IN PARTNERSHIP WITH:
Université Rennes 1

Activity Report 2016

Project-Team PACAP

Pushing Architecture and Compilation for Application Performance

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Architecture, Languages and Compilation

Table of contents

Project-Team PACAP

Creation of the Project-Team: 2016 July 01

Keywords:

Computer Science and Digital Science:

1.1. - Architectures
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
1.1.9. - Fault tolerant systems
1.1.10. - Reconfigurable architectures
1.6. - Green Computing
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
2.3.3. - Real-time systems
4.2. - Correcting codes
4.4. - Security of equipment and software
7.11. - Performance evaluation
7.12. - Computer arithmetic

Other Research Topics and Application Domains:

1. - Life sciences
2. - Health
3. - Environment and planet
4. - Energy
5. - Industry of the future
6. - IT and telecom
7. - Transport and logistics
8. - Smart Cities and Territories
9. - Society and Knowledge

1. Members

Research Scientists

Erven Rohou [Team leader, Inria, Senior Researcher, HDR]

Sylvain Collange [Inria, Researcher]
Pierre Michaud [Inria, Researcher]
André Seznec [Inria, Senior Researcher, HDR]

Faculty Members
Damien Hardy [Univ. Rennes I, Associate Professor]
Isabelle Puaut [Univ. Rennes I, Professor, HDR]

Engineers
Nicolas Kiss [Inria, from Apr 2016]
Imane Lasri [Inria, from Mar 2016]
Sébastien Martinez [Univ. Rennes I, from June 2016]
Arthur Perais [Inria]
Emmanuel Riou [Inria, until Nov 2016]

PhD Students
Arif Ali Ana-Pparakkal [Inria]
Rabab Bouziane [Inria]
Nabil Hallou [Inria, until Nov 2016]
Kleovoulos Kalaitzidis [Inria, from Jun 2016]
Sajith Kalathingal [Inria, until Oct 2016]
Andrea Mondelli [Inria]
Viet Anh Nguyen [Univ. Rennes I]
Benjamin Rouxel [Univ. Rennes I]
Aswinkumar Sridharan [Inria]
Arjun Suresh [Inria, until Jul 2016]

Post-Doctoral Fellows
Fernando Endo [Inria]
Biswabandan Panda [Inria]

Visiting Scientists
Stefano Cherubin [Politecnico di Milano, from Sep 2016 until Oct 2016]
Anita Tino [Ryerson University, from Oct 2016]
Rubens Emilio Alves Moreira [Universidade Federal de Minas Gerais, from Feb 2016 to May 2016]

Administrative Assistant
Virginie Desroches [Inria]

2. Overall Objectives

2.1. Overall Objectives

2.1.1. Long-Term Goal

In brief, the long-term goal of the PACAP project-team is about *performance*, that is: how fast programs run. We intend to contribute to the ongoing race for exponentially increasing performance and for performance guarantees.

Traditionally, the term “performance” is understood as “how much time is needed to complete execution”. *Latency*-oriented techniques focus on minimizing the average-case execution time (ACET). We are also interested in other definitions of performance. *Throughput*-oriented techniques are interested in how many units of computations can be completed per unit of time. This is more relevant on manycores and GPUs where many computing nodes are available, and latency is less critical. Finally, we also study worst-case execution times (WCET). They are extremely important for critical real-time systems where designers must guarantee that deadlines are met, in any situation.

Given the complexity of current systems, simply assessing their performance has become a non-trivial task which we also plan to tackle.

We occasionally consider other metrics related to performance, such as power efficiency, total energy, overall complexity, and real-time response guarantee. Our ultimate goal is to propose solutions that make computing systems more efficient, taking into account current and envisioned applications, compilers, runtimes, operating systems, and microarchitectures. And since increased performance often comes at the expense of another metric, identifying the related trade-offs is of interest to PACAP.

ALF witnessed the end of the "magically" increasing clock frequency and the introduction of commodity multicore processors. PACAP will likely experience the end of Moore's law [1], and the generalization of commodity heterogeneous manycore processors. This impacts how performance is increased and how it can be guaranteed. It is also a time where exogenous parameters should be promoted to first-class citizens:

1. the existence of faults, whose impact is becoming increasingly important when the photo-lithography feature size decreases;
2. the need for security at all levels of computing systems;
3. *green* computing, or the growing concern of power consumption.

2.1.2. Approach

We strive to address performance in a way as transparent as possible for users. For example, instead of proposing any new language, we consider existing applications (written for example in standard C), and we develop compiler optimizations that immediately benefit programmers; we propose microarchitectural features as opposed to changes in processor instruction sets; we analyze and re-optimize binary programs automatically, without any user intervention.

The perimeter of research directions proposed for the PACAP project-team derive from the intersection of two axes: on the one hand, our high-level research objectives, derived from the overall panorama of computing systems, on the other hand the existing expertise and background of the team members on key technology (see illustration on Figure 1). Note that it does not imply that we will systematically explore all intersecting points of the figure, yet all correspond to a sensible research direction. These lists are neither exhaustive, nor final. Operating systems in particular constitute a promising operating point for several of the issues we plan to tackle. Other aspects will likely emerge during the lifespan of the project-team.

2.1.3. Latency-oriented Computing

Improving the ACET of general purpose systems has been the core "business" of CAPS and ALF for two decades. We plan to pursue this line of research, acting at all levels: compilation, dynamic optimizations, and microarchitecture.

2.1.4. Throughput-Oriented Computing

The goal is to maximize the performance-to-power ratio. We will leverage the execution model of throughput-oriented architectures (such as GPUs) and extend it towards general purpose systems. To address the memory wall issue, we will consider bandwidth saving techniques, such as cache and memory compression.

2.1.5. Real-Time Systems – WCET

Designers of real-time systems must provide an upper bound of the worst-case execution time of the tasks within their systems. By definition this bound must be safe (i.e. greater than any possible execution time). To be useful, WCET estimates have to be as tight as possible. The process of obtaining a WCET bound consists in analyzing a binary executable, modeling the hardware, and then maximizing an objective function that takes into account all possible flows of execution and their respective execution times. Our research will consider the following directions:

[1] Moore's law states that the number of transistors in a circuit doubles (approximately) every two years.

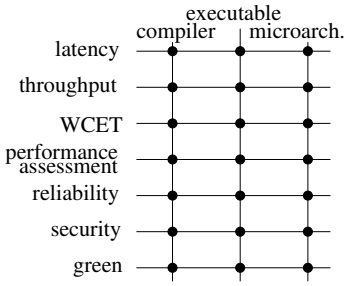

Figure 1. Perimeter of Research Objectives

1. better modeling of hardware to either improve tightness, or handle more complex hardware (e.g. multicores);
2. eliminate unfeasible paths from the analysis;
3. consider probabilistic approaches where WCET estimates are provided with a confidence level.

2.1.6. Performance Assessment

Moore's law drives the complexity of processor micro-architectures, which impacts all other layers: hypervisors, operating systems, compilers and applications follow similar trends. While a small category of experts is able to comprehend (parts of) the behavior of the system, the vast majority of users are only exposed to – and interested in – the bottom line: how fast their applications are actually running. In the presence of virtual machines and cloud computing, multi-programmed workload add yet another degree of non-determinism to the measure of performance. We plan to research how application performance can be characterized and presented to a final user: behavior of the microarchitecture, relevant metrics, possibly visual rendering. Targeting our own community, we also research techniques appropriate for fast and accurate ways to simulate future architectures, including heterogeneous designs, such as latency/throughput platforms.

Once diagnosed, the way bottlenecks are addressed depends on the level of expertise of users. Experts can typically be left with a diagnostic as they probably know better how to fix the issue. Less knowledgeable users must be guided to a better solution. We plan to rely on iterative compilation to generate multiple versions of critical code regions, to be used in various runtime conditions. To avoid the code bloat resulting from multiversioning, we will leverage split-compilation to embed code generation "recipes" to be applied just-in-time, or even at rutime thanks to dynamic binary translation. Finally, we will explore the applicability of auto-tuning, where programmers expose which parameters of their code can be modified to generate alternate versions of the program (for example trading energy consumption for quality of service) and let a global orchestrator make decisions.

2.1.7. Dealing with Faults – Reliability

Semiconductor technology evolution suggests that permanent failure rates will increase dramatically with scaling. While well-known approaches, such as error correcting codes, exist to recover from failures and provide fault-free chips, the exponential growth of the number of faults will make them unaffordable in the future. Consequently, other approaches like fine-grained disabling and reconfiguration of hardware elements (e.g. individual functional units or cache blocks) will become economically necessary. This fine-grained disabling will degrade performance compared to a fault-free execution. This evolution impacts performance (both ACET and WCET). We plan to address this evolution, and propose new techniques, which can be developed at any level. For example, at microarchitecture level, one might consider designing part of a cache in an older technology to guarantee a minimum level of performance; at compile-time, one might generate

redundant code for critical sections; at run-time, one can monitor faults and apply corrective measures to the software, or hardware. Solutions involving multiple levels are also very promising.

2.1.8. Dealing with Attacks – Security

Computer systems are under constant attack, from young hackers trying to show their skills, to "professional" criminals stealing credit card information, and even government agencies with virtually unlimited resources. A vast amount of techniques have been proposed in the literature to circumvent attacks. Many of them cause significant slowdowns due to additional checks and countermeasures. Thanks to our expertise in microarchitecture and compilation techniques, we will be able to significantly improve efficiency, robustness and coverage of security mechanism, as well as to partner with field experts to design innovative solutions.

2.1.9. Green Computing

Power consumption has become a major concern of computing systems, at all form factors, ranging from energy-scavenging sensors for IoT, to battery powered embedded systems and laptops, and up to supercomputers operating in the tens of megawatts. Execution time and energy are often related optimization goals. Optimizing for performance under a given power cap, however, introduces new challenges. It also turns out that technologists introduce new solutions (e.g. magnetic RAM) which, in turn, result in new trade-offs and optimization opportunities.

3. Research Program

3.1. Motivation

Our research program is naturally driven by the evolution of our ecosystem. Relevant recent changes can be classified in the following categories: technological constraints, evolving community, and domain constraints. We hereby summarize these evolutions.

3.1.1. Technological constraints

Until recently, binary compatibility guaranteed portability of programs, while increased clock frequency and improved micro-architecture provided increased performance. However, in the last decade, advances in technology and micro-architecture started translating into more parallelism instead. Technology roadmaps even predict the feasibility of thousands of cores on a chip by 2020. Hundreds are already commercially available. Since the vast majority of applications are still sequential, or contain significant sequential sections, such a trend put an end to the automatic performance improvement enjoyed by developers and users. Many research groups consequently focused on parallel architectures and compiling for parallelism.

The focus of ALF – and the DAL ERC – was paradoxically on Amdahl's law: the performance of applications will ultimately be driven by the performance of the sequential part. Despite a number of advances (some of them contributed by ALF), sequential tasks are still a major performance bottleneck. Addressing it is still on the agenda of the proposed PACAP project-team.

In addition, due to power constraints, only part of the billions of transistors of a microprocessor can be operated at any given time (the *dark silicon* paradigm). A sensible approach consists in specializing parts of the silicon area to provide dedicated accelerators (not run simultaneously). This results in diverse and heterogeneous processor cores. Application and compiler designers are now confronted with a moving target, challenging portability and jeopardizing performance.

Finally, we live in a world where billions of sensors, actuators, and computers play a crucial role in our life: flight control, nuclear plant management, defense systems, banking, or health care. These systems must be reliable, despite the fact that they are subject to faults (for example due to aging, charged particle hit, or random noise). Faults will soon become the new *de facto* standard. The evolutions of the semiconductor industry predict an exponential growth of the number of permanent faults [58]. Reliability considerations usually degrade performance. We will propose solutions to mitigate this impact (for example by limiting overheads to critical sections).

Note on technology.
Technology also progresses at a fast pace. We do not propose to pursue any research on technology *per se*. Recently proposed paradigms (quantum computing, non-Si, brain-inspired) have received lots of attention from the research community. We do *not* intend to invest in those paradigms, but we will continue to investigate compilation and architecture for more conventional programming paradigms. Still, several technological shifts may have consequences for us, and we will closely monitor their developments, they include for example non-volatile memory (impacts security, makes writes longer than loads), 3D-stacking (impacts bandwidth), and photonics (impacts latencies and connection network).

3.1.2. Evolving community

The PACAP project-team will tackle performance-related issues, for conventional programming paradigms. In fact, programming complex environments is no longer reserved to experts in compilation and architecture. A large community now develops applications for a wide range of targets, including mobile "apps", cloud, multicore or heterogeneous processors.

This also includes domain scientists (in biology, medicine, but also social sciences) who started relying heavily on computational resources, gathering huge amounts of data, and requiring considerable amount of processing to analyze them. Our research is motivated by the growing discrepancy between on the one hand the complexity of the workloads and the computing systems, and on the other hand the expanding community of developers at large, with limited expertise to optimize and to map efficiently computations to compute nodes.

3.1.3. Domain constraints

Mobile, embedded systems have become ubiquitous. Many of them have real-time constraints. For this class of systems, correctness implies not only producing the correct result, but also doing so within specified deadlines. In the presence of heterogeneous, complex and highly dynamic systems, producing *tight* (i.e. useful) upper bound to the worst-case execution time has become extremely challenging. Our research will aim at improving the tightness as well as enlarging the set of features that can be safely analyzed.

The ever growing dependence of our economy on computing systems also implies that security has become of utmost importance. Many systems are under constant attacks from intruders. Protection has a cost also in terms of performance. We plan to leverage our background to contribute solutions that minimize this impact.

Note on Applications Domains.
As was already the case for ALF, PACAP will work on fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impacts on any application domain that requires high performance execution (telecommunication, multimedia, biology, health, engineering, environment...), but also on many embedded applications that exhibit other constraints such as power consumption, code size and guaranteed response time.

We strive to extract from active domains the fundamental characteristics that are relevant to our research. For example, *big data* is of interest to PACAP because it relates to the study of hardware/software mechanisms to efficiently transfer huge amounts of data to the computing nodes. Similarly, the *Internet of Things* is of interest because it has implications in terms of ultra low power consumption.

3.2. Research Objectives

Processor micro-architecture and compilation have been at the core of the research carried by the CAPS and ALF project teams for two decades, with undeniable contributions. They will continue to be the foundation of PACAP.

Heterogeneity and diversity of processor architectures now require new techniques to guarantee that the hardware is satisfactorily exploited by the software. We will devise new static compilation techniques (cf. Section 3.2.1), but also build upon iterative [1] and split [2] compilation to continuously adapt software to its environment (Section 3.2.2). Dynamic binary optimization will also play a key role in delivering adapting software and delivering performance.

The end of Moore's law and Dennard's scaling [2] offer an exciting window of opportunity, where performance improvements will no longer derive from additional transistor budget or increased clock frequency, but rather come from breakthroughs in microarchitecture (Section 3.2.3). We will also consider how to reconcile CPU and GPU designs (Section 3.2.4).

Heterogeneity and multicores are also major obstacles to determining tight worst-case execution times of real-time systems (Section 3.2.5), which we plan to tackle.

