RESEARCH CENTRE

Rennes - Bretagne Atlantique

IN PARTNERSHIP WITH:

Université Rennes 1

2020

ACTIVITY REPORT

Project-Team

PACAP

Pushing Architecture and Compilation for Application Performance

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

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Architecture, Languages and Compilation

Contents

Project-Team PACAP

Creation of the Project-Team: 2016 July 01

Keywords

Computer sciences and digital sciences

A1.1. – Architectures

A1.1.1. – Multicore, Manycore

A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)

A1.1.3. – Memory models

A1.1.4. – High performance computing

A1.1.5. – Exascale

A1.1.9. – Fault tolerant systems

A1.1.10. – Reconfigurable architectures

A1.1.11. – Quantum architectures

A1.6. – Green Computing

A2.2. – Compilation

A2.2.1. – Static analysis

A2.2.2. – Memory models

A2.2.4. – Parallel architectures

A2.2.5. – Run-time systems

A2.2.6. – GPGPU, FPGA...

A2.2.7. – Adaptive compilation

A2.2.8. – Code generation

A2.2.9. – Security by compilation

A2.3.1. – Embedded systems

A2.3.3. – Real-time systems

A4.2. – Correcting codes

A4.4. – Security of equipment and software

A5.10.3. – Planning

A5.10.5. – Robot interaction (with the environment, humans, other robots)

A6.2.6. – Optimization

A9.2. – Machine learning

Other research topics and application domains

- B1. – Life sciences
- B2. – Health
- B3. – Environment and planet
- B4. – Energy
- B5. – Industry of the future
- B5.7. – 3D printing
- B6. – IT and telecom
- B7. – Transport and logistics
- B8. – Smart Cities and Territories
- B9. – Society and Knowledge

1 Team members, visitors, external collaborators

Research Scientists

- Erven Rohou [Team leader, Inria, Senior Researcher, HDR]
- Caroline Collange [Inria, Researcher]
- Byron Hawkins [Inria, Starting Research Position, until Nov 2020]
- Pierre Michaud [Inria, Researcher]
- André Seznec [Inria, Senior Researcher, HDR]

Faculty Members

- Damien Hardy [Univ de Rennes I, Associate Professor]
- Isabelle Puaut [Univ de Rennes I, Professor, HDR]

Post-Doctoral Fellow

- Pierre-Yves Péneau [Inria, until Mar 2020]

PhD Students

- Abderaouf Amalou [Univ de Rennes I, from Oct 2020]
- Nicolas Bellec [Univ de Rennes I]
- Arthur Blanleuil [Univ de Rennes I]
- Niloofar Charmchi [Inria, until Mar 2020]
- Kleovoulos Kalaitzidis [Inria, until Feb 2020]
- Kévin Le Bon [Inria]
- Anis Peysieux [Inria]
- Daniel Rodrigues Carvalho [Inria]
- Bahram Yarahmadi [Inria]

Technical Staff

- Loïc Besnard [CNRS, Engineer]
- Alexandre Kouyoumdjian [Inria, Engineer, until Feb 2020]
- Pierre-Yves Péneau [Inria, Engineer, from Apr 2020]

Interns and Apprentices

- Romain Belafia [Univ de Rennes I, from Jun 2020 until Sep 2020]
- Dorian Bescon [Univ de Rennes I, from May 2020 until Aug 2020]
- Antoine Geimer [Univ de Rennes I, from May 2020 / until July 2020]
- Sarah Piet [Univ de Rennes I, from May 2020 until Jul 2020]
- Hugo Raoelina [Inria, from May 2020 until Aug 2020]
- Hugo Texier [École centrale de Lyon, from Jun 2020 until Jul 2020]

Administrative Assistant

- Virginie Desroches [Inria]

2 Overall objectives

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 concerned with how many units of computation 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 time (WCET), which is 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 micro-architectures. And since increased performance often comes at the expense of another metric, identifying the related trade-offs is of interest to PACAP.

The previous decade witnessed the end of the "magically" increasing clock frequency and the introduction of commodity multicore processors. PACAP is experiencing 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.

Approach We strive to address performance in a way that is as transparent as possible to the 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 of the PACAP project-team derives 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 in key technologies (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.

Latency-oriented Computing Improving the ACET of general purpose systems has been the "core business" of PACAP's ancestors (CAPS and ALF) for two decades. We plan to pursue this line of research, acting at all levels: compilation, dynamic optimizations, and micro-architecture.

