), video games with many interactions, physical simulations, etc. For such applications, the synchronous model simplifies system design and implementation, but the expressiveness of the algorithmic part of the language is just as essential, as is the ability to create or stop a process dynamically.

The development of REACTIVEML was started by Louis Mandel during his PhD thesis [57], [55] and is ongoing. The language extends OCAML[6] with Esterel-like synchronous primitives — synchronous composition, broadcast communication, pre-emption/suspension — applying the solution of Boussinot [32] to solve causality issues.

[1] http://www.di.ens.fr/~pouzet/lucid-synchrone
[2] The name is a reference to Lustre which stands for "Lucid Synchrone et Temps réel".
[3] http://www.esterel-technologies.com/products/scade-suite/
[4] http://www.3ds.com/products/catia/portfolio/dymola/overview/
[5] http://rml.lri.fr/
[6] More precisely a subset of OCAML without objects or functors.

Several open problems have been solved by Louis Mandel: the interaction between ML features (higher-order) and reactive constructs with a proper type system; efficient simulation that avoids busy waiting. The latter problem is particularly difficult in synchronous languages because of possible reactions to the absence of a signal. In the REACTIVEML implementation, there is no busy waiting: inactive processes have no impact on the overall performance. It turns out that this enables REACTIVEML to simulate millions of (logical) parallel processes and to compete with the best event-driven simulators [58].

REACTIVEML has been used for simulating routing protocols in ad-hoc networks [54] and large scale sensor networks [69]. The designer benefits from a real programming language that gives precise control of the level of simulation (e.g., each network layer up to the MAC layer) and programs can be connected to models of the physical environment programmed with LUTIN [68]. REACTIVEML is used since 2006 by the synchronous team at VERIMAG, Grenoble (in collaboration with France-Telecom) for the development of low-consumption routing protocols in sensor networks.

3.1.2. Relaxing synchrony with buffer communication

In the data-flow synchronous model, the clock calculus is a static analysis that ensures execution in bounded memory. It checks that the values produced by a node are instantaneously consumed by connected nodes (synchronous constraint). To program Kahn process networks with bounded buffers (as in video applications), it is thus necessary to explicitly place nodes that implement buffers. The buffers sizes and the clocks at which data must be read or written have to be computed manually. In practice, it is done with simulation or successive tries and errors. This task is difficult and error prone. The aim of the n-synchronous model is to automatically compute at compile time these values while insuring the absence of deadlock.

Technically, it allows processes to be composed whenever they can be synchronized through a bounded buffer [35], [36]. The new flexibility is obtained by relaxing the clock calculus by replacing the equality of clocks by a sub-typing rule. The result is a more expressive language which still offers the same guarantees as the original. The first version of the model was based on clocks represented as ultimately periodic binary words [75]. It was algorithmically expensive and limited to periodic systems. In [40], an abstraction mechanism is proposed which permits direct reasoning on sets of clocks that are defined as a rational slope and two shifts. An implementation of the n-synchronous model, named LUCY-N, was developed in 2009 [56], as was a formalization of the theory in COQ [41]. We also worked on low-level compiler and runtime support to parallelize the execution of relaxed synchronous systems, proposing a portable intermediate language and runtime library called ERBIUM [59].

This work started as a collaboration between Marc Pouzet (LIP6, Paris, then LRI and Inria Proval, Orsay), Marc Duranton (Philips Research then NXP, Eindhoven), Albert Cohen (Inria Alchemy, Orsay) and Christine Eisenbeis (Inria Alchemy, Orsay) on the real-time programming of video stream applications in set-top boxes. It was significantly extended by Louis Mandel and Florence Plateau during her PhD thesis [63] (supervised by Marc Pouzet and Louis Mandel). Low-level support has been investigated with Cupertino Miranda, Philippe Dumont (Inria Alchemy, Orsay), Antoniu Pop (Mines ParisTech). Further directions of research and experimentation have been and are being followed through the theses of Léonard Gérard (defended in 2013), Adrien Guatto (defended in January 2016) and Nhat Minh Lê.

3.1.3. Polyhedral compilation and optimizing compilers

Despite decades of progress, the best parallelizing and optimizing compilers still fail to extract parallelism and to perform the necessary optimizations to harness multi-core processors and their complex memory hierarchies. *Polyhedral compilation* aims at facilitating the construction of more effective optimization and parallelization algorithms. It captures the flow of data between individual instances of statements in a loop nest, allowing to accurately model the behavior of the program and represent complex parallelizing and optimizing transformations. Affine multidimensional scheduling is one of the main tools in polyhedral compilation [47]. Albert Cohen, in collaboration with Cédric Bastoul, Sylvain Girbal, Nicolas Vasilache, Louis-Noël Pouchet and Konrad Trifunovic (LRI and Inria Alchemy, Orsay) has initiated to a large number of research, development and transfer activities in this area.

The relation between polyhedral compilation and data-flow synchrony has been identified through data-flow array languages [53], [52], [70], [48] and the study of the scheduling and mapping algorithms for these languages. We would like to deepen the exploration of this link, embedding polyhedral techniques into the compilation flow of data-flow, relaxed synchronous languages.

Our previous work led to the design of a theoretical and algorithmic framework rooted in the polyhedral model of compilation, and to the implementation of a set of tools based on production compilers (Open64, GCC) and source-to-source prototypes (PoCC, http://pocc.sourceforge.net). We have shown that not only does this framework simplify the problem of building complex loop nest optimizations, but also that it scales to real-world benchmarks [37], [49], [66], [65]. The polyhedral model has finally evolved into a mature, production-ready approach to solve the challenges of maximizing the scalability and efficiency of loop-based computations on a variety of high performance and embedded targets.

After an initial experiment with Open64 [38], [37], we ported these techniques to GCC [64], [72], [71] and LLVM [51], applying them to multi-level parallelization and optimization problems, including vectorization and exploitation of thread-level parallelism. Independently, we made significant progress in the design of effective optimization heuristics, working on the interactions between the semantics of the compiler's intermediate representation and the structure of the optimization space [66], [65], [67], [30], [62]. These results open opportunities for complex optimizations that target larger problems, such as the scheduling and placement of process networks, or the offloading of computational kernels to hardware accelerators (such as GPUs). A new framework has been designed, centered on the Integer Set Library (isl, http://freecode.com/projects/isl) and implemented through multiple compiler interfaces (Graphite in GCC, Polly in LLVM) and a source-to-source research compiler (PPCG) [74], [39], [50], [73]. This new framework underlies our collaborative research activities in the CARP and COPCAMS European projects, as well as emerging transfer projects through the TETRACOM European coordination action, and most recently a bilateral industry contract called Polly Labs, fully funded by ARM and aiming for extension to other industry partners.

3.1.4. Automatic compilation of high performance circuits

For both cost and performance reasons, computing systems tightly couple parts realized in hardware with parts realized in software. The boundary between hardware and software keeps moving with the underlying technology and the external economic pressure. Moreover, thanks to FPGA technology, hardware itself has become programmable. There is now a pressing need from industry for hardware/software co-design, and for tools which automatically turn software code into hardware circuits, or more usually, into hybrid code that simultaneously targets GPUs, multiple cores, encryption ASICs, and other specialized chips.

Departing from customary C-to-VHDL compilation, we trust that sharper results can be achieved from source programs that specify bit-wise time/space behavior in a rigorous synchronous language, rather than just the I/O behavior in some (ill-specified) subset of C. This specification allows the designer to also program the (asynchronous) environment in which to operate the entire system, and to profile/measure/control each variable of the design.

At any time, the designer can edit a single specification of the system, from which both the software and the hardware are automatically compiled, and guaranteed to be compatible. Once correct (functionally and with respect to the behavioral specification), the application can be automatically deployed (and tested) on a hard/soft hybrid co-design support.

Key aspects of the advocated methodology were validated by Jean Vuillemin in the design of a PAL2HDTV video sampler [60], [61]. The circuit was automatically compiled from a synchronous source specification, decorated and guided by a few key hints to the hardware back-end, that targetted an FPGA running at real-time video specifications: a tightly-packed highly-efficient design at 240MHz, generated 100% automatically from the application specification source code, and including all run-time/debug/test/validate ancillary software. It was subsequently commercialized on FPGA by LetItWave, and then on ASIC by Zoran. This successful experience underlines our research perspectives on parallel synchronous programming.

4. Highlights of the Year

4.1. Highlights of the Year

Awards

Albert Cohen received a HiPEAC Industry Transfer Award for the Polly Labs initiative, in collaboration with Sven Verdoolaege, Tobias Grosser (now ETH Zürich), and ARM. The award comes with a 1000 euro gift.

Louis Mandel and Marc Pouzet received the price for the "Most influential PPDP'05 paper "ReactiveML: a reactive extension to ML" given at the PPDP conference, in Siena (Italy), in July 2015.

5. New Software and Platforms

5.1. Cmmtest: a tool for hunting concurrency compiler bugs

Participant: Francesco Zappa Nardelli [contact].

Languages, concurrency, memory models, C11/C++11, compiler, bugs.

The Cmmtest tool performs random testing of C and C++ compilers against the C11/C++11 memory model. A test case is any well-defined, sequential C program; for each test case, cmmtest:

1. compiles the program using the compiler and compiler optimisations that are being tested;
2. runs the compiled program in an instrumented execution environment that logs all memory accesses to global variables and synchronisations;
3. compares the recorded trace with a reference trace for the same program, checking if the recorded trace can be obtained from the reference trace by valid eliminations, reorderings and introductions.

Cmmtest identified several mistaken write introductions and other unexpected behaviours in the latest release of the gcc compiler. These have been promptly fixed by the gcc developers.

Cmmtest is available from http://www.di.ens.fr/~zappa/projects/cmmtest/ and a list of bugs reported thanks to cmmtest is available from http://www.di.ens.fr/~zappa/projects/cmmtest/gcc-bugs.html.

5.2. GCC

KEYWORDS: Compilation - Polyhedral compilation
FUNCTIONAL DESCRIPTION

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.

The emphasis as now moved towards LLVM and its Polly framework for polyhedral compilation.

- Participants: Albert Cohen, Riyadh Baghdadi, Mircea Namolaru and Nhat Minh Le
- Contact: Albert Cohen
- URL: http://gcc.gnu.org/

5.3. Heptagon

FUNCTIONAL DESCRIPTION

Heptagon is an experimental language for the implementation of embedded real-time reactive systems. It is developed inside the Synchronics large-scale initiative, in collaboration with Inria Rhones-Alpes. It is essentially a subset of Lucid Synchrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.

- Participants: Adrien Guatto, Marc Pouzet, Cédric Pasteur, Léonard Gérard, Brice Gelineau, Gwenaël Delaval and Eric Rutten
- Contact: Marc Pouzet
- URL: http://heptagon.gforge.inria.fr

5.4. Ott and Lem

lightweight executable mathematics
FUNCTIONAL DESCRIPTION

Ott and Lem are lightweight tools for writing, managing, and publishing large scale semantic definitions, where the scale makes it hard to keep a definition internally consistent, and to keep a tight correspondence between a definition and implementations.

The two tools are complementary. Ott focuses on higher-level programming language semantics. Lem is domain-specific language that resembles a pure subset of Objective Caml, supporting typical functional programming constructs and common logical mechanisms. Both tools can generate OCaml, HOL4, Coq, and Isabelle code. They also generate LaTeX code for inclusion of the language definition in scientific documents. Ott also supports a Lem backend for a tight integration between the two tools.