Finally, we also describe how we plan to address transversal aspects such reliability (Section 3.2.6), power efficiency (Section 3.2.7), and security (Section 3.2.8).

3.2.1. Static Compilation

Static compilation techniques will continue to be relevant to address the characteristics of emerging hardware technologies, such as non-volatile memories, 3D-stacking, or novel communication technologies. These techniques expose new characteristics to the software layers. As an example, non-volatile memories typically have asymmetric read-write latencies (writes are much longer than reads) and different power consumption profiles. PACAP will study the new optimization opportunities and develop tailored compilation techniques for the upcoming compute nodes. New technologies may also be coupled with traditional solutions to offer new trade-offs. We will study how programs can adequately exploit the specific features of the proposed heterogeneous compute nodes.

We propose to build upon iterative compilation [1] to explore how applications perform on different configurations. When possible, Pareto points will be related to application characteristics. The best configuration, however, may actually depend on runtime information, such as input data, dynamic events, or properties that are available only at runtime. Unfortunately a runtime system has little time and means to determine the best configuration. For these reasons, we will also leverage split-compilation [2]: the idea consists in pre-computing alternatives, and embedding in the program enough information to assist and drive a runtime system towards to the best solution.

3.2.2. Software Adaptation

More than ever, software will need to adapt to their environment. In most cases, this environment will remain unknown until runtime. This is already the case when one deploys an application to a cloud, or an "app" to mobile devices. The dilemma is the following: for maximum portability, developers should target the most general device; but for performance they would like to exploit the most recent and advanced hardware features. JIT compilers can handle the situation to some extent, but binary deployment requires dynamic binary rewriting. Our work has shown how SIMD instructions can be upgraded from SSE to AVX [3]. Many more opportunities will appear with diverse and heterogeneous processors, featuring various kinds of accelerators.

On shared hardware, the environment is also defined by other applications competing for the same computational resources. It will become increasingly important to adapt to changing runtime conditions, such as the contention of the cache memories, available bandwidth, or hardware faults. Fortunately, optimizing at runtime is also an opportunity, because this is the first time the program is visible as a whole: executable and libraries (including library versions). Optimizers may also rely on dynamic information, such as actual input data, parameter values, etc. We have already developed a software platform [14] to analyze and optimize programs at runtime, and we started working on automatic dynamic parallelization of sequential code, and dynamic specialization.

We started addressing some of these challenges in ongoing projects such as Nano2017 PSAIC Collaborative research program with STMicroelectronics, as well as within the Inria Project Lab MULTICORE. The starting H2020 FET HPC project ANTAREX will also address these challenges from the energy perspective. We will further leverage our platform and initial results to address other adaptation opportunities. Efficient software adaptation will require expertise from all domains tackled by PACAP, and strong interaction between all team members is expected.

[2] According to Dennard scaling, as transistors get smaller the power density remains constant, and the consumed power remains proportional to the area.

3.2.3. Research directions in uniprocessor microarchitecture

Achieving high single-thread performance remains a major challenge even in the multicore era (Amdahl's law). The members of the PACAP project-team have been conducting research in uniprocessor micro-architecture research for about 20 years covering major topics including caches, instruction front-end, branch prediction, out-of-order core pipeline, branch prediction and value prediction. In particular, in the recent years they have been recognized world leaders in branch prediction [19][9] and in cache prefetching [7] and they have revived the forgotten concept of value prediction [12], [11]. This research was supported by the ERC Advanced grant DAL (2011-2016) and also by Intel. We intend to pursue research on achieving ultimate unicore performance. Below are several non-orthogonal directions that we have identified for mid-term research:

1. management of the memory hierarchy (particularly the hardware prefetching);
2. practical design of very wide issue execution core;
3. speculative execution.

Memory design issues:
Performance of many applications is highly impacted by the memory hierarchy behavior. The interactions between the different components in the memory hierarchy and the out-of-order execution engine have high impact on performance.

The last *Data Prefetching Contest* held with ISCA 2015 has illustrated that achieving high prefetching efficiency is still a challenge for wide-issue superscalar processors, particularly those featuring a very large instruction window. The large instruction window enables an implicit data prefetcher. The interaction between this implicit hardware prefetcher and the explicit hardware prefetcher is still relatively mysterious as illustrated by Pierre Michaud's BO prefetcher (winner of DPC2) [7]. The first objective of the research is to better understand how the implicit prefetching enabled by the large instruction window interacts with the L2 prefetcher and then to understand how explicit prefetching on the L1 also interacts with the L2 prefetcher.

The second objective of the research is related to the interaction of prefetching and virtual/physical memory. On real hardware, prefetching is stopped by page frontiers. The interaction between TLB prefetching (and on which level) and cache prefetching must be analyzed.

The prefetcher is not the only actor in the hierarchy that must be carefully controlled. Significant benefit can also be achieved through careful management of memory access bandwidth, particularly the management of spatial locality on memory accesses, both for reads and writes. The exploitation of this locality is traditionally handled in the memory controller. However, it could be better handled if larger temporal granularity was available. Finally, we also intend to continue to explore the promising avenue of compressed caches. In particular we recently proposed the skewed compressed cache [15]. It offers new possibility for efficient compression schemes.

Ultra wide-issue superscalar.
To effectively leverage memory level parallelism, one requires huge out-of-order execution structures as well as very wide issue superscalar processor. For the two past decades, implementing always wider issue superscalar processor has been challenging. The objective of our research on the execution core is to explore (and revisit) directions to allow the design of a very wide-issue (8-to-16 way) out-of-order execution core while mastering its complexity (silicon area, hardware logic complexity, power/energy consumption).

The first direction that we intend to explore is the use of clustered architecture as in our recent work [8]. Symmetric clustered organization allows to benefit from simpler bypass network, but induce large complexity on the issue queue. One remarkable finding of our study [8] is that, when considering two large clusters (e.g. 8-wide) steering large groups of consecutive instructions (e.g. 64 μops) to the same cluster is quite efficient. This opens opportunities to limit the complexity of the issue queues (monitoring less fewer buses) and register files (reducing number of ports, and number of physical registers) in the clusters, since not all results have to be forwarded to the other cluster.

The second direction that we intend to explore is associated with the approach we developed with Sembrant et al. [16]. It reduces the number of instructions waiting in the instruction queues for the applications benefiting from very large instruction windows. Instructions are dynamically classified as ready (independent from any long latency instruction) or non-ready, and as urgent (part of a dependency chain leading to a long latency instruction) or non-urgent. Non-ready non-urgent instructions can be delayed until the long latency instruction has been executed; this allows to reduce the pressure on the issue queue. This proposition opens the opportunity to consider an asymmetric microarchitecture with a cluster dedicated to the execution of urgent instructions and a second cluster executing the non-urgent instructions. The microarchitecture of this second cluster could be optimized to reduce complexity and power consumption (smaller instruction queue, less aggressive scheduling...)

Speculative execution.
Out-of-order (OoO) execution relies on speculative execution that requires predictions of all sorts: branch, memory dependency, value...

The PACAP members have been major actors of the branch prediction research for the last 20 years; and their proposals have influenced the design of most of the hardware branch predictors in current microprocessors. We will continue to steadily explore new branch predictor designs as for instance [18].

In speculative execution, we have recently revisited value prediction (VP) which was a hot research topic between 1996 and 2002. However it was considered up to recently that value prediction would lead to a huge increase in complexity and power consumption in every stage of the pipeline. Fortunately, we have recently shown that complexity usually introduced by value prediction in the OoO engine can be overcome [12], [11], [19], [9]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [12]. Thus, both prediction validation and recovery by squashing can be done outside the out-of-order engine, at commit time. Furthermore, we propose a new pipeline organization, EOLE ({Early | Out-of-order | Late} Execution), that leverages VP with validation at commit to execute many instructions outside the OoO core, in-order [11]. With EOLE, the issue-width in OoO core can be reduced without sacrificing performance, thus benefiting the performance of VP without a significant cost in silicon area and/or energy. In the near future, we will explore new avenues related to value prediction. These directions include register equality prediction and compatibility of value prediction with weak memory models in multiprocessors.

3.2.4. Towards heterogeneous single-ISA CPU-GPU architectures

Heterogeneous single-ISA architectures have been proposed in the literature during the 2000's [56] and are now widely used in the industry (ARM big.LITTLE, NVIDIA 4+1...) as a way to improve power-efficiency in mobile processors. These architectures include multiple cores whose respective microarchitectures offer different trade-offs between performance and energy efficiency, or between latency and throughput, while offering the same interface to software. Dynamic task migration policies leverage the heterogeneity of the platform by using the most suitable core for each application, or even each phase of processing. However, these works only tune cores by changing their complexity. Energy-optimized cores are either identical cores implemented in a low-power process technology, or simplified in-order superscalar cores, which are far from state-of-the-art throughput-oriented architectures such as GPUs.

We propose to investigate the convergence of CPU and GPU at both architecture and compilation levels.

Architecture.
The architecture convergence between Single Instruction Multiple Threads (SIMT) GPUs and multicore processors that we have been pursuing [36] opens the way for heterogeneous architectures including latency-optimized superscalar cores and throughput-optimized GPU-style cores, which all share the same instruction set. Using SIMT cores in place of superscalar cores will enable the highest energy efficiency on regular sections of applications. As with existing single-ISA heterogeneous architectures, task migration will not necessitate any software rewrite and will accelerate existing applications.

Compilers for emerging heterogeneous architectures.

Single-ISA CPU+GPU architectures will provide the necessary substrate to enable efficient heterogeneous processing. However, it will also introduce substantial challenges at the software and firmware level. Task placement and migration will require advanced policies that leverage both static information at compile time and dynamic information at run-time. We are tackling the heterogeneous task scheduling problem at the compiler level. As a first step, we are prototyping scheduling algorithms on existing multiple-ISA CPU+GPU architectures like NVIDIA Tegra X1.

3.2.5. Real-time systems

Safety-critical systems (e.g. avionics, medical devices, automotive...) have so far used simple unicore hardware systems as a way to control their predictability, in order to meet timing constraints. Still, many critical embedded systems have increasing demand in computing power, and simple unicore processors are not sufficient anymore. General-purpose multicore processors are not suitable for safety-critical real-time systems, because they include complex micro-architectural elements (cache hierarchies, branch, stride and value predictors) meant to improve average-case performance, and for which worst-case performance is difficult to predict. The prerequisite for calculating tight WCET is a deterministic hardware system that avoids dynamic, time-unpredictable calculations at run-time.

Even for multi and manycore systems designed with time-predictability in mind (Kalray MPPA manycore architecture [3], or the Recore manycore hardware [4]) calculating WCETs is still challenging. The following two challenges will be addressed in the mid-term:

1. definition of methods to estimate WCETs tightly on manycores, that smartly analyzes and/or controls shared resources such as buses, NoCs or caches;
2. methods to improve the programmability of real-time applications through automatic parallelization and optimizations from model-based designs.

3.2.6. Fault Tolerance

Technology trends suggest that, in tomorrow's computing world, failures will become commonplace due to many factors, and the expected probability of failure will increase with scaling. While well-known approaches, such as error correcting codes, exist to recover from failures and provide fault-free chips, the exponential growth of the number of faults will make them unaffordable in the future. Consequently, other approaches such as fine-grained disabling and reconfiguration of hardware elements (e.g. individual functional units or cache blocks) will become economically necessary. We are going to enter a new era: functionally correct chips with variable performance among chips and throughout their lifetime [58].

Transient and permanent faults may be detected by similar techniques, but correcting them generally involves different approaches. We are primarily interested in permanent faults, even though we do not necessarily disregard transient faults (e.g. the TMR approach in the next paragraph addresses both kind of faults).

CPU.
Permanent faults can occur anywhere in the processor. The performance implications of faulty cells vary depending on how the array is used in a processor. Most of micro-architectural work aiming at assessing the performance implications of permanently faulty cells relies on simulations with random fault-maps. These studies are, therefore, limited by the fault-maps they use that may not be representative for the average and distributed performance. They also do not consider aging effect.

Considering the memory hierarchy, we have already studied [5] the impact of permanent faults on the average and worst-case performance based on analytical models. We will extend these models to cover other components and other designs, and to analyze the interaction between faulty components.

For identified critical hardware structures, such as the memory hierarchy, we will propose protection mechanisms by for instance using larger cells, or even by selecting a different array organization to mitigate the impact of faults.

[3] http://www.kalrayinc.com
[4] http://www.recoresystems.com/

Another approach to deal with faults is to introduce redundancy at the code level. We propose to consider static compilation techniques focusing on existing hardware. As an example, we plan to leverage SIMD extensions of current instruction sets to introduce redundancy in scalar code at minimum cost. With these instructions, it will be possible to protect the execution from both soft errors by using TMR (triple modular redundancy) with voters in the code itself, and permanent faults without the need of extra hardware support to deconfigure faulty functional units.

Reconfigurable Computing.
In collaboration with the CAIRN project-team, we propose to construct Coarse Grain Reconfigurable Architectures (CGRA) from a sea of basic arithmetic and memory elements organized into clusters and connected through a hierarchical interconnection network. These clusters of basic arithmetic operators (e.g. 8-bit arithmetic and logic units) would be able to be seamlessly configured to various accuracy and data types to adapt the consumed energy to application requirements taking advantage of approximate computations. We propose to add new kinds of error detection (and sometimes correction) directly at the operator level by taking advantage of the massive redundancy of the array. As an example, errors can be tracked and detected in a complex sequence of double floating-point operations by using a reduced-precision version of the same processing.

Such reconfigurable blocks will be driven by compilation techniques, in charge of computing checkpoints, detecting faults, and replaying computations when needed.

Dynamic compilation techniques will help better exploit faulty hardware, by allocating data and computations on correct resources. In case of permanent faults, we will provide a mechanism to reconfigure the hardware, for example by reducing the issue width of VLIW processors implemented in CGRA. Dynamic code generation (JIT compiler) will re-generate code for the new configuration, guaranteeing portability and optimal exploitation of the hardware.

3.2.7. Power efficiency

PACAP will address power-efficiency at several levels. First, we will design static and split compilation techniques to contribute to the race for Exascale computing (the general goal is to reach 10^{18} FLOP/s at less than 20 MW). Second, we will focus on high-performance low-power embedded compute nodes. will research new static and dynamic compilation techniques that fully exploit emerging memory and NoC technologies. Finally, in collaboration with the CAIRN project-team, we will investigate the synergy of reconfigurable computing and dynamic code generation.

Green and heterogeneous high-performance computing.
Concerning HPC systems, our approach consists in mapping, runtime managing and autotuning applications for green and heterogeneous High-Performance Computing systems up to the Exascale level. One key innovation of the proposed approach consists of introducing a separation of concerns (where self-adaptivity and energy efficient strategies are specified aside to application functionalities) promoted by the definition of a Domain Specific Language (DSL) inspired by aspect-oriented programming concepts for heterogeneous systems. The new DSL will be introduced for expressing adaptivity/energy/performance strategies and to enforce at runtime application autotuning and resource and power management. The goal is to support the parallelism, scalability and adaptability of a dynamic workload by exploiting the full system capabilities (including energy management) for emerging large-scale and extreme-scale systems, while reducing the Total Cost of Ownership (TCO) for companies and public organizations.

