IN PARTNERSHIP WITH:
Institut polytechnique de Grenoble

Université Grenoble Alpes

Activity Report 2016

Project-Team CORSE

Compiler Optimization and Run-time SystEms

IN COLLABORATION WITH: Laboratoire d’Informatique de Grenoble (LIG)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Architecture, Languages and Compilation

Table of contents

Project-Team CORSE

Creation of the Team: 2014 November 01, updated into Project-Team: 2016 July 01

CORSE *is located at Giant/Minatec in Grenoble.*

Keywords:

Computer Science and Digital Science:

1.1.1. - Multicore
1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
1.1.3. - Memory models
1.1.4. - High performance computing
1.1.5. - Exascale
1.1.10. - Reconfigurable architectures
1.1.12. - Non-conventional architectures
1.6. - Green Computing
2.1.7. - Distributed programming
2.1.9. - Dynamic languages
2.1.10. - Domain-specific languages
2.2. - Compilation
2.2.1. - Static analysis
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
2.4.1. - Analysis
6.2.7. - High performance computing
7.1. - Parallel and distributed algorithms
7.3. - Optimization
7.6. - Computer Algebra
7.9. - Graph theory

Other Research Topics and Application Domains:

3.2. - Climate and meteorology
3.3.1. - Earth and subsoil
4.5.1. - Green computing
5.3. - Nanotechnology
6.1.2. - Software evolution, maintenance
6.6. - Embedded systems
6.7. - Computer Industry (harware, equipments...)
9.1. - Education
9.6. - Reproducibility

1. Members

Research Scientists

Fabrice Rastello [Team leader, Inria, Research Scientist, Senior Researcher, HDR]
Frederic Desprez [Inria, Research Scientist, Senior Researcher, HDR]

Faculty Members

Florent Bouchez - Tichadou [Univ. Grenoble Alpes, Associate Professor]
Francois Broquedis [INP Grenoble Alpes, Associate Professor]
Ylies Falcone [Univ. Grenoble Alpes, Associate Professor]
Alain Ketterlin [Univ. Strasbourg, Associate Professor]
Jean Francois Mehaut [Univ. Grenoble Alpes, Professor, HDR]

Engineers

Kevin Pouget [Univ. Grenoble Alpes, DEMA/Nano2017]
Cyril Six [Inria, Internship then engineers]

PhD Students

Georgios Christodoulis [Univ. Grenoble I]
Antoine El Hokayem [Univ. Grenoble Alpes]
Luis Felipe Garlet Millani [Univ. Grenoble Alpes, Brazil CNPq]
François Gindraud [Univ. Grenoble Alpes, until Aug 2016]
Fabian Gruber [Univ. Grenoble Alpes]
Raphael Jakse [Univ. Grenoble. Alpes, from Feb 2016]
Thomas Messi Nguele [Cotutelle Univ. Grenoble Alpes, Univ. Yaoundé 1]
Diogo Nunes Sampaio [Inria, UFMG]
Emmanuelle Saillard [Inria, H2020/HPC4E, Post Doctoral Fellow, from Dec 2016]
Duco Van Amstel [Inria, until Jun 2016]
Philippe Virouleau [Inria]
Ye Xia [Orange Labs]
Naweiluo Zhou [Univ. Grenoble Alpes, HPES/Persyval, until Nov 2016]
Nassim Halli [Univ. Grenoble Alpes, Aselta, until Oct 2016]

Post-Doctoral Fellow

Brice Videau [CNRS, FP7/Mont-Blanc, until Oct 2016]

Visiting Scientists

Henrique Cota de Freitas [PUC Minas, Brazil Capes, until Jul 2016]
Rogerio Goncalves [PhD student at University of Sao Paulo, Brazil CNPq, from Apr 2015 until March 2016]
Julien Langou [UC Denver]

Administrative Assistants

Julie Bourget [Inria, until Jul 2016]
Maria Immaculada Presseguer [Inria]

Others

Léa Albert [Inria, Internship, until Jul 2016]
Ali Cherri [UGA, Internship, from Feb 2016 until Aug 2016]
Nils Defauw [UGA, Internship, from Jun 2016 until Jul 2016]
Nora Hagmeyer [Inria, Internship, IPL C2S@Exa, from Aug 2016]
Erick Lavoie [Inria, Internship, from Jun 2016 until Sep 2016]
Antoine Pouille [ENS Lyon, Internship, until Feb 2016]
Nicolas Tollenaere [Inria, Internship then Engineers, from Apr 2016]
Laurent Zominy [Univ. Grenoble Alpes CNRS, FP7/Mont-Blanc, from Apr 2016 until Sep 2016]

2. Overall Objectives

2.1. Overall Objectives

Languages, compilers, and run-time systems are some of the most important components to bridge the gap between applications and hardware. With the continuous increasing power of computers, expectations are evolving, with more and more ambitious, *computational intensive and complex applications*. As desktop PCs are becoming a niche and servers mainstream, three categories of computing impose themselves for the next decade: mobile, cloud, and super-computing. Thus *diversity, heterogeneity* (even on a single chip) and thus also *hardware virtualization* is putting more and more pressure both on compilers and run-time systems. However, because of the energy wall, *architectures* are becoming more and more *complex* and *parallelism ubiquitous* at every level. Unfortunately, the memory-CPU gap continues to increase and energy consumption remains an important issue for future platforms. To address the challenge of *performance and energy consumption* raised by silicon companies, compilers and run-time systems must *evolve* and, in particular, interact, *taking into account the complexity of the target architecture*.

The overall objective of CORSE is to address this challenge by *combining static and dynamic compilation* techniques, with more interactive *embedding of programs and compiler environment in the runtime system*.

3. Research Program

3.1. Scientific Foundations

One of the characteristics of CORSE is to base our researches on diverse advanced mathematical tools. Compiler optimization requires the usage of the several tools around discrete mathematics: combinatorial optimization, algorithmic, and graph theory. The aim of CORSE is to tackle optimization not only for regular but also for irregular applications. We believe that new challenges in compiler technology design and in particular for split compilation should also take advantage of graph labeling techniques. In addition to runtime and compiler techniques for program instrumentation, hybrid analysis and compilation advances will be mainly based on polynomial and linear algebra.

The other specificity of CORSE is to address technical challenges related to compiler technology, runtime systems, and hardware characteristics. This implies mastering the details of each. This is especially important as any optimization is based on a reasonably accurate model. Compiler expertise will be used in modeling applications (e.g. through automatic analysis of memory and computational complexity); Runtime expertise will be used in modeling the concurrent activities and overhead due to contention (including memory management); Hardware expertise will be extensively used in modeling physical resources and hardware mechanisms (including synchronization, pipelines, etc.).

The core foundation of the team is related to the combination of static and dynamic techniques, of compilation, and runtime systems. We believe this to be essential in addressing high-performance and low energy challenges in the context of new important changes shown by current application, software, and architecture trends.

Our project is structured along two main directions. The first direction belongs to the area of runtime systems with the objective of developing strong relations with compilers. The second direction belongs to the area of compiler analysis and optimization with the objective of combining dynamic analysis and optimization with static techniques. The aim of CORSE is to ground those two research activities on the development of the end-to-end optimization of some specific domain applications.

4. Application Domains

4.1. Transfer

The main industrial sector related to the research activities of CORSE is the one of semi-conductor (programmable architectures spanning from embedded systems to servers). Obviously any computing application which has the objective of exploiting as much as possible the resources (in terms of high-performance but also low energy consumption) of the host architecture is intended to take advantage of advances in compiler and runtime technology. These applications are based over numerical kernels (linear algebra, FFT, convolution...) that can be adapted on a large spectrum of architectures. Members of CORSE already maintain fruitful and strong collaborations with several companies such as STMicroelectronics, Bull, Kalray, or Aselta.

Applying our techniques to a specific real application domain is cherished by all members of the team. In particular we believe (multi-scale) computational mechanics (such as fluid mechanics, molecular dynamics) to be a challenging domain that could take advantage both of compiler and run-time technologies that we intend to develop in CORSE. The goal is to provide an end-to-end solution to the automatic optimization (thus targeting portability of optimized code) of a specific application that requires extensive computational power. If we succeed our research should contribute indirectly to advances in that domain. We are still in the process of prospecting for the most appropriate application.

5. New Software and Platforms

5.1. Tirex

TIREX is an extensible, textual intermediate code representation that is intended to be used as an exchange format for compilers and other tools working on low level code. In the scope of the TIREX project we have developed tools for generating TIREX code from higher level languages such as C, as well as a number of static analyses and transformations.

Work on the TIREX project consisted of two main parts, firstly creation of a machine description library for all parts of the TIREX project, secondly, the development of tools for parsing assembly code.

We developed `archinfo`, a LLVM based library that allows programatic access to descriptors for a target CPUs instructions and registers. The focus was to expose information that was not already available from LLVM, such as machine operand types (float or integer, bitwidth, ...) and flags describing the high level behaviour of the instructions.

The, also LLVM based, assembly parser is intended to be used for translating assembly files generated by common compilers to TIREX, but it can also handle a number of idioms usually found in hand written assembly code. It reconstructs some high level information required for the TIREX format, such as the control flow and call graph, from the assembly code. We also started investigating how our existing tools can be extended to directly parse binary code and reconstruct information from them.

5.2. QEMU plugins

We have collaborated with STMicroelectronics on extending the QEMU CPU emulator with a plugin system. These plugins allow users to observe and modify the machine code emitted by QEMUs binary translator.

We have leveraged this to start development on a number of tools for profiling and performance debugging.

- cachesim: A QEMU plugin that feeds memory accesses observed during program execution into the DineroIV cache simulator. This allows estimating the number of a cache misses caused by each instruction of a program. Using this information we can also estimate the amount of memory bandwidth required by a program. This in turn can be used to diagnose if the applications performance is constrained by memory or CPU resources.
- dep-rate: A QEMU plugin that uses a shadow memory to detects data dependencies between instructions and correlates them with cache misses reported by DineroIV to estimate the performance impact of these dependencies.
- cpath: A QEMU plugin that estimates the optimal execution time of a program on an infinitely parallel CPU and compares it to that for a more realistic model of a CPU. This comparison is used to judge the amount of instruction level parallelism existing in a program.

5.3. Givy

Givy is a runtime developed as part of the PhD thesis of François Gindraud. It is designed for architectures with distributed memories, with the Kalray MPPA as the main target. It executes dynamic data-flow task graphs, annotated with memory dependencies. It automatically handles scheduling and placement of tasks (using the memory dependency hints), and generate memory transfers between distributed memory nodes when needed by using a software cache coherence protocol. An important part of the work corresponds on implementing and testing a memory allocator with specific properties that is a building block of the whole run time. This memory allocator is also tuned to work on the MPPA and its constraints, turning with very little memory and being efficient in the context of multith readed calls.

