INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team A3

Analyse Avancée Appliquée à l'optimisation des codes

Rocquencourt - Futurs

THEME 1A

Table of contents

1. Team

Head of project team

Christine Eisenbeis [DR Inria]

Administrative assistants

Nathalie Gaudechoux [SAR Inria, with Fractales]
Stéphanie Meunier [TR Inria, with Gemo]

Staff members, Inria

Philippe Clauss [Professor, Université de Strasbourg, on partial secondment until August 31st, 2002]
Albert Cohen [CR]
François Thomasset [DR]

Junior technical staff

Saurabh Sharma [until August 31st, 2003]
Marc Gonzalez-Sigler [since September 1st, 2003]

Other staff members

Denis Barthou [Maître de conférences, Université de Versailles-Saint-Quentin]
Olivier Temam [Professor, Université Paris-Sud]

Ph. D. students

Pierre Amiranoff [professeur certifié de mathématiques (on secondment), 1/2 ATER CNAM-Paris until August 31th, 2003, PRAG, Université de Nanterre, since September 1st, 2003]
Cédric Bastoul [bourse MENRT, Université de Paris VI until August 31th, 2003, 1/2 ATER, Université de Versailles-Saint-Quentin, since September 1st, 2003]
Patrick Carribault [Université de Versailles-Saint-Quentin, since September 1st, 2003]
Ivan Djelic [bourse MENRT, 1/2 ATER, Université de Versailles St-Quentin, until June 30th, 2003]
Sylvain Girbal [bourse CEA, Université Paris-Sud]

Student interns

Patrick Carribault [DEA, Université Paris-Sud, from March 1st till August 31st, 2003]
Meriem Belguidoum [DESS, CNAM/Paris 6, from April 1st till August 31st, 2003]

2. Overall Objectives

A3 has been a INRIA pre-project since 1996 and was created as a project-team in december 1998. A3 is a joint project with the Prism lab of the Université de Versailles-Saint-Quentin. It was accepted by CNRS in july 1999. The A3 project-team is ending at the end of 2003 and will become the new INRIA FutURs ALCHEMY team.

A3 research focus on program analysis together with its applications in code performance optimization on high performance processors, especially the optimization of memory hierarchy management and instruction-level parallelism. A3 designs methods and tools used by compilers or users for code analysis and optimization that exploit at best as possible architectural features of processors.

Objectives of the A3 project-team are:

- develop new methods for program data-flow analysis,
- generalize traditional methods of static analysis to code optimization,
- take into account architectural features in code analysis,
- develop new methods of code optimization,
- develop new methods and tools for dynamic code analysis, as well as new optimization methods that exploit results of this analysis.

Applications targeted by A3 are:

- optimization of programs known as “computation intensive”, on high performance processors available in personal computers as well as workstations,
- analysis and optimization of programs of specialized processors and/or embedded processors,
- program parallelization on servers or workstations with a limited number of processors.

3. Scientific Foundations

Software or hardware? Computer programming has always been a trade-off between both. Specific processors at one extreme point, general-purpose microprocessors at the opposite, this question is exacerbated when performance issues are concerned. High performance processors architectures are actually constantly evolving and their efficient programmation requires a more and more specialized expertise. While programmers could easily vectorize or parallelize the source program on 1980’ supercomputers, they have today to take into account memory hierarchy and instruction level parallelism, that are typically managed at the assembly code level.

The step of **semantic analysis** of programs, data access patterns, anticipation of dynamic execution behaviours, is a prerequisite of any optimisation. In traditional compilers, code analysis relies on very strong theories of program semantics and applies to any general code. But the resulting information is not very precise, especially in the case of regular data types. At the opposite automatic parallelizers provide a very fine grain analysis of array data accesses. But this kind of analysis is restricted to programs with very regular control patterns as well as regular data memory accesses.

Likewise optimizations performed in traditional compilers are basically aiming at reducing instructions count - for instance removing invariants in loops avoids wasting time in useless computations. These methods apply to any kind of programs. At the opposite optimizing for high performance architecture usually requires changing the schedule of instructions. This is easy and efficient in the very restricted case of straight line code or loop without branchs - for instruction level parallelism - and regular data access patterns - for memory hierarchy management; how to extend the scope of optimizations is still an open problem.

Hence in both code analysis and code optimization two trends are emerging among methods. First kinds of methods are **general purpose** but ignore architectural features. Second kinds of methods are **efficient** but restricted to specific code constructs. How to combine generality and efficiency? Research done in the A3 project team hinges on this issue.