High-performance low-power embedded compute nodes.
We will address the design of next generation energy-efficient high-performance embedded compute nodes. It focuses at the same time on software, architecture and emerging memory and communication technologies in order to synergistically exploit their corresponding features. The approach of the project is organized around three complementary topics: 1) compilation techniques; 2) multicore architectures; 3) emerging memory and communication technologies. PACAP will focus on the compilation aspects, taking as input the software-visible characteristics of the proposed emerging technology, and making the best possible use of the new features (non-volatility, density, endurance, low-power).

Hardware Accelerated JIT Compilation.
Reconfigurable hardware offers the opportunity to limit power consumption by dynamically adjusting the number of available resources to the requirements of the running software. In particular, VLIW processors can adjust the number of available issue lanes. Unfortunately, changing the processor width often requires recompiling the application, and VLIW processors are highly dependent of the quality of the compilation, mainly because of the instruction scheduling phase performed by the compiler. Another challenge lies in the high constraints of the embedded system: the energy and execution time overhead due to the JIT compilation must be carefully kept under control.

We started exploring ways to reduce the cost of JIT compilation targeting VLIW-based heterogeneous many-core systems. Our approach lies on a hardware/software JIT compiler framework. While basic optimizations and JIT management are performed in software, the compilation back-end is implemented by means of specialized hardware. This back-end involves both instruction scheduling and register allocation, which are known to be the most time-consuming stages of such a compiler.

3.2.8. Security

Security is a mandatory concern of any modern computing system. Various threat models have led to a multitude of protection solutions. ALF already has contributions, thanks to the HAVEGE [62] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [55], or thread-based control flow mangling [60]).

We plan to partner with security experts who can provide intuition, know-how and expertise, in particular in defining threat models, and assessing the quality of the solutions. Our background in compilation and architecture will help design more efficient and less expensive protection mechanisms.

We already have ongoing research directions related to security. We also plan to partner with the Inria/CentraleSupelec CIDRE project-team to design a tainting technique based on a just-in-time compiler.

Compiler-based data protection.
We will specify and design error correction codes suitable for an efficient protection of sensitive information in the context of Internet of Things (IoT) and connected objects. We will partner with experts in security and codes to prototype a platform that demonstrates resilient software. PACAP's expertise will be key to select and tune the protection mechanisms developed within the project, and to propose safe, yet cost-effective solutions from an implementation point of view.

JIT-based tainting.
Dynamic information flow control (DIFC, also known as *tainting*) is used used to detect intrusions and to identify vulnerabilities. It consists in attaching metadata (called *taints* or *labels*) to information containers, and to propagate the taints when particular operations are applied to the containers: reads, writes, etc. The goal is then to guarantee that confidential information is never used to generate data sent to an untrusted container; conversely, data produced by untrusted entities cannot be used to update sensitive data.

The containers can be of various granularities: fine-grain approaches can deal with single variables, coarser-grain approaches consider a file as a whole. The CIDRE project-team has developed several DIFC monitors. kBlare is coarse-grain monitor in the Linux kernel. JBlare is a fine-grain monitor for Java applications. Fine-grain monitors provide a better precision at the cost of a significant overhead in execution time.

We propose to combine the expertise of CIDRE in DIFC with our expertise in JIT compilation to design hybrid approaches. An initial static analysis of the program prior to installation or execution will feed information to a dynamic analyzer that propagates taints during just-in-time compilation.

4. Application Domains

4.1. Any computer usage

The PACAP team is working on the fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impacts on any application domain that requires high performance execution (telecommunication, multimedia, biology, health, engineering, environment...), but also on many embedded applications that exhibit other constraints such as power consumption, code size and guaranteed response time. Our research activity implies the development of software prototypes.

5. Highlights of the Year

5.1. Highlights of the Year

André Seznec was elevated as an ACM Fellow in December 2016 with the citation: “For contributions to branch prediction and cache memory design”.

André Seznec won the three tracks of the 5th Championship on Branch Prediction.

5.1.1. Awards

Sajith Kalathingal, Sylvain Collange, Bharath Swamy and André Seznec received the Best Paper award of the SBAC-PAD 2016 conference.

Damien Hardy, Isabelle Puaut, Yiannakis Sazeides won the best paper award of the Embedded Systems Software track at DATE 2016: Probabilistic WCET estimation in presence of hardware for mitigating the impact of permanent faults. Design, Automation and Test in Europe. Dresden, Germany, March 2016.

Aswinkumar Sridharan and André Seznec won the best paper award for “Discrete Cache Insertion Policies for Shared Last Level Cache Management on Large Multicores” at the 30th IEEE International Parallel & Distributed Processing Symposium, May 2016, Chicago.

For his PhD thesis [10] “Increasing the Performance of Superscalar Processors through Value Prediction”, Arthur Perais received:

- Prix de thèse Fondation Rennes 1, 1er Prix de l’école doctorale MATISSE;
- Prix de thèse Gilles Kahn, accessit.

BEST PAPERS AWARDS:

[46]
A. SEZNEC. *TAGE-SC-L Branch Predictors Again*, in "5th JILP Workshop on Computer Architecture Competitions (JWAC-5): Championship Branch Prediction (CBP-5)", Seoul, South Korea, June 2016, https://hal.inria.fr/hal-01354253

[45]
A. SEZNEC. *Exploring branch predictability limits with the MTAGE+SC predictor* *, in "5th JILP Workshop on Computer Architecture Competitions (JWAC-5): Championship Branch Prediction (CBP-5)", Seoul, South Korea, June 2016, 4 p. , https://hal.inria.fr/hal-01354251

[36]
S. KALATHINGAL, S. COLLANGE, B. NARASIMHA SWAMY, A. SEZNEC. *Dynamic Inter-Thread Vectorization Architecture: extracting DLP from TLP*, in "International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD)", Los Angeles, United States, October 2016, https://hal.inria.fr/hal-01356202

[35]

D. HARDY, I. PUAUT, Y. SAZEIDES. *Probabilistic WCET estimation in presence of hardware for mitigating the impact of permanent faults*, in "Design, Automation and Test in Europe", Dresden, Germany, March 2016, https://hal.inria.fr/hal-01259493

[48]
A. SRIDHARAN, A. SEZNEC. *Discrete Cache Insertion Policies for Shared Last Level Cache Management on Large Multicores*, in "30th IEEE International Parallel & Distributed Processing Symposium", Chicago, United States, May 2016, https://hal.inria.fr/hal-01259626

6. New Software and Platforms

6.1. ATMI

KEYWORDS: Analytic model - Chip design - Temperature
SCIENTIFIC DESCRIPTION

Research on temperature-aware computer architecture requires a chip temperature model. General purpose models based on classical numerical methods like finite differences or finite elements are not appropriate for such research, because they are generally too slow for modeling the time-varying thermal behavior of a processing chip.

We have developed an ad hoc temperature model, ATMI (Analytical model of Temperature in MIcroprocessors), for studying thermal behaviors over a time scale ranging from microseconds to several minutes. ATMI is based on an explicit solution to the heat equation and on the principle of superposition. ATMI can model any power density map that can be described as a superposition of rectangle sources, which is appropriate for modeling the microarchitectural units of a microprocessor.
FUNCTIONAL DESCRIPTION

ATMI is a library for modelling steady-state and time-varying temperature in microprocessors. ATMI uses a simplified representation of microprocessor packaging.

- Participant: Pierre Michaud
- Contact: Pierre Michaud
- URL: https://team.inria.fr/pacap/software/atmi/

6.2. Heptane

KEYWORDS: Static analysis - Real time - Performance - WCET - IPET - Worst Case Execution Time
SCIENTIFIC DESCRIPTION

WCET estimation

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v3, with number IDDN.FR.001.510039.000.S.P.2003.000.10600.

The aim of Heptane is to produce upper bounds of the execution times of applications. It is targeted at applications with hard real-time requirements (automotive, railway, aerospace domains). Heptane computes WCETs using static analysis at the binary code level. It includes static analyses of microarchitectural elements such as caches and cache hierarchies.

For more information, please contact Damien Hardy or Isabelle Puaut.
FUNCTIONAL DESCRIPTION

In a hard real-time system, it is essential to comply with timing constraints, and Worst Case Execution Time (WCET) in particular. Timing analysis is performed at two levels: analysis of the WCET for each task in isolation taking account of the hardware architecture, and schedulability analysis of all the tasks in the system. Heptane is a static WCET analyser designed to address the first issue.

- Participants: Isabelle Puaut, Damien Hardy, Loïc Besnard
- Partner: Université de Rennes 1
- Contact: Isabelle Puaut
- URL: https://team.inria.fr/pacap/software/heptane/

6.3. Tiptop

KEYWORDS: HPC - Performance - CPU - Cache - Cycles - Instructions - Branch predictor
SCIENTIFIC DESCRIPTION

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display.

Performance, hardware counters, analysis tool.

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2, with number IDDN.FR.001.450006.000.S.P.2011.000.10800. Current version is 2.3, released July 2015.

Tiptop has been integrated in major Linux distributions, such as Fedora, Debian, Ubuntu.

Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.

No privilege is required, any user can run tiptop
FUNCTIONAL DESCRIPTION

Today's microprocessors have become extremely complex. To better understand the multitude of internal events, manufacturers have integrated many monitoring counters. Tiptop can be used to collect and display the values from these performance counters very easily. Tiptop may be of interest to anyone who wants to optimise the performance of their HPC applications.

- Participant: Erven Rohou
- Contact: Erven Rohou
- URL: http://tiptop.gforge.inria.fr

6.4. ATC

Address Trace Compression
KEYWORDS: Compressing - Decompressing - Address traces
FUNCTIONAL DESCRIPTION

ATC is a utility and a C library for compressing/decompressing address traces. It implements a new lossless transformation, Bytesort, that exploits spatial locality in address traces. ATC leverages existing general-purpose compressors such as gzip and bzip2. ATC also provides a lossy compression mode that yields higher compression ratios while preserving certain important characteristics of the original trace.

- Participant: Pierre Michaud
- Contact: Pierre Michaud
- URL: https://team.inria.fr/pacap/software/atc/

6.5. Barra

Modelisation of a GPU architecture
KEYWORDS: Simulator - GPU - Computer architecture
SCIENTIFIC DESCRIPTION

Research on throughput-oriented architectures demands accurate and representative models of GPU architectures in order to be able to evaluate new architectural ideas, explore design spaces and characterize applications. The Barra project is a simulator of the NVIDIA Tesla GPU architecture.

Barra builds upon knowledge acquired through micro-benchmarking, in order to provide a baseline model representative of industry practice. The simulator provides detailed statistics to identify optimization opportunities and is fully customizable to experiment ideas of architectural modifications. Barra incorporates both a functional model and a cycle-level performance model.
FUNCTIONAL DESCRIPTION

Barra simulates CUDA programs at the assembly language level (Tesla ISA). Its ultimate goal is to provide a 100 % bit-accurate simulation, offering bug-for-bug compatibility with NVIDIA G80-based GPUs. It works directly with CUDA executables, neither source modification nor recompilation is required.

Barra is primarily intended as a tool for research in computer architecture, although it can also be used to debug, profile and optimize CUDA programs at the lowest level.

- Participants: Sylvain Collange, David Defour, Alexandre Kouyoumdjian and Fabrice Mouhartem
- Contact: Sylvain Collange
- URL: http://barra.gforge.inria.fr/

6.6. If-memo

KEYWORD: Performance, function memoization, dynamic optimization
Status: Ongoing development, early prototype. Registered with APP (Agence de Protection des Programmes) under number IDDN.FR.001.250013.000.S.P.2015.000.10800.

SCIENTIFIC DESCRIPTION

Memoization is the technique of saving result of executions so that future executions can be omitted when the inputs repeat. Memoization has been proposed in previous literature at the instruction level, basic block level and function level using hardware as well as pure software level approaches including changes to programming language.

We proposed software memoization of pure functions for procedural languages. We rely on the operating system loader, taking advantage of the LD_PRELOAD feature of UNIX systems. By setting this variable to the path of a shared library, we instruct the loader to first look to missing symbols in that library. Our library redefines the functions we wish to intercept. The interception code is very straightforward: it receives the same parameter as the target function and checks in a table (a software cache) if this value is readily available. In the favorable case, the result value is immediately returned. Otherwise, we invoke the original function, and store the result in the cache before returning it.

Our technique does not require the availability of source code and thus can be applied even to commercial applications as well as applications with legacy codes. As far as users are concerned, enabling memoization is as simple as setting an environment variable. We validated If-memo with x86-64 platform using both GCC and icc compiler tool-chains, and ARM cortex-A9 platform using GCC.

- Participants: Erven Rohou and Arjun Suresh
- Contact: Erven Rohou

6.7. Padrone

KEYWORDS: Legacy code - Optimization - Performance analysis - Dynamic Optimization

Status: Registered with APP (Agence de Protection des Programmes) under number IDDN.FR.001.250013.000.S.P.2015.000.10800.

FUNCTIONAL DESCRIPTION

Padrone is new platform for dynamic binary analysis and optimization. It provides an API to help clients design and develop analysis and optimization tools for binary executables. Padrone attaches to running applications, only needing the executable binary in memory. No source code or debug information is needed. No application restart is needed either. This is especially interesting for legacy or commercial applications, but also in the context of cloud deployment, where actual hardware is unknown, and other applications competing for hardware resources can vary. The profiling overhead is minimum.

- Participants: Erven Rohou and Emmanuel Riou
- Contact: Erven Rohou
- https://team.inria.fr/pacap/software/Padrone/

6.8. STiMuL

Steady temperature in Multi-Layers components
FUNCTIONAL DESCRIPTION

STiMuL is a C library for modeling steady-state heat conduction in microprocessors. It can be used to obtain temperature from power density or power density from temperature. It can also be used to model stacked dies. STiMuL does not model time-varying temperature. For time-varying temperature, other models must be used, such as ATMI.

- Participant: Pierre Michaud
- Contact: Pierre Michaud
- URL: https://team.inria.fr/pacap/software/stimul/

6.9. TPCalc

Throughput calculator
KEYWORDS: Architecture - Performance analysis
FUNCTIONAL DESCRIPTION

TPCalc is a throughput calculator for microarchitecture studies concerned with multi-program workloads consisting of sequential programs. Because microarchitecture simulators are slow, it is difficult to simulate throughput experiments where a multicore executes many jobs that enter and leave the system. The usual practice of measuring instantaneous throughput on independent coschedules chosen more or less randomly is not a rigorous practice because it assumes that all the coschedules are equally important, which is not always true. TPCalc can compute the average throughput of a throughput experiment without actually doing the throughput experiment. The user first defines the workload heterogeneity (number of different job types), the multicore configuration (number of cores and symmetries). TPCalc provides a list of base coschedules. The user then simulates these coschedules, using some benchmarks of his choice, and feeds back to TPCalc the measured execution rates (e.g., instructions per cycle or instructions per second).TPCalc eventually outputs the average throughput.