5.4. Dynamic Dependence Graph (DDG)

By instrumenting the memory accesses, at the LLVM IR level, of a hand selected region of a program, the DDG tool builds a graph with all dynamic instructions. Each instruction, i.e. a node in the graph, is identified by a statement identifier, mapping the dynamic instruction to a static statement, and an induction vector, containing the trip counters of loops surrounding the related statement. Edges connecting these nodes represent either data dependence, reuse or anti-dependence among the instructions, obtained by using the shadow memory technique, that labels ownership to a given written memory position to a dynamic instruction, and creating relationship to it to instructions that read the exact same memory position. Instructions that have a statically known formula (SCEVs) are not tracked, allowing our technique to remove, for example, obvious dependencies from a loop iteration to the next, and still track integer instructions. As the number of dynamic instructions, even in very simple applications, grows extremely fast, the generated graph does not to fit in main memory just after a few hundred loop iterations, our tool allows limiting the number of loop iterations that are tracked. Dependencies between iterations outside the observed iteration space can either be ignored or clamped as being generated by a single instruction. The generated graph can be used to guide loop optimizers, that could not extract precise dependencies. It can also be used by performance debugging tools, in order to determine if it is possible to obtain a new instruction schedule that would improve locality.

5.5. Integer polynomial Fourier-Motzkin elimination

Quantifier elimination is the process of removing existential variables of a given formula, obtaining one that is simpler in the number of variables, and that is implied by the original formula. A very well known algorithm is the Fourier-Motzkin elimination process, that given a system (or formula) of inequalities removes variables by combining all upper and lower bounds of such variables. At each step a variable is selected and eliminated. The very first limitation of this algorithm is the fact that it is designed for linear systems, where all coefficients of the variable being eliminated are numeric values, and the inequality can be classified as either a upper or lower bound. When dealing with polynomials, all possible values, positive, negative, or zero, for an coefficient, that is, a symbolic expression, must be explored. To avoid this requirement we use the positiveness algorithm, proposed by Mark Schweighofer, to retrieve symbolic coefficient signs. In fact, this algorithm is of major importance when resolving system over integer variables, instead of reals, as it is used in many other techniques required to preserve the precision of the simplified formula, such as symbolic normalization, convex hull detection, redundancy removing. Our C++ implementation uses GiNaC for symbolic expressions manipulation.

5.6. BOAST: Metaprogramming of Computing Kernels

BOAST aims at providing a framework to metaprogram, benchmark and validate computing kernels. BOAST is a programming framework dedicated to code generation and autotuning. This software allows the transformation from code written in the BOAST DSL to classical HPC targets like FORTRAN, C, OpenMP, OpenCL or CUDA. It also enables the meta-programming of optimization that can be (de)activated when needed. BOAST can also benchmark and do non regression tests on the generated kernels. This approach gives, both, performance gains and improved performance portability.

BOAST can be dowloaded at this address https://forge.imag.fr/projects/boast/.

BOAST was already used to generate and optimize the computing kernels of three scientific applications:

- BigDFT: A massively parallel electronic structure code using wavelet basis set.
- SPECFEM: Computational Infrastructure for Geodynamics.
- Gysela: Fusion plasma simulations.

BOAST is currently used in the context of the European H2020/HPC4E project. The computing kernels of two scientific applications are currently studied with BOAST:

- Alya: Large Scale Computational Mechanics.
- Hou10ni: Solutions to accoustics wave propagation problems. This code is developed by the Magique3D Inria team (Pau, Julien Diaz).

Frédéric Desprez presented BOAST at the CSCD workshop http://www.netlib.org/utk/people/JackDongarra/CCDSC-2016/ in October 2016. After this workshop, a paper was submited at the Internationaj Journal on High Performance Computing Applications (IJHPCA).

BOAST was also used in the Bulldog project during the last CERMACS summer school http://smai.emath.fr/cemracs/cemracs16/ in July 2016. A joint paper with CEA researchers from Cadarache and Maison de la Simulation was also submitted to present the results of the Bulldog project.

5.7. mcGDB: Interactive debugging of OpenMP programs

MCGDB introduced the concept of *programming-model centric* source-level interactive debugging as an extension of the traditional language-level interactive debugging. The idea was to integrate into debuggers the notion of *programming models*, as abstract machines running over the physical ones. These abstract machines, implemented by runtime libraries and programming frameworks, provide the high-level primitives required for the implementation of today's parallel applications.

We developed a proof-of-concept, mcGDB, as a Python extension of GDB, the debugger of the GNU project. mcGDB was initially developed by Kevin Pouget during his thesis with STMicroelectronics. mcGDB is currently extended with the Nano2017/DEMA project.

We proposed the new support of mcGDB for OpenMP task-based programming. This support consists of task-based execution representation and control improvements, in cooperation with Temanejo graphical debugger. We also studied import implementation details of mcGDB, related to the support of multiple OpenMP environments and CPU architectures; the separation of cross-cutting concerns (user interaction and execution representing) through aspect-oriented programming, and the first steps of mcGDB micro-benchmarking.

mcGDB [30] was presented at the second OpenMPCon developpers conference in Nara.

6. New Results

6.1. Simplification and Run-time Resolution of Data Dependence Constraints for Loop Transformations

Participants: Diogo Nunes Sampaio, Alain Ketterlin [Inria CAMUS], Louis-Noël Pouchet [CSU, USA], Fabrice Rastello.

Loop optimizations such as tiling, thread-level parallelization or vectorization are essential transformations to improve performance. It is needed to compute dependence information at compile-time to assess their validity, but in many real situations, static dependence analysis fails to provide precise enough information. Part of the reason for this failure comes from the need to handle polynomial constraints in the dependence computation problem: such polynomial constraints can arise from linearized array accesses, typical in compilers IR such as LLVM-IR. In this scenario, the compiler will often be unable to apply aggressive transformations due to lack of conclusive static dependence analysis. This work tackles the problem of eliminating quantifiers in systems of inequalities using polynomial constraints. In particular, we design a quantifier elimination scheme on integer multivariate-polynomials, which can aid application of off-the-shelf polyhedral transformations on a larger class of programs, that holds polynomial memory access and affine loop bounds. We make a significant leap in accuracy compared to prior approaches, enabling to implement a hybrid optimizing compilation scheme. In this scheme, a test is evaluated at run-time to determine the legality of the program transformation chosen by the compiler, falling back to executing the original code if the test fails. This test integrates all may-dependences, involving polynomial inequalities, and is simplified by quantifier elimination at compile-time using our techniques. The preciseness of the presented scheme and the low run-time overhead of the test are key to make this approach realistic. We experimentally validate our technique on 25 benchmarks using complex loop transformations, achieving negligible overhead. Preciseness is assessed by the observed success of generated test in practical cases. We compare our variable elimination technique to other existing tools and demonstrate we achieve better precision when dealing with polynomial memory accesses.

This work is the fruit of the collaboration 8.4 with OSU.

6.2. A bounded memory allocator for software-defined global address spaces

Participants: François Gindraud, Fabrice Rastello, Albert Cohen [ENS Ulm], Francois Broquedis.

This work is about the design of a memory allocator targeting manycore architectures with distributed memory. Among the family of Multi Processor System on Chip (MPSoC), these devices are composed of multiple nodes linked by an on-chip network; most nodes have multiple processors sharing a small local memory. While MPSoC typically excel on their performance-per-Watt ratio, they remain hard to program due to multilevel parallelism, explicit resource and memory management, and hardware constraints (limited memory, network topology).

Typical programming frameworks for MPSoC leave much target-specific work to the programmer: combining threads or node-local OpenMP, software caching, explicit message passing (and sometimes, routing), with non-standard interfaces. More abstract, automatic frameworks exist, but they target large-scale clusters and do not model the hardware constraints of MPSoC.

This memory allocator is one component of a larger runtime system, called Givy 5.3, to support dynamic task graphs with automatic software caching and data-driven execution on MPSoC. To simplify the programmer's view of memory, both runtime and program data objects live in a Global Address Space (GAS). To avoid address collisions when objects are dynamically allocated, and to manage virtual memory mappings across nodes, a GAS-aware memory allocator is required. This work proposes such an allocator with the following properties: (1) it is free of inter-node synchronizations; (2) its node-local performance match that of state-of-the-art shared-memory allocators; (3) it provides node-local mechanisms to implement inter-node software caching within a GAS; (4) it is well suited for small memory systems (a few MB per node).

This work has been presented at the international conference ISMM 2016 [16].

6.3. On Fusing Recursive Traversals of K-d Trees

Participants: Samyam Rajbhandari [OSU, USA], Jinsung Kim [OSU, USA], Sriram Krishnamoorthy [PNNL, USA], Louis-Noel Pouchet [CSU, USA], Fabrice Rastello, Robert J. Harrison [Stony Brook, USA], P. Sadayappan [OSU, USA].

Loop fusion is a key program transformation for data locality optimization that is implemented in production compilers. But optimizing compilers for imperative languages currently cannot exploit fusion opportunities across a set of recursive tree traversal computations with producer-consumer relationships. In this work, we develop a compile-time approach to dependence characterization and program transformation to enable fusion across recursively specified traversals over k-d trees. We present the FuseT source-to-source code transformation framework to automatically generate fused composite recursive operators from an input program containing a sequence of primitive recursive operators. We use our framework to implement fused operators for MADNESS, Multiresolution Adaptive Numerical Environment for Scientific Simulation. We show that locality optimization through fusion can offer significant performance improvement.

This work is the fruit of the collaboration 8.4 with OSU. The specific work on FuseT has been presented to the international conference CC 2016 [32] and the more general work on the improvement of MADNESS at the ACM/IEEE international conference SC 2016 [20].

6.4. Effective Padding of Multidimensional Arrays to Avoid Cache Conflict Misses

Participants: Changwan Hong [OSU, USA], Wenlei Bao [OSU, USA], Albert Cohen [Inria PARKAS], Sriram Krishnamoorthy [PNNL, USA], Louis-Noel Pouchet [CSU, USA], Fabrice Rastello, J. Ramanujam [LSU, USA], P. Sadayappan [OSU, USA].

Caches are used to significantly improve performance. Even with high degrees of set associativity, the number of accessed data elements mapping to the same set in a cache can easily exceed the degree of associativity. This can cause conflict misses and lower performance, even if the working set is much smaller than cache capacity. Array padding (increasing the size of array dimensions) is a well-known optimization technique that can reduce conflict misses. In this work, we develop the first algorithms for optimal padding of arrays aimed at a set-associative cache for arbitrary tile sizes. In addition, we develop the first solution to padding for nested tiles and multi-level caches. Experimental results with multiple benchmarks demonstrate a significant performance improvement from padding.

This work is the fruit of the collaboration 8.4 with OSU. It has been presented at the ACM international conference PLDI 2016 [29].

6.5. PolyCheck: Dynamic Verification of Iteration Space Transformations on Affine Programs

Participants: Sriram Krishnamoorthy [PNNL], Bao Wenlei [OSU], Louis-Noël Pouchet [UCLA], P. Sadayappan [OSU], Fabrice Rastello.

High-level compiler transformations, especially loop transformations, are widely recognized as critical optimizations to restructure programs to improve data locality and expose parallelism. Guaranteeing the correctness of program transformations is essential, and to date three main approaches have been developed: proof of equivalence of affine programs, matching the execution traces of programs, and checking bit-by-bit equivalence of program outputs. Each technique suffers from limitations in the kind of transformations supported, space complexity, or the sensitivity to the testing dataset. In this work, we take a novel approach that addresses all three limitations to provide an automatic bug checker to verify any iteration reordering transformations on affine programs, including non-affine transformations, with space consumption proportional to the original program data and robust to arbitrary datasets of a given size. We achieve this by exploiting the structure of affine program control- and data-flow to generate at compile-time lightweight checker code to be executed within the transformed program. Experimental results assess the correctness and effectiveness of our method and its increased coverage over previous approaches.