- Participants: Francesco Zappa Nardelli, Scott Owens, Peter Sewell
- Contact: Francesco Zappa Nardelli
- URL: http://www.cl.cam.ac.uk/~pes20/ott/ and http://www.cl.cam.ac.uk/~pes20/lem/

5.5. Lucid Synchrone

FUNCTIONAL DESCRIPTION

Lucid Synchrone is a language for the implementation of reactive systems. It is based on the synchronous model of time as provided by Lustre combined with features from ML languages. It provides powerful extensions such as type and clock inference, type-based causality and initialization analysis and allows to arbitrarily mix data-flow systems and hierarchical automata or flows and valued signals.

- Contact: Marc Pouzet
- URL: http://www.di.ens.fr/~pouzet/lucid-synchrone/

5.6. Lucy-n

Lucy-n: an n-synchronous data-flow programming language
FUNCTIONAL DESCRIPTION

Lucy-n is a language to program in the n-synchronous model. The language is similar to Lustre with a buffer construct. The Lucy-n compiler ensures that programs can be executed in bounded memory and automatically computes buffer sizes. Hence this language allows to program Kahn networks, the compiler being able to statically compute bounds for all FIFOs in the program.

- Participants: Albert Cohen, Adrien Guatto, Marc Pouzet and Louis Mandel
- Contact: Albert Cohen
- URL: https://www.lri.fr/~mandel/lucy-n/

5.7. PPCG

FUNCTIONAL DESCRIPTION

PPCG is our source-to-source research tool for automatic parallelization in the polyhedral model. It serves as a test bed for many compilation algorithms and heuristics published by our group, and is currently the best automatic parallelizer for CUDA and OpenCL (on the Polybench suite).

- Participants: Sven Verdoolaege, Tobias Grosser, Michael Kruse, Chandan Reddy, Riyadh Baghdadi and Albert Cohen
- Contact: Sven Verdoolaege
- URL: http://repo.or.cz/w/ppcg.git

5.8. ReactiveML

FUNCTIONAL DESCRIPTION

ReactiveML is a programming language dedicated to the implementation of interactive systems as found in graphical user interfaces, video games or simulation problems. ReactiveML is based on the synchronous reactive model due to Boussinot, embedded in an ML language (OCaml).

The Synchronous reactive model provides synchronous parallel composition and dynamic features like the dynamic creation of processes. In ReactiveML, the reactive model is integrated at the language level (not as a library) which leads to a safer and a more natural programming paradigm.

- Participants: Guillaume Baudart, in collaboration with Louis Mandel now at IBM Research
- Contact: Guillaume Baudart
- URL: http://rml.lri.fr

5.9. SundialsML

Sundials/ML
KEYWORDS: Simulation - Mathematics - Numerical simulations
FUNCTIONAL DESCRIPTION

Sundials/ML is a comprehensive OCaml interface to the Sundials suite of numerical solvers (CVODE, CVODES, IDA, IDAS, KINSOL, ARKODE). Its structure mostly follows that of the Sundials library, both for ease of reading the existing documentation and for adapting existing source code, but several changes have been made for programming convenience and to increase safety, namely:

- solver sessions are mostly configured via algebraic data types rather than multiple function calls;
- errors are signalled by exceptions not return codes (also from user-supplied callback routines);
- user data is shared between callback routines via closures (partial applications of functions);
- vectors are checked for compatibility (using a combination of static and dynamic checks); and
- explicit free commands are not necessary since OCaml is a garbage-collected language.

OCaml versions of the standard examples usually have an overhead of about 50% compared to the original C versions, and almost never more than 100%.

NEW PROGRESS

The current version of Sundials/ML comprises about 37,000 lines of OCaml (plus 15,000 lines of api documentation) and 16,000 lines of C (plus 1600 lines of commentary). This year we worked on updating the interface to support Sundials 2.6.x. This involved adding support for a new solver (ARKODE), new modules for sparse matrices (SuperLU_MT and KLU), new nvectors (pthreads and OpenMP), and new linear solvers (SPFGMR and PCG), as well as treating several other new or modified features. This work is almost complete and will be released early in 2016. The technical developments required to interface OCaml with this library are explained in a report which has been sumitted as a deliverable in the MODRIO project: "D.4.2.16—OCaml interface to the Sundials suite of numerical solvers". This text will be developed and submitted for journal publication in early 2016.

- Participants: Marc Pouzet and Timothy Bourke
- Partner: UPMC, AIST (Jun Inoue)
- Contact: Timothy Bourke
- URL: http://inria-parkas.github.io/sundialsml/

5.10. Zelus

SCIENTIFIC DESCRIPTION

The Zélus implementation has two main parts: a compiler that transforms Zélus programs into OCaml programs and a runtime library that orchestrates compiled programs and numeric solvers. The runtime can use the Sundials numeric solver, or custom implementations of well-known algorithms for numerically approximating continuous dynamics.

FUNCTIONAL DESCRIPTION

Zélus is a new programming language for hybrid system modeling. It is based on a synchronous language but extends it with Ordinary Differential Equations (ODEs) to model continuous-time behaviors. It allows for combining arbitrarily data-flow equations, hierarchical automata and ODEs. The language keeps all the fundamental features of synchronous languages: the compiler statically ensure the absence of deadlocks and critical races, it is able to generate statically scheduled code running in bounded time and space and a type-system is used to distinguish discrete and logical-time signals from continuous-time ones. The ability to combines those features with ODEs made the language usable both for programming discrete controllers and their physical environment.

NEW PROGRESS

- Development and release of a comprehensive manual: http://zelus.di.ens.fr/man/
- Progress on the interaction of multiple numeric solvers (masters internship of V. Andreani).
- New causality analysis (detection of algebraic loops).
- Participants: Marc Pouzet and Timothy Bourke
- Contact: Marc Pouzet
- http://zelus.di.ens.fr

5.11. isl

FUNCTIONAL DESCRIPTION

isl is a library for manipulating sets and relations of integer points bounded by linear constraints. Supported operations on sets include intersection, union, set difference, emptiness check, convex hull, (integer) affine hull, integer projection, transitive closure (and over-approximation), computing the lexicographic minimum using parametric integer programming. It includes an ILP solver based on generalized basis reduction, and a new polyhedral code generator. isl also supports affine transformations for polyhedral compilation, and increasingly abstract representations to model source and intermediate code in a polyhedral framework.

- Participants: Sven Verdoolaege, Michael Kruse and Albert Cohen
- Contact: Sven Verdoolaege
- URL: http://repo.or.cz/w/isl.git

5.12. LaTeX package: Checklistings

FUNCTIONAL DESCRIPTION

User manuals and papers about programming languages usually contain many code samples, often with accompanying compiler messages giving the types of declarations or error messages explaining why certain declarations are invalid.

The checklistings package augments the fancyvrb and listings packages for including source code in LaTeX documents with a way to pass the source code through a compiler and also include the resulting messages in the document. It also integrates with the HeVeA tool developed in the Gallium team: http://hevea.inria.fr.

The motivation is to check the code samples in a document for syntax and typing errors and to facilitate the inclusion of inferred types and compiler warnings or errors in a text. This package is intentionally very lightweight and unlike packages like python it is not intended for interacting with an interpretor or including the execution traces of code. While checklistings does not focus on a specific programming language, it is designed to work well with ML-like languages.

We developed this package to improve the quality of our papers and presentations on the Zélus programming language, but it is designed to be general purpose and also works, for instance, with OCaml programs.

- Participants: Timothy Bourke and Marc Pouzet
- Contact: Timothy Bourke
- URL: http://www.ctan.org/pkg/checklistings

6. New Results

6.1. Reasoning about C11 Program Transformations

Participants: Francesco Zappa Nardelli, Robin Morisset.

We have shown that the weak memory model introduced by the 2011 C and C++ standards does not permit many of common source-to-source program transformations (such as expression linearisation and "roach motel" reordering) that modern compilers perform and that are deemed to be correct. As such it cannot be used to define the semantics of intermediate languages of compilers, as, for instance, LLVM aimed to. We consider a number of possible local fixes, some strengthening and some weakening the model. We have evaluated the proposed fixes by determining which program transformations are valid with respect to each of the patched models. We have provided formal Coq proofs of their correctness or counterexamples as appropriate.

A paper on this work has been accepted in [18]. In collaboration with Viktor Vafeiadis (MPI-SWS, Germany) and Thibaut Balabonski (U. Paris Sud).

6.2. Language design on top of JavaScript

Participant: Francesco Zappa Nardelli.

This research project aims at improving the design of the JavaScript language. We propose a typed extension of JavaScript combining dynamic types, concrete types and like types to let developers pick the level of guarantee that is appropriate for their code. We have implemented our type system in the V8 JavaScirpt engine and we have explored the performance and software engineering benefits.

A paper on this work has been accepted in ECOOP 2015 [21].

With Gregor Richards (Waterloo University) and Jan Vitek (Northeastern University).

6.3. Synchronous Functional Language with Integer Clocks

Participant: Adrien Guatto.

Adrien Guatto defended his PhD thesis on the modular description of space/time tradeoffs at the language level. His thesis work extends the n-synchronous framework proposed by Cohen, Mandel, Plateau, Pouzet and others. Clocks now feature arbitrary positive integers that model bursty communication between subprograms: “integer clocks”. The activation conditions of Lustre are revisited in this new setting to become “local time scales” that allow subprograms to perform several steps atomically relative to their context. The thesis details the integration of these features in a clock type system for a higher-order functional language, giving full formal treatment of its metatheory and compilation to finite-state digital circuits.

6.4. Fidelity in Real-Time Programming

Participants: Guillaume Baudart, Timothy Bourke.

In this work we study embedded systems with a significant mix of discrete reactive behaviours and ‘physical’ timing constraints. The idea is to make the most of the advantages of synchronous languages for precisely specifying discrete behaviours but to adapt or extend them to treat real-time constraints more abstractly, that is, without an *a priori* definition of an eventual sampling interval.

This year we concluded our study of the Loosely Timed-Triggered Architectures (LTTA) by developing simplified models of the underlying implementations and protocols. This enabled us to improve the protocols, simplify the correctness and performance arguments, and compare them to systems built using modern clock synchronization algorithms. We developed our models in the Zélus programming language which enables (instances of) them to be compiled for simulation and contributes to our work on better exploiting synchronous languages for real-time specification and analysis. This work was presented at the EMSOFT conference and a journal article has been submitted.

This year we also concluded our study of the Quasi-synchronous Approach to modelling real-time distributed systems. We formalized the relation between the discrete abstraction proposed by Paul Caspi and the real-time archictectures for which it is intended. This enabled us to precisely state a correctness requirement for the abstraction and to show that it is sound for systems of two nodes (a typical case explored in other publications) but not for general systems of three or more nodes. Our formalization clarifies the relation between the causality of traces of the real-time system and the causality introduced by the synchronous abstraction. This enables us to state and show necessary and sufficient restrictions on the communication topologies and timing characteristics of systems to ensure soundness. A paper explaining this result has been drafted and will be submitted early in 2016.

6.5. Verified compilation of Lustre

Participants: Timothy Bourke, Marc Pouzet.

Synchronous dataflow languages and their compilers are increasingly used to develop safety-critical applications, like fly-by-wire controllers in aircraft and monitoring software for power plants. A striking example is the SCADE Suite tool of ANSYS/Esterel Technologies which is DO-178B/C qualified for the aerospace and defense industries. This tool allows engineers to develop and validate systems at the level of abstract block diagrams that are automatically compiled into executable code.

Formal modelling and verification in an interactive theorem prover can potentially complement the industrial certification of such tools to give very precise definitions of language features and increased confidence in their correct compilation; ideally, right down to the binary code that actually executes.