- Participant: Pierre Michaud
- Partner: Ghent University
- Contact: Pierre Michaud
- URL: https://team.inria.fr/pacap/software/tpcalc/

6.10. Parasuite

Participants: Sylvain Collange, Imane Lasri, Erven Rohou, André Seznec.

Parasuite: parallel benchmarks for multi-core CPUs, clusters and accelerators

Despite the ubiquity of parallel architectures in all computing segments, the research community often lacks benchmarks representative of parallel applications. The Inria Parallel Benchmark Suite (Parasuite) seeks to address this need by providing a set of representative parallel benchmarks for the architecture, compiler and system research communities. Parasuite targets the main contemporary parallel programming technologies: shared-memory multi-thread parallelism for multi-core, message-passing parallelism for clusters and fine-grained data-level parallelism for GPU architectures and SIMD extensions.

All benchmarks come with input datasets of various sizes, to accommodate use cases ranging from microarchitecture simulation to large-scale performance evaluation. Correctness checks on the computed results enable automated regression testing. In order to support computer arithmetic optimization and approximate computing research scenarios, the correctness checks favor accuracy metrics evaluating domain-specific relevance rather than bit-exact comparisons against an arbitrary reference output.

Visit: http://parasuite.inria.fr/

6.11. Simty

Participant: Sylvain Collange.

Simty: A Synthesizable General-Purpose SIMT Processor.

Simty is a massively multi-threaded processor core that dynamically assembles SIMD instructions from scalar multi-thread code. It runs the RISC-V (RV32-I) instruction set. Unlike existing SIMD or SIMT processors like GPUs, Simty takes binaries compiled for general-purpose processors without any instruction set extension or compiler changes. Simty is described in synthesizable VHDL.

Visit: http://team.inria.fr/pacap/simty

7. New Results

7.1. Compiler, vectorization, interpretation

Participants: Erven Rohou, Emmanuel Riou, Arjun Suresh, André Seznec, Nabil Hallou, Sylvain Collange, Rabab Bouziane, Arif Ali Ana-Pparakkal, Stefano Cherubin.

7.1.1. Improving sequential performance through memoization

Participants: Erven Rohou, Emmanuel Riou, André Seznec, Arjun Suresh.

Many applications perform repetitive computations, even when properly programmed and optimized. Performance can be improved by caching results of pure functions, and retrieving them instead of recomputing a result (a technique called memoization).

We propose [20] a simple technique for enabling software memoization of any dynamically linked pure function and we illustrate our framework using a set of computationally expensive pure functions – the transcendental functions.

Our technique does not need the availability of source code and thus can be applied even to commercial applications as well as applications with legacy codes. As far as users are concerned, enabling memoization is as simple as setting an environment variable.

Our framework does not make any specific assumptions about the underlying architecture or compiler toolchains, and can work with a variety of current architectures.

We present experimental results for x86-64 platform using both gcc and icc compiler tool-chains, and for ARM cortex-A9 platform using gcc. Our experiments include a mix of real world programs and standard benchmark suites: SPEC and Splash2x. On standard benchmark applications that extensively call the transcendental functions we report memoization benefits of upto 16 %, while much higher gains were realized for programs that call the expensive Bessel functions. Memoization was also able to regain a performance loss of 76 % in *bwaves* due to a known performance bug in the gcc libm implementation of *pow* function.

Initial work has been published in ACM TACO 2015 [20] and accepted for presentation at the International Conference HiPEAC 2016 in Prague.

Further developments have been accepted for publication at the Compiler Construction Conference 2017 [49].

This research is described in the PhD thesis of Arjun Suresh [24].

7.1.2. Optimization in the Presence of NVRAM

Participants: Erven Rohou, Rabab Bouziane.

Energy-efficiency is one of the most challenging design issues in both embedded and high-performance computing domains. The aim is to reduce as much as possible the energy consumption of considered systems while providing them with the best computing performance. Finding an adequate solution to this problem certainly requires a cross-disciplinary approach capable of addressing the energy/performance trade-off at different system design levels.

We proposed [42] an empirical impact analysis of the integration of Spin Transfer Torque Magnetic Random Access Memory (STT-MRAM) technologies in multicore architectures when applying some existing compiler optimizations. For that purpose, we use three well-established architecture and NVM evaluation tools: NVSim, gem5 and McPAT. Our results show that the integration of STT-MRAM at cache memory levels enables a significant reduction of the energy consumption (up to 24.2 % and 31 % on the considered multicore and monocore platforms respectively) while preserving the performance improvement provided by typical code optimizations. We also identified how the choice of the clock frequency impacts the relative efficiency of the considered memory technologies.

This research is done in collaboration with Abdoulaye Gamatié at LIRMM (Montpellier) within the context the the ANR project CONTINUUM.

7.1.3. Hardware/Software JIT Compiler

Participant: Erven Rohou.

Dynamic Binary Translation (DBT) is often used in hardware/software co-design to take advantage of an architecture model while using binaries from another one. The co-development of the DBT engine and of the execution architecture leads to architecture with special support to these mechanisms. We proposed a hardware accelerated dynamic binary translation where the first steps of the DBT process are fully accelerated in hardware. Results shows that using our hardware accelerators leads to a speed-up of 8× and a cost in energy 18× lower, compared with an equivalent software approach.

An initial version of this work has been presented at Compas'16 [51]. The latest results have been accepted for publication at DATE 2017 [44].

This research is done in collaboration with Steven Derrien and Simon Rokicki from the CAIRN team.

7.1.4. Dynamic Parallelization of Binary Programs

Participants: Erven Rohou, Emmanuel Riou, Nabil Hallou.

We address runtime automatic parallelization of binary executables, assuming no previous knowledge on the executable code. The Padrone platform is used to identify candidate functions and loops. Then we disassemble the loops and convert them to the intermediate representation of the LLVM compiler. This allows us to leverage the power of the polyhedral model for auto-parallelizing loops. Once optimized, new native code is generated just-in-time in the address space of the target process.

Our approach enables user transparent auto-parallelization of legacy and/or commercial applications with auto-parallelization.

This work has been accepted for publication in the Springer journal IJPP: “Runtime Vectorization Transformations of Binary Code”.

This work is done in collaboration with Philippe Clauss (Inria CAMUS).

7.1.5. Dynamic Function Specialization

Participants: Erven Rohou, Arif Ali Ana-Pparakkal.

Compilers can do better optimization with the knowledge of run-time behaviour of the program. *Function Specialization* is an optimization technique in which different versions of a function are created according to the value of its arguments. It can be difficult to predict the exact value/behaviour of arguments during static compilation and so it is difficult for a static compiler to do efficient function specialization. In our *dynamic function specialization* technique, we capture the actual value of arguments during execution of the program and, when profitable, create specialized versions and include them at runtime.

This research is done within the context of the Nano 2017 PSAIC collaborative project.

7.1.6. Application Autotuning for Performance and Energy

Participants: Erven Rohou, Stefano Cherubin, Imane Lasri.

Due to the increasing complexity of both applications behaviors and underlying hardware, achieving reasonable (not to mention best) performance can hardly be done at compile time. Autotuning is an approach where a runtime manager is able to adapt the software to the runtime conditions. We have developed a framework and shown through a domain specific application initial exploration scenarios [32], [47].

We started characterizing applications – in particular the Parasuite benchmarks – and we will rely on split-compilation [2] embed hints and heuristics inside a binary program for dynamic adaptation and optimization.

This research is done within the context of the H2020 FET HPC collaborative project ANTAREX.

7.1.7. Customized Precision Computing

Participants: Erven Rohou, Stefano Cherubin, Imane Lasri.

Customized precision originates from the fact that many applications can tolerate some loss of quality during computation, as in the case of media processing (audio, video and image), data mining, machine learning, etc. Error-tolerating applications are increasingly common in the emerging field of real-time HPC. Thus, recent works have investigated this line of research in the HPC domain as a way to provide a breakthrough in power and performance for the Exascale era.

We aim at leveraging existing, HPC-oriented hardware architectures, while including in the precision tuning an adaptive selection of floating and fixed-point arithmetic. It is part of a wider effort to provide the programmers with an easy way to manage extra-functional properties of programs, including precision, power, and performance.

We explore tradeoffs between precision and time-to-solution, as well as precision and energy-to-solution.

This is done within the context of the ANTAREX project in collaboration with Stefano Cherubin, Cristina Silvano and Giovanni Agosta from Politecnico di Milano, and Olivier Sentieys from the CAIRN team.

7.1.8. SPMD Function Call Re-Vectorization

Participant: Sylvain Collange.

SPMD programming languages for SIMD hardware such as C for CUDA, OpenCL or ISPC have contributed to increase the programmability of SIMD accelerators and graphics processing units. However, SPMD languages still lack the flexibility offered by low-level SIMD programming on explicit vectors. To close this expressiveness gap while preserving the SPMD abstraction, we introduce the notion of Function Call Re-Vectorization (CREV) [38]. CREV allows changing the dimension of vectorization during the execution of an SPMD kernel, and exposes it as a nested parallel kernel call. CREV affords a programmability close to dynamic parallelism, a feature that allows the invocation of kernels from inside kernels, but at much lower cost. In this paper, we present a formal semantics of CREV, and an implementation of it on the ISPC compiler. To validate our idea, we have used CREV to implement some classic algorithms, including string matching, depth first search and Bellman-Ford, with minimum effort. These algorithms, once compiled by ISPC to Intel-based vector instructions, are as fast as state-of-the-art implementations, yet much simpler. As an example, our straightforward implementation of string matching beats the Knuth-Morris-Pratt algorithm by 12 %.

This work was done during the internship of Rubens Emilio in Rennes in collaboration with Sylvain Collange and Fernando Pereira (UFMG) as part of the Inria PROSPIEL Associate Team.

7.1.9. SPMD Function Call Fusion

Participant: Sylvain Collange.

The increasing popularity of Graphics Processing Units (GPUs) has brought renewed attention to old problems related to the Single Instruction, Multiple Data execution model. One of these problems is the reconvergence of divergent threads. A divergence happens at a conditional branch when different threads disagree on the path to follow upon reaching this split point. Divergences may impose a heavy burden on the performance of parallel programs.

We have proposed a compiler-level optimization to mitigate the performance loss due to branch divergence on GPUs [21]. This optimization consists in merging function call sites located at different paths that sprout from the same branch. We show that our optimization adds negligible overhead on the compiler. When not applicable, it does not slow down programs and it accelerates substantially those in which it is applicable. As an example, we have been able to speed up the well known SPLASH Fast Fourier Transform benchmark by 11 %.

This work is done in collaboration with Douglas do Couto Teixeira and Fernando Pereira from UFMG as part of the Inria PROSPIEL Associate Team.

7.1.10. SIMD programming in SPMD: application to multi-precision computations

Participant: Sylvain Collange.

GPUs are an important hardware development platform for problems where massive parallel computations are needed. Many of these problems require a higher precision than the standard double floating-point (FP) available. One common way of extending the precision is the multiple-component approach, in which real numbers are represented as the unevaluated sum of several standard machine precision FP numbers. This representation is called a FP expansion and it offers the simplicity of using directly available and highly optimized FP operations. We propose new data-parallel algorithms for adding and multiplying FP expansions specially designed for extended precision computations on GPUs [34]. These are generalized algorithms that can manipulate FP expansions of different sizes (from double-double up to a few tens of doubles) and ensure a certain worst case error bound on the results.

This work is done in collaboration with Mioara Joldes (CNRS/LAAS), Jean-Michel Muller (CNRS/LIP) and Valentina Popescu (ENS Lyon/LIP).

7.2. Processor Architecture

Participants: Pierre Michaud, Sylvain Collange, Erven Rohou, André Seznec, Arthur Perais, Sajith Kalathingal, Andrea Mondelli, Aswinkumar Sridharan, Biswabandan Panda, Fernando Endo, Kleovoulos Kalaitzidis.

Processor, cache, locality, memory hierarchy, branch prediction, multicore, power, temperature

7.2.1. Microarchitecture

7.2.1.1. Branch prediction

Participant: André Seznec.

IMLI-based predictors

The wormhole (WH) branch predictor was recently introduced to exploit branch outcome correlation in multidimensional loops. For some branches encapsulated in a multidimensional loop, their outcomes are correlated with those of the same branch in neighbor iterations, but in the previous outer loop iteration. In [18], we introduced practical predictor components to exploit this branch outcome correlation in multidimensional loops: the IMLI-based predictor components. The iteration index of the inner most loop in an application can be efficiently monitored at instruction fetch time using the Inner Most Loop Iteration (IMLI) counter. The outcomes of some branches are strongly correlated with the value of this IMLI counter. Our experiments show that augmenting a state-of-the-art global history predictor such as TAGE-SC-L [45] with IMLI-based components outperforms previous state-of-the-art academic predictors leveraging local and global history at much lower hardware complexity (i.e., smaller storage budget , smaller number of tables and simpler management of speculative states).

This study was accepted in the special issue Top Picks of the best papers in 2015 computer architecture conferences in IEEE Micro [30].

This research was done in collaboration with Joshua San Miguel and Jorge Albericio from University of Toronto

Championship Branch Prediction

The 5th Championship Branch Prediction was organized in Seoul in June 2016. The predictors submitted by the PACAP-team, respectively TAGE-SC-L and MTAGE-SC, for limited storage budgets and infinite storage budgets won the three tracks of the competition [46], [45]. These predictors are derived from our reference work [17].

7.2.1.2. Revisiting Value Prediction

Participants: Arthur Perais, André Seznec.

Value prediction was proposed in the mid 90's to enhance the performance of high-end microprocessors. From 2013 to 2016, we have progressively revived the interest in value prediction. At a first step, we showed that all predictors are amenable to very high accuracy at the cost of some loss on prediction coverage [12]. Furthermore, we proposed EOLE [13]. EOLE leverages Value Prediction to *Early Execute* simple instructions whose operands are ready in parallel with Rename and to *Late Execute* to simple predicted instructions just before Commit. EOLE allows to reduce the out-of-order issue-width by 33% without impeding performance.

An extension of the initial EOLE paper [13] was published in ACM TOCS [27].

7.2.1.3. Physical register sharing

Participants: Arthur Perais, André Seznec.

Sharing a physical register between several instructions is needed to implement several microarchitectural optimizations. However, register sharing requires modifications to the register reclaiming process: Committing a single instruction does not guarantee that the physical register allocated to the previous mapping of its architectural destination register is free-able anymore. Consequently, a form of register reference counting must be implemented. While such mechanisms (e.g., dependency matrix, per register counters) have been described in the literature, we argue that they either require too much storage, or that they lengthen branch misprediction recovery by requiring sequential rollback. As an alternative, we present the Inflight Shared Register Buffer (ISRB), a new structure for register reference counting [41]. The ISRB has low storage overhead and lends itself to checkpoint-based recovery schemes, therefore allowing fast recovery on pipeline flushes. We illustrate our scheme with Move Elimination (short-circuiting moves) and an implementation of Speculative Memory Bypassing (short-circuiting store-load pairs) that makes use of a TAGE-like predictor to identify memory dependencies. We show that the whole potential of these two mechanisms can be achieved with a small register tracking structure.