This work is the fruit of the collaboration 8.4 with OSU and was presented at ACM POPL'16 [14].

6.6. Modularizing Crosscutting Concerns in Component-Based Systems

Participants: Antoine El-Hokayem, Yliès Falcone, Mohamad Jaber [American University of Beirut, Lebanon].

We define a method to modularize crosscutting concerns in the Behavior Interaction Priority (BIP) component-based framework. Our method is inspired from the Aspect Oriented Programming (AOP) paradigm which was initially conceived to support the separation of concerns during the development of monolithic systems. BIP has a formal operational semantics and makes a clear separation between architecture and behavior to allow for compositional and incremental design and analysis of systems. We thus distinguish local from global aspects. Local aspects model concerns at the component level and are used to refine the behavior of components. Global aspects model concerns at the architecture level, and hence refine communications (synchronization and data transfer) between components. We formalize global aspects as well as their integration into a BIP system through rigorous transformation primitives and overview local aspects. We present AOP-BIP, a tool for Aspect-Oriented Programming of BIP systems, and demonstrate its use to modularize logging, security, and fault-tolerance in a network protocol.

This work results of the collaboration with American University of Beirut (Lebanon) and was presented at SEFM 2016 [15].

6.7. Predictive runtime enforcement

Participants: Srinivas Pinisetty [Aalto University, Finland], Viorel Preoteasa [Aalto University, Finland], Stavros Tripakis [Aalto University, Finland], Thierry Jéron [Inria Rennes, France], Yliès Falcone, Hervé Marchand [Inria Rennes, France].

Runtime enforcement (RE) is a technique to ensure that the (untrustworthy) output of a black-box system satisfies some desired properties. In RE, the output of the running system, modeled as a stream of events, is fed into an enforcement monitor. The monitor ensures that the stream complies with a certain property, by delaying or modifying events if necessary. This work deals with predictive runtime enforcement, where the system is not entirely black-box, but we know something about its behavior. This a-priori knowledge about the system allows to output some events immediately, instead of delaying them until more events are observed, or even blocking them permanently. This in turn results in better enforcement policies. We also show that if we have no knowledge about the system, then the proposed enforcement mechanism reduces to a classical non-predictive RE framework. All our results are formalized and proved in the Isabelle theorem prover.

This work was presented at SAC-SVT 2016 [19].

6.8. Third International Competition on Runtime Verification

Participants: Giles Reger [University of Manchester, UK], Sylvain Hallé [The University of Québec at Chicoutimi, Canada], Yliès Falcone.

We report on the Third International Competition on Runtime Verification (CRV-2016). The competition was held as a satellite event of the 16th International Conference on Runtime Verification (RV'16). The competition consisted of two tracks: offline monitoring of traces and online monitoring of Java programs. The intention was to also include a track on online monitoring of C programs but there were too few participants to proceed with this track. This report describes the format of the competition, the participating teams, the submitted benchmarks and the results. We also describe our experiences with transforming trace formats from other tools into the standard format required by the competition and report on feedback gathered from current and past participants and use this to make suggestions for the future of the competition.

This work was presented at RV 2016 [13].

6.9. Monitoring Multi-threaded Component-Based Systems

Participants: Hosein Nazarpour [Verimag, France], Yliès Falcone, Saddek Bensalem [Verimag, France], Marius Bozga [Verimag, France], Jacques Combaz [Verimag, France].

This work addresses the monitoring of logic-independent linear-time user-provided properties on multi-threaded component-based systems. We consider intrinsically independent components that can be executed concurrently with a centralized coordination for multiparty interactions. In this context, the problem that arises is that a global state of the system is not available to the monitor. A naive solution to this problem would be to plug a monitor which would force the system to synchronize in order to obtain the sequence of global states at runtime. Such solution would defeat the whole purpose of having concurrent components. Instead, we reconstruct on-the-fly the global states by accumulating the partial states traversed by the system at runtime. We define formal transformations of components that preserve the semantics and the concurrency and, at the same time, allow to monitor global-state properties. Moreover, we present RVMT-BIP, a prototype tool implementing the transformations for monitoring multi-threaded systems described in the BIP (Behavior, Interaction, Priority) framework, an expressive framework for the formal construction of heterogeneous systems. Our experiments on several multi-threaded BIP systems show that RVMT-BIP induces a cheap runtime overhead.

This work was presented at iFM 2016 [18].

6.10. Decentralized Enforcement of Artifact Lifecycles

Participants: Sylvain Hallé [The University of Québec at Chicoutimi, Canada], Raphaël Khoury [The University of Québec at Chicoutimi, Canada], Antoine El-Hokayem, Yliès Falcone.

Artifact-centric workflows describe possible executions of a business process through constraints expressed from the point of view of the documents exchanged between principals. A sequence of manipulations is deemed valid as long as every document in the workflow follows its prescribed lifecycle at all steps of the process. So far, establishing that a given workflow complies with artifact lifecycles has mostly been done through static verification, or by assuming a centralized access to all artifacts where these constraints can be monitored and enforced. We propose an alternate method of enforcing document lifecycles that requires neither static verification nor single-point access. Rather, the document itself is designed to carry fragments of its history, protected from tampering using hashing and public-key encryption. Any principal involved in the process can verify at any time that a document's history complies with a given lifecycle. Moreover, the proposed system also enforces access permissions: not all actions are visible to all principals, and one can only modify and verify what one is allowed to observe.

This work was presented at EDOC 2016 [17].

6.11. Runtime enforcement of regular timed properties by suppressing and delaying events

Participants: Yliès Falcone, Thierry Jéron [Inria Rennes, France], Hervé Marchand [Inria Rennes, France], Srinivas Pinisetty [Aalto University, Finland].

Runtime enforcement is a verification/validation technique aiming at correcting possibly incorrect executions of a system of interest. In this work, we consider enforcement monitoring for systems where the physical time elapsing between actions matters. Executions are thus modelled as timed words (i.e., sequences of actions with dates). We consider runtime enforcement for timed specifications modelled as timed automata. Our enforcement mechanisms have the power of both delaying events to match timing constraints, and suppressing events when no delaying is appropriate, thus possibly allowing for longer executions. To ease their design and their correctness-proof, enforcement mechanisms are described at several levels: enforcement functions that specify the input–output behaviour in terms of transformations of timed words, constraints that should be satisfied by such functions, enforcement monitors that describe the operational behaviour of enforcement functions, and enforcement algorithms that describe the implementation of enforcement monitors. The feasibility of enforcement monitoring for timed properties is validated by prototyping the synthesis of enforcement monitors from timed automata.

This work was published in the journal Science of Computer Programming [8].

6.12. Organising LTL monitors over distributed systems with a global clock

Participants: Christian Colombo [University of Malta, Malta], Yliès Falcone.

Users wanting to monitor distributed systems often prefer to abstract away the architecture of the system by directly specifying correctness properties on the global system behaviour. To support this abstraction, a compilation of the properties would not only involve the typical choice of monitoring algorithm, but also the organisation of submonitors across the component network. Existing approaches, considered in the context of LTL properties over distributed systems with a global clock, include the so-called orchestration and migration approaches. In the orchestration approach, a central monitor receives the events from all subsystems. In the migration approach, LTL formulae transfer themselves across subsystems to gather local information. We propose a third way of organising submonitors: choreography, where monitors are organised as a tree across the distributed system, and each child feeds intermediate results to its parent. We formalise choreography-based decentralised monitoring by showing how to synthesise a network from an LTL formula, and give a decentralised monitoring algorithm working on top of an LTL network. We prove the algorithm correct and implement it in a benchmark tool. We also report on an empirical investigation comparing these three approaches on several concerns of decentralised monitoring: the delay in reaching a verdict due to communication latency, the number and size of the messages exchanged, and the number of execution steps required to reach the verdict.

This work was published in the journal Formal Methods in System Design [6].

6.13. Decentralised LTL monitoring

Participants: Andreas Bauer [TU Munich, Software and Systems Engineering Munich, Germany], Yliès Falcone.

Users wanting to monitor distributed or component-based systems often perceive them as monolithic systems which, seen from the outside, exhibit a uniform behaviour as opposed to many components displaying many local behaviours that together constitute the system's global behaviour. This level of abstraction is often reasonable, hiding implementation details from users who may want to specify the system's global behaviour in terms of a linear-time temporal logic (LTL) formula. However, the problem that arises then is how such a specification can actually be monitored in a distributed system that has no central data collection point, where all the components' local behaviours are observable. In this case, the LTL specification needs to be decomposed into sub-formulae which, in turn, need to be distributed amongst the components' locally attached monitors, each of which sees only a distinct part of the global behaviour. The main contribution of this work is an algorithm for distributing and monitoring LTL formulae, such that satisfaction or violation of specifications can be detected by local monitors alone. We present an implementation and show that our algorithm introduces only a negligible delay in detecting satisfaction/violation of a specification. Moreover, our practical results show that the communication overhead introduced by the local monitors is generally lower than the number of messages that would need to be sent to a central data collection point. Furthermore, our experiments strengthen the argument that the algorithm performs well in a wide range of different application contexts, given by different system/communication topologies and/or system event distributions over time.

This work was published in the journal Formal Methods in System Design [4].

6.14. Using data dependencies to improve task-based scheduling strategies on NUMA architectures

Participants: Philippe Virouleau, François Broquedis, Thierry Gautier [Inria, AVALON], Fabrice Rastello.

The recent addition of data dependencies to the OpenMP 4.0 standard provides the application programmer with a more flexible way of synchronizing tasks. Using such an approach allows both the compiler and the runtime system to know exactly which data are read or written by a given task, and how these data will be used through the program lifetime. Data placement and task scheduling strategies have a significant impact on performances when considering NUMA architectures. While numerous studies focus on these topics, none of them has made extensive use of the information available through dependencies. One can use this information to modify the behavior of the application at several levels : during initialization to control data placement and during the application execution to dynamically control both the task placement and the tasks stealing strategy, depending on the topology. This work introduces several heuristics for these strategies, their implementations in the xkaapi OpenMP runtime system and the performances on linear algebra applications executed on a 192-core NUMA machine. Such approaches report noticeable performance improvement when considering both the architecture topology and the tasks data dependencies.

This work has been presented at the international conference EuroPar'2016 [22].

6.15. Description, Implementation and Evaluation of an Affinity Clause for Task Directives

Participants: Philippe Virouleau, Adrien Roussel [IFPEN], François Broquedis, Thierry Gautier [Inria, AVALON], Fabrice Rastello, Jean-Marc Gratien [IFPEN].

This work extends the affinity-based scheduling we proposed at the Europar 2016 conference to fit the philosophy of OpenMP programming. On this topic, OpenMP does not provide a lot of flexibility to the programmer yet, which lets the runtime system decide where a task should be executed. In this work, we propose our own interpretation of the new affinity clause for the task directive, which is being discussed by the OpenMP Architecture Review Board. This clause enables the programmer to give hints to the runtime about tasks placement during the program execution, which can be used to control the data mapping on the architecture. In our proposal, the programmer can express affinity between a task and the following resources: a thread, a NUMA node, and a data. We provide an implementation of this proposal in the Clang-3.8 compiler, and an implementation of the corresponding extensions in the xkaapi OpenMP runtime system.