This year we picked up on previous work in the PARKAS team to develop a verified compiler for a Lustre/SCADE-like synchronous language. We focused on the critical and until now unresolved compiler stage that transforms dataflow equations into imperative code. We developed, in Coq, a prototype compiler for the core language (without modular resets or tuples) and showed its correctness with respect to a dataflow semantics based on functions from natural numbers to present or absent values. This required the development of a novel intermediate model for relating delayed dataflow streams to imperative memories in such a way that a critical induction could be stated and proved. We further showed how to justify a post-transformation

optimization that is essential for the efficiency of clock-directed code generation. We are preparing a paper describing these results. Work continues on both semantic questions (existence of a semantics for well-type and well-clocked programs, treatment of resets, etc.) and compilation issues (integration with the verified CompCert compiler).

In collaboration with Pierre-Évariste Dagand (CNRS) and Lionel Reig (Collège de France).

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

Technology Transfer Project, partly funded by the TETRACOM grant and by Kalray.

7.2. Bilateral Grants with Industry

Polly Labs initiative. Fully funded by ARM.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

ANR WMC project (program “jeunes chercheuses, jeunes chercheurs”), 2012–2016, 200 Keuros. F. Zappa Nardelli is the main investigator.

ANR Boole project (program “action blanche”), 2009-2014.

ANR CAFEIN, 2013-2015. Marc Pouzet.

8.1.2. Investissements d’avenir

Sys2Soft contract (Briques Génériques du Logiciel Embarqué). Partenaire principal: Dassault-Systèmes, etc. Inria contacts are Benoit Caillaud (HYCOMES, Rennes) and Marc Pouzet (PARKAS, Paris).

ManycoreLabs contract (Briques Génériques du Logiciel Embarqué). Partenaire principal: Kalray. Inria contacts are Albert Cohen (PARKAS, Paris), Alain Darte (COMPSYS, Lyon), Fabrice Rastello (CORSE, Grenoble).

8.1.3. Others

Marc Pouzet is scientific advisor for the Esterel-Technologies/ANSYS company.

8.2. European Initiatives

8.2.1. FP7 & H2020 Projects

8.2.1.1. Eurolab-4-HPC

Title: EuroLab-4-HPC: Foundations of a European Research Center of Excellence in High Performance Computing Systems

Program: H2020

Duration: September 2015 - September 2017

Coordinator: CHALMERS TEKNISKA HOEGSKOLA AB

Partners:

Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)

Chalmers Tekniska Hoegskola (Sweden)

Ecole Polytechnique Federale de Lausanne (Switzerland)

Eidgenoessische Technische Hochschule Zuerich (Switzerland)

Foundation for Research and Technology Hellas (Greece)

Universitaet Stuttgart (Germany)

Rheinisch-Westfaelische Technische Hochschule Aachen (Germany)

Technion - Israel Institute of Technology (Israel)

Universitaet Augsburg (Germany)

The University of Edinburgh (United Kingdom)

Universiteit Gent (Belgium)

The University of Manchester (United Kingdom)

Inria contact: Albert Cohen

Europe has built momentum in becoming a leader in large parts of the HPC ecosystem. It has brought together technical and business stakeholders from application developers via system software to exascale systems. Despite such gains, excellence in high performance computing systems is often fragmented and opportunities for synergy missed. To compete internationally, Europe must bring together the best research groups to tackle the longterm challenges for HPC. These typically cut across layers, e.g., performance, energy efficiency and dependability, so excellence in research must target all the layers in the system stack. The EuroLab-4-HPC project's bold overall goal is to build connected and sustainable leadership in high-performance computing systems by bringing together the different and leading performance orientated communities in Europe, working across all layers of the system stack and, at the same time, fuelling new industries in HPC.

8.2.1.2. TETRACOM

Title: Technology Transfer in Computing Systems

Program: FP7

Duration: September 2013 - August 2016

Coordinator: RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN

Partners:

Imperial College of Science, Technology and Medicine (United Kingdom)

Rheinisch-Westfaelische Technische Hochschule Aachen (Germany)

Technische Universiteit Delft (Netherlands)

Tty-Saatio (Finland)

Universita di Pisa (Italy)

Inria contact: Albert Cohen

The mission of the TETRACOM Coordination Action is to boost European academia-to-industry technology transfer (TT) in all domains of Computing Systems. While many other European and national initiatives focus on training of entrepreneurs and support for start-up companies, the key differentiator of TETRACOM is a novel instrument called Technology Transfer Project (TTP). TTPs help to lower the barrier for researchers to make the first steps towards commercialisation of their research results. TTPs are designed to provide incentives for TT at small to medium scale via partial funding of dedicated, well-defined, and short term academia-industry collaborations that bring concrete R&D results into industrial use. This will be implemented via competitive Expressions-of-Interest (EoI) calls for TTPs, whose coordination, prioritization, evaluation, and management are the major actions of TETRACOM. It is expected to fund up to 50 TTPs. The TTP activities will be complemented by Technology Transfer Infrastructures (TTIs) that provide training, service, and dissemination actions. These are designed to encourage a larger fraction of the R&D community to

engage in TTPs, possibly even for the first time. Altogether, TETRACOM is conceived as the major pilot project of its kind in the area of Computing Systems, acting as a TT catalyst for the mutual benefit of academia and industry. The projects primary success metrics are the number and value of coordinated TTPs as well as the amount of newly introduced European TT actors. It is expected to acquire around more than 20 new contractors over the project duration. TETRACOM complements and actually precedes the use of existing financial instruments such as venture capital or business angels based funding.

8.2.1.3. COPCAMS

Title: COgnitive & Perceptive CAMeraS

Program: FP7

Duration: April 2013 - March 2016

Coordinator: ___COORDINATOR___???

Partners:

Aselsan Elektroniknayi Ve Ticaret A.S. (Turkey)

Application Solutions (electronics and Vision) Ltd (United Kingdom)

Bs Spolka Z Ograniczona Odpowiedzialnoscia Spolka Komandytowa (Poland)

Concatel Sl (Spain)

Commissariat A L Energie Atomique et Aux Energies Alternatives (France)

Centre Tecnologic de Telecomunicacions de Catalunya (Spain)

Politechnika Gdanska (Poland)

Information and Image Management Systems (Spain)

Institut Jozef Stefan (Slovenia)

Iquadrat Informatica Sl (Spain)

"kolektor Group D.O.O., Vodenje in Upravljanje Družb" (Slovenia)

Queen Mary University of London (United Kingdom)

Danmarks Tekniske Universitet (Denmark)

Sogilis (France)

Squadrone System (France)

Stmicroelectronics Grenoble 2s (France)

Fundacion Tecnalia Research & Innovation (Spain)

Tedesys Global Sociedad Limitada (Spain)

Thales Communications & Securitys (France)

Thales (France)

Thales Research & Technology (uk) (United Kingdom)

Universidad de Cantabria (Spain)

Wavelens (France)

Inria contact: Albert Cohen

'Vision systems are becoming ubiquitous in our daily lives. Complex analysis of images from multiple cameras will become the norm in the future, from cars to industrial systems, from smart cities to facility monitoring, aimed at extracting meaningful, context-dependent information. Today's market is dominated by a combination of relatively simple, fixed function, configurable cameras that stream video to PC-based (and in some cases small embedded) gateways. These systems cannot scale beyond a certain size because of power consumption and the aggregate networking bandwidth required to stream videos to servers, where aggregated video analysis is performed. So the trend for visual analytics functions is that they get executed at the edge of these complex vision systems, e.g. in the cameras themselves. The Cognitive and Perceptive Camera Systems (COPCAMS) proposal leverages recent advances in embedded computing platforms to design, prototype and field-test full large-scale vision systems. It aims at exploiting a new many-core programmable accelerator platform to power a new generation of vision related devices (smart cameras and gateways), able to extract relevant information from captured images and autonomously react to the sensed environment by interoperating at large scale in a distributed manner. Date of approval by ARTEMIS JU: 7/04/2015.'

8.2.1.4. EMC2

Title: Embedded Multi-Core Systems for Mixed Criticality Applications in Dynamic and Changeable Real-Time Environments

Program: FP7

Duration: April 2014 - March 2917

Coordinator: Infineon Technologies

Partners:

Aicas (Germany)

Avl Software and Functions (Germany)

Denso Automotive Deutschland (Germany)

Elektrobit Automotive (Germany)

Evision Systems (Germany)

Nxp Semiconductors Germany (Germany)

Tttech Computertechnik (Austria)

"kompetenzzentrum - Das Virtuelle Fahrzeug, Forschungsgesellschaft Mbh" (Austria)

Frequentis (Austria)

Thales Austria (Austria)

Blueice Bvba (Belgium)

Freescale Polovodice Ceska Republika Sro (Czech Republic)

Institut Mikroelektronickych Aplikaci S.R.O. (Czech Republic)

Sysgo Sro (Czech Republic)

Silkan Rt (France)

"united Technologies Research Centre Ireland," (Ireland)

Mbda Italia Spa (Italy)

Fornebu Consulting As (Norway)

Westerngeco As (Norway)

Simula Research Laboratory As (Norway)

Ixion Industry and Aerospace Sl (Spain)

Visure Solutions Sl (Spain)

Seven Solutions Sl (Spain)

Telvent Energia (Spain)
Instituto Tecnologico de Informatica (Spain)
Ambar Telecomunicaciones Sl (Spain)
Sics Swedish Ict (Sweden)
Arcticus Systems (Sweden)
Arccore (Sweden)
Xdin Stockholm (Sweden)
Systemite (Sweden)
Stichting Imec Nederland (Netherlands)
Tomtom International Bv (Netherlands)
Infineon Technologies Uk Ltd (United Kingdom)
Sundance Multiprocessor Technology Ltd (United Kingdom)
Systonomy (United Kingdom)
Ensilica Ltd (United Kingdom)
Test and Verification Solutions Ltd (United Kingdom)
Abb (Sweden)
Ait Austrian Institute of Technology (Austria)
Alenia Aermacchi Spa (Italy)
Avl List (Austria)
Airbus Defence and Space (Germany)
Bayerische Motoren Werke Aktiengesellschaft (Germany)
Consorzio Interuniversitario Nazionale Per l'Informatica (Italy)
Critical Software (Portugal)
Chalmers Tekniska Hoegskola (Sweden)
Danfoss Power Electronics As (Denmark)
Ericsson (Sweden)
Centro Ricerche Fiat (Italy)
Fraunhofer-Gesellschaft Zur Foerderung Der Angewandten Forschung E.V (Germany)
Hi Iberia Ingenieria Y Proyectos Sl (Spain)
Harokopio University (Greece)
Infineon Technologies Austria (Austria)
"inesc Id - Instituto de Engenhariade Sistemas E Computadores, Investigacao E Desenvolvimento Em Lisboa Associacao" (Portugal)
Infineon Technologies (Germany)
Integrasys (Spain)
Instituto Superior de Engenharia Do Porto (Portugal)
Kungliga Tekniska Hoegskolan (Sweden)
Lulea Tekniska Universitet (Sweden)
Magillem Design Servicess (France)
Nxp Semiconductors Netherlands Bv (Netherlands)
Offis E.V. (Germany)
Politecnico di Torino (Italy)

Philips Medical Systems Nederland Bv (Netherlands)

Quobis Networks Sl (Spain)

Rockwell Collins France (France)

Rigas Tehniska Universitate (Latvia)

Selex Es Spa (Italy)

Siemens Aktiengesellschaft (Germany)

Systematic Paris Region Association (France)

Sysgo (Germany)

Thales Alenia Space Italia Spa (Italy)

"thales Alenia Space Espana," (Spain)

Technolution B.V. (Netherlands)

Thales Avionicss (France)

Nederlandse Organisatie Voor Toegepast Natuurwetenschappelijk Onderzoek Tno (Netherlands)