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

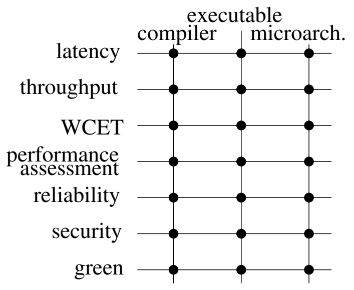

Figure 1: Perimeter of Research Objectives

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.

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. 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.

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 workloads 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 micro-architecture, 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.

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 micro-architecture and compilation techniques, we will be able to significantly improve efficiency, robustness and coverage of security mechanisms, as well as to partner with field experts to design innovative solutions.

Green Computing – Power Concerns 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 puts an end to the automatic performance improvement enjoyed by developers and users. Many research groups consequently focused on parallel architectures and compiling for parallelism.

Still, 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 members of the team), sequential tasks are still a major performance bottleneck. Addressing it is still on the agenda of the 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 thus confronted with a moving target, challenging portability and jeopardizing performance.

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 (non-Silicon, 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), quantum computing (impacts the entire software stack).

3.1.2 Evolving community

The PACAP project-team tackles performance-related issues, for conventional programming paradigms. In fact, programming complex environments is no longer the exclusive domain of 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 a 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 a *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.
PACAP works on fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impact 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 members of the project teams for two decades, with undeniable contributions. They 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. One of our goals is to devise new static compilation techniques (cf. Section 3.2.1), but also build upon iterative [1] and split [33] 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 increased 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 micro-architecture (Section 3.2.3). Reconciling CPU and GPU designs (Section 3.2.4) is one of our objectives.

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 as power efficiency (Section 3.2.6), and security (Section 3.2.7).

3.2.1 Static Compilation

Static compilation techniques continue to be relevant in addressing the characteristics of emerging hardware technologies, such as non-volatile memories, 3D-stacking, or novel communication technolo-

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

gies. 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 studies new optimization opportunities and develops tailored compilation techniques for upcoming compute nodes. New technologies may also be coupled with traditional solutions to offer new trade-offs. We 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 are 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 also leverage split-compilation [33]: 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 needs to adapt to its environment. In most cases, this environment remains 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 transparently [2]. 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 becomes 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 [40] 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 H2020 FET HPC project ANTAREX also addresses these challenges from the energy perspective. We further leverage our platform and initial results to address other adaptation opportunities. Efficient software adaptation requires expertise from all domains tackled by PACAP, and strong interaction between all team members is expected.

3.2.3 Research directions in uniprocessor micro-architecture

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, and value prediction. In particular, in recent years they have been recognized as world leaders in branch prediction [42] [38] and in cache prefetching [6] and they have revived the forgotten concept of value prediction [9][8][25]. This research was supported by the ERC Advanced grant DAL (2011-2016) and also by Intel. We 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 cores;

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 *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) [6]. The first research objective 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 research objective 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 benefits 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 [11]. It offers new possibilities 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 processors. For the two past decades, implementing ever wider issue superscalar processors has been challenging. The objective of our research on the execution core is to explore (and revisit) directions that 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 are exploring is the use of clustered architectures [7]. Symmetric clustered organization allows to benefit from a simpler bypass network, but induce large complexity on the issue queue. One remarkable finding of our study [7] 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 fewer buses) and register files (fewer ports and physical registers) in the clusters, since not all results have to be forwarded to the other cluster.

The second direction that we are exploring is associated with the approach that we developed with Sembrant et al. [41]. 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 micro-architecture with a cluster dedicated to the execution of urgent instructions and a second cluster executing the non-urgent instructions. The micro-architecture 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 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 [43].

In speculative execution, we have recently revisited value prediction (VP) which was a hot research topic between 1996 and 2002. However it was considered until 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 [9][8] [42] [38]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [9]. 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 [8]. 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 [36] 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 micro-architectures 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 investigate the convergence of CPU and GPU at both architecture and compiler levels.

Architecture.

The architecture convergence between Single Instruction Multiple Threads (SIMT) GPUs and multicore processors that we have been pursuing [35] 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.

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 analyze and/or control shared resources such as buses, NoCs or caches;

[3]http://www.kalrayinc.com
[4]https://www.hipeac.net/network/institutions/2428/recore/

2. methods to improve the programmability of real-time applications through automatic parallelization and optimizations from model-based designs.