7.2.1.4. Register Sharing for Equality Prediction

Participants: Arthur Perais, Fernando Endo, André Seznec.

Recently, Value Prediction (VP) has been gaining renewed traction in the research community. VP speculates on the result of instructions to increase Instruction Level Parallelism (ILP). In most embodiments, VP requires large tables to track predictions for many static instructions. However, in many cases, it is possible to detect that the result of an instruction is produced by an older inflight instruction, but not to predict the result itself. Consequently it is possible to rely on predicting register equality and handle speculation through the renamer. To do so, we propose to use Distance Prediction [40], a technique that was previously used to perform Speculative Memory Bypassing (short-circuiting def-store-load-use chains). Distance Prediction attempts to determine how many instructions separate the instruction of interest and the most recent older instruction that produced the same result. With this information, the physical register identifier of the older instruction can be retrieved from the ROB and provided to the renamer. The implementation of Distance Prediction necessitates a hardware mechanism to handle the sharing of physical registers as the ISRB [41].

7.2.1.5. Storage-Free Memory Dependency Prediction

Participants: Arthur Perais, André Seznec.

Memory Dependency Prediction (MDP) is paramount to good out-of-order performance, but decidedly not trivial as a all instances of a given static load may not necessarily depend on all instances of a given static store. As a result, for a given load, MDP should predict the exact store instruction the load depends on, and not only whether it depends on an inflight store or not, i.e., ideally, prediction should not be binary. However, we first argue that given the high degree of sophistication of modern branch predictors, the fact that a given dynamic load depends on an inflight store can be captured using the binary prediction capabilities of the branch predictor, providing coarse MDP at zero storage overhead. Second, by leveraging hysteresis counters, we show that the precise producer store can in fact be identified. This embodiment of MDP yields performance levels that are on par with state-of-the-art, and requires less than 70 additional bits of storage over a baseline without MDP at all [28].

7.2.1.6. Compressed Caches

Participants: André Seznec, Biswabandan Panda.

The YACC compressed cache

Cache memories play a critical role in bridging the latency, bandwidth, and energy gaps between cores and off-chip memory. However, caches frequently consume a significant fraction of a multicore chip's area, and thus account for a significant fraction of its cost. Compression has the potential to improve the effective capacity of a cache, providing the performance and energy benefits of a larger cache while using less area. The design of a compressed cache must address two important issues: i) a low-latency, low-overhead compression algorithm that can represent a fixed-size cache block using fewer bits and ii) a cache organization that can efficiently store the resulting variable-size compressed blocks. This paper focuses on the latter issue. We propose YACC (Yet Another Compressed Cache), a new compressed cache design that targets improving effective cache capacity with a simple design [29]. YACC uses super-blocks to reduce tag overheads, while packing variable-size compressed blocks to reduce internal fragmentation. YACC achieves the benefits of two state-of-the art compressed caches, Decoupled Compressed Cache (DCC) [61] and Skewed Compressed Cache (SCC) [15], with a more practical and simpler design. YACC's cache layout is similar to conventional caches, with a largely unmodified tag array and unmodified data array.

This study was done in collaboration with Somayeh Sardashti and David Wood from University of Wisconsin.

The DISH compression scheme

The effectiveness of a compressed cache depends on three features: i) the compression scheme, ii) the compaction scheme, and iii) the cache layout of the compressed cache. Both SCC [15] and YACC [29] use compression techniques to compress individual cache blocks, and then a compaction technique to compact multiple contiguous compressed blocks into a single data entry. The primary attribute used by these techniques for compaction is the compression factor of the cache blocks, and in this process, they waste cache space. We propose dictionary sharing (DISH), a dictionary based cache compression scheme that reduces this wastage [39]. DISH compresses a cache block by keeping in mind that the block is a potential candidate for the compaction process. DISH encodes a cache block with a dictionary that stores the distinct 4-byte chunks of a cache block and the dictionary is shared among multiple neighboring cache blocks. The simple encoding scheme of DISH also provides a single cycle decompression latency and it does not change the cache layout of compressed caches. Compressed cache layouts that use DISH outperforms the compression schemes, such as BDI and CPACK+Z, in terms of compression ratio, system performance , and energy efficiency.

7.2.1.7. Clustered microarchitecture

Participants: Andrea Mondelli, Pierre Michaud, André Seznec.

In the last 10 years, the clock frequency of high-end superscalar processors did not increase significantly. Performance keeps being increased mainly by integrating more cores on the same chip and by introducing new instruction set extensions. However, this benefits only to some applications and requires rewriting and/or recompiling these applications. A more general way to increase performance is to increase the IPC, the number of instructions executed per cycle.

In [8], we argue that some of the benefits of technology scaling should be used to increase the IPC of future superscalar cores. Starting from microarchitecture parameters similar to recent commercial high-end cores, we show that an effective way to increase the IPC is to increase the issue width. But this must be done without impacting the clock cycle. We propose to combine two known techniques: clustering and register write specialization. The objective of past work on clustered microarchitecture was to allow a higher clock frequency while minimizing the IPC loss. This led researchers to consider narrow-issue clusters. Our objective, instead, is to increase the IPC without impacting the clock cycle, which means wide-issue clusters. We show that, on a wide-issue dual cluster, a very simple steering policy that sends 64 consecutive instructions to the same cluster, the next 64 instructions to the other cluster, and so on, permits tolerating an inter-cluster delay of several cycles. We also propose a method for decreasing the energy cost of sending results of one cluster to the other cluster.

This study published in ACM TACO in 2015 [8] and was presented at the HIPEAC 2016 conference.

7.2.1.8. Hardware data prefetching

Participant: Pierre Michaud.

Hardware prefetching is an important feature of modern high-performance processors. When an application's working set is too large to fit in on-chip caches, disabling hardware prefetchers may result in severe performance reduction. We propose a new hardware data prefetcher, the Best-Offset (BO) prefetcher. The BO prefetcher is an offset prefetcher using a new method for selecting the best prefetch offset taking into account prefetch timeliness. The hardware required for implementing the BO prefetcher is very simple. A version of the BO prefetcher won the 2015 Data Prefetching Championship. A comprehensive study of the BO prefetcher was presented at the HPCA 2016 conference [37].

7.2.1.9. Exploiting loops for lower energy consumption

Participants: Andrea Mondelli, Pierre Michaud, André Seznec.

Recent superscalar processors use a loop buffer to decrease the energy consumption in the front-end. The energy savings comes from the branch predictor, instruction cache and instruction decoder being idle when micro-ops are delivered to the back-end from the loop buffer. We explored the possibility to exploit loop behaviors for decreasing energy consumption further, in the back-end, without impacting performance. We proposed two independent optimizations requiring little extra hardware. The first optimization detects and removes from the execution redundant micro-ops producing the same result in every loop iteration. The second optimization focuses on loop loads and detects situations where a loop load needs accessing only the data cache, or only the store queue, not both.

7.2.2. Microarchitecture Performance Modeling

7.2.2.1. Optimal cache replacement

Participant: Pierre Michaud.

A cache replacement policy is an algorithm, implemented in hardware, selecting a block to evict to make room for an incoming block. This research topic has been revitalized recently, as level-2 and level-3 caches were integrated on chip. A cache replacement policy cannot be optimal in general unless it has the knowledge of future references. Unfortunately, practical replacement policies do not have this knowledge. Still, optimal replacement is an important benchmark for understanding replacement policies. Moreover, some new replacement policies proposed recently are directly inspired from algorithms for determining hits and misses under optimal replacement. Hence it is important to improve our understanding of optimal replacement.

The OPT policy, which evicts the block referenced furthest in the future, was proved optimal by Mattson et al. [57]. However, their proof is long and somewhat complicated. In collaboration with some researchers from Inha University, we found a shorter and more intuitive proof of optimality for OPT [6].

An intriguing aspect of optimal replacement, seldom mentioned in the literature, is the fact that Belady's MIN algorithm determines OPT hits and misses without the knowledge of future references [54]. Starting from this fact, we searched and found a new algorithm, different from MIN, for determining OPT hits and misses. This algorithm provides new insights about optimal replacement. We show that traces of OPT stack distances have a distinctive structure. In particular, we prove that OPT miss curves are always convex. We show that, like an LRU cache, an OPT cache cannot experience more misses as the reuse distance of references is decreased. Consequently, accessing data circularly is the worst access pattern for OPT, like it is for LRU. We discovered an equivalence between an OPT cache of associativity N with bypassing allowed and an OPT cache of associativity N+1 with bypassing disabled. A paper deriving these results was accepted in ACM TACO and will be presented at the HiPEAC 2017 conference [25].

7.2.2.2. Adaptive Intelligent Memory Systems

Participants: André Seznec, Aswinkumar Sridharan.

Multi-core processors employ shared Last Level Caches (LLC). This trend will continue in the future with large multi-core processors (16 cores and beyond) as well. At the same time, the associativity of this LLC tends to remain in the order of sixteen. Consequently, with large multicore processors, the number of cores that share the LLC becomes larger than the associativity of the cache itself. LLC management policies have been extensively studied for small scale multi-cores (4 to 8 cores) and associativity degree in the 16 range. However, the impact of LLC management on large multi-cores is essentially unknown, in particular when the associativity degree is smaller than the number of cores.

In [48], we introduce Adaptive Discrete and deprioritized Application PrioriTization (ADAPT), an LLC management policy addressing the large multi-cores where the LLC associativity degree is smaller than the number of cores. ADAPT builds on the use of the Footprint-number metric. Footprint-number is defined as the number of unique accesses (block addresses) that an application generates to a cache set in an interval of time. We propose a monitoring mechanism that dynamically samples cache sets to estimate the Footprint-number of applications and classifies them into discrete (distinct and more than two) priority buckets. The cache replacement policy leverages this classification and assigns priorities to cache lines of applications during cache replacement operations. Footprint-number is computed periodically to account the dynamic changes in applications behavior. We further find that de- prioritizing certain applications during cache replacement is beneficial to the overall performance. We evaluate our proposal on 16, 20 and 24-core multi-programmed workloads and discuss other aspects in detail.

[48] got the best paper award at the IPDPS 2016 conference.

7.2.2.3. Augmenting superscalar architecture for efficient many-thread parallel execution

Participants: Sylvain Collange, André Seznec, Sajith Kalathingal.

Threads of Single-Program Multiple-Data (SPMD) applications often exhibit very similar control flows, i.e. they execute the same instructions on different data. In [36] we propose the Dynamic Inter-Thread Vectorization Architecture (DITVA) to leverage this implicit data-level parallelism in SPMD applications by assembling dynamic vector instructions at runtime. DITVA extends an in-order SMT processor with SIMD units with an inter-thread vectorization execution mode. In this mode, multiple scalar threads running in lockstep share a single instruction stream and their respective instruction instances are aggregated into SIMD instructions. To balance thread-and data-level parallelism, threads are statically grouped into fixed-size independently scheduled warps. DITVA leverages existing SIMD units and maintains binary compatibility with existing CPU architectures. Our evaluation on the SPMD applications from the PARSEC and Rodinia OpenMP benchmarks shows that a 4-warp $\times$ 4-lane 4-issue DITVA architecture with a realistic bank-interleaved cache achieves 1.55$\times$ higher performance than a 4-thread 4-issue SMT architecture with AVX instructions while fetching and issuing 51 % fewer instructions, achieving an overall 24 % energy reduction.

Our paper [36] received the Best Paper Award of the SBAC-PAD conference.

7.2.2.4. Generalizing the SIMT execution model to general-purpose instruction sets

Participant: Sylvain Collange.

The *Single Instruction, Multiple Threads* (SIMT) execution model as implemented in NVIDIA Graphics Processing Units (GPUs) associates a multi-thread programming model with an SIMD execution model [59]. It combines the simplicity of scalar code from the programmer's and compiler's perspective with the efficiency of SIMD execution units at the hardware level. However, current SIMT architectures demand specific instruction sets. In particular, they need specific branch instructions to manage thread divergence and convergence. Thus, SIMT GPUs have remained incompatible with traditional general-purpose CPU instruction sets.

We designed Simty, an SIMT processor proof of concept that lifts the instruction set incompatibility between CPUs and GPUs [50]. Simty is a massively multi-threaded processor core that dynamically assembles SIMD instructions from scalar multi-thread code. It runs the RISC-V (RV32-I) instruction set. Unlike existing SIMD or SIMT processors like GPUs, Simty takes binaries compiled for general-purpose processors without any instruction set extension or compiler changes. Simty is described in synthesizable RTL. A FPGA prototype validates its scaling up to 2048 threads per core with 32-wide SIMD units.

7.3. WCET estimation and optimization

Participants: Isabelle Puaut, Damien Hardy, Viet Anh Nguyen, Benjamin Rouxel, Sébastien Martinez, Erven Rohou.

7.3.1. WCET estimation for many core processors

Participants: Viet Anh Nguyen, Damien Hardy, Sébastien Martinez, Isabelle Puaut, Benjamin Rouxel.

7.3.1.1. Optimization of WCETs by considering the effects of local caches

The overall goal of this research is to defined WCET estimation methods for parallel applications running on many-core architectures, such as the Kalray MPPA machine.

Some approaches to reach this goal have been proposed, but they assume the mapping of parallel applications on cores already done. Unfortunately, on architectures with caches, task mapping requires a priori known WCETs for tasks, which in turn requires knowing task mapping (i.e., co-located tasks, co-running tasks) to have tight WCET bounds. Therefore, scheduling parallel applications and estimating their WCET introduce a chicken and egg situation.

We address this issue by developing both optimal and heuristic techniques for solving the scheduling problem, whose objective is to minimize the WCET of a parallel application. Our proposed static partitioned non-preemptive mapping strategies address the effect of local caches to tighten the estimated WCET of the parallel application. Experimental results obtained on real and synthetic parallel applications show that co-locating tasks that reuse code and data improves the WCET.

This research is part of the PIA Capacités project.

7.3.1.2. Accounting for shared resource contentions to minimize WCETs

Accurate WCET analysis for multi-cores is known to be challenging, because of concurrent accesses to shared resources, such as communication through busses or Networks on Chips (NoC). Since it is impossible in general to guarantee the absence of resource conflicts during execution, current WCET techniques either produce pessimistic WCET estimates or constrain the execution to enforce the absence of conflicts, at the price of a significant hardware under-utilization. In addition, the large majority of existing works consider that the platform workload consists of independent tasks. As parallel programming is the most promising solution to improve performance, we envision that within only a few years from now, real-time workloads will evolve toward parallel programs. The WCET behavior of such programs is challenging to analyze because they consist of *dependent* tasks interacting through complex synchronization/communication mechanisms.

In this work, we propose techniques that account for interferences to access shared ressources, in order to minimize the WCET of parallel applications. An optimal and a heuristic method are proposed to map and schedule tasks on multi-cores. These methods take the structure of applications (synchronizations/communications) into consideration to tightly identify shared resource interferences and consequently tighten WCET estimates.