Technische Universitaet Wien (Austria)

Technische Universiteit Eindhoven (Netherlands)

Technische Universitat Braunschweig (Germany)

Technische Universiteit Delft (Netherlands)

Technische Universitat Dortmund (Germany)

Universitetet I Oslo (Norway)

Technische Universitaet Kaiserslautern (Germany)

University of Limerick (Ireland)

Universita Degli Studi di Genova (Italy)

Universita Degli Studi Dell'aquila (Italy)

University of Bristol (United Kingdom)

The University of Manchester (United Kingdom)

"ustav Teorie Informace A Automatizace Av Cr, V.V.I." (Czech Republic)

Vector Fabrics Bv (Netherlands)

Volvo Technology (Sweden)

Vysoke Uceni Technicke V Brne (Czech Republic)

Inria contact: Albert Cohen

Embedded systems are the key innovation driver to improve almost all mechatronic products with cheaper and even new functionalities. Furthermore, they strongly support today's information society as inter-system communication enabler. Consequently boundaries of application domains are alleviated and ad-hoc connections and interoperability play an increasing role. At the same time, multi-core and many-core computing platforms are becoming available on the market and provide a breakthrough for system (and application) integration. A major industrial challenge arises facing (cost) efficient integration of different applications with different levels of safety and security on a single computing platform in an open context. The objective of the EMC^2 project (Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments) is to foster these changes through an innovative and sustainable service-oriented architecture approach for mixed criticality applications in dynamic and changeable real-time environments. The EMC2 project focuses on the industrialization of European research outcomes and builds on the results of previous ARTEMIS, European and National projects. It provides the paradigm shift to a new and sustainable system architecture which is suitable to handle open dynamic systems. EMC^2 is part of

the European Embedded Systems industry strategy to maintain its leading edge position by providing solutions for: . Dynamic Adaptability in Open Systems . Utilization of expensive system features only as Service-on-Demand in order to reduce the overall system cost. . Handling of mixed criticality applications under real-time conditions . Scalability and utmost flexibility . Full scale deployment and management of integrated tool chains, through the entire lifecycle Approved by ARTEMIS-JU on 12/12/2013 for EoN. Minor mistakes and typos corrected by the Coordinator, finally approved by ARTEMIS-JU on 24/01/2014. Amendment 1 changes approved by ECSEL-JU on 31/03/2015.

8.2.2. Collaborations in European Programs, except FP7 & H2020

8.2.2.1. EMC2

Title: Affordable Safe & Secure Mobility Evolution – ASSUME

Program: Eureka ITEA3

Duration: April 2014 - March 2917

Coordinator: Siemens

Partners:

Inria

ENS Paris

Thales RT

Airbus

Esterel Technologies

Kalray

And many European partners

Inria contact: Dumitru Potop-Butucaru

Future mobility solutions will increasingly rely on smart components that continuously monitor the environment and assume more and more responsibility for a convenient, safe and reliable operation. In order to realize this vision, the need for computing power will drastically increase beyond what can be provided by conventional sequential single-core hardware. While the required efficiency and scalability makes it mandatory for future embedded micro-controllers to rely on multi- and many-core architectures, the change in hardware architecture also entails fundamental changes to state of the art software development methodology. Replacing today's essentially sequential technology by omnipresent communication between cores poses the tremendous challenge in software development to identify and exploit opportunities for concurrency in a way which still guarantees reliable and predictable behavior. Aside from the evolution of new hardware architectures, software development must address the increasing level of complexity of new highly automatic mobility solutions. For automotive, the self-driving car is the next big revolution and it is still unclear how functional and non-functional guarantees can be given for this new class of assistance functions. European industry heavily relies on the premium market segments. In these segments, innovative functions are the most important factor to influence buying decisions. New competitors, e.g. Google, enter the stage and challenge the established industry with eager visions. However, the single most important roadblock for this market is the ability to come up with an affordable, safe multi-core development methodology that allows industry to deliver trustworthy new functions at competitive prices. The ASSUME algorithm portfolio will be the key technology to bring innovative solutions from sandboxes into consumers' daily lives. ASSUME provides a seamless engineering methodology to overcome this roadblock. The problem is addressed on the constructive and on the analytic side. For efficient construction and synthesis of embedded systems, the project provides new tools, standards and methodologies to cover most of the challenges by design. In addition, ASSUME provides a well-integrated sound static analysis solution that allows proving the absence of problems even in a multi-core environment. New algorithms will be integrated in exploitable tools. New interoperability standards and requirements formalization standards will facilitate cooperation between different

market players. The ASSUME consortium includes leading European industry partners for mobility solutions, tool and service providers for embedded system development as well as leading research institutes for static analysis for model-driven and traditional embedded systems development.

8.2.3. Collaborations with Major European Organizations

Albert Cohen is an external member of the ARTEMIS-IA Working Group. Collaborating on the writing of the association's Strategic Research Agenda (SRA), and the ECSEL JU Multi-Annual Research and Innovation Agenda (MASRIA).

https://artemis-ia.eu

8.3. International Initiatives

8.3.1. Inria Associate Teams not involved in an Inria International Labs

8.3.1.1. POLYFLOW

Title: Polyhedral Compilation for Data-Flow Programming Languages

International Partner (Institution - Laboratory - Researcher):

IISc Bangalore (India) - Department of Computer Science and Automation (CSA) - Uday Kumar Reddy Bondhugula

Start year: 2013

See also: http://polyflow.gforge.inria.fr

Polyhedral techniques for program transformation are now used in several proprietary and open source compilers. However, most of the research on polyhedral compilation has focused on imperative languages such as C, where computation is specified in terms of statements with zero or more nested loops and other control structures around them. Graphical data-flow languages, where there is no notion of statements or a schedule specifying their relative execution order, have so far not been studied using a powerful transformation or optimization approach. These languages are extremely popular in system analysis, modeling and design, in embedded reactive control. They also underline the construction of many domain-specific languages and compiler intermediate representations. The copy and execution semantics of data-flow languages impose a different set of challenges. We plan to bridge this gap by studying techniques that could enable extraction of a polyhedral representation from data-flow programs, transform them, and synthesize them from their equivalent polyhedral representation.

An extension for 3 more years has been requested. It may be partly funded by CEFIPRA.

8.3.2. Inria International Partners

8.3.2.1. Informal International Partners

Prof. Uday Bondhugula, CSA department, Indian Institute of Science, India. See POLYFLOW associate team for details.

Prof. P. Sadayappan, CS department, Ohio State University, USA. Joint publications, frequent visits, occasionally for several weeks.

Prof. M. Sheeran, Computer Science and Engineering Department, Chalmers University of Technology, Sweden. Regular visits. Continuing exchanges on languages and compilation for synchronous and hybrid systems.

Prof. C. Tinelli, CS department, University of IOWA, USA. Regular visits. Continuing exchanges on the verification of synchronous languages and programs.

Prof. R. von Hanxleden, Director at the Department of Computer Science, Head of the Real-Time and Embedded Systems Group, Kiel University, Germany. Regular visits and scientific collaboration.

Prof. M. Mendler, Head of the Informatics Theory Group, Bamberg University, Germany. Regular visits and scientific collaboration.

Dr. Sven Verdoolaege, CS department, K. U. Leuven, Belgium. Joint steering of the Polly Labs initiative and contractual cooperation in this context.

Dr. Tobias Grosser in the group of Prof. Torsten Hoeffler, ETH Zürich. Joint steering of the Polly Labs initiative. See Polly Labs for details.

Pr. Peter Sewell, Computer Laboratory, University of Cambridge, UK. Regular visits and scientific collaboration.

Pr. Jan Vitek, College of Computer & Information Science Northeastern University, USA. Regular visits and scientific collaboration.

8.3.3. Participation In other International Programs

The POLYFLOW associate team has been extended for up to 3 years on January 1st 2016, in collaboration with CEFIPRA (http://cefipraonline.in).

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Prof. Michael Mendler, Univ. Bamberg, Germany, spent one month as an invited professor in the team in March 2015.

Dr. Artjoms Sinkarovs, Heriot Watt University, UK, spent 2 months as an visiting scholar in Summer 2015.

8.4.1.1. Internships

Abhishek Jain, 4th year student from IIT Delhi, visited us for 1 and a half months in January 2015.

Chaitanya Malaviya, 3rd year student from Nanyang Technological University, visited us for 2 months in July and August 2015.

8.4.2. Visits to International Teams

8.4.2.1. Research stays abroad

Marc Pouzet spent 15 days in the group of Prof. M. Mendler at Bamberg University in July 2015.

Albert Cohen spent 1 month in the group of Prof. P. Sadayappan at Ohio State University, in April–May 2015. One paper was accepted to the ACM PLDI 2016 conference as a result of this collaboration.

Timothy Bourke spent 1 week in the group of Prof. C. Tinelli at The University of Iowa in December 2015.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. General chair, scientific chair

Albert Cohen was the general chair of the 20^{th} ACM Conference on the Principles and Practice of Parallel Computing, PPoPP 2015, San Francisco (ranked A+). 186 participants. Coordination of the three co-located conferences. 700 participants.

Albert Cohen will be the general chair of the ACM Conference on Programming language Design and Implementation, PLDI 2017 conference in Barcelona (ranked A+). 600 participants expected.

9.1.1.2. Member of the organizing committees

Albert Cohen is in the steering committee of CC 2016, now separated from ETAPS. The number of submissions doubled after the renovation of the conference organization.

Albert Cohen is the sponsor chair of HiPEAC 2015 and 2016.

9.1.2. Scientific events selection

9.1.2.1. Chair of conference program committees

- Albert Cohen was program chair of the ARCS 2015 conference: http://arcs2014.iti.uni-luebeck.de

9.1.2.2. Member of the conference program committees

- Timothy Bourke was a member of the PC of EMSOFT 2015, Modelica 2015, and RTAS 2016.
- Albert Cohen was a member of the PC of PLDI 2016, PACT 2016, 2015; IPDPS 2015; ICPP 2015 and of the ERC of PLDI 2015 (ranked A+); ASPLOS 2016 (ranked A+); ISCA 2016 (ranked A+);
- Marc Pouzet was a member of the PC of DAC 2015; EOOLT 2015; FLOPS 2015; and two french speaking conferences AFADL 2015 and MSR 2015. Member of the jury for "Prix de these GPL" (PhD thesis award in software engineering).
- Francesco Zappa Nardelli is a member of the PC of POPL 2016 and of the ERC of ECOOP 2015.

9.1.2.3. Reviewer

- Timothy Bourke was a reviewer for DAC 2015 (Design Automation Conference), DATE 2015 (Design, Automation and Test in Europe), ETAPS/FASE 2015 (Fundamental Approaches to Software Engineering), FLOPS 2015 (Symposium on Functional and Logic Programming), JFLA 2015 (Journées Francophones des Langages Applicatifs), and NFM 2015 (NASA Formal Methods Symposium).

9.1.3. Journal

9.1.3.1. Member of the editorial boards

- Albert Cohen is an Associate Editor of the ACM Transactions on Architecture and Code Optimization (TACO), since 2013.
- Albert Cohen is on the editorial board of the Int. Journal on Parallel Programming (IJPP, Springer), since 2011.