3.2.6 Power efficiency

PACAP addresses power-efficiency at several levels. First, we 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 focus on high-performance low-power embedded compute nodes. Within the ANR project Continuum, in collaboration with architecture and technology experts from LIRMM and the SME Cortus, we research new static and dynamic compilation techniques that fully exploit emerging memory and NoC technologies. Finally, in collaboration with the CAIRN project-team, we 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 in 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 manycore systems. Our approach relies 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.

3.2.7 Security

Security is a mandatory concern of any modern computing system. Various threat models have led to a multitude of protection solutions. Members of PACAP already contributed in the past, thanks to the HAVEGE [44] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [34], or thread-based control flow mangling [39]). Still, security is not core competence of PACAP members.

Our strategy consists in partnering 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 expertise in compilation and architecture helps design more efficient and less expensive protection mechanisms.

Examples of collaborations so far include the following:

Compilation: We partnered with experts in security and codes to prototype a platform that demonstrates resilient software. They designed and proposed advanced masking techniques to hide sensitive data in application memory. PACAP's expertise is 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.

Dynamic Binary Rewriting: Our expertise in dynamic binary rewriting combines well with the expertise of the CIDRE team in protecting application. Security has a high cost in terms of performance, and static insertion of counter measures cannot take into account the current threat level. In collaboration with CIDRE, we propose an adaptive insertion/removal of countermeasures in a running application based of dynamic assessment of the threat level.

WCET Analysis: Designing real-time systems requires computing an upper bound of the worst-case execution time. Knowledge of this timing information opens an opportunity to detect attacks on the control flow of programs. In collaboration with CIDRE, we are developing a technique to detect such attacks thanks to a hardware monitor that makes sure that statically computed time information is preserved (CAIRN is also involved in the definition of the hardware component).

4 Application domains

4.1 Domains

The PACAP team is working on fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impact 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 Awards

André Seznec received the 2019 Intel outstanding researcher award for his work on “Design tradeoffs for extreme cores”.

André Seznec received the 2020 IEEE B. Ramakrishna Rau award[5] for “pioneering contributions to cache design and branch prediction”.

André Seznec received the second prize at the first Instruction Prefetching Championship [24].

5.2 Startup

After 3.5 years in PACAP (first as a postdoc, then on a *Starting Research Position*) working on dynamic binary rewriting, Byron Hawkins decided to create a startup. He was accepted to Inria Startup Studio, where he has been developing his business since Jan 2021.

Efficient techniques for trace recording, analysis and replay can improve accuracy and reduce computational effort across a diverse spectrum of realtime and offline evaluation tasks. The potential of trace-based program introspection remains largely untapped because it involves generating, storing and analyzing trace files that quickly grow to unwieldy proportions, introducing prohibitive demands on system resources and overwhelming analysis algorithms. In the entrepreneurial venture “the Padrone Platform”, both challenges are addressed simultaneously by new algorithms for compressing the trace

[5] https://www.computer.org/volunteering/awards/rau

into a format that permits typical analysis to be conducted without decompression. Performance and effectiveness of data flow sampling and analysis can also be improved on the basis of the same algorithms.

Products under development include (1) a flexible trace record/replay framework for software and hardware performance analysis, (2) scalability analysis for high-throughput applications, and (3) detailed and complete execution trail audit, to improve security on systems ranging from application servers to IoT controllers that may be exposed to adversarial influence. The Padrone Platform is supported by Inria Startup Studio and its algorithms were developed by the Inria PACAP team.

6 New software and platforms

6.1 New software

6.1.1 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.

ATMI (Analytical model of Temperature in MIcroprocessors) is an ad hoc temperature model 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.

URL: https://team.inria.fr/pacap/software/atmi/

Author: Pierre Michaud

Contact: Pierre Michaud

Participant: Pierre Michaud

6.1.2 HEPTANE

Keywords: IPET, WCET, Performance, Real time, Static analysis, Worst Case Execution Time

Scientific Description: WCET estimation

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.

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.