This work has been presented at the international workshop on OpenMP IWOMP'2016 [23].

6.16. Design methodology for workload-aware loop scheduling strategies based on genetic algorithm and simulation

Participants: Pedro H. Penna [PUC Minas], Márcio Castro [UFSC], Henrique C. Freitas [PUC Minas], François Broquedis, Jean-François Méhaut.

In high-performance computing, the application's workload must be evenly balanced among threads to deliver cutting-edge performance and scalability. In OpenMP, the load balancing problem arises when scheduling loop iterations to threads. In this context, several scheduling strategies have been proposed, but they do not take into account the input workload of the application and thus turn out to be suboptimal. In this work, we introduce a design methodology to propose, study, and assess the performance of workload-aware loop scheduling strategies. In this methodology, a genetic algorithm is employed to explore the state space solution of the problem itself and to guide the design of new loop scheduling strategies, and a simulator is used to evaluate their performance. As a proof of concept, we show how the proposed methodology was used to propose and study a new workload-aware loop scheduling strategy named smart round-robin (SRR). We implemented this strategy into GNU Compiler Collection's OpenMP runtime. We carry out several experiments to validate the simulator and to evaluate the performance of SRR. Our experimental results show that SRR may deliver up to 37.89% and 14.10% better performance than OpenMP's dynamic loop scheduling strategy in the simulated environment and in a real-world application kernel, respectively.

This work is presented in the CCPE journal [9].

6.17. The Mont-Blanc prototype: An Alternative Approach for HPC Systems

Participants: Brice Videau, Kevin Pouget, Jean-François Méhaut.

The evolution of High-Performance Computing (HPC) systems is driven by the need of reducing time-to-solution and increasing the resolution of models and problems being solved by a particular program. Important milestones from the HPC system performance perspective were achieved using commodity technology. Examples are the ASCI Red and the Roadrunner supercomputers, which broke the 1 TFLOPS and 1 PFLOPS barriers, respectively. These systems showed how commodity technology could be used to take the next step in HPC system architecture.

Driven by a much larger market, commodity components evolve faster than their special-purpose counterparts, eventually achieving the same performance and eventually surpassing or replacing them. For this reason, RISC processors displaced vector processors, and x86 displaced RISC.

Nowadays commodity is in the embedded / mobile processor segment. Mobile processors develop fast, and are still not at a point of diminishing performance improvements from new designs. Furthermore, they progressively incorporate the capabilities required for HPC.

The embedded market size and endless customer requirements allow for constant investments into innovative designs, and rapid testing and adoption of new technologies. For example, LPDDR memory technology was first introduced in the mobile domain and has recently been proposed as a memory solution for energy proportional servers.

The Mont-Blanc project aims at providing an alternative HPC system solution based on the current commodity technology: mobile chips. As a demonstrator of such an approach, the project designed, built, and set-up a 1080-node HPC cluster made of Samsung Exynos 5250 SoCs. The Mont-Blanc project established the following goals: to design and deploy a sufficiently large HPC prototype system based on the current mobile commodity technology; to port and optimize the software stack, and enable its use for HPC; to port and optimize a set of HPC applications to be run at this HPC system.

Comparing the Mont-Blanc prototype to a contemporary supercomputer, MareNostrum III, reveals that a single-socket Mont-Blanc node is 9x slower than a dual-socket MareNostrum III node, while saving up to 40% of energy. MPI parallel applications show a 3.5x slowdown when running with the same number of MPI ranks on both machines, while consuming 9% less energy on the Mont-Blanc prototype on average. When targeting the same execution time, the Mont-Blanc prototype offers 12.5% space savings.

This work was funded by the European Commission with the Mont-Blanc projects 8.3.1.1. This scientific result was presented at the SuperComputing Conference SC'2016 in Salt Lake City [31]. The paper was selected as a *best paper finalist*.

6.18. Control of Autonomid Parallelism on Software Transactional Memory

Participants: Naweiluo Zhou, Gwenaël Delaval [Univ. Grenoble Alpes, Associate Professor, Ctrl-A Inria team], Bogdan Robu [Univ. Grenoble Alpes, Associate Professor, Gipsa Laboratory], Eric Rutten [Inria, Rsearcher, Ctrl-A Inria team], Jean-François Méhaut.

Parallel programs need to manage the trade-off between the time spent in synchronization and computation. A high parallelism may decrease computing time while increase synchronization cost among threads. A way to improve program performance is to adjust parallelism to balance conflicts among threads. However, there is no universal rule to decide the best parallelism for a program from an offline view. Furthermore, an offline tuning is error-prone. Hence, it becomes necessary to adopt a dynamic tuning-configuration strategy to better manage a STM system. Software Transactional Memory (STM) has emerged as a promising technique, which bypasses locks, to address syn- chronization issues through transactions. Autonomic computing offers designers a framework of methods and techniques to build automated systems with well-mastered behaviours. Its key idea is to implement feedback control loops to design safe, efficient and predictable controllers, which enable monitoring and adjusting controlled systems dynamically while keeping overhead low. We propose to design feedback control loops to automate the choice of parallelism level at runtime to diminish program execution time.

This work is funded by the Persyval laboratory (LabEx) and the HPES team 8.1.2. This scientific result is part of the Naweiluo Zhou's thesis. The thesis was defended in October 2016 [2]. This work was presented in the HPCS conference [25]. The paper was selected as *best paper finalist*. The Naweiluo Zhou's work is also presented at the ICAC conference.

6.19. Evaluating the SEE sensitivity of a 45nm SOI Multi-core Processor due to 14 MeV Neutrons

Participants: Pablo Ramos [Univ. Grenoble Alpes and ESPE Ecuador, PhD student TIMA Laboratory], Vanessa Vargas [Univ. Grenoble Alpes and ESPE Ecuador, PhD student TIMA Laboratory], Maud Baylac [CNRS, IN2P3, LSPSC Laboratory], Francesca Villa [CNRS, IN2P3, LSPSC Laboratory], Nacer-Eddine Zergainoh [Univ. Grenoble Alpes, Associate Professor, TIMA Laboratory], Jean-François Méhaut, Raoul Velazco [CNRS, Senior Scientist, TIMA Laboratory].

The aim of this work is to evaluate the SEE sensitivity of a multi-core processor having implemented ECC and parity in their cache memories. Two different application scenarios are studied. The first one configures the multi-core in Asymmetric Multi-Processing mode running a memory-bound application, whereas the second one uses the Symmetric Multi-Processsing mode running a CPU-bound application. The experiments were validated through radiation ground testing performed with 14 MeV neutrons on the Freescale P2041 multi-core manufactured in 45nm SOI technology. A deep analysis of the observed errors in cache memories was carried-out in order to reveal vulnerabilities in the cache protection mechanisms. Critical zones like tag addresses were affected during the experiments. In addition, the results show that the sensitivity strongly depends on the application and the multi-processsing mode used.

This work is part of the STIC Amsud EnergySFE project 8.4.3. These results are published in the IEEE Transactions on Nuclear Science [10].

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Grants with Industry

- PSAIC Nano2017 is a bilateral Grant with STMicroelectronics. CORSE is involved in the development of trace analysis and hybrid compilation.
- DEMA Nano2017 is a bilateral Grant with STMicroelectronics. CORSE is involved in the development of debugging of multithreaded applications.

7.2. CIFRE contracts

- CORSE is involved in a contract with Kalray associated with the CIFRE PhD of Duco van Amstel who defended in Spring 2016. The subject of the collaboration is related to fine grain scheduling.
- CORSE is involved in a contract with Aselta for the CIFRE thesis of Nassim Halli. Nassim Halli was advised by Henri-Pierre Charles (CEA LIST, Grenoble and Jean-François Méhaut. The subject of this thesis is the code optimization of Java Applications. The thesis was defended in October 2016.
- CORSE is also involved in a contract with STMicroelectronics for the CIFRE thesis of Oleg Iegorov. The subject of this thesis is a Data Mining Approach to Temporal Debugging of Embedded Streaming Applications. Oleg Iegorov was advised by the SLIDE LIG team and the CORSE Inria team. The thesis was defended in April 2016.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. HEAVEN Persyval Project

- Title: HEterogenous Architectures: Versatile Exploitation and programiNg
- HEAVEN leaders: François Broquedis, Olivier Muller[TIMA lab]
- CORSE participants: François Broquedis, Frédéric Desprez, Georgios Christodoulis
- Computer architectures are getting more and more complex, exposing massive parallelism, hierarchically-organized memories and heterogeneous processing units. Such architectures are extremely difficult to program as they most of the time make application programmers choose between portability and performance.

  While standard programming environments like OpenMP are currently evolving to support the execution of applications on different kinds of processing units, such approaches suffer from two main issues. First, to exploit heterogeneous processing units from the application level, programmers need to explicitly deal with hardware-specific low-level mechanisms, such as the memory transfers between the host memory and private memories of a co-processor for example. Second, as the evolution of programming environments towards heterogeneous programming mainly focuses on CPU/GPU platforms, some hardware accelerators are still difficult to exploit from a general-purpose parallel application.

  FPGA is one of them. Unlike CPUs and GPUs, this hardware accelerator can be configured to fit the application needs. It contains arrays of programmable logic blocks that can be wired together to build a circuit specialized for the targeted application. For example, FPGAs can be configured to accelerate portions of code that are known to perform badly on CPUs or GPUs. The energy efficiency of FPGAs is also one of the main assets of this kind of accelerators compared to GPUs, which encourages the scientific community to consider FPGAs as one of the building blocks of large scale low-power heterogeneous multicore platforms.

  However, only a fraction of the community considers programming FPGAs for now, as configurations must be designed using low-level description languages such as VHDL that application programmers are not experienced with.

  The main objective of this project is to improve the accessibility of heterogeneous architectures containing FPGA accelerators to parallel application programmers. The proposed project focuses on three main aspects:

  - Portability: we don't want application programmers to redesign their applications completely to benefit from FPGA devices. This means extending standard parallel programming environments like OpenMP to support FPGA. Improving application portability also means leveraging most of the hardware-specific low-level mechanisms at the runtime system level ;
  - Performance: we want our solution to be flexible enough to get the most out of any heterogeneous platforms containing FPGA devices depending on specific performance needs, like computation throughput or energy consumption for example ;
  - Experiments: Experimenting with FPGA accelerators on real-life scientific applications is also a key element of our project proposal. In particular, the solutions developed in this project will allow comparisons between architectures on real-life applications from different domains like signal processing and computational finance.

  Efficient programming and exploitation of heterogeneous architectures implies the development of methods and tools for system design, embedded or not. The HEAVEN project proposal fits in the PCS research action of the PERSYVAL-lab. The PhD of Georgios Christodoulis is funded by this project.

8.1.2. HPES Persyval Project

- Title: High Performance Embedded Systems
- HPES leader: Henri-Pierre Charles [CEA List, CRI PILSI]

- HPES participants: Suzane Lesecq [CEA Leti], Laurent Fesquet [TIMA Lab], Stéphane Mancini [TIMA Lab], Eric Ruten [Inria/CtrlA], Nicolas Marchand [Gipsa Lab], Bogdan Robu [Gipsa Lab]
- CORSE participants: Naweiluo Zhou [PhD Persyval], Fabrice Rastello, Jean-François Méhaut
- The computing area has been recently deeply modified by the emergence of the so-called multicore processor. Within the same chip, several computing units are implemented. This architectural concept allows meeting the performance requirements under stringent energy consumption constraints. Multicores are used for laptops, Graphical Processor Units (GPU), High Performance Computing (HPC) platforms, but also for embedded systems su ch as mobile phones. Moreover, low-power high performance multicores developed for embedded systems will be soon used in data centers for HPC. This raises new scientific challenges to architecture, systems and application designers that have face massively parallel computing platforms.