4. Application Domains

Applications targeted in A3 are basically code optimization in high performance architectures. This includes general purpose microprocessors and specialized embedded processors or DSP (*Digital Signal Processor*), as well as servers with a small number of processors or supercomputers with shared memory programming model.

As for programs A3 targets performance critical applications. This includes scientific computations (european MHAOTEU project, 1998-2001), multimedia embedded applications (european OCEANS project, 1996-1999), or videographics applications (SANDRA project described in section 7.1).

The objective of performance can be understood with various meanings: maximization of execution speed, minimization of data transfers or memory access, minimization of power consumption or power dissipation, minimization of the size of the generated code, minimization of the cost of design of the processor...

Methods and algorithms developed in the A3 team apply to any level in the programmation process:

- programming environment (code reengineering, interactive optimization tools, code transformation toolbox). The MHAOTEU ESPRIT project (1998-2001) focused on this kind of applications for memory hierarchy optimization.

- source to source preprocessor: for instance PAF (*Paralléliseur Automatique de FORTRAN*) or TOPS (source to source software pipelining, section 5.8).
- compiler ;
- postprocessor of assembly code optimization, such as in the SALTO framework of the IRISA CAPS team in which the PILO and LORA software were integrated;
- moreover, A3 research work makes it possible to characterize the difficulty of use of each device of the specialized processors, and thus to influence their architecture, as in the case of the project SANDRA.

The european OCEANS ESPRIT project, that ended in 1999, hinged on the interaction between the 2 phases of pre- and post-processing for VLIW architectures.

5. Software

5.1. DiST

distributed simulation of microprocessors with dynamic warm-up and trace partitioning.

5.2. Tuareg

extensive environment for writing, running and debugging OCaml programs in Emacs/XEmacs. Thousands of installations worldwide, distributed as part of Debian GNU/Linux 3.0.

5.3. DigLC2

Participants: Albert Cohen, Olivier Temam.

gate-level simulator for the LC-2 processor (*Little Computer 2* from Yale Patt and Sanjay Patel) [4]; dedicated to computer architecture education, based on Chipmunk's DigLog system.

5.4. CLooG

Participant: Cédric Bastoul.

CLOOG (*Chunky LOOp Generator*) is a software and a library that generates the loop code for scanning integer points of polyhedra http://www.prism.uvsq.fr/~cedb/bastools/cloog.html.

5.5. PipLib

Participants: Paul Feautrier, Cédric Bastoul.

PIP is a now well-known solver of parameterized problems of integer linear programming http://www.prism.uvsq.fr/~cedb/bastools/piplib.html. PIP finds the lexicographically minimal integer point of a convex polyhedra, even if these polyhedra depend linearly on one or more parameters.

5.6. PiLo : pipeline logiciel, software pipelining

PILO is a software pipelining package. It was implemented by Antoine Sawaya for his PhD [14]. The loop is given by its dependency graph that specifies dependence distances and execution latencies constraints. Architectural features are modelized by resource reservation tables, functional units counts, number of available registers. PILO is based on the *DESP* method (Decomposed Software Pipelining [15]), improved in [14]. It is used in the Sage++ programming environment (section 5.8) as well as in SALTO (OCEANS project).

5.7. LoRA : Loop Register Allocation

LoRA [9] is a package for loop register allocation that was designed by Sylvain Lelait for his PhD [12]. LoRA is based on the *meeting graph* and trades off register pressure against loop unrolling degree. LoRA is connected to PiLo (see above) and was also integrated in MOST (Modulo Scheduling Testbed) developed in McGill University (Montreal).

5.8. TOPS : source to source software pipelining

TOPS is a tool for loop software pipelining in FORTRAN programs. User or compiler specifies which loops should be software pipelined by inserting directives. One can also specify, based on analysis or expertise which data should be prefetched for avoiding processor stalls in the case of cache miss. TOPS was designed by Min Dai for his PhD [32]. It is based on the Sage++ environment of program manipulation.

5.9. Tools for symbolic computation and polyedra manipulation

Participant: François Thomasset.

A number of interface tools have been designed by François Thomasset, these tools connect Maple, MuPad and the Polylib and Piplib libraries.