This work is performed in cooperation with Steven Derrien, Angeliki Kritikakou and Imen Fassi from the CAIRN research group and is part of the ARGO H2020 project.

7.3.2. Cache-Persistence-Aware Response-Time Analysis for Fixed-Priority Preemptive Systems

Participants: Damien Hardy, Isabelle Puaut.

A task can be preempted by several jobs of higher priority tasks during its execution. Assuming the worst-case memory demand for each of these jobs leads to pessimistic worst-case response time (WCRT) estimations. Indeed, there is a big chance that a large portion of the instructions and data associated with the preempting task τ_j are still available in the cache when τ_j releases its next jobs. Accounting for this observation allows the pessimism of WCRT analysis to be significantly reduced, which is not considered by existing work.

The four main contributions of this work are: 1) The concept of persistent cache blocks is introduced in the context of WCRT analysis, which allows re-use of cache blocks to be captured, 2) A cache-persistence-aware WCRT analysis for fixed-priority preemptive systems exploiting the PCBs to reduce the WCRT bound, 3) A multi-set extension of the analysis that further improves the WCRT bound and 4) An evaluation showing that our cache-persistence-aware WCRT analysis results in up to 10 % higher schedulability than state-of-the-art approaches.

This work [43] appeared at ECRTS 2016 and was selected as an outstanding paper in this conference.

This work was performed in cooperation with Syed Aftab Rashid, Geoffrey Nelissen, Benny Akesson and Eduardo Tovar from ISEP (Polytechnic Institute of Porto), Portugal.

7.4. Fault Tolerance

7.4.1. WCET estimation for architectures with faulty caches

Participants: Damien Hardy, Isabelle Puaut.

Fine-grained disabling and reconfiguration of hardware elements (functional units, cache blocks) will become economically necessary to recover from permanent failures, whose rate is expected to increase dramatically in the near future. This fine-grained disabling will lead to degraded performance as compared to a fault-free execution.

Until recently, all static worst-case execution time (WCET) estimation methods were assuming fault-free processors, resulting in unsafe estimates in the presence of faults. The first static WCET estimation technique dealing with the presence of permanent faults in instruction caches was proposed in [4]. This study probabilistically quantified the impact of permanent faults on WCET estimates. It demonstrated that the probabilistic WCET (pWCET) estimates of tasks increase rapidly with the probability of faults as compared to fault-free WCET estimates.

New results show that very simple reliability mechanisms allow mitigating the impact of faulty cache blocks on pWCETs. Two mechanisms, that make part of the cache resilient to faults are analyzed. Experiments show that the gain in pWCET for these two mechanisms are on average 48 % and 40 % as compared to an architecture with no reliability mechanism.

This work [35] appeared at DATE 2016 (best paper award for the embedded systems track).

This is joint work with Yannakis Sazeides from University of Cyprus.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

8.1.1. Nano 2017 PSAIC

Participants: Arif Ali Ana-Pparakkal, Erven Rohou, Emmanuel Riou.

Nano 2017 PSAIC is a collaborative R&D program involving Inria and STMicroelectronics. The PSAIC (Performance and Size Auto-tuning through Iterative Compilation) project concerns the automation of program optimization through the combination of several tools and techniques such as: compiler optimization, profiling, trace analysis, iterative optimization and binary analysis/rewriting. For any given application, the objective is to devise through a fully automated process a compiler profile optimized for performance and code size. For this purpose, we are developing instrumentation techniques that can be focused and specialized to a specific part of the application aimed to be monitored.

The project involves the Inria teams PACAP, AriC, CAMUS and CORSE. PACAP contributes program analyses at the binary level, as well as binary transformations. We will also study the synergy between static (compiler-level) and dynamic (run-time) analyses.

8.2. Bilateral Grants with Industry

8.2.1. Intel research grant INTEL2014-8957

Participants: André Seznec, Biswabandan Panda, Arthur Perais, Fernando Endo.

Intel is supporting the research of the PACAP project-team on “Mixing branch and value prediction to enable high sequential performance”.

8.2.2. Intel research grant INTEL2016-11174

Participants: André Seznec, Pierre Michaud, Kleovoulos Kalaitzidis.

Intel is supporting the research of the PACAP project-team on “Design tradeoffs for extreme cores”.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. Capacités: Projet “Investissement d’Avenir”, 1/11/14 to 31/01/2018

Participants: Damien Hardy, Isabelle Puaut, Viet Anh Nguyen, Sébastien Martinez.

The project objective is to develop a hardware and software platform based on manycore architectures, and to demonstrate the relevance of these manycore architectures (and more specifically the Kalray manycore) for several industrial applications. The Kalray MPPA manycore architecture is currently the only one able to meet the needs of embedded systems simultaneously requiring high performance, lower power consumption, and the ability to meet the requirements of critical systems (low latency I/O, deterministic processing times, and dependability). The project partners are Kalray (lead), Airbus, Open-Wide, Safran Sagem, IS2T, Real Time at Work, Dassault Aviation, Eurocopter, MBDA, ProbaYes, IRIT, Onera, Verimag, Inria, Irisa, Tima and Armines.

9.1.2. Multicore: Inria Project Lab, 2013-2016

Participants: Erven Rohou, Nabil Hallou.

Multicore is an Inria Project Lab (IPL, formerly *Action d'Envergure*) started in 2013. It is entitled "Large scale multicore virtualization for performance scaling and portability". Partner project-teams include: PACAP, ALGORILLE, CAMUS, REGAL, RUNTIME, as well as DALI. This project aims to build collaborative virtualization mechanisms that achieve essential tasks related to parallel execution and data management. We want to unify the analysis and transformation processes of programs and accompanying data into one unique virtual machine.

9.1.3. ANR Continuum 2015–2019

Participants: Erven Rohou, Rabab Bouziane.

The CONTINUUM project aims to address the energy-efficiency challenge in future computing systems by investigating a design continuum for compute nodes, which seamlessly goes from software to technology levels via hardware architecture. Power saving opportunities exist at each of these levels, but the real measurable gains will come from the synergistic focus on all these levels as considered in this project. Then, a cross-disciplinary collaboration is promoted between computer science and microelectronics, to achieve two main breakthroughs: i) combination of state-of-the-art heterogeneous adaptive embedded multicore architectures with emerging communication and memory technologies and, ii) power-aware dynamic compilation techniques that suitably match such a platform.

Continuum started on Oct 1st 2015. Partners are LIRMM and Cortus SAS.

9.1.4. ANR CHIST-ERA SECODE 2016-2018

Participants: Nicolas Kiss, Damien Hardy, Erven Rohou.

In this project, we specify and design error correction codes suitable for an efficient protection of sensitive information in the context of Internet of Things (IoT) and connected objects. Such codes mitigate passive attacks, like memory disclosure, and active attacks, like stack smashing. The innovation of this project is to leverage these codes for protecting against both cyber and physical attacks. The main advantage is a full coverage of attacks of the connected embedded systems, which is considered as a smart connected device and also a physical device. The outcome of the project is first a method to generate and execute cyber-resilient software, and second to protect data and its manipulation from physical threats like side-channel attacks. Theses results are demonstrated by using a smart sensor application with hardened embedded firmware and tamper-proof hardware platform.

Partners are Télécom Paris Tech, Université Paris 8, University of Sabancı(Turkey), and Université Catholique de Louvain (Belgium).

9.1.5. ANR W-SEPT 2012-2016

Participants: Isabelle Puaut, Erven Rohou.

Critical embedded systems are generally composed of repetitive tasks that must meet drastic timing constraints, such as termination deadlines. Providing an upper bound of the worst-case execution time (WCET) of such tasks at design time is thus necessary to prove the correctness of the system. Static WCET estimation methods, although safe, may produce largely over-estimated values. The objective of the project is to produce tighter WCET estimates by discovering and transforming flow information at all levels of the software design process, from high level-design models (e.g. Scade, Simulink) down to binary code. The ANR W-SEPT project partners are Verimag Grenoble, IRIT Toulouse, Inria Rennes. A case study is provided by Continental Toulouse.

9.1.6. PEPS INS2I gDGA

Participant: Sylvain Collange.

This interdisciplinary project aims at extending the definition and the range of applicability of distance geometry, with a particular attention to its discretization. As it is already possible to remark from recent publications in the scientific literature, the distance geometry can nowadays be seen as a classical problem in operational research, with a wide range of potential applications. Among the possible extensions, this project will mainly focus on dynamical problems, motivated by a certain number of novel applications that we have identified. These include interaction motion adaptation, the simulation of crowd behaviors, and the conception of recommender systems that are able to satisfy modern privacy regulations. The classical application of the distance geometry arising in the biological field will also be considered in this project. The necessity of a strong computational power for the mentioned applications motivates the need of implementing our algorithms in environments capable of exploiting the resources in GPU cards.

Partners are: Inria, Unviersité de Rennes 2, INSA Rennes, Université d'Avignon, CNRS.

9.2. European Initiatives

9.2.1. FP7 & H2020 Projects

9.2.1.1. ANTAREX

Participants: Erven Rohou, Imane Lasri.

Title: Auto-Tuning and Adaptivity appRoach for Energy efficient exascale HPC Systems

Programm: H2020

Duration: September 2015 - September 2018

Coordinator: Politecnico di Milano, Italy (POLIMI)

Partners:

Consorzio Interuniversitario Cineca (Italy)

Dompé Farmaceutici Spa (Italy)

Eidgenoessische Technische Hochschule Zürich (Switzerland)

Vysoka Skola Banska - Technicka Univerzita Ostrava (Czech Republic)

Politecnico di Milano (Italy)

Sygic As (Slovakia)

Universidade do Porto (Portugal)

Inria contact: Erven Rohou

Energy-efficient heterogeneous supercomputing architectures need to be coupled with a radically new software stack capable of exploiting the benefits offered by the heterogeneity at all the different levels (supercomputer, job, node) to meet the scalability and energy efficiency required by Exascale supercomputers. ANTAREX will solve these challenging problems by proposing a disruptive holistic approach spanning all the decision layers composing the supercomputer software stack and exploiting effectively the full system capabilities (including heterogeneity and energy management). The main goal of the ANTAREX project is to provide a breakthrough approach to express application self-adaptivity at design-time and to runtime manage and autotune applications for green and heterogenous High Performance Computing (HPC) systems up to the Exascale level.

9.2.1.2. Eurolab-4-HPC

Participant: André Seznec.

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

Programm: H2020

Duration: September 2015 - September 2017

Coordinator: CHALMERS TEKNISKA HOEGSKOLA AB

Partners:

Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)

Chalmers Tekniska Hoegskola (Sweden)

École Polytechnique Federale de Lausanne (Switzerland)

Foundation for Research and Technology Hellas (Greece)

Universität Stuttgart (Germany)

Rheinisch-Westfaelische Technische Hochschule Aachen (Germany)

Technion - Israel Institute of Technology (Israel)

Universitaet Augsburg (Germany)

The University of Edinburgh (United Kingdom)

Universiteit Gent (Belgium)

The University of Manchester (United Kingdom)

Inria contact: Albert Cohen (Inria Paris)

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

9.2.1.3. DAL

Participants: Pierre Michaud, Sylvain Collange, Erven Rohou, André Seznec, Arthur Perais, Sajith Kalathingal, Andrea Mondelli, Aswinkumar Sridharan.

Title: DAL: Defying Amdahl's Law

Program: FP7

Type: ERC

Duration: April 2011 - March 2016

Coordinator: Inria

Inria contact: André Seznec

Multicore processors have now become mainstream for both general-purpose and embedded computing. Instead of working on improving the architecture of the next generation multicore, with the DAL project, we deliberately anticipate the next few generations of multicores. While multicores featuring 1000's of cores might become feasible around 2020, there are strong indications that sequential programming style will continue to be dominant. Even future mainstream parallel applications will exhibit large sequential sections. Amdahl's law indicates that high performance on these sequential sections is needed to enable overall high performance on the whole application. On many (most) applications, the effective performance of future computer systems using a 1000-core processor chip will significantly depend on their performance on both sequential code sections and single thread. We envision that, around 2020, the processor chips will feature a few complex cores and many (may be 1000's) simpler, more silicon and power effective cores. In the DAL research project, we will explore the microarchitecture techniques that will be needed to enable high performance on such heterogeneous processor chips. Very high performance will be required on both sequential sections -legacy sequential codes, sequential sections of parallel applications- and critical threads on parallel applications -e.g. the main thread controlling the application. Our research will focus on enhancing single process performance. On the microarchitecture side, we will explore both a radically new approach, the sequential accelerator, and more conventional processor architectures. We will also study how to exploit heterogeneous multicore architectures to enhance sequential thread performance.

9.2.1.4. ARGO

Participants: Isabelle Puaut, Damien Hardy.

Title: Argo: WCET-Aware Parallelization of Model-Based Applications for Heterogeneous Parallel Systems
Program: H2020
Type: RIA
Duration: Jan 2016 - Dec 2018
Coordinator: Karlsruher Institut fuer Technologie (KIT)
Université Rennes I contact: Steven Derrien
Partners:

Karlsruher Institut fuer Technologie (KIT)
SCILAB enterprises SAS
Recore Systems BV
Université de Rennes 1
Technologiko Ekpaideftiko Idryma (TEI) Dytikis Elladas
Absint GmbH
Deutsches Zentrum fuer Luft - und Raumfahrt EV
Fraunhofer

Increasing performance and reducing costs, while maintaining safety levels and programmability are the key demands for embedded and cyber-physical systems in European domains, e.g. aerospace, automation, and automotive. For many applications, the necessary performance with low energy consumption can only be provided by customized computing platforms based on heterogeneous many-core architectures. However, their parallel programming with time-critical embedded applications suffers from a complex toolchain and programming process. Argo (WCET-Aware PaRallelization of Model-Based Applications for HeteroGeneOus Parallel Systems) will address this challenge with a holistic approach for programming heterogeneous multi- and many-core architectures using automatic parallelization of model-based real-time applications. Argo will enhance WCET-aware automatic parallelization by a crosslayer programming approach combining automatic tool-based and user-guided parallelization to reduce the need for expertise in programming parallel heterogeneous architectures. The Argo approach will be assessed and demonstrated by prototyping comprehensive time-critical applications from both aerospace and industrial automation domains on customized heterogeneous many-core platforms.

Argo also involves Steven Derrien, Angeliki Kritikakou, and Imen Fassi from the CAIRN team.

9.2.2. Collaborations in European Programs, Except FP7 & H2020

9.2.2.1. COST Action TACLe - Timing Analysis on Code-Level 10-2012/09-2016

Participants: Damien Hardy, Isabelle Puaut, Benjamin Rouxel.

Embedded systems increasingly permeate our daily lives. Many of those systems are business- or safety-critical, with strict timing requirements. Code-level timing analysis (used to analyze software running on some given hardware w.r.t. its timing properties) is an indispensable technique for ascertaining whether or not these requirements are met. However, recent developments in hardware, especially multi-core processors, and in software organization render analysis increasingly more difficult, thus challenging the evolution of timing analysis techniques.