  The number of cores on a chip is increasing quickly. At the same time, the memory bandwidth is increasing too slowly to ensure the performance such multicore platforms should attain. This phenomenon is known as “Memory Wall” and at the moment no efficient solution to exceed this limitation exists. With the increase in the number of cores, cache coherency is becoming as well a tremendous challenge.

  Power consumption is also a huge challenge as it imposes strong constraints on the computing platform, whatever the application domain. The first machine ranked in the Green500 has an energy performance ratio of 2 Gflops per watt. This ratio has to be improved by 30 when exascale computing is considered. The multi-core processor might help to improve this ratio; however, the software stack should as well evolve to boost this improvement.

8.1.3. AGIR DEREVES

- Title: DEcentralised Runtime Verification and Enforcement of distributed and cyber-physical Systems
- DEREVES leader: Ylies Falcone
- CORSE participants: Ylies Falcone, Antoine El-Hokayem, Raphaël Jakse
- DEREVES aims at advancing the theory of decentralised runtime verification and enforce- ment for distributed systems, with the objective of proposing realistic monitoring and monitor-synthesis algorithms for expressive specifications that can be used for the efficient monitoring of multithreaded, dis- tributed and cyber-physical systems. The project shall help transferring runtime verification and enforcement to a wider audience of programmers of distributed systems by providing them techniques and tools to help them guaranteeing the correctness of their systems.

8.2. National Initiatives

8.2.1. IPL C2S@Exa

- Title: Computer and Computational Sciences at Exascale
- C2S@Exa leader: Stéphane Lanteri
- CORSE participants: François Broquedis, Frédéric Desprez, Jean-François Méhaut, Brice Videau, Philippe Virouleau, Nora Hagmeyer
- The C2S@Exa Inria large-scale initiative is concerned with the development of numerical modeling methodologies that fully exploit the processing capabilities of modern massively parallel architectures in the context of a number of selected applications related to important scientific and technological challenges for the quality and the security of life in our society. At the current state of the art in technologies and methodologies, a multidisciplinary approach is required to overcome the challenges raised by the development of highly scalable numerical simulation software that can exploit computing platforms offering several hundreds of thousands of cores. Hence, the main objective of the C2S@Exa Inria large-scale initiative is the establishment of a continuum of expertise in the

computer science and numerical mathematics domains, by gathering researchers from Inria project-teams whose research and development activities are tightly linked to high performance computing issues in these domains. More precisely, this collaborative effort involves computer scientists that are experts of programming models, environments and tools for harnessing massively parallel systems, algorithmists that propose algorithms and contribute to generic libraries and core solvers in order to take benefit from all the parallelism levels with the main goal of optimal scaling on very large numbers of computing entities and, numerical mathematicians that are studying numerical schemes and scalable solvers for systems of partial differential equations in view of the simulation of very large-scale problems.

8.2.2. PIA ELCI

- Title: Environnement logiciel pour le calcul intensif
- ELCI leader: Corinne Marchand (BULL SAS)
- CORSE participants: François Broquedis, Philippe Virouleau
- Duration: from Sept. 2014 to Sept. 2017
- The ELCI project main goal is to develop a highly-scalable new software stack to tackle high-end supercomputers, from numerical solvers to programming environments and runtime systems. In particular, the CORSE team is studying the scalability of OpenMP runtime systems on large scale shared memory machines through the PhD of Philippe Virouleau, co-advised by researchers from the CORSE and AVALON Inria teams. This work intends to propose new approaches based on a compiler/runtime cooperation to improve the execution of scientific task-based programs on NUMA platforms. The PhD of Philippe Virouleau is funded by this project.

8.3. European Initiatives

8.3.1. FP7 & H2020 Projects

8.3.1.1. Mont-Blanc2

Title: Mont-Blanc (European scalable and power efficient HPC platform based on low-power embedded technology)

Program FP7

Duration: 01/10/2013 - 31/01/2017

Coordinator: Barcelona Supercomputing Center (BSC)

Mont-Blanc consortium: BSC, Bull, Arm, Juelich, LRZ, USTUTT, Cineca, CNRS, Inria, CEA Leti, Univ. Bristol, Allinea

CORSE contact: Jean-François Méhaut

CORSE participants: Brice Videau, Kevin Pouget

The Mont-Blanc project aims to develop a European Exascale approach leveraging on commodity power-efficient embedded technologies. The project has developed a HPC system software stack on ARM, and is deployed the first integrated ARM-based HPC prototype by 2014, and is also working on a set of 11 scientific applications to be ported and tuned to the prototype system.

The rapid progress of Mont-Blanc towards defining a scalable power efficient Exascale platform has revealed a number of challenges and opportunities to broaden the scope of investigations and developments. Particularly, the growing interest of the HPC community in accessing the Mont-Blanc platform calls for increased efforts to setup a production-ready environment. The Mont-Blanc 2 proposal has 4 objectives:

1. To complement the effort on the Mont-Blanc system software stack, with emphasis on programmer tools (debugger, performance analysis), system resiliency (from applications to architecture support), and ARM 64-bit support

2. To produce a first definition of the Mont-Blanc Exascale architecture, exploring different alternatives for the compute node (from low-power mobile sockets to special-purpose high-end ARM chips), and its implications on the rest of the system
3. To track the evolution of ARM-based systems, deploying small cluster systems to test new processors that were not available for the original Mont-Blanc prototype (both mobile processors and ARM server chips)
4. To provide continued support for the Mont-Blanc consortium, namely operations of the original Mont-Blanc prototype, the new developer kit clusters and hands-on support for our application developers

Mont-Blanc 2 contributes to the development of extreme scale energy-efficient platforms, with potential for Exascale computing, addressing the challenges of massive parallelism, heterogeneous computing, and resiliency. Mont-Blanc 2 has great potential to create new market opportunities for successful EU technology, by placing embedded architectures in servers and HPC.

8.3.1.2. EoCoE

Title: Energy oriented Centre of Excellence for computer applications

Programm: H2020

Duration: October 2015 - October 2018

Coordinator: CEA

Partners:

Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)

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

Centre Europeen de Recherche et de Formation Avancee en Calcul Scientifique (France)

Consiglio Nazionale Delle Ricerche (Italy)

The Cyprus Institute (Cyprus)

Agenzia Nazionale Per le Nuove Tecnologie, l’energia E Lo Sviluppo Economico Sostenibile (Italy)

Fraunhofer Gesellschaft Zur Forderung Der Angewandten Forschung Ev (Germany)

Instytut Chemii Bioorganicznej Polskiej Akademii Nauk (Poland)

Forschungszentrum Julich (Germany)

Max Planck Gesellschaft Zur Foerderung Der Wissenschaften E.V. (Germany)

University of Bath (United Kingdom)

Universite Libre de Bruxelles (Belgium)

Universita Degli Studi di Trento (Italy)

Inria contact: Michel Kern

The aim of the present proposal is to establish an Energy Oriented Centre of Excellence for computing applications, (EoCoE). EoCoE (pronounce “Echo”) will use the prodigious potential offered by the ever-growing computing infrastructure to foster and accelerate the European transition to a reliable and low carbon energy supply. To achieve this goal, we believe that the present revolution in hardware technology calls for a similar paradigm change in the way application codes are designed. EoCoE will assist the energy transition via targeted support to four renewable energy pillars: Meteo, Materials, Water and Fusion, each with a heavy reliance on numerical modelling. These four pillars will be anchored within a strong transversal multidisciplinary basis providing high-end expertise in applied mathematics and HPC. EoCoE is structured around a central Franco-German hub coordinating a pan-European network, gathering a total of 8 countries and 23 teams. Its partners are strongly engaged in both the HPC and energy fields; a prerequisite for the long-term sustainability of EoCoE and also ensuring that it is deeply integrated in the overall European strategy for HPC. The primary goal of EoCoE is to create a new, long lasting and sustainable community around computational energy science. At the same time, EoCoE is committed to deliver high-impact results within the first three years. It will resolve current bottlenecks in application codes, leading to new modelling capabilities and scientific advances among the four user communities; it will develop cutting-edge mathematical and numerical methods, and tools to foster the usage of Exascale computing. Dedicated services for laboratories and industries will be established to leverage this expertise and to foster an ecosystem around HPC for energy. EoCoE will give birth to new collaborations and working methods and will encourage widely spread best practices.

8.3.1.3. HPC4E

Title: HPC for Energy (HPC4E)

Programm: H2020

Duration: December 2015 - November 2017

Program FP7

Coordinator: Barcelona Supercomputing Center

Partners:

Centro de Investigaciones Energeticas, Medioambientales Y Tecnologicas-Ciemat (Spain)

Iberdrola Renovables Energia (Spain)

Repsol (Spain)

Total S.A. (France)

Lancaster University (United Kingdom)

Inria contact: Stephane Lanteri

CORSE particpants: Jean-François Méhaut, Frédéric Desprez, Emmanuelle Saillard (Post-Doct since Dec 2016)

This project aims to apply the new exascale HPC techniques to energy industry simulations, customizing them, and going beyond the state-of-the-art in the required HPC exascale simulations for different energy sources: wind energy production and design, efficient combustion systems for biomass-derived fuels (biogas), and exploration geophysics for hydrocarbon reservoirs. For wind energy industry HPC is a must. The competitiveness of wind farms can be guaranteed only with accurate wind resource assessment, farm design and short-term micro-scale wind simulations to forecast the daily power production. The use of CFD LES models to analyse atmospheric flow in a wind farm capturing turbine wakes and array effects requires exascale HPC systems. Biogas, i.e. biomass-derived fuels by anaerobic digestion of organic wastes, is attractive because of its wide availability, renewability and reduction of CO2 emissions, contribution to diversification of energy supply, rural development, and it does not compete with feed and food feedstock. However, its use in practical systems is still limited since the complex fuel composition might lead to unpredictable combustion performance and instabilities in industrial combustors. The next generation of exascale HPC systems will be able to run combustion simulations in parameter regimes relevant to industrial applications using alternative fuels, which is required to design efficient furnaces, engines, clean burning vehicles and power plants. One of the main HPC consumers is the oil & gas (O&G) industry. The computational requirements arising from full wave-form modelling and inversion of seismic and electromagnetic data is ensuring that the O&G industry will be an early adopter of exascale computing technologies. By taking into account the complete physics of waves in the subsurface, imaging tools are able to reveal information about the Earth’s interior with unprecedented quality.

8.3.2. Collaborations in European Programs, Except FP7 & H2020

Program: COST

Project acronym: ArVI

Project title: Runtime Verification beyond Monitoring

Duration: December 2014 - May 2017

Coordinator: Martin Leucker, University of Lubeck

Abstract: Runtime verification (RV) is a computing analysis paradigm based on observing a system at runtime to check its expected behavior. RV has emerged in recent years as a practical application of formal verification, and a less ad-hoc approach to conventional testing by building monitors from formal specifications.