9.1.4. Invited talks

- February, T. Bourke presented "Exploiting and extending synchronous languages for hybrid modelling" at the MODPROD Workshop in Linköping, Sweden.
- February, T. Bourke presented "A synchronous approach to designing and compiling hybrid modelling languages" in the Chalmer's Functional Programming group in Gothenburg, Sweden.
- February, T. Bourke presented "Mechanization of a mesh network routing protocol and the proof of its loop freedom in Isabelle/HOL" in the Chalmer's Formal Methods group in Gothenburg, Sweden.
- September, T. Bourke participated in the Modelica conference in Versailles, France.
- October, T. Bourke participated in the Embedded Systems Week conferences in Amsterdam, The Netherlands.
- October, T. Bourke presented "Some recent improvements in modelling Loosely Time-Triggered Architectures" at the Journées Nationales GEOCAL-LAC-LTP 2015 in Nancy, France.
- November, T. Bourke presented "Zélus : un langage synchrone avec équations différentielles ordinaires" in the tool session of the 10th colloquium on the Modélisation des Systèmes Réactifs (MSR 2015) in Nancy, France.

- December, T. Bourke presented "Verification of code generation for a Lustrish language" at the SYNCHRON workshop in Kiel, Germany.
- December, T. Bourke presented "Showing safety properties of a layered reactive model in Isabelle/HOL" at the University of Iowa, USA.
- December, T. Bourke presented "Towards the verified compilation of Lustre" at the University of Iowa, USA.
- October, M. Pouzet gave an invited talk at the EMSOFT'2015 conference, in Amsterdam "Building a Hybrid Systems Modeler from Synchronous Language Principles".
- April, F. Zappa Nardelli gave a Keynote at the EuroLLVM international conference, London, UK.
- July, F. Zappa Nardelli gave a Keynote at the TAP international conference, L'Aquila, Italy.

9.1.5. Leadership within the scientific community

Marc Pouzet is responsible with Sandrine Blazy (Prof., Univ. Rennes) of the group LTP (Language, Types, Proof) of the French GDR (Groupe de Recherche) GPL (Genie de la Programmation) from the CNRS. Two one-day seminars are organised every year. In 2015, one seminar in Bordeaux; the second was 12-14 october 2015, in Nancy, and together with the groups LAC and GeoCal.

Albert Cohen is a member of the Steering Committee of the ACM PPoPP and Compiler Construction conferences.

Albert Cohen is a Steering Committee member of the HiPEAC 3 network of excellence.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master: F. Zappa Nardelli: "A Programmer's introduction to Computer Architectures and Operating Systems" (M1), 45h, École Polytechnique, France

Master: A. Cohen: "Operating Systems Principles and Programming" (M1), 38h, École Polytechnique, France

Master: A. Cohen & F. Zappa Nardelli, "Semantics, languages and algorithms for multicore programming", Lecture, 9h+12h, M2, MPRI: Ecole normale supérieure and Université Paris Diderot, France

Licence : F. Zappa Nardelli: "Introduction à l'informatique" (L3), TDs, 40h, École Polytechnique, France

Licence : F. Zappa Nardelli: "Programmation Concurrente et Distribuée" (L3), TDs, 32h, École Polytechnique, France

Master : M. Pouzet & T. Bourke: "Synchronous Systems" (M2), Lectures and TDs, MPRI, France

Master: T. Bourke participated in reviewing the M1 internships of students at the ENS, France.

Licence : M. Pouzet & T. Bourke: "Operating Systems" (L3), Lectures and TDs, ENS, France.

Licence : J. Vuillemin & T. Bourke, "Digital Systems" (L3), Lectures and TDs, ENS, France

Marc Pouzet is Director of Studies for the CS department, at ENS.

9.2.2. Supervision

PhD in progress : Guillaume Baudart, 3rd year, supervised by T. Bourke and M. Pouzet

PhD in progress : Robin Morisset, 3rd year, supervised by F. Zappa Nardelli.

PhD in progress : Adrien Guatto, 4th year, supervised by A. Cohen and M. Pouzet, defended in January 2016.

PhD in progress : Riyadh Baghdadi, 4th year, supervised by A. Cohen and S. Verdoolaege, defended in September 2015.

PhD in progress : Nhat Minh Lê, 3rd year, supervised by A. Cohen.

PhD in progress : Ivan Llopard, 4th year, supervised by A. Cohen, defense in February 2016.

PhD in progress : Jie Zhao, 1st year, supervised by A. Cohen.

PhD in progress : Chandan Reddy, 1st year, supervised by A. Cohen.

M2 Internship: Virgile Andreani (Ecole normale supérieure, MPRI) supervised by T. Bourke and M. Pouzet.

L3 Internship: Chaitanya Malaviya (Nanyang Technological University) supervised by T. Bourke and M. Pouzet.

M1 Internship: Léo Testard (UPMC) - June/July 2015. Supervised by M. Pouzet

M1 Internship: Valentin Lorentz (Ecole normale suérieure de Lyon) - April/July 2015. Supervised by M. Pouzet

9.2.3. Juries

Albert Cohen was in the committees of

- Habilitation thesis, as reviewer: Marc Pérache (2015, U. Versailles and CEA DAM),
  Vincent Gramoli (2015, U. Pierre et Marie Curie et U. New South Wales, Australia);
- Habilitation thesis, as examiner: Dumitru Potop-Butucaru (2015, U. Pierre et Marie Curie and Inria),
  Stéphane Louise (2015, U. Versailles and CEA LIST);
- PhD thesis, as a reviewer: Diego Caballero (2015, U. Politècnica de Cataluña),
  Mohamed Bergach (2015, U. Nice Sophia-Antipolis),
  Lokesh Gidra (2015, U. Pierre et Marie Curie),
  Zakaria Bendifallah (2015, U. Versailles),
  Manuel Selva (2015, U. Lyon),
  Stefan Geuns (2015, U. Twente, The Netherlands),
  Jarryd Beck (2015, U. New South Wales, Australia),
  Jean-François Dollinger (2015, U. Strasbourg),
  Artem Shinkarov (2015, U. Heriot Watt, UK),
  Eugene Yip (2015, U. Auckland, New Zealand);
- PhD thesis, as an examiner:
  Andi Drebes (2015, U. Pierre et Marie Curie),
  Alexandre Aminot (2015, U. Grenoble).

Marc Pouzet was member of the PhD jury of Arlen Cox (2015, Univ. of Boulder).

Marc Pouzet was reviewer of the PhD thesis of Bernhard Thiele (2015, TUM, Munich)

Marc Pouzet was a member of the recruiting jury for two assistant professor positions at Chalmers University, Sueden, during spring 2015.

Marc Pouzet was a member of the recruiting jury for two Prof. position at University Paris Diderot, during spring 2015.

10. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] S. G. BHASKARACHARYA, U. BONDHUGULA, A. COHEN. *Automatic Storage Optimization for Arrays*, in "ACM Trans. On Programming Languages and Systems (TOPLAS)", 2016, Original submission, candidate for presentation at PLDI 2016, https://hal.archives-ouvertes.fr/hal-01257223

[2] U. BONDHUGULA, A. ACHARYA, A. COHEN. *The Pluto+ Algorithm: A Practical Approach for Parallelization and Locality Optimization of Affine Loop Nests*, in "ACM Trans. On Programming Languages and Systems (TOPLAS)", 2016, https://hal.archives-ouvertes.fr/hal-01257226

[3] T. BOURKE, R. J. VAN GLABBEEK, P. HÖFNER. *Mechanizing a Process Algebra for Network Protocols*, in "Journal of Automated Reasoning", 2016 [*DOI :* 10.1007/s10817-015-9358-9], https://hal.inria.fr/hal-01250185

[4] M. H. FOROUZANFAR, L. ALEXANDER, H. R. ANDERSON, V. F. BACHMAN, S. BIRYUKOV, M. BRAUER, R. BURNETT, D. CASEY, M. M. COATES, A. COHEN, K. DELWICHE, K. ESTEP, J. J. FROSTAD, K. C. ASTHA, H. H. KYU, M. MORADI-LAKEH, M. NG, E. L. SLEPAK, B. A. THOMAS, J. WAGNER, G. M. AASVANG, C. ABBAFATI, A. ABBASOGLU OZGOREN, F. ABD-ALLAH, S. F. ABERA, V. ABOYANS, B. ABRAHAM, J. P. ABRAHAM, I. ABUBAKAR, N. M. E. ABU-RMEILEH, T. C. ABURTO, T. ACHOKI, A. ADELEKAN, K. ADOFO, A. K. ADOU, J. C. ADSUAR, A. AFSHIN, E. E. AGARDH, M. J. AL KHABOURI, F. H. AL LAMI, S. S. ALAM, D. ALASFOOR, M. I. ALBITTAR, M. A. ALEGRETTI, A. V. ALEMAN, Z. A. ALEMU, R. ALFONSO-CRISTANCHO, S. ALHABIB, R. ALI, M. K. ALI, F. ALLA, P. ALLEBECK, P. J. ALLEN, U. ALSHARIF, E. ALVAREZ, N. ALVIS-GUZMAN, A. A. AMANKWAA, A. T. AMARE, E. A. AMEH, O. AMELI, H. AMINI, W. AMMAR, B. O. ANDERSON, C. A. T. ANTONIO, P. ANWARI, S. ARGESEANU CUNNINGHAM, J. ARNLÖV, V. S. A. ARSENIJEVIC, A. ARTAMAN, R. J. ASGHAR, R. ASSADI, L. S. ATKINS, C. ATKINSON, M. A. AVILA, B. AWUAH, A. BADAWI, M. C. BAHIT, T. BAKFALOUNI, K. BALAKRISHNAN, S. BALALLA, R. K. BALU, A. BANERJEE, R. M. BARBER, S. L. BARKER-COLLO, S. BARQUERA, L. BARREGARD, L. H. BARRERO, T. BARRIENTOS-GUTIERREZ, A. C. BASTO-ABREU, A. BASU, S. BASU, M. O. BASULAIMAN, C. BATIS RUVALCABA, J. BEARDSLEY, N. BEDI, T. BEKELE, M. L. BELL, C. BENJET, D. A. BENNETT, H. BENZIAN, E. BERNABÉ, T. J. BEYENE, N. BHALA, A. BHALLA, Z. A. BHUTTA, B. BIKBOV, A. A. BIN ABDULHAK, J. D. BLORE, F. M. BLYTH, M. A. BOHENSKY, B. BORA BAŞARA, G. BORGES, N. M. BORNSTEIN, D. BOSE, S. BOUFOUS, R. R. BOURNE, M. BRAININ, A. BRAZINOVA, N. J. BREITBORDE, H. BRENNER, A. D. M. BRIGGS, D. M. BRODAY, P. M. BROOKS, N. G. BRUCE, T. S. BRUGHA, B. BRUNEKREEF, R. BUCHBINDER, L. N. BUI, G. BUKHMAN, A. G. BULLOCH, M. BURCH, P. G. J. BURNEY, I. R. CAMPOS-NONATO, J. C. CAMPUZANO, A. J. CANTORAL, J. CARAVANOS, R. CÁRDENAS, E. CARDIS, D. O. CARPENTER, V. CASO, C. A. CASTAÑEDA-ORJUELA, R. E. CASTRO, F. CATALÁ-LÓPEZ, F. CAVALLERI, A. ÇAVLIN, V. K. CHADHA, J.-C. CHANG, F. J. CHARLSON, H. CHEN, W. CHEN, Z. CHEN, P. P. CHIANG, O. CHIMED-OCHIR, R. CHOWDHURY, C. A. CHRISTOPHI, T.-W. CHUANG, S. S. CHUGH, M. CIRILLO, T. K. D. CLASSEN, V. COLISTRO, M. COLOMAR, S. M. COLQUHOUN, A. G. CONTRERAS, C. COOPER, K. COOPERRIDER, L. T. COOPER, J. CORESH, K. J. COURVILLE, M. H. CRIQUI, L. CUEVAS-NASU, J. DAMSERE-DERRY, H. DANAWI, L. DANDONA, R. DANDONA, P. I. DARGAN, A. DAVIS, D. V. DAVITOIU, A. DAYAMA, E. F. DE CASTRO, V. DE LA CRUZ-GÓNGORA, D. DE LEO, G. DE LIMA, L. DEGENHARDT, B. DEL POZO-CRUZ, R. P. DELLAVALLE, K. DERIBE, S. DERRETT, D. C. DES JARLAIS, M. DESSALEGN, G. A. DEVEBER, K. M. DEVRIES, S. D. DHARMARATNE, M. K. DHERANI, D. DICKER, E. L. DING, K. DOKOVA, E. R. DORSEY, T. R. DRISCOLL, L. DUAN, A. M. DURRANI, B. E. EBEL, R. G. ELLENBOGEN, Y. M.