- http://www-rocq.inria.fr/~thomasse/Maple-MuPad/ Conversion of Maple programs to MuPAD;
- http://www-rocq.inria.fr/~thomasse/mupad-ehrhart Interface between MuPAD and the Polylib (Ehrhart) library;
- http://www-rocq.inria.fr/~thomasse/mupad-pip/ Interface between MuPAD and the Piplib library.

5.10. SAREQ : testing the equivalence of systems of recurrence equations

Participants: Denis Barthou, Paul Feautrier, Xavier Redon.

The SAREQ software was developed by Xavier Redon (LIFL, Université of Lille I) in a collaborative work with Denis Barthou and P. Feautrier on the recognition of algorithms. It makes it possible to test the equivalence of two closely connected systems of equations of recurrence. It is written in Ocaml and uses the « Polylib » et « Omega » polyhedric libraries.

6. New Results

6.1. Semantic analysis of programs

6.1.1. Instancewise program analysis

Participants: Pierre Amiranoff, Albert Cohen, Paul Feautrier.

We achieved the formalization and the extension of Instancewise Analysis fo Recursive Programs [17][25]. We have defined a language dedicated to the precise static computation of inductive variables. This language constraints the programming model. It focuses on the variables and data structures included in the model. It is called MOGUL, for MOnoid GUided Language, as properties of the control and data flows are expressed in the frame of finite type monoïds. For the MOGUL language, we have built a complete implementation (from parsing to the computation of the rational transducers that describe dependences in OCaml. In the special case where the programs operates over arrays, we built an hybrid dependence test combining Formal Language Theory with a decidable sub-class of Minsky machines. We have shown the NP-completion of this test, but we exhibit an efficient algorithm in practice, for all the programs we encountered, with the help of Integer Linear Programming. Recently, we have extended the model in order to take loop bounds and conditional tests into account, for a subclass of MOGUL programs.

6.1.2. Analysis of predicated code

Participants: Albert Cohen, Ivan Djelic.

Following our preliminary results [6], we strenghtened the analysis of predicate relations in guarded assembly code (e.g., on EPIC architectures), with a generalized setting in abstract interpretation [31] for arbitrary bit-vector data-flow problems. The applications we considered include partial redundancy elimination, reaching-definition analysis, dependence testing and register allocation have, for which we provided specific lattices and specializations of the general setting.

6.1.3. Analysis of regularities in program traces

Participants: Philippe Clauss, Bénédicte Kenmei Youta, Odile Rousselet.

Execution tracing is a convenient way of understanding the program behaviour while going beyond limits of static analysis. But it can be difficult to express traces in a comprehensible way or to represent them through a model being used for simulation, prediction and optimization. In this article, we propose to model traces of programs by nests of loops by identifying repetitive and periodic patterns. The generated loops can then be used for all the objectives quoted above while benefitting from the developments carried out within the framework of the polyhedric model [22].

6.1.4. A Symbolic Approach to Bernstein Expansion for Program Analysis and Optimization

Participants: Philippe Clauss, Irina Tchoupaeva.

Several mathematical tools for static analysis of programs were developed in the last decades. Although these tools are particularly useful, they have of course limits. In particular, integer multi-variable polynomials appear in many cases of analysis methods, that are unable to handle them. Although specific methods were already proposed, they consider only subsets of such expressions. This article presents a general and original approach of a symbolic version of the Bernstein expansion. This approach makes it possible to give bounds to the values taken by a multivariable polynomial on a "box" (interval), and is in general more precise than the traditional methods of approximation by intervals [29].

6.2. Optimization of programs performance

6.2.1. Iterative optimization meets the polytope model

Participants: Cédric Bastoul, Albert Cohen, Sylvain Girbal, Marc Gonzalez-Sigler, Saurabh Sharma, Olivier Temam.

Static cost models have a hard time coping with hardware components exhibiting complex run-time behaviors, calling for alternative solutions. Iterative optimization is emerging as a promising research direction, but currently, it is mostly limited to finding the parameters of program transformations [34][33]. We want to extend the scope and efficiency of iterative optimization techniques by searching not only for the appropriate parameter of a given transformation, but for the program transformations themselves, and especially for *compositions* of program transformations.