URL: https://team.inria.fr/pacap/software/heptane/

Authors: Damien Hardy, Isabelle Puaut, Benjamin Lesage, Thomas Piquet, François Joulaud

Contact: Isabelle Puaut

Participants: Benjamin Lesage, Loïc Besnard, Damien Hardy, François Joulaud, Isabelle Puaut, Thomas Piquet

Partner: Université de Rennes 1

6.1.3 tiptop

Keywords: Instructions, Cycles, Cache, CPU, Performance, HPC, Branch predictor

Scientific Description: Tiptop is a simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+) and displays them in a way simple to the Linux "top" utility. The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. No privilege is required, any user can run tiptop.

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

Current version is 2.3.1, released October 2017. Tiptop has been integrated in major Linux distributions, such as Fedora, Debian, Ubuntu, CentOS.

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 optimize the performance of their HPC applications.

URL: https://team.inria.fr/pacap/software/tiptop/

Author: Erven Rohou

Contact: Erven Rohou

Participant: Erven Rohou

6.1.4 PADRONE

Keywords: Legacy code, Optimization, Performance analysis, Dynamic Optimization

Functional Description: Padrone is a 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.

URL: https://team.inria.fr/pacap/software/padrone

Authors: Emmanuel Riou, Erven Rohou

Contact: Erven Rohou

Participants: Emmanuel Riou, Erven Rohou

6.1.5 If-memo

Keyword: Performance

Functional Description: If-memo is a linker-based technique for enabling software memorizing of any dynamically linked pure function by function interception. Typically, this framework is useful to intercept the computationally expensive pure functions - the transcendental functions from the math library. Our technique does not need the availability of source code and thus can even be applied 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 tool-chains, and can work with a variety of current architectures.

URL: https://team.inria.fr/pacap/software/if-memo/

Authors: Arjun Suresh, Erven Rohou

Contact: Erven Rohou

Participants: Arjun Suresh, Erven Rohou

6.1.6 Simty

Name: Simty

Keywords: GPU, Softcore, FPGA, SIMT, Multi-threading, RISC-V

Functional Description: 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.

URL: https://gitlab.inria.fr/collange/simty

Author: Caroline Collange

Contact: Caroline Collange

6.1.7 sigmask

Keywords: Compilation, Side-channel, Masking, Security, Embedded systems

Functional Description: Sigmask is a compiler plugin based on the LLVM infrastructure that automatically protects secret information in programs, such as encryption keys, against side-channel attacks. The programmer annotates their source code to highlight variables containing sensitive data. The compiler automatically analyzes the program and computes all memory locations potentially derived from the secret. It then applies a masking scheme to avoid information leakage. Sigmask provides several schemes: ODSM (Orthogonal Direct Sum Masking), IP (Inner Product) Masking, and simple random bit masking. The programmer may also provide their own masking scheme through a well-defined API.

Authors: Nicolas Kiss, Damien Hardy, Erven Rohou

Contact: Erven Rohou

Participants: Nicolas Kiss, Damien Hardy, Erven Rohou

6.1.8 GATO3D

Keywords: Code optimisation, 3D printing

Functional Description: GATO3D stands for "G-code Analysis Transformation and Optimization". It is a library that provides an abstraction of the G-code, the language interpreted by 3D printers, as well as an API to manipulate it easily. First, GATO3D reads a file in G-code format and builds its representation in memory. This representation can be transcribed into a G-code file at the end of the manipulation. The software also contains client codes for the computation of G-code properties, the optimization of displacements, and a graphical rendering.

Authors: Damien Hardy, Erven Rohou

Contacts: Erven Rohou, Damien Hardy

6.1.9 OPTILARA

Keywords: Optimisation, Pure function, Loop splitting, Memoization

Functional Description: Set of program transformations, developed in LARA (aspect language) and integrated into the CLAVA development environment.

Author: Loïc Besnard

Contact: Loïc Besnard

Partner: Université de Porto

7 New results

7.1 Compilation and Optimization

Participants Damien Hardy, Byron Hawkins, Erven Rohou, Bahram Yarahmadi.

7.1.1 Optimization in the Presence of NVRAM

Participants Erven Rohou, Bahram Yarahmadi.

A large and increasing number of Internet-of-Things devices are not equipped with batteries and harvest energy from their environment. Many of them cannot be physically accessed once they are deployed (embedded in civil engineering structures, sent in the atmosphere or deep in the oceans). When they run out of energy, they stop executing and wait until the energy level reaches a threshold. Programming such devices is challenging in terms of ensuring memory consistency and guaranteeing forward progress.