There is a great potential applicability of RV beyond software reliability, if one allows monitors to interact back with the observed system, and generalizes to new domains beyond computers programs (like hardware, devices, cloud computing and even human centric systems). Given the European leadership in computer based industries, novel applications of RV to these areas can have an enormous impact in terms of the new class of designs enabled and their reliability and cost effectiveness.

This Action aims to build expertise by putting together active researchers in different aspects of runtime verification, and meeting with experts from potential application disciplines. The main goal is to overcome the fragmentation of RV research by (1) the design of common input formats for tool cooperation and comparison; (2) the evaluation of different tools, building a growing sets benchmarks and running tool competitions; and (3) by designing a road-map and grand challenges extracted from application domains.

8.4. International Initiatives

8.4.1. Inria International Labs

- JLESC (Joint Laboratory on Exascale Computing)
  The CORSE team is involved in the JLESC with collaborations with UIUC (Sanjay Kalé) and BSC (Mont-Blanc projects). Kevin Pouget, Brice Videau and Jean-François Méhaut attended to the two JLESC workshops (Barcelona and Bonn) in 2015.
  - **Energy Efficiency and Load Balancing**
  - The power consumption of High Performance Computing (HPC) systems is an increasing concern as large-scale systems grow in size and, consequently, consume more energy. In response to this challenge, we propose new energy-aware load balancers that aim at reducing the energy consumption of parallel platforms running imbalanced scientific applications without degrading their performance. Our research explores dynamic load balancing, low power manycore platforms and DVFS techniques in order to reduce power consumption.
  - We propose the improvement of the performance and scalability of parallel seismic wave models through dynamic load balancing. These models suffer from load imbalance for two reasons. First, they add a specific numerical condition at the borders of the domain, in order to absorb the outgoing energy. The decomposition of the domain into a grid of subdomains, which are distributed among tasks, creates load differences between the tasks that simulate the borders and those responsible for the central subdomains. Second, the propagation of waves in the simulated area changes the workload on the subdomains on different time-steps. Therefore causing dynamic load imbalance. In order to evaluate the use of dynamic load balancing, we ported a seismic wave simulator to Adaptive MPI, to benefit from its load balancing framework. Our experimental results show that dynamic load balancers can adapt to load variations during the application's execution and improve performance by 36%.
  - we also focus on reducing the energy consumption of imbalanced applications through a combination of load balancing and Dynamic Voltage and Frequency Scaling (DVFS). Our strategy employs an Energy Daemon Tool to gather power information and a load balancing module that benefits from the load balancing framework available in the CHARM++

runtime system. We propose two variants of our energy-aware load balancer (ENERGYLB) to save energy on imbalanced workloads without considerably impacting the overall system performance. The first one, called Fine- Grained EnergyLB (FG-ENERGYLB), is suitable for plat- forms composed of few tens of cores that allow per-core DVFS. The second one, called Coarse-Grained EnergyLB (CG-ENERGLB) is suitable for current HPC platforms composed of several multi-core processors that feature per-chip DVFS.

8.4.2. Inria Associate Teams Not Involved in an Inria International Labs

8.4.2.1. IOComplexity

Title: Automatic characterization of data movement complexity

International Partner (Institution - Laboratory - Researcher):

Ohio State University (United States) - P. Sadayappan

Start year: 2015

See also: https://team.inria.fr/corse/iocomplexity/

The goal of this project is to develop new techniques and tools for the automatic characterization of the data movement complexity of an application. The expected contributions are both theoretical and practical, with the ambition of providing a fully automated approach to I/O complexity characterization, in starking contrast with all known previous work that are stricly limited to pen-and-paper analysis.

I/O complexity becomes a critical factor due in large part to the increasing dominance of data movement over computation in energy consumption for current and emerging architectures. This project aims at enabling: 1. the selection of algorithms according to this new criteria (as opposed to the criteria on arithmetic complexity that has been used up to now); 2. the design of specific architectures in terms of cache size, memory bandwidth, GFlops etc. based on application-specific bounds on memory traffic; 3. higher quality feedback to the user, the compiler, or the run-time system about data traffic, a major performance and energy factor.

8.4.2.2. PROSPIEL

- Title: Profiling and specialization for locality
- International Partner (Institution - Laboratory - Researcher):

  Universidade Federal de Minas Gerais (Brazil) - Computer Science Department - Fernando Magno Quintão Pereira
- Start year: 2015
- See also: https://team.inria.fr/alf/prospiel/
- The PROSPIEL project aims at optimizing parallel applications for high performance on new throughput-oriented architectures: GPUs and many-core processors. Traditionally, code optimization is driven by a program analysis performed either statically at compile-time, or dynamically at run-time. Static program analysis is fully reliable but often over-conservative. Dynamic analysis provides more accurate data, but faces strong execution time constraints and does not provide any guarantee. By combining profiling-guided specialization of parallel programs with runtime checks for correctness, PROSPIEL seeks to capture the advantages of both static analysis and dynamic analysis. The project relies on the polytope model, a mathematical representation for parallel loops, as a theoretical foundation. It focuses on analyzing and optimizing performance aspects that become increasingly critical on modern parallel computer architectures: locality and regularity.

8.4.2.3. Exase

Title: Exascale Computing Scheduling Energy

See also: https://team.inria.fr/exase/

Inria leader: Jean-Marc Vincent (Mescal)

Inria teams: Mescal, Moais, CORSE

CORSE participants: Jean-François Méhaut, François Broquedis, Frédéric Desprez

International Partner (Institution - Laboratory - Researcher):

> Federal University of Rio Grande do Soul (UFRGS, Porto Alegre, Brazil) - Informatics Faculty - L. Schnoor, N. Maillard, P. Navaux
>
> Pontifical University Minas (PUC Minas, Belo Horizonte, Brazil) - Computer Science faculty, Henrique Freitas
>
> University of Sao Paulo (USP, Sao Paulo, Brazil), IME faculty, Alfredo Goldman

Start year: 2014

The main scientific goal of Exase for the three years is the development of state-of- the-art energy-aware scheduling algorithms for exascale systems. As previously stated, issues on energy are fundamental for next generation parallel platforms and all scheduling decisions must be aware of that. Another goal is the development of trace analysis techniques for the behavior analysis of schedulers and the applications running on exascale machines. We list below specific objectives for each development axis presented in the previous section. analysis.

- Fundamentals for the scaling of schedulers
- Design of schedulers for large-scale infrastructures
- Tools for the analysys of large scale schedulers

8.4.3. Participation in Other International Programs

- LICIA (LIG, UFRGS Brazil)
- EnergySFE (STIC Amsud)
  - Leader: University Federal of Santa Catarina (UFSC): Màrcio Castro
  - Partners: UFSC (Florianapolis, Brazil), UFRGS (Porto Alegre, Brazil), ESPE (Ecuador), CNRS (LIG/CORSE, TIMA, LSPSC)
  - Duration: January 2016 - December 2017
  - CORSE participants: Jean-François Méhaut, François Broquedis, Frédéric Desprez
  - The main goal of the EnergySFE research project is to propose fast and scalable energy-aware scheduling and fault tolerance techniques and algorithms for large-scale highly parallel architectures. To achieve this goal, it will be crucial to answer the following research questions:
    - How to schedule tasks and threads that compete for resources with different constraints while considering the complex hierarchical organization of future Exascale supercomputers?
    - How to tolerate faults without incurring in too much overhead in future Exascale supercomputers?
    - How scheduling and fault tolerance approaches can be adapted to be energy-aware?

  The first EnergySFE workshop was organized by the CORSE team a the Inria Minatec building in September 2016.

8.5. International Research Visitors

8.5.1. Visits of International Scientists

- Louis-Noël Pouchet (OSU), visited CORSE two times one month
- Julien Langou (UCDenver) is visiting professor since September 2016

- Mohamad Jaber (AUB) visited CORSE two weeks in January 2016
- Sylvain Hallé (U of Québec) visited CORSE one week in August 2016
- Christian Colombo (U of Malta) visited CORSE two weeks in March 2016
- Henrique Freitas (PUC Minas) visited CORSE one year since July 2015 until July 2016

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. General Chair, Scientific Chair

- Ylies Falcone: 1st international summer school on Runtime Verification; 3rd international Competition on Runtime Verification
- Frédéric Desprez: EuroPAR 2016 (co-chair and workshop chair)

9.1.1.2. Member of the Organizing Committees

- Fabrice Rastello: Program Committee ACM/IEEE CGO 2015; Steering Committee Journées française de la compilation; Steering Committee ACM/IEEE CGO

9.1.2. Scientific Events Selection

9.1.2.1. Chair of Conference Program Committees

- Fabrice Rastello: Program Chair ACM/IEEE CGO 2016; Program Chair “Journées française de la compilation”, Aussois, 2016
- Ylies Falcone: Program Chair RV 2016

9.1.2.2. Member of the Conference Program Committees

- Fabrice Rastello: ACM CC 2016, ACM SRC SC 2016, ACM/IEEE SRC SC 2016
- Alain Ketterlin: ACM/IEEE CGO 2016
- Ylies Falcone: CARI 2016, SSS 2016, RV 2016, Pre-Post’16, SAC-SVT’16
- Frédéric Desprez: Closer 2016, CCGrid 2016, HPC 2016, EuroPAR 2016, CloudCom 2016

9.1.3. Journal

9.1.3.1. Reviewer - Reviewing activities

- Fabrice Rastello: ACM TACO
- Ylies Falcone: Formal Aspects of Computing, ACM Transactions on Automatic and Control, Acta Informatica, Formal Methods in System Design, International Journal of Information and Computer Security, Science of Computer Programming, Software Tools for Technology Transfer, Journal of Systems and Software, NFM 2016

9.1.4. Invited talks

- Fabrice Rastello: UCDenver: “Toward Automatic Characterisation of the Data Access Complexity of Programs”
- Ylies Falcone: American University of Beirut: “On the Runtime Enforcement of Timed Properties”
- Ylies Falcone: LAAS Toulouse: “On the Runtime Enforcement of Timed Properties”
- Frédéric Desprez: Inria Alumni: “Internet des objets, Où sont les ruptures? Activités à l’Inria”
- Frédéric Desprez: SUCCES Workshop: “CIMENT, GRICAD, Grid’5000: La synergie grenobloise”

- Frédéric Desprez: CCDSC Workshop: “BOAST: Performance Portability Using Meta-Programming and Auto-Tuning”
- Frédéric Desprez: Eurecom Seminar 2016: “Challenges and Issues of Next Cloud Computing Platforms”
- Frédéric Desprez: European Commission, Brussels: “Research Issues for Future Cloud Infrastructures”
- Frédéric Desprez: CIRM, CEMRACS 2016 summer school: “OpenCL Introduction”
- François Broquedis: CIRM, CEMRACS 2016 summer school: “A Gentle Introduction to OpenMP Programming”
- Jean-François Méhaut: CEMRACS 2016 summer school: “Overview of architectures and programming language for parallel computing”