This work is one of the cornerstones of our *Center for Program Tuning* (RNTL project, 2004–2006) described in section 7.2. The main goal is to facilitate the expression and search of compositions of program transformations. This framework relies on a unified polyhedral representation of loops and statements. The key to our framework is to clearly separate the impact of each program transformation on the following three components: the iteration domain, the statements schedule and the memory access functions. Within this framework, composing a long sequence of program transformations induces no code explosion. As a result, searching for compositions of transformations is not hampered by the multiplicity of compositions, and ultimately, it is equivalent to testing different values of the matrices parameters in many cases. Our techniques have been implemented ot top of the Open64/ORC compiler. In addition, we are beginning the design of a robust infrastructure for iterative optimization, based on machine learing techniques (operation research, e.g., genetic algorithms). This infrastructure distributes simulations, dynamic profiles, compilations,

transformations, while interacting with a machine-learning component or with an expert user. Validation of these concepts and application of the tools will be a critical issue in the center for program tuning.

6.2.2. Optimization of pattern-matching programs

Participants: Patrick Carribault, Albert Cohen.

To achieve the best performance on single processors, optimizations need to target most components of the architecture simultaneously, focusing on the memory hierarchy (including registers), branch prediction and instruction-level parallelism. Typical examples of good candidates for aggressive optimization technologies include regular and numerical computations from scientific, signal processing or multimedia applications.

More irregular programs can also be data and compute intensive, but less architecture-aware optimizations have been proposed for such programs. Still, speculative and very complex transformations are available for such codes in the context of massively parallel computers. We investigated the applicability and extension/adaptation of some of these techniques for the optimization on uniprocessors, and our results were extremely promising in the case of two approximate string-matching codes (for computational biology). Hybrid static-dynamic optimizations for such programs are also being considered, driving the selection of optimization parameters at run-time through the fine-grain tracking of the behaviour of the application (performance counters). Other codes will be considered in the future, including additional bioinformatics examples (matching for sequencing, protein and RNA folding), as well as shape recognition algorithms (classification), data-mining and irregular numerical codes, e.g., meshes.

6.2.3. Improving loop data locality

Participants: Cédric Bastoul, Paul Feautrier.

Cache designers based themselves on probabilistic arguments : each program addresses the main memory randomly. If this model is appropriate for the usual irregular programs, it fails for the regular programs stemming from scientific computation or digital signal processing. Moreover embedded applications require that the execution times be predictible, and thus that the cache misses are kept under control. In our approach, we propose to bypass the replacement mechanism that is hard to plan in advance. The program is divided of pieces (*chunks*). At the beginning of each chunk, the cache is empty. The size of each chunk is adjusted so that its data exactly hold in the cache. Hence the replacement strategy has no effect. At the end of each chunk, the cache is flushed before starting the next one. In this model, it is possible to give an asymptotic estimation of data traffic and we use this information to calculate an optimal chunking, that may induce transformation of the source program.

In the context of his PhD, C. Bastoul defined the chunking algorithms for the most significant cases: group reuse, self reuse, multiple reuse, and has implemented the CHUNKY prototype. The program is then rebuilt by the CLOOG code generator (section 5.4). The experiments are very encouraging for both data locality improvement [21] and code generation [20] where new significant results have been achieved. This work can be transposed to software managed local memories or scratchpad in the context of embedded memories.

6.3. Synchronous languages and real-time videoprocessing applications

Participants: Meriem Belguidoum, Albert Cohen, Christine Eisenbeis, Marc Pouzet.

In the continuity of the SANDRA project (section 7.1) about high performance architectures for videoprocessing we consider the problem of specifying, controlling, verifying physical time properties directly in programs. This year we have worked with Marc Pouzet of the *Université de Paris 6* and have started to specify the HPN (Hierarchical Process Network) [3] model in his LUCID SYNCHRONE synchrounous and functional programming language. There are two main issues that had to be clarified in order to use LUCID SYNCHRONE. The first one was to be able to handle the data flow sequences by blocks. This implies keeping track of their recent history, with possibly parameterized length. We succeeded in specifying block computations in pure LUCID SYNCHRONE. Additionally the size of the blocks can be itself a flow. The second issue was to specify the burstiness property of HPN, meaning that arrival of data can be delayed but only within a time window.

We could specify the burstiness as well but at the price of specifying the times of arrivals by a specific clock. Therefore on the latter case the clock can not be synthetized. The follow-up of this work will be to consider if the mechanism used in LUCID SYNCHRONE for clock computing can be adapted for accounting for other parameters of implementation such as ressource usage. This work was done by Meriem Belguidoum for her DESS degree [27].

7. Contracts and Grants with Industry

7.1. The Sandra project

Participants: Albert Cohen, Marc Duranton, Christine Eisenbeis.