ELSHREK, M. ENDRES, S. P. ERMAKOV, H. E. ERSKINE, B. ESHRATI, A. ESTEGHAMATI, S. FAHIMI, E. J. A. FARAON, F. FARZADFAR, D. F. J. FAY, V. L. FEIGIN, A. B. FEIGL, S.-M. FERESHTEHNEJAD, A. J. FERRARI, C. P. FERRI, A. D. FLAXMAN, T. D. FLEMING, N. FOIGT, K. J. FOREMAN, U. F. PALEO, R. C. FRANKLIN, B. GABBE, L. GAFFIKIN, E. GAKIDOU, A. GAMKRELIDZE, F. G. GANKPÉ, R. T. GANSEVOORT, F. A. GARCÍA-GUERRA, E. GASANA, J. M. GELEIJNSE, B. D. GESSNER, P. GETHING, K. B. GIBNEY, R. F. GILLUM, I. A. M. GINAWI, M. GIROUD, G. GIUSSANI, S. GOENKA, K. GOGINASHVILI, H. GOMEZ DANTES, P. GONA, T. GONZALEZ DE COSIO, D. GONZÁLEZ-CASTELL, C. C. GOTAY, A. GOTO, H. N. GOUDA, R. L. GUERRANT, H. C. GUGNANI, F. GUILLEMIN, D. GUNNELL, R. GUPTA, R. GUPTA, R. A. GUTIÉRREZ, N. HAFEZI-NEJAD, H. HAGAN, M. HAGSTROMER, Y. A. HALASA, R. R. HAMADEH, M. HAMMAMI, G. J. HANKEY, Y. HAO, H. L. HARB, T. N. HAREGU, J. M. HARO, R. HAVMOELLER, S. I. HAY, M. T. HEDAYATI, I. B. HEREDIA-PI, L. HERNANDEZ, K. R. HEUTON, P. HEYDARPOUR, M. HIJAR, H. W. HOEK, H. J. HOFFMAN, J. C. HORNBERGER, H. D. HOSGOOD, D. G. HOY, M. HSAIRI, G. HU, H. HU, C. HUANG, J. J. HUANG, B. J. HUBBELL, L. HUIART, A. HUSSEINI, M. L. IANNARONE, K. M. IBURG, B. T. IDRISOV, N. IKEDA, K. INNOS, M. INOUE, F. ISLAMI, S. ISMAYILOVA, K. H. JACOBSEN, H. A. JANSEN, D. L. JARVIS, S. K. JASSAL, A. JAUREGUI, S. JAYARAMAN, P. JEEMON, P. N. JENSEN, V. JHA, F. JIANG, G. JIANG, Y. JIANG, J. B. JONAS, K. JUEL, H. KAN, S. S. KANY ROSELINE, N. E. KARAM, A. KARCH, C. K. KAREMA, G. KARTHIKEYAN, A. KAUL, N. KAWAKAMI, D. S. KAZI, A. H. KEMP, A. P. KENGNE, A. KEREN, Y. S. KHADER, S. E. A. H. KHALIFA, E. A. KHAN, Y.-H. KHANG, S. KHATIBZADEH, I. KHONELIDZE, C. KIELING, D. KIM, S. KIM, Y. KIM, R. W. KIMOKOTI, Y. KINFU, J. M. KINGE, B. M. KISSELA, M. KIVIPELTO, L. D. KNIBBS, A. K. KNUDSEN, Y. KOKUBO, M. R. KOSE, S. KOSEN, A. KRAEMER, M. KRAVCHENKO, S. KRISHNASWAMI, H. KROMHOUT, T. KU, B. KUATE DEFO, B. KUCUK BICER, E. J. KUIPERS, C. KULKARNI, V. S. KULKARNI, G. A. KUMAR, G. F. KWAN, T. LAI, A. LAKSHMANA BALAJI, R. LALLOO, T. LALLUKKA, H. LAM, Q. LAN, V. C. LANSINGH, H. J. LARSON, A. LARSSON, D. O. LARYEA, P. M. LAVADOS, A. E. LAWRYNOWICZ, J. L. LEASHER, J.-T. LEE, J. LEIGH, R. LEUNG, M. LEVI, Y. LI, Y. LI, J. LIANG, X. LIANG, S. S. LIM, M. P. LINDSAY, S. E. LIPSHULTZ, S. LIU, Y. LIU, B. K. LLOYD, G. LOGROSCINO, S. J. LONDON, N. LOPEZ, J. LORTET-TIEULENT, P. A. LOTUFO, R. LOZANO, R. LUNEVICIUS, J. MA, S. MA, V. M. P. MACHADO, M. F. MACINTYRE, C. MAGIS-RODRIGUEZ, A. A. MAHDI, M. MAJDAN, R. MALEKZADEH, S. MANGALAM, C. C. MAPOMA, M. MARAPE, W. MARCENES, D. J. MARGOLIS, C. MARGONO, G. B. MARKS, R. V. MARTIN, M. B. MARZAN, M. T. MASHAL, F. MASIYE, A. J. MASON-JONES, K. MATSUSHITA, R. MATZOPOULOS, B. M. MAYOSI, T. T. MAZORODZE, A. C. MCKAY, M. MCKEE, A. MCLAIN, P. A. MEANEY, C. MEDINA, M. M. MEHNDIRATTA, F. MEJIA-RODRIGUEZ, W. MEKONNEN, Y. A. MELAKU, M. MELTZER, Z. A. MEMISH, W. MENDOZA, G. A. MENSAH, A. MERETOJA, F. A. MHIMBIRA, R. MICHA, T. R. MILLER, E. J. MILLS, A. MISGANAW, S. MISHRA, N. MOHAMED IBRAHIM, K. A. MOHAMMAD, A. H. MOKDAD, G. L. MOLA, L. MONASTA, J. C. MONTAÑEZ HERNANDEZ, M. MONTICO, A. R. MOORE, L. MORAWSKA, R. MORI, J. MOSCHANDREAS, W. N. MOTURI, D. MOZAFFARIAN, U. O. MUELLER, M. MUKAIGAWARA, E. C. MULLANY, K. S. MURTHY, M. NAGHAVI, Z. NAHAS, A. NAHEED, K. S. NAIDOO, L. NALDI, D. NAND, V. NANGIA, K. M. V. NARAYAN, D. NASH, B. NEAL, C. NEJJARI, S. P. NEUPANE, C. R. NEWTON, F. N. NGALESONI, J. D. D. NGIRABEGA, G. NGUYEN, N. T. NGUYEN, M. J. NIEUWENHUIJSEN, M. I. NISAR, J. R. NOGUEIRA, J. M. NOLLA, S. NOLTE, O. F. NORHEIM, R. E. NORMAN, B. NORRVING, L. NYAKARAHUKA, I.-H. OH, T. OHKUBO, B. O. OLUSANYA, S. B. OMER, J. N. OPIO, R. OROZCO, R. S. PAGCATIPUNAN, A. W. PAIN, J. D. PANDIAN, C. I. A. PANELO, C. PAPACHRISTOU, E.-K. PARK, C. D. PARRY, A. J. PATERNINA CAICEDO, S. B. PATTEN, V. K. PAUL, B. I. PAVLIN, N. PEARCE, L. S. PEDRAZA, A. PEDROZA, L. PEJIN STOKIC, A. PEKERICLI, D. M. PEREIRA, R. PEREZ-PADILLA, F. PEREZ-RUIZ, N. PERICO, S. A. L. PERRY, A. PERVAIZ, K. PESUDOVS, C. B. PETERSON, M. PETZOLD, M. R. PHILLIPS, H. P. PHUA, D. PLASS, D. POENARU, G. V. POLANCZYK, S. POLINDER, C. D. POND, C. A. POPE, D. POPE, S. POPOVA, F. POURMALEK, J. POWLES, D. PRABHAKARAN, N. M. PRASAD, D. M. QATO, A. D. QUEZADA, D. A. A. QUISTBERG, L. RACAPÉ, A. RAFAY, K. RAHIMI, V. RAHIMI-MOVAGHAR, S. U. RAHMAN, M. RAJU, I. RAKOVAC, S. M. RANA, M. RAO, H. RAZAVI, K. S. REDDY, A. H. REFAAT, J. REHM, G. REMUZZI, A. L. RIBEIRO, P. M. RICCIO, L. RICHARDSON, A. RIEDERER, M. ROBINSON, A. ROCA, A. RODRIGUEZ, D. ROJAS-RUEDA, I. ROMIEU, L. RONFANI, R. ROOM, N. ROY, G. M. RUHAGO, L.

RUSHTON, N. SABIN, R. L. SACCO, S. SAHA, R. SAHATHEVAN, M. A. SAHRAIAN, J. A. SALOMON, D. SALVO, U. K. SAMPSON, J. R. SANABRIA, L. M. SANCHEZ, T. G. SÁNCHEZ-PIMIENTA, L. SANCHEZ-RIERA, L. SANDAR, I. S. SANTOS, A. SAPKOTA, M. SATPATHY, J. E. SAUNDERS, M. SAWHNEY, M. I. SAYLAN, P. SCARBOROUGH, J. C. SCHMIDT, I. J. C. SCHNEIDER, B. SCHÖTTKER, D. C. SCHWEBEL, J. G. SCOTT, S. SEEDAT, S. G. SEPANLOU, B. SERDAR, E. E. SERVAN-MORI, G. SHADDICK, S. SHAHRAZ, T. S. LEVY, S. SHANGGUAN, J. SHE, S. SHEIKHBAHAEI, K. SHIBUYA, H. H. SHIN, Y. SHINOHARA, R. SHIRI, K. SHISHANI, I. SHIUE, I. D. SIGFUSDOTTIR, D. H. SILBERBERG, E. P. SIMARD, S. SINDI, A. SINGH, G. M. SINGH, J. A. SINGH, V. SKIRBEKK, K. SLIWA, M. SOLJAK, S. SONEJI, K. SØREIDE, S. SOSHNIKOV, L. A. SPOSATO, C. T. SREERAMAREDDY, N. J. C. STAPELBERG, V. STATHOPOULOU, N. STECKLING, D. J. STEIN, M. B. STEIN, N. STEPHENS, H. STÖCKL, K. STRAIF, K. STROUMPOULIS, L. STURUA, B. F. SUNGUYA, S. SWAMINATHAN, M. SWAROOP, B. L. SYKES, K. M. TABB, K. TAKAHASHI, R. T. TALONGWA, N. TANDON, D. TANNE, M. TANNER, M. TAVAKKOLI, B. J. TE AO, C. M. TEIXEIRA, M. M. TÉLLEZ ROJO, A. S. TERKAWI, J. L. TEXCALAC-SANGRADOR, S. V. THACKWAY, B. THOMSON, A. L. THORNE-LYMAN, A. G. THRIFT, G. D. THURSTON, T. TILLMANN, M. TOBOLLIK, M. TONELLI, F. TOPOUZIS, J. A. TOWBIN, H. TOYOSHIMA, J. TRAEBERT, B. X. TRAN, L. TRASANDE, M. TRILLINI, U. TRUJILLO, Z. T. DIMBUENE, M. TSILIMBARIS, E. M. TUZCU, U. S. UCHENDU, K. N. UKWAJA, S. B. UZUN, S. VAN DE VIJVER, R. VAN DINGENEN, C. H. VAN GOOL, J. VAN OS, Y. Y. VARAKIN, T. J. VASANKARI, A. M. N. VASCONCELOS, M. S. VAVILALA, L. J. VEERMAN, G. VELASQUEZ-MELENDEZ, N. VENKETASUBRAMANIAN, L. VIJAYAKUMAR, S. VILLALPANDO, F. S. VIOLANTE, V. V. VLASSOV, S. E. VOLLSET, G. R. WAGNER, S. G. WALLER, M. T. WALLIN, X. WAN, H. WANG, J. WANG, L. WANG, W. WANG, Y. WANG, T. S. WAROUW, C. H. WATTS, S. WEICHENTHAL, E. WEIDERPASS, R. G. WEINTRAUB, A. WERDECKER, K. R. WESSELLS, R. WESTERMAN, H. A. WHITEFORD, J. D. WILKINSON, H. C. WILLIAMS, T. N. WILLIAMS, S. M. WOLDEYOHANNES, C. D. A. WOLFE, J. Q. WONG, A. D. WOOLF, J. L. WRIGHT, B. WURTZ, G. XU, L. L. YAN, G. YANG, Y. YANO, P. YE, M. YENESEW, G. K. YENTÜR, P. YIP, N. YONEMOTO, S.-J. YOON, M. Z. YOUNIS, Z. YOUNOUSSI, C. YU, M. E. ZAKI, Y. ZHAO, Y. ZHENG, M. ZHOU, J. ZHU, S. ZHU, X. ZOU, J. R. ZUNT, A. D. LOPEZ, T. VOS, C. J. MURRAY. *Global, regional, and national comparative risk assessment of 79 behavioural, environmental and occupational, and metabolic risks or clusters of risks in 188 countries, 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013*, in "Lancet", December 2015, vol. 386, n^{o} 10010, pp. 2287-323, https://hal-unilim.archives-ouvertes.fr/hal-01262737