Checkpoint Placement based Worst-Case Energy Consumption Previous work has proposed to insert checkpoints in the program so that execution can resume from well-defined locations. We propose to define these checkpoint locations based on worst-case energy consumption of code sections, with limited additional effort for programmers. As our method is based upon worst-case energy consumption, we can guarantee memory consistency and forward progress.

This work has been presented at the SAMOS 2020 conference [27].

Dynamic Adaptive Checkpoint Placement Previous work has proposed to back-up the volatile states which are necessary for resuming the program execution after power failures. They either do it at compile time by placing checkpoints into the control flow of the program or at runtime by leveraging voltage monitoring facilities and interrupts, so that execution can resume from well-defined locations after power failures. We propose for the first time a dynamic checkpoint placement strategy which delays checkpoint placement and specialization to the runtime and takes decisions based on the past power failures and execution paths that are taken. We evaluate our work on a TI MSP430 device, with different types of benchmarks as well as different uninterrupted intervals, and we measure the execution time.

This research is done within the context of the project IPL ZEP.

7.1.2 Dynamic Binary Analysis and Optimization

Participants Byron Hawkins, Erven Rohou.

Dynamic Abstract Interpretation With the wide diffusion of multicore systems, the need for automatic parallelization becomes more pronounced, particularly for legacy programs, where the source code is not generally available. An essential operation in any parallelization system is detecting data dependence among parallelization candidate instructions. Conducting dependence analysis at the binary-level is more challenging than at the source-level due to the much lower semantics of the binary code.

We proposed [17] using the elaborate "static" abstract interpretation analysis, for the first time, at runtime for data dependence detection. Specifically, our system interprets instructions in a hot region, while at the same time, collecting programs semantics for visited program points, thereby conducting abstract interpretation analysis dynamically. The analysis is guaranteed to be correct as long as execution does not exit the region prematurely. Moreover, successive hot region re-entries will resume previous analysis, albeit much faster in case of no major change in the program semantics. Such an approach enables a more powerful analysis than other simple dynamic analyses which would typically miss parallelization opportunities. The proposed approach also does not require any hardware support, availability of the source code, as well as any code re-compilation. To study the performance and accuracy of our approach, we have extended the Padrone dynamic code modification framework, and conduct an initial study on a set of PolyBench kernels and selected programs from SPEC CPU. Experimental results show accurate dependence detection with low overhead.

Alternatively, speculative parallelization is one of the most popular automatic parallelization techniques, based on detecting *probably*-parallelizable code parts. We proposed [18] a runtime data-dependence detection technique that is based on abstract interpretation at the intermediate representation (IR) level. Unlike most existing approaches in which data analysis occurs at compile time, our proposed method conducts the analysis while interpreting the code, focusing on potentially parallelized loops. This process is based on abstract interpretation. It first computes the abstract domain for each hot loop program points. Each abstract domain is incrementally computed, until a fixed point is achieved for all program points, and correspondingly the analysis terminates in order to consecutively detect the existence of data dependence. Once the analysis result reports a parallelization possibility for the finished hot loop, the interpreter invokes the compiler to resume the execution in a parallel fashion as recommended by our proposed approach. The proposed technique is implemented on LLVM compiler, then used to test the dependence detection for a set of kernels on the Polybench framework, and the data dependence analysis required for each kernel is studied in terms of the computation overhead.

Finally, we proposed [20] employing a simple abstract single-trace analysis method using intervals and congruent modulo domains to track dependencies at lower time and memory costs. The method gathers and abstracts the set of all memory reference addresses for each static memory access instruction. This method removes the need for keeping a large shadow memory and only requires a single pair-wise analysis pass to detect dependencies among memory instructions through simple intersection operations. Moreover, the combination of interval and congruent domains improves precision when compared with only using an interval domain representation, mainly when the data is not accessed in a dense access pattern. We further improve precision through partitioning memory space into blocks, where references in each block are abstracted independently. An initial performance study is conducted on SPEC CPU-2006 benchmark programs and polyhedral benchmark suite. Results show that the method reduces execution time overhead by 1.4x for polyhedral and 10.7x for SPEC2006 on average; and significantly reduces memory by 109780x and 6981x for polyhedral and SPEC2006 respectively; the method has an average precision of 99.05 % and 61.37 % for polyhedral and SPEC respectively. Using memory partitioning resulted in improving mean precision to be 82.25 % and decreasing memory reduction to be 47x for SPEC2006 suite.