New principles for building "timing-composable" embedded systems are needed in order to make timing analysis tractable in the future. This requires improved contacts within the timing analysis community, as well as with related communities dealing with other forms of analysis such as model-checking and type-inference, and with computer architectures and compilers. The goal of this COST Action is to gather these forces in order to develop industrial-strength code-level timing analysis techniques for future-generation embedded systems, through several working groups:

- WG1 Timing models for multi-cores and timing composability
- WG2 Tooling aspects
- WG3 Early-stage timing analysis
- WG4 Resources other than time

Isabelle Puaut is in the management committee of the COST Action TACLe - Timing Analysis on Code-Level (http://www.tacle.eu). She is responsible of Short Term Scientific Missions (STSM) within TACLe.

9.2.3. Collaborations with Major European Organizations

9.2.3.1. HiPEAC4 NoE

Participants: Pierre Michaud, Erven Rohou, André Seznec.

P. Michaud, A. Seznec and E. Rohou are members of the European Network of Excellence HiPEAC4.

HiPEAC4 addresses the design and implementation of high-performance commodity computing devices in the 10+ year horizon, covering both the processor design, the optimizing compiler infrastructure, and the evaluation of upcoming applications made possible by the increased computing power of future devices.

9.3. International Initiatives

9.3.1. PHC IMHOTEP

Participant: Erven Rohou.

Title: Thoth – An Automatic Dynamic Binary Parallelisation System

International Partner (Institution - Laboratory - Researcher):

Egypt-Japan University of Science and Technology - Prof. Ahmed ElMahdy.

Dates: 2016–2017

With the current global trend towards utilizing cloud computing and smart devices, executing the same application across becomes a necessity. Moreover, parallelism is now abundant with various forms that include thread- and data-parallel execution models. Such diversity in ISA and explicit parallelism makes software development cost prohibitive, especially for natively optimized binaries. This project leverages dynamic binary translation technology to provide for exploiting the underlying parallel resources without the need of having the source code of the application. In particular the project integrates low overhead dynamic profiling, novel OSR parallel de-optimization and a retargetable parallelization modules to allow for dynamic parallelization of binaries.

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

9.3.2.1. PROSPIEL

Participant: Sylvain Collange.

Title: Profiling and specialization for locality

International Partner (Institution - Laboratory - Researcher):

Universidade Federal de Minas Gerais (Brazil) - DCC - Fernando Magno Quintão Pereira

Start year: 2015

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

9.3.3. Inria International Partners

9.3.3.1. Informal International Partners

The PACAP project-team has informal collaborations (visits, common publications) with University of Wisconsin at Madison (Pr Wood), University of Toronto (Pr Moshovos), University of Ghent (Dr Eyerman), University of Uppsala (Pr Hagersten), University of Cyprus (Pr Sazeides), the Egyptian-Japanese University of Science and Technology (Pr Ahmed El-Mahdy), Intel Haifa (Dr Zaks, Eng Nuzman), Barcelona Supercomputing Center (Dr Cazorla, Dr Abella), ISEP Porto (Dr Nelissen, Dr Nélis).

9.4. International Research Visitors

9.4.1. Visits of International Scientists

9.4.1.1. Internships

Rubens Emilio Alves Moreira, student at Universidade Federal de Minas Gerais, visited from Feb 2016 to May 2016 within the context of the PROSPIEL associated team.

Stefano Cherubin, PhD student at Politecnico di Milano for one month in Oct 2016, within the context of the ANTAREX H2020 project.

Anita Tino, PhD student at Ryerson University, visited from Oct 2016 within the context of a MITACS grant.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. Member of the Organizing Committees

A. Seznec is member of the ACM/IEEE PACT conference steering committee.

A. Seznec is member of the ACM/IEEE ISCA symposium steering committee.

10.1.2. Scientific Events Selection

10.1.2.1. Chair of Conference Program Committees

Isabelle Puaut is Program Chair of the 2017 IEEE Real-Time Systems Symposium (RTSS).

A. Seznec was PC chair of the 2016 ACM/IEEE ISCA symposium.

10.1.2.2. Member of the Conference Program Committees

Isabelle Puaut is member of the program committees of the Euromicro Conference on Real Time Systems (ECRTS) 2016 and 2017, the IEEE Real-Time Systems Symposium (RTSS) 2016, the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2017 and the WCET workshop 2016.

A. Seznec is a member of IEEE Micro 2017 Top Picks selection committee.

A. Seznec was a member of the SAMOS 2016 conference program committee.

Damien Hardy was a member of RTNS 2016 and WCET 2016 program committees.

Pierre Michaud was a member of the program committees of the HPCA 2017 conference and of the 5th JILP Workshop on Computer Architecture Competitions (JWAC-5).

Sylvain Collange was PC member of ISCA 2016 and of Compas'2016.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

Isabelle Puaut is Associate Editor for IEEE Transactions on Computers (IEEE TC).

A. Seznec is a member of the editorial boards of IEEE Micro and ACM Transactions on Architecture and Compiler Optimization.

10.1.4. Invited Talks

Damien Hardy was invited to give a tutorial on Heptane at Tutor16 (1st Tutorial on Tools for Real-Time Systems) in conjunction with the Cyber-Physical Systems week 2016.

Damien Hardy was invited from August 31st to September 2nd at the Barcelona Supercomputing Center group. He presented an invited talk.

A. Seznec presented invited talks at Intel Bangalore (compressed caches, branch prediction, value prediction) in Sept. 2016.

A . Seznec presented the PACAP work on compressed caches at the ARM research summit in Sept. 2016.

A. Seznec presented the PACAP work on compressed caches and register equality prediction at the Intel low latency ISRA workshop in Dec. 2016.

10.1.5. Scientific Expertise

Erven Rohou was an expert for the ANR review process.

10.1.6. Research Administration

Isabelle Puaut is responsible of Short Term Scientific Missions (STSM) withing the European COST action Tacle (Timing Analysis at Code LEvel) (http://www.tacle.eu).

Isabelle Puaut is member of the steering committee of RTNS (Real-Time Networks and Systems).

Isabelle Puaut is member of the steering committee of the Worst Case Execution Time (WCET) workshop, held in conjunction with the Euromicro Conference on Real Time Systems (ECRTS).

Isabelle Puaut is member of the scientific council of University of Rennes 1.

Isabelle Puaut is member of the administration council of the computer science and electrical engineering department of University of Rennes 1.

A. Seznec is an elected member of the Administration Council of Inria.

Erven Rohou is a member of the Inria CDT (Commission du Développement Technologique).

As “correspondant scientifique des relations internationales” for Inria Rennes Bretagne Atlantique, Erven Rohou is a member of the Inria COST GTRI (Groupe de Travail "Relations Internationales" du Comité d’Orientation Scientifique et Technologique).

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Master: Isabelle Puaut, Operating systems, 1st year of master, total of 110 hours

Master: Isabelle Puaut, Damien Hardy, Real-time systems, 1st year of master, total of 58 hours

Master: Isabelle Puaut, Erven Rohou, Writing of scientific publications, 2nd year of master and PhD students, total of 24 hours

Licence: Damien Hardy, Real-time systems, L3 Université de Rennes I, total of 60 hours

Master: Damien Hardy, Operating systems, M2 Université de Rennes I, total of 60 hours

Master: Damien Hardy, Operating systems, M1 Université de Rennes I, total of 60 hours

Master: S. Collange, Programmation parallèle, 22 hours, M1, Université de Rennes I, France

10.2.2. Supervision

PhD: Sajith Kalathingal, "Transforming TLP into DLP with the Dynamic Inter-Thread Vectorization Architecture", Université Rennes 1, Dec 2016, co-advisors S. Collange and A. Seznec

PhD: Aswinkumar Sridharan, "Adaptive and Intelligent Memory Systems", Université Rennes 1, Dec. 2016, advisor A. Seznec

PhD: Arjun Suresh, "Intercepting Functions for Memoization", Université Rennes 1, May 2016, co-advisors E. Rohou and A. Seznec

PhD in progress, Viet Anh Nguyen, Worst-Case Execution Time (WCET) Estimation for Many-core Architectures, started in january 2015. Supervised by Isabelle Puaut and Damien Hardy.

PhD in progress, Benjamin Rouxel, Code optimizations for WCET calculation on many-core platforms, started in october 2015. Supervised by Isabelle Puaut and Steven Derrien from the CAIRN group.

PhD in progress: Nabil Hallou, Université Rennes 1, Feb 2013, co-advisors E. Rohou and P. Clauss (EPI Camus Inria Strasbourg)

PhD in progress: Andrea Mondelli, Université Rennes 1, Oct 2013, co-advisors P. Michaud and A. Seznec

PhD in progress: Rabab Bouziane, Université Rennes 1, Nov 2015, advisor E. Rohou and Abdoulaye Gamatié (LIRMM, Montpellier)

PhD in progress: Arif Ali Ana-Pparakkal, Université Rennes 1, Feb 2015, advisor E. Rohou

PhD in progress: Simon Rokicki, Université Rennes 1, Sep 2015, co-advisors E. Rohou and Steven Derrien (CAIRN)

PhD in progress: Kleovoulos, Kalitzidis, "Ultrawide Issue Superscalar Processors", Université Rennes 1, Dec. 2016, advisor A. Seznec

10.2.3. Juries

Isabelle Puaut was a member of the following committees:

- PhD: Pierre Wilke, Formally Verified Compilation of Low-Level C code, Université de Rennes 1, Nov 2016
- PhD: Guillaume Phavorin, Hard Real-Time Scheduling subjected to Cache-Related Preemption Delays, Université de Poitiers, Sep 2016 (rapporteur)
- PhD: Vincent Mussot, Automates d’annotation de flot pour l’expression et l’intégration de propriétés dans l’analyse de WCET, Université Paul Sabatier, Toulouse, Dec 2016 (rapporteur)
- HDR: Mathieu Jan, Contributions au paradigme par cadencement temporel (TT) et à l’embarquabilité des systèmes temps réel, Université Paris Sud, Dec 2016 (rapporteur)

Erven Rohou was a member of the following committees:

- PhD: Juan Manuel Martinez Caamano, Strasbourg
- PhD: Lénaïc Bagnères, Orsay
- PhD: Michele Scandale, Politecnico di Milano, Milan, Italy
- PhD: Amir Ashouri, Politecnico di Milano, Milan, Italy
- PhD: Sébastien Martinez, Télécom Bretagne, Brest
- PhD: Reem ElKhouly, Egypt-Japan University of Science and Technology, Alexandrie, Egypt

10.2.3.1. Assistant professor hiring committees

Isabelle Puaut: University of Toulouse (computer architecture and real-time systems)

Isabelle Puaut: University of Chalmers, Sweden (real-time systems)

10.2.3.2. Professor hiring committee:

Isabelle Puaut: UBO (Université de Bretagne Occidentale) - real-time systems

10.3. Popularization

Erven Rohou discussed the research axes of the team in the “émergences” newsletter http://emergences.inria.fr/2016/newsletter_n43/L42-PACAP.

Nicolas Kiss, Damien Hardy and Erven Rohou presented a poster at the “Rencontres inter-UMRs-DGA”, of the “Pôle d’excellence Cyber”.

Erven Rohou and Isabelle Puaut presented a poster (with ANR W-SEPT colleagues) at “Les rencontres du numérique de l’ANR”.

11. Bibliography

Major publications by the team in recent years

[1] F. BODIN, T. KISUKI, P. M. W. KNIJNENBURG, M. F. P. O’BOYLE, E. ROHOU. *Iterative Compilation in a Non-Linear Optimisation Space*, in "Workshop on Profile and Feedback-Directed Compilation (FDO-1), in conjunction with PACT ’98", October 1998

[2] A. COHEN, E. ROHOU. *Processor Virtualization and Split Compilation for Heterogeneous Multicore Embedded Systems*, in "DAC", June 2010, pp. 102–107

[3] N. HALLOU, E. ROHOU, P. CLAUSS, A. KETTERLIN. *Dynamic Re-Vectorization of Binary Code*, in "SAMOS", July 2015, https://hal.inria.fr/hal-01155207

[4] D. HARDY, I. PUAUT. *Static probabilistic Worst Case Execution Time Estimation for architectures with Faulty Instruction Caches*, in "21st International Conference on Real-Time Networks and Systems", Sophia Antipolis, France, October 2013 [*DOI :* 10.1145/2516821.2516842], https://hal.inria.fr/hal-00862604

[5] D. HARDY, I. SIDERIS, A. SAIDI, Y. SAZEIDES. *EETCO: A tool to estimate and explore the implications of datacenter design choices on the tco and the environmental impact*, in "Workshop on Energy-efficient Computing for a Sustainable World in conjunction with the 44th Annual IEEE/ACM International Symposium on Microarchitecture (Micro-44)", 2011

[6] M.-K. LEE, P. MICHAUD, J. S. SIM, D. NYANG. *A simple proof of optimality for the MIN cache replacement policy*, in "Information Processing Letters", September 2015, 3 p. [*DOI :* 10.1016/J.IPL.2015.09.004], https://hal.inria.fr/hal-01199424

[7] P. MICHAUD. *A Best-Offset Prefetcher* ***Champion***, in "2nd Data Prefetching Championship", Portland, OR, USA, June 2015, https://hal.inria.fr/hal-01165600

[8] P. MICHAUD, A. MONDELLI, A. SEZNEC. *Revisiting Clustered Microarchitecture for Future Superscalar Cores: A Case for Wide Issue Clusters*, in "ACM Transactions on Architecture and Code Optimization (TACO) ", August 2015, vol. 13, n^{o} 3, 22 p. [*DOI :* 10.1145/2800787], https://hal.inria.fr/hal-01193178

[9] P. MICHAUD, A. SEZNEC. *Pushing the branch predictability limits with the multi-poTAGE+SC predictor :* ***Champion in the unlimited category***, in "4th JILP Workshop on Computer Architecture Competitions (JWAC-4): Championship Branch Prediction (CBP-4)", Minneapolis, United States, June 2014, https://hal.archives-ouvertes.fr/hal-01087719

[10] A. PERAIS. *Increasing the performance of superscalar processors through value prediction*, Université Rennes 1, September 2015, https://tel.archives-ouvertes.fr/tel-01282474

[11] A. PERAIS, A. SEZNEC. *EOLE: Paving the Way for an Effective Implementation of Value Prediction*, in "International Symposium on Computer Architecture", Minneapolis, MN, United States, ACM/IEEE, June 2014, vol. 42, pp. 481 - 492 [*DOI :* 10.1109/ISCA.2014.6853205], https://hal.inria.fr/hal-01088130

[12] A. PERAIS, A. SEZNEC. *Practical data value speculation for future high-end processors*, in "International Symposium on High Performance Computer Architecture", Orlando, FL, United States, IEEE, February 2014, pp. 428 - 439 [*DOI :* 10.1109/HPCA.2014.6835952], https://hal.inria.fr/hal-01088116

[13] A. PERAIS, A. SEZNEC. *EOLE: Toward a Practical Implementation of Value Prediction*, in "IEEE Micro", June 2015, vol. 35, n^{o} 3, pp. 114 - 124 [*DOI :* 10.1109/MM.2015.45], https://hal.inria.fr/hal-01193287