The SANDRA project is a collaborative work between INRIA A3 and Philips Research (Eindhoven). It started with the PRF (Philips Research France) in year 2003.

The topic is to design a specification and programming environment for video-graphics processors that are of growing importance todays with the advent of digital and high definition television. In this context the flow of images are coming from different sources at possibly different rates. They have to be combined on a single screen or other drivers such as videorecorders after some processing steps from simple scaling upto more 2D or 3D complex transformations. In past years we have designed a programming language, schedulers and the HPN environment for real-time properties verification [16][26]. In 2003 we have considered using the LUCID SYNCHRONE language for clock verification (see 6.3). We have also worked with the INRIA Compose project-team in order to set up a collaboration on domain-specific languages for videographics processing.

7.2. The Cop project

Participants: Cédric Bastoul, Albert Cohen, Sylvain Girbal, Marc Gonzalez-Sigler, Saurabh Sharma, Olivier Temam.

We submitted an exploratory RNTL project (long-term academic-industrial research project, funding from the ministry of research) called "Centre d'Optimisation de Programmes" (COP) or "Center for Program Tuning" (CPT), in January 2004. This submission was successful and the project was ranked 1st (ex aequo with 5 other projects). The project is coordinated by Olivier Temam from LRI, Paris-South University, with partners from IRIT (Toulouse), CEA Saclay, STMicroelectronics Crolles and HP France. Funding starts in January 2004.

8. Other Grants and Activities

8.1. National Initiatives

Véronique Donzeau-Gouge, professor at CNAM is the official supervisor of Pierre Amiranoff.

A3 organizes a joint seminar with CRI (Centre de Recherches en Informatique, Ecole des Mines de Paris) and LRI (Laboratoire de Recherches en Informatique, Université Paris-Sud). Talks of 2003 are given below.

- january 7th, 2003 :*Simulating a $2M Commercial Server on a $2K PC*, Mark D. Hill (University of Wisconsin-Madison);
- july 15th, 2003 :*Optimisation de code de Pattern Matching sur l'architecture EPIC : tranformation, expériences, automatisation*, Patrick Carribault, INRIA/A3;
- july 21st, 2003 :*Reconnaissance de templates d'algorithmes*, Christophe Alias, PRISM, Université de Versailles-Saint-Quentin;
- august 28th, 2003 :*Analysis of Induction Variables Using Chains of Recurrences: Extensions*, Sebastian Pop, ICPS (Université Louis Pasteur, Strasbourg) and CRI;
- november 4th, 2003 :*Validation de Codes Assembleurs Relogeables par Interprétation Abstraite*, Matthieu Martel, CEA, Laboratoire de Sûreté des Logiciels (LSL);
- september 11th, 2003 :*Multivariate techniques for benchmark selection*, Koen de Bosschoere, Ghent University, Belgium.

8.2. European initiatives

8.2.1. Proposals

Olivier Temam is in the steering committee of the proposal of the european HiPeac (High Performance architectures and compilers) Network of Excellence. Scientists of the A3 project-team are members of this network as well. The aim is to federate the european research in current and future processor architectures and compilers.

A3 has participated in the elaboration of the SpotLight proposal for a european STREP [1] project targeting tools for data-flow based applications on reconfigurable architectures.

8.2.2. Other collaborations

PAI Procope with Prof. Christian Lengauer at the University of Passau, Germany: Christoph Herrmann, associate researcher, visited INRIA Rocquencourt in september (two weeks); Peter Faber, PhD student, visited INRIA Futurs in december (one week).

A new Procope contract has been agreed for two years (2004–2006), between Albert Cohen and Christoph Herrmann, to collaborate and promote common research on metaprogramming and domain-specific program optimization.

In the context of the SANDRA project (voir 7.1), Marc Duranton (Philips Research, Eindhoven) visited INRIA on regular basis.

8.3. International initiatives

We collaborate with Jean-Luc Gaudiot (University of Irvine at Los Angeles) and Guang Gao (University of Delaware) on "I-structures" and their use in program optimization. J.-L. Gaudiot visited INRIA on April 22nd and Gao on April 24th.

8.4. Visiting scientists

They are listed in the sections 8.1, 8.2 and 8.3.

9. Dissemination

9.1. Leadership within scientific community

Albert Cohen is a member of the program comittee for the DATE'04 conference; topic B11 "design aspects of emerging technologies and applications".