This research was partially done in cooperation with the Egypt-Japan University of Science and Technology, within the context of the project PHC IMHOTEP.

Single Pass Trace Compression Execution tracing has been applied across a diverse range of application domains to improve visibility into the internal behavior of a program or an entire system. For specialized problems such as performance optimization and security protection and audit, detailed solutions can be derived from a simple integration of data flow sampling and analysis. Practical application of both execution and data flow tracing, however, has been limited by the prohibitive quantity of trace data, which quickly becomes difficult to store or transfer across a network, and in many important cases can easily overwhelm analysis algorithms. Techniques have been proposed to compress such traces, but so far only a subset of essential challenges are addressed. Efficient compression schemes are opaque, requiring time and space consuming decompression into the original form, which remains intractable for many important classes of analysis. While transparent compression can eliminate this costly phase, it must be preceded by analysis to discover the trace structure, which becomes difficult for programs of signficant size and complexity where trace sources require timely consumption of the trace stream. For example, SPEC CPU 2017 benchmark mct generates 40 MB of trace data per second for an Intel PT source, which discards any fragment of trace data that overflows its circular RAM buffer.

To resolve these limitations and enable complete execution tracing that scales to complex programs and long-running executions, we propose the Single Pass Trace Compression algorithm, which delegates to dedicated algorithms for incremental control and data flow analysis to perform lossless compression into a transparent format on the fly, without ever storing the original trace in memory or on disk. A diverse range of typical program analysis algorithms can be adapted to operate directly on the compressed trace, improving performance of the analysis while avoiding costly decompression. For data flow sampling and analysis, the same algorithms can reduce complexity and resource demands by integrating an incremental schema of path constraints into recorded traces. It is well known that a complete and precise set of path constraints can reduce intractable factors of data flow analysis to manageable complexity. Although complete path constraints are also intractable for general programs, by starting with an efficient but incomplete path analysis, the incremental structure of these algorithms can gradually expand towards complete path constraints by efficiently integrating new information on demand during analysis, either from additional data flow samples or from partial conclusions derived by a path constraint solver. We report experiments with complex programs such as xalancbmk from the SPEC CPU 2017 suite, which demonstrate efficient and effective trace recording for industrial scale program executions involving complex object-oriented structure and highly irregular functions such as malloc.

Elastic Spanning Tree The depth-first spanning tree has been a central component of detailed graph analysis that focuses on hierarchical dependencies across the graph as a whole, such as identification of strongly connected components or topological sorting. When graphs model external domains such as logical formulas, linear constraint systems and weighted schedules, dependency analysis leveraging the depth-first spanning tree reveals essential relations among the corresponding dependencies of the external domain. As streaming scenarios increase in importance throughout the global computing ecosystem, it has become useful to continuously maintain a depth-first spanning tree over an increasing graph, where vertices and edges are regularly added but never removed. Incremental adaptation of traditional depth-first traversal algorithms has been limited to O(mn) for an eventual graph of m edges and n vertices, because the hierarchical structure of the spanning tree is maintained as a linear order across all vertices, referred to as the "depth-first numbering", which frequently changes at a worst-case cost of $O(n)$ per update.

To improve both worst-case and average performance of incremental depth-first construction, we propose the Elastic Spanning Tree (EST) along with a hybrid spanning tree that employs EST within designated subtrees. Since depth-first numbering is not necessary for EST, the scope of traditional depth-first numbering in the containing tree is reduced. An EST subtree employs a sibling-relative depth first (SRDF) structure that performs best over sparsely connected graphs where the traditional spanning tree performs worst. Vertex addition at any position in an SRDF spanning subtree is performed in $O(1)$, and in most configurations, any new edge within an SRDF can be integrated in $O(\log s)$ for an SRDF subtree of s vertices. At any point during graph construction, an SRDF subtree can be decomposed into an ordinary subtree of the traditional (containing) spanning tree, and an SRDF subtree can likewise be instantiated on demand in place of a traditional subtree, together allowing for SRDF to be employed for optimal advantage at each step of incremental graph construction. Although the worst-case performance of EST remains $O(mn)$, this scenario can be proven limited to a small number of special case input sequences that are not relevant in practice, such as continuously prepending a single path having no branches. Light constraints on the input permit worst-case $O(m\sqrt{n})$ performance for an eventual graph of size (m,n), and we show these constraints to be compatible with the natural structure of graphs occurring in important problem domains.