9.1.5. Scientific expertise

- Frédéric Desprez: European project in the FP7 framework
- Frédéric Desprez: Comité d’orientation stratégique de CIRRUS (COMUE Paris)
- Frédéric Desprez: Groupe Technique GENCI
- Frédéric Desprez: Conseil Scientifique GIS France Grille
- Frédéric Desprez: GENCI, expert for grants of computing resources (CT6)
- Ylies Falcone: Representative of France in the COST Action ARVI
- Ylies Falcone: COST Action ARVI, co-leader of Working Group on Core Runtime Verification
- Jean-François Mehaut: Eurolab-4-HPC, expert for cross site mobility research grants
- Jean-François Mehaut: GENCI, expert for grants of computing resources (CT6)

9.1.6. Research administration

- Frédéric Desprez: Deputy Scientific Director at Inria
- Frédéric Desprez: Director of the GIS GRID5000
- Frédéric Desprez: Conseil Scientifique ESIEE Paris

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master II: Fabrice Rastello, Advanced Compilers, 12 hours, ENS Lyon

Master I: Jean-François Méhaut, Operating System Design, 50 hours, Polytech Grenoble

L3: Jean-François Méhaut, Numerical Methods, 50 hours, Polytech Grenoble,

L3: Jean-François Méhaut, Advanced Algorithms, 50 hours, Polytech Grenoble

L3: François Broquedis, Imperative programming using python, 40 hours, Grenoble Institute of Technology (Ensimag)

L3: François Broquedis, C programming, 80 hours, Grenoble Institute of Technology (Ensimag)

M1: François Broquedis, Operating systems and concurrent programming, 40 hours, Grenoble Institute of Technology (Ensimag)

M1: François Broquedis, Operating Systems Development Project - Fundamentals, 20 hours, Grenoble Institute of Technology (Ensimag)

M1: François Broquedis, Operating Systems Project, 20 hours, Grenoble Institute of Technology (Ensimag)

Master: Florent Bouchez Tichadou, Compilation project, 15 hours, M1 Info & M1 MoSig

Licence: Florent Bouchez Tichadou, C programming, 24 hours, L3, Grenoble Institute of Technology (Ensimag)

Master: Florent Bouchez Tichadou, Algorithmic Problem Solving, 41 hours, M1 MoSIG

Licence: Florent Bouchez Tichadou, Algorithms languages and programming, 121 hours, L2 UGA

Licence: Florent Bouchez Tichadou is responsible of the second year of INF (informatique) and MIN (mathématiques et informatique) students at UGA

Master I: Ylies Falcone Proof Techniques and Logic Reminders, MoSIG, 3 hours

Master I: Ylies Falcone Recaps on Object-Oriented Programming, MoSIG, 3 hours

Master II: Ylies Falcone Introduction to Runtime Verification, MoSIG HECS, 8 hours.

Master I: Ylies Falcone Programming Language Semantics and Compiler Design, MoSIG, 66 hours

License: Ylies Falcone Languages and Automata, UJF, 105 hours

Master: Ylies Falcone is co-responsible of the first year of the International Master of Computer Science (Univ. Grenoble Alpes and INP ENSIMAG)

9.2.2. Supervision

9.2.2.1. Fabrice Rastello

PhD defended [3]: Duco van Amstel, Scheduling and optimization for memory locality of dataflow programs on many-core processors, advised by Fabrice Rastello and Benoit Dupont-de-Dinechin

PhD defended [1]: Diogo Sampaio, Profiling Guided Hybrid Compilation, October 8 2013, advised by Fabrice Rastello

PhD defended: Venmugil Elango, Dynamic Analysis for Characterization of Data Locality Potential, advised by Fabrice Rastello and P. Sadayappan.

PhD in progress: François Gindraud, Semantics and compilation for a data-flow model with a global address space and software cache coherency, January 1st 2013, advised by Fabrice Rastello and Albert Cohen.

PhD in progress: Fabian Grüber, Interactive & iterative performance debugging, September 2016, advised by Fabrice Rastello and Ylies Falcone.

PhD in progress: Philippe Virouleau, *Improving the performance of task-based runtime systems on large scale NUMA machines*, co-advised by Thierry Gautier (Inria/AVALON), Fabrice Rastello, François Broquedis

9.2.2.2. Jean-François Méhaut

PhD defended (April 2016): Oleg Iegorov, advised by Alexandre Termier (Dream/Irisa), Vincent Leroy (SLIDE/LIG) and Jean-François Méhaut

PhD defended (October 2016): Nassim Halli, CIFRE with Asselta, advised by Henri-Pierre Charles (CEA/DRT List), Jean-François Méhaut

PhD defended [36]: Naweiluo Zhou, advised by Eric Rutten (Inria, CtrlA), Gwenael Delaval (UGA, CtrlA), Jean-François Méhaut

PhD in progress: Thomas Messi Nguelé, advised by Maurice Tchuenté (Yaoundé I, LIRIMA) and Jean-François Méhaut

PhD in progress: Thomas Goncalves, advised by Marc Perache (CEA/DAM), Frédéric Desprez, Jean-François Méhaut

PhD in progress: Luis Felipe Milani, advised by Lucas Schnoor (UFRGS), François Broquedis and Jean-François Méhaut

PhD in progress: Vanessa Vargas, advised by Raoul Velazco (CNRS, TIMA) and Jean-François Méhaut

PhD in progress: Raphaël Jakse, Monitoring and Debugging Component-Based Systems, advised by Jean-François Mehaut and Ylies Falcone.

9.2.2.3. Frédéric Desprez

PhD defended (October 2016): Jonathan Pastor, advised by Frédéric Desprez, Adrien Lèbre (EMN Nantes, Ascola team)

PhD in progress: Pedro Silva, advised by Frédéric Desprez, C. Perez (Inria, Avalon team)

PhD in progress: Georgios Christodoulis, advised by Frederic Desprez, Olivier Muller (TIMA/SLS) and François Broquedis

PhD in progress: Thomas Goncalves, advised by Marc Perache (CEA/DAM), Frédéric Desprez, Jean-François Méhaut

PhD in progress: Ye Xia, advised by Thierry Coupaye (Orange), Frédéric Desprez, Xavier Etchevers (Orange)

9.2.2.4. François Broquedis

PhD in progress: Georgios Christodoulis, *Adaptation of a heterogeneous runtime system to efficiently exploit FPGA* advised by Frederic Desprez, Olivier Muller (TIMA/SLS) and François Broquedis

PhD in progress: Philippe Virouleau, *Improving the performance of task-based runtime systems on large scale NUMA machines*, co-advised by Thierry Gautier (Inria/AVALON), Fabrice Rastello, François Broquedis

9.2.2.5. Ylies Falcone

PhD in progress: Hosein Nazarpour, Monitoring Multithreaded and Distributed Component-based Systems, advised by Saddek Bensalem (Vérimag) and Ylies Falcone.

PhD in progress: Antoine El-Hokayem, Decentralised and Distributed Monitoring of Cyber-Physical Systems, advised by Ylies Falcone.

PhD in progress: Fabian Grüber, Interactive & iterative performance debugging, September 2016, advised by Fabrice Rastello and Ylies Falcone.

PhD in progress: Raphaël Jakse, Monitoring and Debugging Component-Based Systems, advised by Jean-François Mehaut and Ylies Falcone.

9.2.3. Juries

9.2.3.1. Fabrice Rastello

Venmugil Elango, Advisor, *Dynamic Analysis for Characterization of Data Locality Potential*, PhD of OSU, 06/01/2016

Arjun Suresh, Reviewer, *Intercepting Functions for Memoization*, PhD of Université de Rennes, 10/04/2016

Duco Van-Amstel, Advisor, *Scheduling and optimization for memory locality of dataflow programs on many-core processors*, Université Grenoble Alpes, 11/07/2016.

Juan Manuel Martinez Caamano, Reviewer, *Fast and Flexible Compilation Techniques for Effective Speculative Polyhedral Parallelization*, Université de Strasrbourg, 29/09/2016

Pierre Guillou, Reviewer, *Compilation efficace d'applications de traitement d'images pour processeurs manycore*, Université de recherche PAris Sciences et Lettres, 30/11/2016

9.2.3.2. Jean-François Méhaut

Oleg Iegorov, Advisor,*Data Mining Approach to Temporal Debugging of Embedded Streaming Applications*, PhD of Université Grenoble Alpes, April 2016

Nassim Halli, Advisor, *Code Optimizations of High Performance Java Applications*, PhD of Université Grenoble Alpes, October 2016

Naweiluo Zhou, Advisor, *Autonomic Thread Parallelism and Mapping Control for Software Transactional Memory System*, PhD of Université Grenoble Alpes, October 2016

Marc Sergent, Reviewer, *Passage à l'échelle d'un support d'exécution à base de tâches pour l'algèbre linéaire creuse*, PhD of Université de Bordeaux, October 2016

Jean-Charles Papin, Reviewer, *A Scheduling and Partitioning Model for Stencil-based Applications on ManyCore Devices*, PhD of Ecole Normale Supérieure de Cachan, July 2016

9.2.3.3. Frédéric Desprez

Jean-Marie Couteyen, Reviewer, *Parallélisation et passage à l'échelle du code FLUSEPA*, PhD of Université de Bordeaux, September 2016

Jonathan Pastor, Advisor, *Contributions à la mise en place d'une infrastructure de Cloud Computing à large échelle* , Ecole des Mines de Nantes, October 2016

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] D. N. SAMPAIO. *Profile Guided Hybrid Compilation*, Université Grenoble-Alpes, December 2016, https://hal.inria.fr/tel-01428425

[2] N. ZHOU. *Autonomic Thread Parallelism and Mapping Control for Software Transactional Memory*, UJF Grenoble-1 ; Inria Grenoble, October 2016, https://hal.archives-ouvertes.fr/tel-01408450

[3] D. VAN AMSTEL. *Data Locality on Manycore Architectures*, Université Grenoble-Alpes, July 2016, https://hal.inria.fr/tel-01358312

Articles in International Peer-Reviewed Journals

[4] A. BAUER, Y. FALCONE. *Decentralised LTL Monitoring*, in "Formal Methods in System Design", May 2016, vol. 48, n^o 1-2, 48 p. [*DOI :* 10.1007/S10703-016-0253-8], https://hal.inria.fr/hal-01313730

[5] M. CASTRO, E. FRANCESQUINI, F. DUPROS, H. AOCHI, P. NAVAUX, J.-F. MEHAUT. *Seismic Wave Propagation Simulations on Low-power and Performance-centric Manycores*, in "Parallel Computing", 2016 [*DOI :* 10.1016/J.PARCO.2016.01.011], https://hal.archives-ouvertes.fr/hal-01273153

[6] C. COLOMBO, Y. FALCONE. *Organising LTL Monitors over Distributed Systems with a Global Clock*, in "Formal Methods in System Design", May 2016, vol. 49, n^o 1-2, 50 p. [*DOI :* 10.1007/S10703-016-0251-X], https://hal.inria.fr/hal-01315776

[7] Y. FALCONE, M. JABER. *Fully-automated Runtime Enforcement of Component-based Systems with Formal and Sound Recovery*, in "Software Tools for Technology Transfer (STTT)", February 2016, https://hal.inria.fr/hal-01262658

[8] Y. FALCONE, T. JÉRON, H. MARCHAND, S. PINISETTY. *Runtime Enforcement of Regular Timed Properties by Suppressing and Delaying Events*, in "Science of Computer Programming", March 2016 [*DOI :* 10.1016/J.SCICO.2016.02.008], https://hal.inria.fr/hal-01281727