[5] M. POUZET, C. PASTEUR, L. MANDEL. *Time Refinement in a Functional Synchronous Language*, in "Science of Computer Programming", July 2015, https://hal.inria.fr/hal-01242733

[6] S. VERDOOLAEGE, T. GROSSER, A. COHEN. *Polyhedral AST generation is more than scanning polyhedra*, in "ACM Letters on Programming Languages and Systems", 2015, vol. 37, n^{o} 4, Original submission, candidate for presentation at PLDI 2016, https://hal.archives-ouvertes.fr/hal-01257239

International Conferences with Proceedings

[7] R. BAGHDADI, U. BEAUGNON, A. COHEN, T. GROSSER, M. KRUSE, C. REDDY, S. VERDOOLAEGE, A. BETTS, A. F. DONALDSON, J. KETEMA, J. ABSAR, R. DÁVID, S. V. HAASTREGT, A. KRAVETS, A. LOKHMOTOV, E. HAJIYEV. *PENCIL: a Platform-Neutral Compute Intermediate Language for Accelerator Programming*, in "Parallel Architectures and Compilation Techniques (PACT)", San Francisco, California, United States, 2015, https://hal.archives-ouvertes.fr/hal-01257236

[8] G. BAUDART, T. BOURKE, A. BENVENISTE. *Loosely Time-Triggered Architectures: Improvements and Comparisons*, in "Proceedings of the 12th International Conference on Embedded Software (EMSOFT '15)", Amsterdam, Netherlands, October 2015 [*DOI :* 10.1109/EMSOFT.2015.7318263], https://hal.inria.fr/hal-01243005

[9] S. G. BHASKARACHARYA, U. BONDHUGULA, A. COHEN. *SMO: An Integrated Approach To Intra-Array and Inter-Array Storage Optimization*, in "Symp. on Principles of Programming Languages (POPL)", St Petersburg, FL, United States, 2016, https://hal.archives-ouvertes.fr/hal-01257228

[10] A. COHEN. *Challenges in Correct-by-Construction Multiprocessor Programming*, in "EPFL-Inria Workshop", Lausanne, Switzerland, 2015, https://hal.archives-ouvertes.fr/hal-01257233

[11] A. COHEN. *Streaming Task Parallelism*, in "Intl. Conf. on Supercomputing (ICS)", Newport Beach, California, United States, 2015, Keynote presentation, https://hal.archives-ouvertes.fr/hal-01257230

[12] A. COHEN, A. DARTE, P. FEAUTRIER. *Static Analysis of OpenStream Programs*, in "6th International Workshop on Polyhedral Compilation Techniques (IMPACT'16), held with HIPEAC'16", Prague, Czech Republic, Proceedings of the IMPACT series, http://impact.gforge.inria.fr/, Michelle Strout and Tomofumi Yuki, January 2016, https://hal.inria.fr/hal-01251845

[13] A. COHEN, V. PERRELLE, D. POTOP-BUTUCARU, M. POUZET, E. SOUBIRAN, Z. ZHANG. *Hard Real Time and Mixed Time Criticality on Off-The-Shelf Embedded Multi-Cores*, in "International Conference on Embedded and Real-Time Software and Systems (ERTS^2)", Toulouse, France, 2016, https://hal.archives-ouvertes.fr/hal-01257225

[14] X. K. DO, S. LOUISE, A. COHEN. *Managing the Latency of Data-Dependent Tasks in Embedded Streaming Applications*, in "IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC 2015)", Turin, Italy, 2015, https://hal.archives-ouvertes.fr/hal-01257231

[15] X. K. DO, S. LOUISE, A. COHEN. *Transaction Parameterized Dataflow: A Model for Context-Dependent Streaming Applications*, in "Design, Automation & Test in Europe Conference & Exhibition (DATE)", Dresden, Germany, 2016, https://hal.archives-ouvertes.fr/hal-01257227

[16] X. K. DO, S. LOUISE, A. COHEN, P. DUBRULLE, T. GOUBIER, L. CUDENNEC, P. DORÉ. *An Empirical Evaluation of a Programming Model for Context-Dependent Real-time Streaming Applications*, in "Workshop on Architecture, Languages, Compilation and Hardware support for Emerging ManYcore systems (ALCHEMY 2015, associated with ICCS)", Reykjavik, Iceland, 2015, https://hal.archives-ouvertes.fr/hal-01257237

[17] M. POUZET, T. BOURKE, C. PASTEUR, P. BRUNO, J.-L. COLAÇO. *A Synchronous-based Code Generator For Explicit Hybrid Systems Languages*, in "International Conference on Compiler Construction (CC)", London, United Kingdom, LNCS, July 2015, https://hal.inria.fr/hal-01242732

[18] V. VAFEIADIS, T. BALABONSKI, S. CHAKRABORTY, R. MORISSET, F. ZAPPA NARDELLI. *Common compiler optimisations are invalid in the C11 memory model and what we can do about it*, in "POPL 2015 - 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages", Mumbai, India, January 2015, https://hal.inria.fr/hal-01089047

[19] S. VERDOOLAEGE, A. COHEN. *Live Range Reordering*, in "6þ Workshop on Polyhedral Compilation Techniques (IMPACT, associated with HiPEAC)", Prag, Czech Republic, 2016, https://hal.archives-ouvertes.fr/hal-01257224

Conferences without Proceedings

[20] A. COHEN, V. PERRELLE, D. POTOP-BUTUCARU, M. POUZET, E. SOUBIRAN, Z. ZHANG. *Hard Real Time and Mixed Time Criticality on Off-The-Shelf Embedded Multi-Cores*, in "8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)", Toulouse, France, January 2016, https://hal.archives-ouvertes.fr/hal-01259157

[21] G. RICHARDS, F. ZAPPA NARDELLI, V. JAN. *Concrete Types for TypeScript*, in "29th European Conference on Object-Oriented Programming (ECOOP 2015)", Prague, Czech Republic, July 2015 [*DOI :* 10.4230/LIPICS.ECOOP.2015.76], https://hal.inria.fr/hal-01239167

Scientific Books (or Scientific Book chapters)

[22] *Pencil A Platform-Neutral Compute Intermediate Language for DSL Compilers*, January 2015, https://hal.archives-ouvertes.fr/hal-01165017

[23] S. POP, A. COHEN. *SSA-based Compiler Design*, in "SSA-based Compiler Design", F. RASTELLO (editor), springer, August 2016, vol. Loop tree and induction variables, ISBN 978-1-4419-6201-0, https://hal.archives-ouvertes.fr/hal-01257229

Books or Proceedings Editing

[24] A. COHEN (editor). *28th International Conference on the Architecture of Computing Systems (ARCS)*, LNCS, Springer, Porto, Portugal, March 2015, n° 9017, https://hal.archives-ouvertes.fr/hal-01257238

[25] A. COHEN, D. GROVE (editors). *Proceedings of the 20th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, ACM, Burlingame, California, United States, February 2015, https://hal.archives-ouvertes.fr/hal-01257234

Research Reports

[26] R. BAGHDADI, A. COHEN, T. GROSSER, S. VERDOOLAEGE, A. LOKHMOTOV, J. ABSAR, S. VAN HAASTREGT, A. KRAVETS, A. DONALDSON. *PENCIL Language Specification*, Inria, May 2015, n° RR-8706, 37 p. , https://hal.inria.fr/hal-01154812

[27] G. BAUDART, T. BOURKE, M. POUZET. *Soundness of the Quasi-Synchronous Abstraction*, Inria Paris-Rocquencourt ; Inria, August 2015, n° RR-8755, 19 p. , https://hal.inria.fr/hal-01175571

[28] A. COHEN, A. DARTE, P. FEAUTRIER. *Static Analysis of OpenStream Programs*, CNRS ; Inria ; ENS Lyon, January 2016, n° RR-8764, 26 p. , Corresponding publication at IMPACT'16 (http://impact.gforge.inria.fr/impact2016), https://hal.inria.fr/hal-01184408

[29] S. VERDOOLAEGE. *PENCIL support in pet and PPCG*, Inria Paris-Rocquencourt ; Inria, March 2015, n° RT-0457, https://hal.inria.fr/hal-01133962

References in notes

[30] R. BAGHDADI, A. COHEN, S. VERDOOLAEGE, K. TRIFUNOVIĆ. *Improved Loop Tiling based on the Removal of Spurious False Dependences*, in "ACM Transactions on Architecture and Code Optimization", 2013, vol. 9, n° 4, Selected for presentation at the HiPEAC 2013 Conf [*DOI :* 10.1145/2400682.2400711], https://hal.inria.fr/hal-00786674

[31] D. BIERNACKI, J.-L. COLAÇO, G. HAMON, M. POUZET. *Clock-directed Modular Code Generation of Synchronous Data-flow Languages*, in "ACM International Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)", Tucson, Arizona, June 2008

[32] F. BOUSSINOT, R. DE SIMONE. *The SL synchronous language*, in "IEEE Transaction on Software Engineering", 1996

[33] P. CASPI, M. POUZET. *Synchronous Kahn Networks*, in "ACM SIGPLAN International Conference on Functional Programming (ICFP)", Philadelphia, Pensylvania, May 1996

[34] P. CASPI, M. POUZET. *A Co-iterative Characterization of Synchronous Stream Functions*, in "Coalgebraic Methods in Computer Science (CMCS'98)", Electronic Notes in Theoretical Computer Science, March 1998, Extended version available as a VERIMAG tech. report no. 97–07