Albert Cohen was in charge of the interaction with research groups at INRIA Rocquencourt until december, for the preparation of the 6th Framework Program of the European Union.

Albert Cohen is a member of the department committee of the PRiSM laboratory, University of Versailles.

Christine Eisenbeis served on the program committe of SCOPES 2003 (7th International Workshop on Software and Compilers for Embedded Systems, Vienna , Austria , September 2003), CGO 2004 (International Symposium on Code Generation and Optimization with Special Emphasis on Feedback-Directed and Runtime Optimization, Palo Alto, March 2004) and CC 2004 (International Conference on Compiler Construction, Barcelona, March 2004). Christine Eisenbeis has organized In February 2003 a one-week workshop in Dagstuhl, Germany with Mary-Lou Soffa and Tom Conte, "Emerging Technologies: can optimization technology meet their demands?" (http://www.dagstuhl.de/03071/index.en.phtml).

Ph. Clauss, A. Cohen, Ch. Eisenbeis et P. Feautrier are in the steering committee of the CNRS "réseau thématique pluridisciplinaire" (RTP) that was initiated in september 2002 on "Architecture and compilers", with Ph. Clauss as the co-leader together with Pascal Sainrat (IRIT, Toulouse). Ph. Clauss et Ch. Eisenbeis are the co-leaders of the CNRS "action spécifique" (AS) on "Compilers for embedded systems"; they have

[1] Specific Targeted Research Project

organized 3 one or two-days meeting in 2003. A. Cohen and Ch. Eisenbeis are also members of the CNRS "action spécifique" on "Future technologies and future paradigms for processor architectures", whose leader is Nathalie Drach-Temam (LRI).

9.2. Teaching at university

Pierre Amiranoff was ATER at CNAM (algorithmics, programming, Ada) and then PRAG (initiation to computer tools, initiation to programming, Pascal).

C. Bastoul teached as a "moniteur" then as an "ATER" at the Université de Versailles-Saint-Quentin ("programming" and "algorithmics").

Ph. Clauss teaches at the DEA of the Université Louis Pasteur of Strasbourg (« Parallelism » and « High performance compilers »).

Albert Cohen teaches at the Master of Computer Science of the Paris South University (compilation and optimization for high performance and embedded systems) and at École Polytechnique (third year computer architecture and first year Java programming labs).

9.3. Workshops, seminars, invitations

The project-team members have given the following talks and attended the following conferences:

- Pierre Amiranoff, Cédric Bastoul, Albert Cohen, Christine Eisenbeis: participation to the CPC'03 workshop, Amsterdam, june 2003 (talk of Pierre Amiranoff, "Instancewise Array Dependence Test for Recursive Programs", talk of Cédric Bastoul, "Reordering methods for data locality improvement").
- Sylvain Girbal: article and presentation at the ACM SIGMetrics'03 conference, San Diego, june 2003, "DiST: A Simple, Reliable and Scalable Method to Significantly Reduce Processor Architecture Simulation Time".
- Albert Cohen, Sylvain Girbal and Olivier Temam: participation to the joint FCRC'03 conference meeting in San-Diego, june 2003, and more specifically conferences PLDI, ISCA, SIGMetrics, PPoPP, SAS, LCTES, and workshops IVME, WCAE, WCED and NSC.
- Albert Cohen: article and presentation to the LCPC'O3 conference (Languages and Compilers for Parallel Computers), College-Station, Texas, october 2003, "Putting Polyhedral Loop Transformations to Work".
- Cédric Bastoul and Albert Cohen: participation to the LCPC'O3 conference (Languages and Compilers for Parallel Computers), College-Station, Texas, october 2003.
- Albert Cohen and Sid-Ahmed-Ali Touati: invited participation and presentations to the Dagstuhl seminar on future challenges in compilation and optimization (organized by Christine Eisenbeis, Mary-Lou Soffa and Tom Conte), february 2003.
- Albert Cohen and Paul Feautrier: invited participation and presentations to the Dagstuhl seminar on metaprogramming, skeletons, generative programming and domain-specific languages (organized by Don Batory, Christian Lengauer and Charles Consel), march 2003.
- Albert Cohen: invited participation and presentation to the Dagstuhl seminar on memory consistency models (organized by Sam Midkiff, Jens Knoop, David Padua and Jae-Jin Lee), october 2003.
- Albert Cohen: presentations at HP Labs Palo Alto (june 2003), University of Illinois at Urbana-Champaign (june 2003), Philips National Labs Eindhoven (december 2003), University of Passau (december 2003).
- Cédric Bastoul: article and presentation at the IEEE ISPDC'03 conference (Parallel and Distributed Computing), Ljubljana, october 2003, "Efficient Code Generation for Automatic Parallelization and Optimization".
- Cédric Bastoul: article and presentation at the CC'03 conference (Compiler Construction), Warsaw, april 2003, "Improving data locality by chunking".
- Cédric Bastoul: seminar presentation at École Normale Supérieure de Lyon, september 2003.