[9] P. H. Penna, M. Castro, H. C. Freitas, F. Broquedis, J.-F. Méhaut. *Design methodology for workload-aware loop scheduling strategies based on genetic algorithm and simulation*, in "Concurrency and Computation: Practice and Experience", 2016 [*DOI :* 10.1002/cpe.3933], https://hal.archives-ouvertes.fr/hal-01354028

[10] P. Ramos, V. Vargas, M. Baylac, F. Villa, S. Rey, J. A. Clemente, N.-E. Zergainoh, J.-F. Méhaut, R. Velazco. *Evaluating the SEE sensitivity of a 45nm SOI Multi-core Processor due to 14 MeV Neutrons*, in "IEEE Transactions on Nuclear Science", March 2016, vol. 63, n^{o} 4, pp. 2193 - 2200 [*DOI :* 10.1109/TNS.2016.2537643], https://hal.archives-ouvertes.fr/hal-01280648

[11] M. A. Souza, P. H. Penna, M. M. Queiroz, A. D. Pereira, L. F. W. Góes, H. C. Freitas, M. Castro, P. O. Navaux, J.-F. Méhaut. *CAP Bench: a benchmark suite for performance and energy evaluation of low-power many-core processors*, in "Concurrency and Computation: Practice and Experience", 2016 [*DOI :* 10.1002/cpe.3892], https://hal.archives-ouvertes.fr/hal-01330543

Invited Conferences

[12] C. Colombo, Y. Falcone. *First International Summer School on Runtime Verification: as part of the ArVi COST Action 1402*, in "Sixteenth International Conference on Runtime Verification", Madrid, Spain, September 2016, https://hal.inria.fr/hal-01428838

[13] G. Reger, S. Hallé, Y. Falcone. *Third International Competition on Runtime Verification CRV 2016*, in "Sixteenth International Conference on Runtime Verification", Madrid, Spain, September 2016, https://hal.inria.fr/hal-01428834

International Conferences with Proceedings

[14] W. Bao, K. Sriram, L.-N. Pouchet, F. Rastello, S. Ponnuswamy. *PolyCheck: Dynamic Verification of Iteration Space Transformations on Affine Programs*, in "Proceedings of the 43nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016", St Petersburg, United States, ACM, January 2016, https://hal.inria.fr/hal-01234104

[15] A. El-Hokayem, Y. Falcone, M. Jaber. *Modularizing Crosscutting Concerns in Component-Based Systems*, in "14th International Conference on Software Engineering and Formal Methods", Vienne, Austria, July 2016, https://hal.inria.fr/hal-01305083

[16] F. Gindraud, F. Rastello, A. Cohen, F. Broquedis. *A bounded memory allocator for software-defined global address spaces*, in "ISMM 2016 - 2016 ACM SIGPLAN International Symposium on Memory Management", Santa Barbara, United States, June 2016, https://hal.inria.fr/hal-01412919

[17] S. Hallé, R. Khoury, A. El-Hokayem, Y. Falcone. *Decentralized Enforcement of Artifact Lifecycles*, in "EDOC 2016", Vienne, Austria, Proceedings of the twentieth entreprise computing conference, September 2016, https://hal.inria.fr/hal-01365315

[18] H. Nazarpour, Y. Falcone, S. Bensalem, M. Bozga, J. Combaz. *Monitoring Multi-Threaded Component-Based Systems*, in "12th International Conference on integrated Formal Methods", Reykjavik, Finland, Proceedings of the 12th International Conference on integrated Formal Methods, June 2016, https://hal.inria.fr/hal-01285579

[19] S. PINISETTY, V. PREOTEASA, S. TRIPAKIS, T. JÉRON, Y. FALCONE, H. MARCHAND. *Predictive Runtime Enforcement* *, in "SAC 2016 31st ACM Symposium on Applied Computing", Pisa, Italy, ACM, April 2016, 6 p. [*DOI :* 10.1145/2851613.2851827], https://hal.inria.fr/hal-01244369

[20] R. SAMYAM, K. JINSUNG, K. SRIRAM, F. RASTELLO, L.-N. POUCHET, R. J. HARRISON, S. PONNUSWAMY. *A domain-specific compiler for a parallel multiresolution adaptive numerical simulation environment*, in "SC 2016 - International Conference for High Performance Computing, Networking, Storage and Analysis", Salt-Lake City, United States, November 2016, https://hal.inria.fr/hal-01412903

[21] P. SILVA, C. PÉREZ, F. DESPREZ. *Efficient Heuristics for Placing Large-Scale Distributed Applications on Multiple Clouds*, in "16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'16)", Cartagena, Colombia, 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 2016 [*DOI :* 10.1109/CCGRID.2016.77], https://hal.archives-ouvertes.fr/hal-01301382

[22] P. VIROULEAU, F. BROQUEDIS, T. GAUTIER, F. RASTELLO. *Using data dependencies to improve task-based scheduling strategies on NUMA architectures*, in "Euro-Par 2016", Grenoble, France, Euro-Par 2016, August 2016, https://hal.inria.fr/hal-01338761

[23] P. VIROULEAU, A. ROUSSEL, F. BROQUEDIS, T. GAUTIER, F. RASTELLO, J.-M. GRATIEN. *Description, Implementation and Evaluation of an Affinity Clause for Task Directives*, in "IWOMP 2016", Nara, Japan, IWOMP 2016 - LLCS 9903, October 2016, https://hal.inria.fr/hal-01343442

[24] N. ZHOU, G. DELAVAL, B. ROBU, E. RUTTEN, J.-F. MÉHAUT. *Autonomic Parallelism and Thread Mapping Control on Software Transactional Memory* , in "13th IEEE International Conference on Autonomic Computing (ICAC 2016)", Wuerzburg, Germany, July 2016, pp. 189 - 198 [*DOI :* 10.1109/ICAC.2016.54], https://hal.archives-ouvertes.fr/hal-01309681

[25] N. ZHOU, G. DELAVAL, B. ROBU, E. RUTTEN, J.-F. MÉHAUT. *Control of Autonomic Parallelism Adaptation on Software Transactional Memory*, in "International Conference on High Performance Computing & Simulation (HPCS 2016)", Innsbruck, Austria, July 2016, pp. 180-187 [*DOI :* 10.1109/HPCSIM.2016.7568333], https://hal.archives-ouvertes.fr/hal-01309195

[26] N. ZHOU, G. DELAVAL, B. ROBU, É. RUTTEN, J.-F. MÉHAUT. *Autonomic Parallelism Adaptation for Software Transactional Memory*, in "Conférence d'informatique en Parallélisme, Architecture et Système (COMPAS)", Lorient, France, July 2016, https://hal.inria.fr/hal-01312786

National Conferences with Proceedings

[27] R. JAKSE, Y. FALCONE, J.-F. MÉHAUT, K. POUGET. *Vérification interactive de propriétés à l'exécution d'un programme avec un débogueur*, in "Compas'2016", Lorient, France, Compas'2016 : Parallélisme / Architecture / Système Lorient, France, du 5 au 8 juillet 2016, July 2016, https://hal.inria.fr/hal-01331973

Conferences without Proceedings

[28] Ł. DOMAGAŁA, D. VAN AMSTEL, F. RASTELLO. *Generalized cache tiling for dataflow programs*, in "Conference on Languages, Compilers, Tools, and Theory for Embedded Systems", Santa Barbara, United States, Proceedings of the 17th ACM SIGPLAN/SIGBED Conference on Languages, Compilers, Tools, and Theory for Embedded Systems, June 2016, 10 p. [*DOI :* 10.1145/2907950.2907960], https://hal.inria.fr/hal-01336172

[29] C. Hong, W. Bao, A. Cohen, S. Krishnamoorthy, L.-N. Pouchet, F. Rastello, J. Ramanujam, S. Ponnuswany. *Effective padding of multidimensional arrays to avoid cache conflict misses*, in "PLDI 2016: Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation", Santa Barbara, United States, June 2016, https://hal.inria.fr/hal-01335346

[30] K. Pouget, M. Santana, J.-F. Méhaut. *Programming-Model Centric Debugging for OpenMP*, in "2nd OpenMPCon Developpers Conference", Nara, Japan, October 2016, https://hal.archives-ouvertes.fr/hal-01351561

[31] N. Rajovic, A. Rico, F. Mantovani, D. Ruiz, J. Vilarrubi, C. Gomez, D. Nieto, H. Servat, X. Martorell, J. Labarta, C. Adeniyi-Jones, S. Derradji, H. Gloaguen, P. Lanucara, N. Sanna, J.-F. Méhaut, K. Pouget, B. Videau, E. Boyer, M. Allalen, A. Auweter, D. Brayford, D. Tafani, V. Weinberg, D. Brömmel, R. Halver, J. Meinke, R. Beivide, M. Benito, E. Vallejo, M. Valero, A. Ramirez. *The Mont-Blanc prototype: An Alternative Approach for HPC Systems*, in "International Conference for High Performance Computing, Networking, Storage and Analysis (SC)", Salt Lake City, United States, November 2016, https://hal.archives-ouvertes.fr/hal-01354939

[32] R. Samyam, K. Jinsung, S. Krishnamoorthy, L.-N. Pouchet, F. Rastello, R. J. Harrison, S. Ponnuswany. *On fusing recursive traversals of K-d trees*, in "Proceedings of the 25th International Conference on Compiler Construction, CC 2016", Barcelona, Spain, March 2016, https://hal.inria.fr/hal-01335355

[33] P. Virouleau. *Amélioration des stratégies d'ordonnancement sur architectures NUMA à l'aidedes dépendances de données*, in "Compas 2016", Lorient, France, July 2016, https://hal.inria.fr/hal-01338750

Scientific Books (or Scientific Book chapters)

[34] L. Genovese, B. Videau, D. Caliste, J.-F. Méhaut, S. Goedecker, T. Deutsch. *Wavelet-Based Density Functional Theory on Massively Parallel Hybrid Architectures*, in "Electronic Structure Calculations on Graphics Processing Units: From Quantum Chemistry to Condensed Matter Physics", R. Walker (editor), Wiley-Blackwell, February 2016, https://hal.archives-ouvertes.fr/hal-01239245

Research Reports

[35] C. Alias, F. Rastello, A. Plesco. *High-Level Synthesis of Pipelined FSM from Loop Nests*, Inria, April 2016, n^o^ 8900, 18 p. , https://hal.inria.fr/hal-01301334

[36] N. Zhou, G. Delaval, B. Robu, É. Rutten, J.-F. Méhaut. *Autonomic Parallelism Adaptation on Software Transactional Memory*, Univ. Grenoble Alpes ; Inria Grenoble, March 2016, n^{o} RR-8887, 24 p. , https://hal.inria.fr/hal-01279599

Other Publications

[37] D. Margery, F. Desprez. *On the sustainability of large-scale computer science testbeds: the Grid'5000 case*, February 2016, working paper or preprint, https://hal.inria.fr/hal-01273170

[38] T. Messi Nguélé, M. Tchuente, J.-F. Méhaut. *Social network ordering based on communities to reduce cache misses*, April 2016, working paper or preprint, https://hal.archives-ouvertes.fr/hal-01304968

[39] M. RENARD, Y. FALCONE, A. ROLLET. *Optimal Enforcement of (Timed) Properties with Uncontrollable Events*, February 2016, working paper or preprint, https://hal.archives-ouvertes.fr/hal-01262444