[35] A. COHEN, M. DURANTON, C. EISENBEIS, C. PAGETTI, F. PLATEAU, M. POUZET. *Synchroning Periodic Clocks*, in "ACM International Conference on Embedded Software (EMSOFT'05)", Jersey city, New Jersey, USA, September 2005

[36] A. COHEN, M. DURANTON, C. EISENBEIS, C. PAGETTI, F. PLATEAU, M. POUZET. *N-Synchronous Kahn Networks: a Relaxed Model of Synchrony for Real-Time Systems*, in "ACM International Conference on Principles of Programming Languages (POPL'06)", Charleston, South Carolina, USA, January 2006

[37] A. COHEN, S. GIRBAL, D. PARELLO, M. SIGLER, O. TEMAM, N. VASILACHE. *Facilitating the Search for Compositions of Program Transformations*, in "Intl. Conf. on Supercomputing (ICS'05)", Boston, Massachusetts, June 2005, pp. 151–160

[38] A. COHEN, S. GIRBAL, O. TEMAM. *A Polyhedral Approach to Ease the Composition of Program Transformations*, in "Euro-Par'04", Pisa, Italy, LNCS, Springer-Verlag, August 2004, n^{O} 3149, pp. 292–303

[39] A. COHEN, T. GROSSER, P. H. J. KELLY, J. RAMANUJAM, P. SADAYAPPAN, S. VERDOOLAEGE. *Split Tiling for GPUs: Automatic Parallelization Using Trapezoidal Tiles to Reconcile Parallelism and Locality, avoiding Divergence and Load Imbalance*, in "GPGPU 6 - Sixth Workshop on General Purpose Processing Using GPUs", Houston, United States, March 2013, https://hal.inria.fr/hal-00786812

[40] A. COHEN, L. MANDEL, F. PLATEAU, M. POUZET. *Abstraction of Clocks in Synchronous Data-flow Systems*, in "The Sixth ASIAN Symposium on Programming Languages and Systems (APLAS)", Bangalore, India, December 2008

[41] A. COHEN, L. MANDEL, F. PLATEAU, M. POUZET. *Relaxing Synchronous Composition with Clock Abstraction*, 2009, Workshop on Hardware Design using Functional languages (HFL 09) - ETAPS, http://hal.inria.fr/hal-00645333

[42] J.-L. COLAÇO, G. HAMON, M. POUZET. *Mixing Signals and Modes in Synchronous Data-flow Systems*, in "ACM International Conference on Embedded Software (EMSOFT'06)", Seoul, South Korea, October 2006

[43] J.-L. COLAÇO, B. PAGANO, M. POUZET. *A Conservative Extension of Synchronous Data-flow with State Machines*, in "ACM International Conference on Embedded Software (EMSOFT'05)", Jersey city, New Jersey, USA, September 2005

[44] J.-L. COLAÇO, M. POUZET. *Clocks as First Class Abstract Types*, in "Third International Conference on Embedded Software (EMSOFT'03)", Philadelphia, Pennsylvania, USA, october 2003

[45] J.-L. COLAÇO, M. POUZET. *Type-based Initialization Analysis of a Synchronous Data-flow Language*, in "International Journal on Software Tools for Technology Transfer (STTT)", August 2004, vol. 6, n^{o} 3, pp. 245–255

[46] P. CUOQ, M. POUZET. *Modular Causality in a Synchronous Stream Language*, in "European Symposium on Programming (ESOP'01)", Genova, Italy, April 2001

[47] P. FEAUTRIER. *Some Efficient Solutions to the Affine Scheduling Problem, Part II, multidimensional time*, in "Intl. J. of Parallel Programming", December 1992, vol. 21, n^{o} 6, pp. 389-420, See also Part I, one dimensional time, 21(5):315–348

[48] A. GAMATIÉ, E. RUTTEN, H. YU, P. BOULET, J.-L. DEKEYSER. *Synchronous Modeling and Analysis of Data Intensive Applications*, in "EURASIP Journal on Embedded Systems", 2008

[49] S. GIRBAL, N. VASILACHE, C. BASTOUL, A. COHEN, D. PARELLO, M. SIGLER, O. TEMAM. *Semi-Automatic Composition of Loop Transformations for Deep Parallelism and Memory Hierarchies*, in "Intl. J. of Parallel Programming", June 2006, vol. 34, n^{o} 3, pp. 261–317, Special issue on Microgrids

[50] T. GROSSER, A. COHEN, J. HOLEWINSKI, P. SADAYAPPAN, S. VERDOOLAEGE. *Hybrid Hexagonal/Classical Tiling for GPUs*, in "Intl. Symp. on Code Generation and Optimization (CGO)", Orlando, FL, United States, February 2014, https://hal.inria.fr/hal-00911177

[51] T. GROSSER, A. GRÖSSLINGER, C. LENGAUER. *Polly - Performing Polyhedral Optimizations on a Low-Level Intermediate Representation*, in "Parallel Processing Letters", 2012, vol. 22, n^{o} 4

[52] A.-C. GUILLOU, F. QUILLERÉ, P. QUINTON, S. RAJOPADHYE, T. RISSET. *Hardware Design Methodology with the Alpha Language*, in "FDL'01", Lyon, France, September 2001

[53] H. LEVERGE, C. MAURAS, P. QUINTON. *The* ALPHA *language and its use for the design of systolic arrays*, in "J. of VLSI Signal Processing", 1991, vol. 3, pp. 173–182

[54] L. MANDEL, F. BENBADIS. *Simulation of Mobile Ad hoc Network Protocols in ReactiveML*, in "Proceedings of Synchronous Languages, Applications, and Programming (SLAP'05)", Edinburgh, Scotland, Electronic Notes in Theoretical Computer Science, April 2005, Workshop ETAPS 2005

[55] L. MANDEL. *Conception, Sémantique et Implantation de ReactiveML : un langage à la ML pour la programmation réactive*, Université Paris 6, 2006

[56] L. MANDEL, F. PLATEAU, M. POUZET. *Lucy-n: a n-Synchronous Extension of Lustre*, in "10th International Conference on Mathematics of Program Construction (MPC'10)", Manoir St-Castin, Québec, Canada, Springer LNCS, June 2010

[57] L. MANDEL, M. POUZET. *ReactiveML, a Reactive Extension to ML*, in "ACM International Conference on Principles and Practice of Declarative Programming (PPDP)", Lisboa, July 2005

[58] F. MARANINCHI, N. BERTHIER, O. BEZET, G. FUNCHAL. *Writing Simulators with Synchronous Languages*, 2008, Synchron 2008: International Open Workshop on Synchronous Programming

[59] C. MIRANDA, A. POP, P. DUMONT, A. COHEN, M. DURANTON. *Erbium: A Deterministic, Concurrent Intermediate Representation to Map Data-Flow Tasks to Scalable, Persistent Streaming Processes*, in "Intl. Conf. on Compilers Architectures and Synthesis for Embedded Systems (CASES'10)", October 2010

[60] J.-B. NOTE, M. SHAND, J. VUILLEMIN. *Realtime video pixel matching*, in "International Conference on Field Programmable Logic and Applications", 2006, pp. 507 – 512

[61] J.-B. NOTE, J. VUILLEMIN. *Towards automatically compiling efficient FPGA hardware*, in "International Workshop on Design and Functional Languages", IEEE, 2007, pp. 115 – 124

[62] E. PARK, J. CAVAZOS, L.-N. POUCHET, C. BASTOUL, A. COHEN, P. SADAYAPPAN. *Predictive Modeling in a Polyhedral Optimization Space*, in "International Journal of Parallel Programming", 2013, vol. 41, n^{o} 5, pp. 704–750 [*DOI :* 10.1007/s10766-013-0241-1], https://hal.inria.fr/hal-00918653

[63] F. PLATEAU. *Modèle n-synchrone pour la programmation de réseaux de Kahn à mémoire bornée*, Université Paris-Sud 11, Orsay, France, 6 janvier 2010, https://www.lri.fr/~mandel/lucy-n/~plateau/these/

[64] S. POP, A. COHEN, C. BASTOUL, S. GIRBAL, G.-A. SILBER, N. VASILACHE. *GRAPHITE: Loop Optimizations Based on the Polyhedral Model for GCC*, in "Proc. of the 4þ GCC Developper's Summit", Ottawa, Canada, June 2006

[65] L.-N. POUCHET, C. BASTOUL, A. COHEN, J. CAVAZOS. *Iterative Optimization in the Polyhedral Model: Part II, Multidimensional Time*, in "ACM Conf. on Programming Language Design and Implementation (PLDI'08)", Tucson, Arizona, June 2008

[66] L.-N. POUCHET, C. BASTOUL, A. COHEN, N. VASILACHE. *Iterative Optimization in the Polyhedral Model: Part I, One-Dimensional Time*, in "Intl. Symp. on Code Generation and Optimization (CGO'07)", San Jose, California, March 2007

[67] L.-N. POUCHET, U. BONDHUGULA, C. BASTOUL, A. COHEN, J. RAMANUJAM, P. SADAYAPPAN. *Combined Iterative and Model-driven Optimization in an Automatic Parallelization Framework*, in "ACM Supercomputing Conf. (SC'10)", New Orleans, Lousiana, November 2010, 11 p.

[68] P. RAYMOND, Y. ROUX, E. JAHIER. *Lutin: a language for specifying and executing reactive scenarios*, in "EURASIP Journal on Embedded Systems", 2008, vol. 2008, Article ID 753821

[69] L. SAMPER, F. MARANINCHI, L. MOUNIER, L. MANDEL. *GLONEMO: Global and Accurate Formal Models for the Analysis of Ad hoc Sensor Networks*, in "Proceedings of the First International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense'06)", Nice, France, May 2006

[70] J. SOULA, P. MARQUET, J.-L. DEKEYSER, A. DEMEURE. *Compilation principle of a specification language dedicated to signal processing*, in "Intl. Conf. on Parallel Computing Technologies", Novosibirsk, Russia, LNCS, Springer-Verlag, September 2001, vol. 2127, pp. 358–370

[71] K. TRIFUNOVIĆ, A. COHEN, D. EDELSOHN, F. LI, T. GROSSER, H. JAGASIA, R. LADELSKI, S. POP, J. SJÖDIN, R. UPADRASTA. *GRAPHITE Two Years After: First Lessons Learned From Real-World Polyhedral Compilation*, in "GCC Research Opportunities Workshop (GROW'10)", Pisa, Italy, January 2010

[72] K. TRIFUNOVIĆ, D. NUZMAN, A. COHEN, A. ZAKS, I. ROSEN. *Polyhedral-Model Guided Loop-Nest Auto-Vectorization*, in "Parallel Architectures and Compilation Techniques (PACT'09)", Raleigh, North Carolina, September 2009

[73] S. VERDOOLAEGE, S. GUELTON, T. GROSSER, A. COHEN. *Schedule Trees*, in "IMPACT - 4th Workshop on Polyhedral Compilation Techniques, associated with HiPEAC", Vienna, Austria, ACM, January 2014, https://hal.inria.fr/hal-00911894

[74] S. VERDOOLAEGE, J. C. JUEGA, A. COHEN, J. I. GÓMEZ, C. TENLLADO, F. CATTHOOR. *Polyhedral Parallel Code Generation for CUDA*, in "ACM Transactions on Architecture and Code Optimization", 2013, vol. 9, n^{o} 4, Selected for presentation at the HiPEAC 2013 Conf. [*DOI :* 10.1145/2400682.2400713], https://hal.inria.fr/hal-00786677

[75] J. VUILLEMIN. *On Circuits and Numbers*, Digital, Paris Research Laboratory, 1993