10. Bibliography

Major publications by the team in recent years

[1] P. CLAUSS. *Counting Solutions to Linear and Nonlinear Constraints through Ehrhart polynomials: Applications to Analyze and Transform Scientific Programs.* in « ACM Int. Conf. on Supercomputing », ACM, May, 1996.

[2] A. COHEN. *Program Analysis and Transformation: from the Polytope Model to Formal Languages / Analyse et transformation de programmes : du modèle polyédrique aux langages formels.* Ph. D. Thesis, Université Versailles-Saint-Quentin-en-Yvelines, December, 1999.

[3] A. COHEN, D. GENIUS, A. KORTEBI, Z. CHAMSKI, M. DURANTON, P. FEAUTRIER. *Multi-periodic Process Networks: Prototyping and Verifying Stream-Processing Systems.* volume 2400, pages 137–146, 2002, http://link.springer-ny.com/link/service/series/0558/bibs/2400/24000137.htm ; http://link.springer-ny.com/link/service/series/0558/papers/2400/24000137.pdf.

[4] A. COHEN, O. TEMAM. *Digital LC-2: From Bits & Gates to a Little Computer.* in « Proc. of the 9þ Workshop on Computer Architecture Education (WCAE'02) », Anchorage, Alaska, USA, May, 2002.

[5] J.-F. COLLARD, D. BARTHOU, P. FEAUTRIER. *Fuzzy array dataflow analysis.* in « Proc. of 5th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming », Santa Barbara, CA, July, 1995.

[6] I. DJELIC. *Élimination de redondances partielles pour architectures EPIC.* in « Technique et science informatiques », février, 2002.

[7] C. EISENBEIS, W. JALBY, A. LICHNEWSKY. *Compiler techniques for optimizing memory and register usage on the CRAY2.* in « International Journal on High Speed Computing », number 2, volume 2, 1990, pages 193-222, http://www.inria.fr/rrrt/rr-1302.html, appeared also as INRIA Research Report no 1302 October 1990.

[8] C. EISENBEIS, W. JALBY, D. WINDHEISER, F. BODIN. *A Strategy for Array Management in Local Memory.* in « Mathematical Programming », volume 63, 1994, pages 331-370, Special Issue on Applications of Discrete Optimization in Computer Science.

[9] C. EISENBEIS, S. LELAIT. *LoRA: a Package for Loop Optimal Register Allocation.* Rapport de recherche, number 3709, INRIA, Rocquencourt, juin, 1999, http://www.inria.fr/rrrt/rr-3709.html.

[10] P. FEAUTRIER. *Parametric integer programming.* in « RAIRO Recherche Opérationnelle », number 3, volume 22, 1988, pages 243-268.

[11] P. FEAUTRIER. *Dataflow Analysis of Scalar and Array References.* in « Int. J. of Parallel Programming », number 1, volume 20, February, 1991, pages 23-53.

[12] S. LELAIT. *Contribution à l'allocation de registres dans les boucles.* Thèse de Doctorat, Université d'Orléans, January, 1996.

[13] K. S. MCKINLEY, O. TEMAM. *A Quantitative Analysis of Loop Nest Locality.* in « ASPLOS' 96 », Cambridge, Massachussets, October, 1996.

[14] A. SAWAYA. *Pipeline Logiciel: Découplage et Contraintes de Registres.* Ph. D. Thesis, Université de Versailles - INRIA Rocquencourt, 1997.

[15] J. WANG, C. EISENBEIS, M. JOURDAN, B. SU. *DEcomposed Software Pipelining: a New Perspective and a New Approach.* in « International Journal on Parallel Processing », number 3, volume 22, 1994, pages 357-379, Special Issue on Compilers and Architectures for Instruction Level Parallel Processing.