[14] E. RIOU, E. ROHOU, P. CLAUSS, N. HALLOU, A. KETTERLIN. *PADRONE: a Platform for Online Profiling, Analysis, and Optimization*, in "Dynamic Compilation Everywhere", Vienna, Austria, January 2014

[15] S. SARDASHTI, A. SEZNEC, D. A. WOOD. *Skewed Compressed Caches*, in "47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014", Minneapolis, United States, December 2014, https://hal.inria.fr/hal-01088050

[16] A. SEMBRANT, T. CARLSON, E. HAGERSTEN, D. BLACK-SHAFFER, A. PERAIS, A. SEZNEC, P. MICHAUD. *Long Term Parking (LTP): Criticality-aware Resource Allocation in OOO Processors*, in "International Symposium on Microarchitecture, Micro 2015", Honolulu, United States, Proceeding of the International Symposium on Microarchitecture, Micro 2015, ACM, December 2015, https://hal.inria.fr/hal-01225019

[17] A. SEZNEC, P. MICHAUD. *A case for (partially)-tagged geometric history length predictors*, in "Journal of Instruction Level Parallelism", April 2006, http://www.jilp.org/vol8

[18] A. SEZNEC, J. SAN MIGUEL, J. ALBERICIO. *The Inner Most Loop Iteration counter: a new dimension in branch history* , in "48th International Symposium On Microarchitecture", Honolulu, United States, ACM, December 2015, 11 p. , https://hal.inria.fr/hal-01208347

[19] A. SEZNEC. *TAGE-SC-L Branch Predictors:* ***Champion in 32Kbits and 256 Kbits category***, in "JILP - Championship Branch Prediction", Minneapolis, United States, June 2014, https://hal.inria.fr/hal-01086920

[20] A. SURESH, B. NARASIMHA SWAMY, E. ROHOU, A. SEZNEC. *Intercepting Functions for Memoization: A Case Study Using Transcendental Functions*, in "ACM Transactions on Architecture and Code Optimization (TACO)", July 2015, vol. 12, n° 2, 23 p. [*DOI :* 10.1145/2751559], https://hal.inria.fr/hal-01178085

[21] D. D. C. TEIXEIRA, S. COLLANGE, F. M. Q. PEREIRA. *Fusion of calling sites*, in "International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD)", Florianópolis, Santa Catarina, Brazil, October 2015 [*DOI :* 10.1109/SBAC-PAD.2015.16], https://hal.archives-ouvertes.fr/hal-01410221

Publications of the year

Doctoral Dissertations and Habilitation Theses

[22] S. KALATHINGAL. *Transforming TLP into DLP with the Dynamic Inter-Thread Vectorization Architecture*, Université Rennes 1, December 2016, https://tel.archives-ouvertes.fr/tel-01426915

[23] A. SRIDHARAN. *Adaptive and Intelligent Memory Systems*, Inria Rennes - Bretagne Atlantique and University of Rennes 1, France, December 2016, https://hal.inria.fr/tel-01442465

[24] A. SURESH. *Intercepting Functions for Memoization*, Université de Rennes 1, May 2016, https://hal.inria.fr/tel-01410539

Articles in International Peer-Reviewed Journals

[25] P. MICHAUD. *Some mathematical facts about optimal cache replacement*, in "ACM Transactions on Architecture and Code Optimization (TACO) ", December 2016, vol. 13, n° 4 [*DOI :* 10.1145/3017992], https://hal.inria.fr/hal-01411156

[26] B. PANDA. *SPAC: A Synergistic Prefetcher Aggressiveness Controller for Multi-core Systems*, in "IEEE Transactions on Computers", 2016 [*DOI :* 10.1109/TC.2016.2547392], https://hal.inria.fr/hal-01307538

[27] A. PERAIS, A. SEZNEC. *EOLE: Combining Static and Dynamic Scheduling through Value Prediction to Reduce Complexity and Increase Performance*, in "TOCS - ACM Transactions on Computer Systems", February 2016, 34 p. , https://hal.inria.fr/hal-01259139

[28] A. PERAIS, A. SEZNEC. *Storage-Free Memory Dependency Prediction*, in "IEEE Computer Architecture Letters", November 2016, pp. 1 - 4 [*DOI :* 10.1109/LCA.2016.2628379], https://hal.inria.fr/hal-01396985

[29] S. SARDASHTI, A. SEZNEC, D. A. WOOD. *Yet Another Compressed Cache: a Low Cost Yet Effective Compressed Cache*, in "ACM Transactions on Architecture and Code Optimization", September 2016, 25 p. , https://hal.inria.fr/hal-01354248

[30] A. SEZNEC, J. SAN MIGUEL, J. ALBERICIO. *Practical Multidimensional Branch Prediction*, in "IEEE Micro", 2016 [*DOI :* 10.1109/MM.2016.33], https://hal.inria.fr/hal-01330510

Invited Conferences

[31] R. IAKYMCHUK, D. DEFOUR, S. COLLANGE, S. GRAILLAT. *Reproducible and Accurate Algorithms for Numerical Linear Algebra*, in "PP: Parallel Processing for Scientific Computing", Paris, France, SIAM, April 2016, https://hal-lirmm.ccsd.cnrs.fr/lirmm-01268048

[32] C. SILVANO, G. AGOSTA, S. CHERUBIN, D. GADIOLI, G. PALERMO, A. BARTOLINI, L. BENINI, J. MARTINOVIČ, M. PALKOVIČ, K. SLANINOVÁ, J. BISPO, J. M. P. CARDOSO, R. ABREU, P. PINTO, C. CAVAZZONI, N. SANNA, A. R. BECCARI, R. CMAR, E. ROHOU. *The ANTAREX Approach to Autotuning and Adaptivity for Energy Efficient HPC Systems*, in "ACM International Conference on Computing Frontiers 2016", Como, Italy, May 2016 [*DOI :* 10.1145/2903150.2903470], https://hal.inria.fr/hal-01341826

International Conferences with Proceedings

[33] A. BONENFANT, F. CARRIER, H. CASSÉ, P. CUENOT, D. CLARAZ, N. HALBWACHS, H. LI, C. MAIZA, M. DE MICHIEL, V. MUSSOT, C. PARENT-VIGOUROUX, I. PUAUT, P. RAYMOND, E. ROHOU, P. SOTIN. *When the worst-case execution time estimation gains from the application semantics*, in "8th European Congress on Embedded Real-Time Software and Systems", Toulouse, France, January 2016, https://hal.inria.fr/hal-01235781

[34] S. COLLANGE, M. JOLDES, J.-M. MULLER, V. POPESCU. *Parallel floating-point expansions for extended-precision GPU computations*, in "The 27th Annual IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)", London, United Kingdom, July 2016, https://hal.archives-ouvertes.fr/hal-01298206

[35] *Best Paper*
D. HARDY, I. PUAUT, Y. SAZEIDES. *Probabilistic WCET estimation in presence of hardware for mitigating the impact of permanent faults*, in "Design, Automation and Test in Europe", Dresden, Germany, March 2016, https://hal.inria.fr/hal-01259493.

[36] *Best Paper*
S. KALATHINGAL, S. COLLANGE, B. NARASIMHA SWAMY, A. SEZNEC. *Dynamic Inter-Thread Vectorization Architecture: extracting DLP from TLP*, in "International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD)", Los Angeles, United States, October 2016, https://hal.inria.fr/hal-01356202.

[37] P. MICHAUD. *Best-Offset Hardware Prefetching*, in "International Symposium on High-Performance Computer Architecture", Barcelona, Spain, March 2016 [*DOI :* 10.1109/HPCA.2016.7446087], https://hal.inria.fr/hal-01254863

[38] R. E. A. MOREIRA, S. COLLANGE, F. M. Q. PEREIRA. *Function Call Re-Vectorization*, in "ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)", Austin, Texas, United States, February 2017, https://hal.archives-ouvertes.fr/hal-01410186

[39] B. PANDA, A. SEZNEC. *Dictionary Sharing: An Efficient Cache Compression Scheme for Compressed Caches*, in "49th Annual IEEE/ACM International Symposium on Microarchitecture, 2016", Taipei, Taiwan, IEEE/ACM, October 2016, https://hal.archives-ouvertes.fr/hal-01354246

[40] A. PERAIS, F. A. ENDO, A. SEZNEC. *Register Sharing for Equality Prediction*, in "International Symposium on Microarchitecture", Taipei, Taiwan, October 2016, https://hal.inria.fr/hal-01354267

[41] A. PERAIS, A. SEZNEC. *Cost Effective Physical Register Sharing*, in "International Symposium on High Performance Computer Architecture", Barcelona, Spain, IEEE, March 2016, vol. 22 [*DOI :* 10.1109/HPCA.2016.7446105], https://hal.inria.fr/hal-01259137

[42] P.-Y. PÉNEAU, R. BOUZIANE, A. GAMATIÉ, E. ROHOU, F. BRUGUIER, G. SASSATELLI, L. TORRES, S. SENNI. *Loop Optimization in Presence of STT-MRAM Caches: a Study of Performance-Energy Tradeoffs*, in "26th International Workshop on Power and Timing Modeling, Optimization and Simulation", Bremen, Germany, Proceedings of the 26th International Workshop on Power and Timing Modeling, Optimization and Simulation, September 2016, 8 p. , https://hal.inria.fr/hal-01347354

[43] S. A. RASHID, G. NELISSEN, D. HARDY, B. AKESSON, I. PUAUT, E. TOVAR. *Cache-Persistence-Aware Response-Time Analysis for Fixed-Priority Preemptive Systems*, in "28th Euromicro Conference on Real-Time Systems (ECRTS)", Toulouse, France, IEEE, July 2016 [*DOI :* 10.1109/ECRTS.2016.25], https://hal.inria.fr/hal-01393220

[44] S. ROKICKI, E. ROHOU, S. DERRIEN. *Hardware-Accelerated Dynamic Binary Translation*, in "IEEE/ACM Design, Automation & Test in Europe Conference & Exhibition (DATE)", Lausanne, Switzerland, March 2017, https://hal.inria.fr/hal-01423639

[45] *Best Paper*
A. SEZNEC. *Exploring branch predictability limits with the MTAGE+SC predictor* *, in "5th JILP Workshop on Computer Architecture Competitions (JWAC-5): Championship Branch Prediction (CBP-5)", Seoul, South Korea, June 2016, 4 p. , https://hal.inria.fr/hal-01354251.

[46] *Best Paper*
A. SEZNEC. *TAGE-SC-L Branch Predictors Again*, in "5th JILP Workshop on Computer Architecture Competitions (JWAC-5): Championship Branch Prediction (CBP-5)", Seoul, South Korea, June 2016, https://hal.inria.fr/hal-01354253.

[47] C. SILVANO, G. AGOSTA, A. BARTOLINI, A. R. BECCARI, L. BENINI, J. BISPO, R. CMAR, J. M. P. CARDOSO, C. CAVAZZONI, J. MARTINOVIČ, G. PALERMO, M. PALKOVIČ, P. PINTO, E. ROHOU, N. SANNA, K. SLANINOVÁ. *AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems: the ANTAREX Approach*, in "Design, Automation, and Test in Europe", Dresden, Germany, Design, Automation, and Test in Europe, March 2016, https://hal.inria.fr/hal-01235741

[48] *Best Paper*
A. SRIDHARAN, A. SEZNEC. *Discrete Cache Insertion Policies for Shared Last Level Cache Management on Large Multicores*, in "30th IEEE International Parallel & Distributed Processing Symposium", Chicago, United States, May 2016, https://hal.inria.fr/hal-01259626.

[49] A. SURESH, E. ROHOU, A. SEZNEC. *Compile-Time Function Memoization*, in "26th International Conference on Compiler Construction", Austin, United States, February 2017, https://hal.inria.fr/hal-01423811

National Conferences with Proceedings

[50] S. COLLANGE. *A Synthesizable General-Purpose SIMT Processor*, in "Conférence d'informatique en Parallélisme, Architecture et Système (Compas)", Lorient, France, July 2016, https://hal.inria.fr/hal-01345070

[51] S. ROKICKI, E. ROHOU, S. DERRIEN. *Hybrid-JIT : Compilateur JIT Matériel/Logiciel pour les Processeurs VLIW Embarqués*, in "Confe´rence d'informatique en Paralle´lisme, Architecture et Syste`me (Compas)", Lorient, France, July 2016, https://hal.archives-ouvertes.fr/hal-01345306

Research Reports

[52] S. COLLANGE. *Simty: a Synthesizable General-Purpose SIMT Processor*, Inria Rennes Bretagne Atlantique, August 2016, n° RR-8944, https://hal.inria.fr/hal-01351689

[53] S. SARDASHTI, A. SEZNEC, D. A. WOOD. *Yet Another Compressed Cache: a Low Cost Yet Effective Compressed Cache*, Inria, February 2016, n° RR-8853, 23 p. , https://hal.inria.fr/hal-01270792

References in notes

[54] L. A. BELADY. *A study of replacement algorithms for a virtual-storage computer*, in "IBM Systems Journal", 1966, vol. 5, n° 2, pp. 78-101

[55] M. HATABA, A. EL-MAHDY, E. ROHOU. *OJIT: A Novel Obfuscation Approach Using Standard Just-In-Time Compiler Transformations*, in "International Workshop on Dynamic Compilation Everywhere", January 2015

[56] R. KUMAR, D. M. TULLSEN, N. P. JOUPPI, P. RANGANATHAN. *Heterogeneous chip multiprocessors*, in "IEEE Computer", nov. 2005, vol. 38, n° 11, pp. 32–38

[57] R. L. MATTSON, J. GECSEI, D. R. SLUTZ, I. L. TRAIGER. *Evaluation techniques for storage hierarchies*, in "IBM Systems Journal", 1970, vol. 9, n° 2, pp. 78-117

[58] S. NASSIF, N. MEHTA, Y. CAO. *A resilience roadmap*, in "Design, Automation Test in Europe Conference Exhibition (DATE), 2010", March 2010, pp. 1011-1016

[59] J. NICKOLLS, W. J. DALLY. *The GPU computing era*, in "Micro, IEEE", 2010, vol. 30, n° 2, pp. 56–69

[60] R. OMAR, A. EL-MAHDY, E. ROHOU. *Arbitrary control-flow embedding into multiple threads for obfuscation: a preliminary complexity and performance analysis*, in "Proceedings of the 2nd international workshop on Security in cloud computing", ACM, 2014, pp. 51–58

[61] S. SARDASHTI, D. A. WOOD. *Decoupled compressed cache: exploiting spatial locality for energy-optimized compressed caching*, in "The 46th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-46, Davis, CA, USA, December 7-11, 2013", 2013, pp. 62–73, http://doi.acm.org/10.1145/2540708.2540715

[62] A. SEZNEC, N. SENDRIER. *HAVEGE: A user-level software heuristic for generating empirically strong random numbers*, in "ACM Transactions on Modeling and Computer Simulation (TOMACS)", 2003, vol. 13, n° 4, pp. 334–346