Articles in referred journals and book chapters

[16] Z. CHAMSKI, M. DURANTON, A. COHEN, C. EISENBEIS, P. FEAUTRIER, D. GENIUS. *Ambient Intelligence: Impact on Embedded-System Design.* To appear in Kluwer Academic Press, 2003, chapter Application Domain-Driven System Design for Pervasive Video Processing.

Publications in Conferences and Workshops

[17] P. AMIRANOFF, A. COHEN, P. FEAUTRIER. *Instancewise Array Dependence Test for Recursive Programs.* in « Workshop on Compilers for Parallel Computers », Amsterdam, january, 2003.

[18] C. BASTOUL. *Efficient code generation for automatic parallelization and optimization.* in « ISPDC'2 IEEE International Symposium on Parallel and Distributed Computing, proceedings to appear », Ljubjana, october, 2003.

[19] C. BASTOUL, A. COHEN, S. GIRBAL, S. SHARMA, O. TEMAM. *Putting Polyhedral Loop Transformations to Work.* in « Workshop on Languages and Compilers for Parallel Computing (LCPC'03) », series LNCS, College Station, Texas, October, 2003.

[20] C. BASTOUL, P. FEAUTRIER. *Improving data locality by chunking.* in « CC'12 International Conference on Compiler Construction, LNCS 2622 », pages 320–335, Warsaw, april, 2003.

[21] C. BASTOUL, P. FEAUTRIER. *Reordering methods for data locality improvement.* in « Workshop on Compilers for Parallel Computers », pages 187–196, Amsterdam, january, 2003.

[22] P. CLAUSS, B. K. YOUTA, O. ROUSSELET. *Modeling Program Traces with Nested Loops.* in « Workshop on Exploring the Trace Space for Dynamic Optimization Techniques (in conjunction with ICS'03 », San Francisco, june, 2003.

[23] S. GIRBAL, G. MOUCHARD, A. COHEN, O. TEMAM. *DiST: A Simple, Reliable and Scalable Method to Significantly Reduce Processor Architecture Simulation Time.* in « (SIGMETRICS'03) », San Diego, California, June, 2003.

[24] S.-A.-A. TOUATI, C. EISENBEIS. *Early Control of Register Pressure for Software Pipelined Loops.* in « International Conference on Compiler Construction », series Lecture Notes in Computer Science, Springer-Verlag, G. HEDIN, editor, Warsaw, Poland, 2003.

Internal Reports

[25] P. AMIRANOFF. *Instancewise Program Analysis.* Technical report, INRIA, 2004, to appear.

[26] Z. CHAMSKI, A. COHEN, M. DURANTON, C. EISENBEIS, P. FEAUTRIER, D. GENIUS, L. PASQUIER, V. RIVIERRE-VIER, F. THOMASSET, Q. ZHAO. *The SANDRA project: cooperative architecture/compiler technology for embedded real-time streaming applications.* Technical report, number 4773, INRIA, mars, 2003, http://www.inria.fr/rrrt/rr-4773.html.

Miscellaneous

[27] M. BELGUIDOUM. *Spécification de noyaux de calcul de traitement vidéo en Lucid synchrone.* rapport de DESS, CNAM/Paris 6, septembre, 2003.

[28] P. CARRIBAULT. *Optimisation de code de Pattern Matching sur l'architecture EPIC : transformations, expériences, automatisation.* rapport de DEA, Université Paris-Sud, septembre, 2003.

[29] P. CLAUSS, I. TCHOUPAEVA. *A Symbolic Approach to Bernstein Expansion for Program Analysis and Optimization.* à paraître, 2004, CC'04, International Conference on Compiler Construction.

[30] *Action spécifique 82 du CNRS, Compilation pour les systèmes embarqués, rapport final de synthèse.* décembre, 2003.

Bibliography in notes

[31] P. COUSOT, R. COUSOT. *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.* in « 4th POPL, Los Angeles, CA », pages 238-252, January, 1977.

[32] M. DAI. *Transformation de code de haut niveau.* Ph. D. Thesis, Université Versailles-Saint-Quentin-en-Yvelines, 9 mai 2000.

[33] D. S. K. D. COOPER AND, L. TORCZON. *Adaptive optimizing compilers for the 21st century.* in « J. of Supercomputing », 2002.

[34] M. O'BOYLE, P. KNIJNENBURG, G. FURSIN. *Feedback Assisted Iterative Compiplation.* in « Parallel Architectures and Compilation Techniques (PACT'01) », IEEE Computer Society Press, October, 2001.