7.1.3 Optimization of G-code for faster 3D printing

Participants Damien Hardy, Erven Rohou.

From May to July 2020, as part of the fight against Covid-19, we studied the possibility of increasing the production capacity of professional players using fused deposition modeling 3D printing, without requiring any modification of existing production infrastructures. Our feasibility study [30] presents several optimization opportunities to achieve this objective. A first optimization is implemented to reduce airtime movements. Experiments in real conditions show a good potential for reducing printing times without affecting the quality on simple 3D models.

*This study was done with the contribution of Guillermo Andrade-Barroso, Loïc Besnard, and Laurent Garnier from SED (*Service d'Expérimentation et de Développement*).*

7.2 Processor Architecture

Participants Arthur Blanleuil, Niloofar Charmchi, Caroline Collange, Kleovoulos Kalaitzidis, Pierre Michaud, Anis Peysieux, Daniel Rodrigues Carvalho, André Seznec.

7.2.1 Value prediction

Participants Kleovoulos Kalaitzidis, André Seznec.

Value Prediction (VP) has been recently gaining interest in the research community, since prior work has established practical solutions for its implementation that provide meaningful performance gains. A constant challenge of contemporary context-based value predictors is to sufficiently capture value redundancy and exploit the predictable execution paths. To do so, modern context-based VP techniques tightly associate recurring values with instructions and contexts by building confidence upon them after a plethora of repetitions. However, when execution monotony exists in the form of intervals, the potential prediction coverage is limited, since prediction confidence is reset at the beginning of each new interval. In [15], we address this challenge by introducing the notion of Equality Prediction (EP), which represents the binary facet of VP. Following a twofold decision scheme (similar to branch prediction), at fetch time EP makes use of control-flow history to predict equality between the last committed result for this instruction and the result of the currently fetched occurrence. When equality is predicted with high confidence, the last committed value is used. Our simulation results show that this technique obtains the same level of performance as previously proposed state-of-the-art context-based value predictors. However, by virtue of exploiting equality patterns that are not captured by previous VP schemes, our design can improve the speedup of standard VP by 19 % on average, when combined with contemporary prediction models.

In [25] we explore the performance limits of value prediction for unlimited size predictors in the context of the Championship Value Prediction evaluation framework (CVP). The CVP framework assumes a processor with a large instruction window (256-entry ROB), an aggressive instruction front-end fetching 16 instructions per cycle, an unlimited number of functional units, and a large value misprediction penalty with a complete pipeline flush at commit on a value misprediction. Our proposition ES-HC-VS-VT combines four predictor components which do not use the speculative results of the inflight occurrences of the instruction to compute the prediction. The four components are respectively the E-stride predictor (ES), the HCVP, Heterogeneous-Context Value, predictor (HC), the VSEP, Value Speculative Equality Predictor (VS) and the VTAGE predictor (VT). Prediction is computed as, first VTAGE prediction, if not high confidence: VSEP prediction, if not high confidence: HCVP prediction, if not high confidence: E-stride prediction. E-Stride computes its prediction from the last committed occurrence of the instruction and the number of speculative inflight occurrences of the instruction in the pipeline. HCVP computes the predicted value through two successive contexts; first the PC and the global history are used to read a stride history, then this stride history is used to obtain a value and a stride. On VTAGE and VSEP the predicted value is the value directly read at prediction time on the predictor tables. As for EVES, for ES-HC-VS-VT, we optimize the algorithms to assign confidence to predictions on each predictor component depending on the expected benefit/loss of a prediction. On the 135 traces from CVP, ES-HC-VS-VT achieves 4.030 IPC against 3.881 IPC achieved by the previous leader of the CVP, HCVP+E-stride.

7.2.2 Compressed caches

Participants Niloofar Charmchi, Caroline Collange, Daniel Rodrigues Carvalho, André Seznec.

The speed gap between CPU and memory is impairing performance. Cache compression and hardware prefetching are two techniques that could confront this bottleneck by decreasing last level cache misses. However, compression and prefetching have positive interactions, as prefetching benefits from higher cache capacity and compression increases the effective cache size.

This study proposes *Compressed cache Layout Aware Prefetching* (CLAP) to leverage the recently proposed sector-based compressed cache layouts, such as SCC or YACC, to create a synergy between compressed caches and prefetching [28]. The idea of this approach is to prefetch contiguous blocks that can be compressed and co-allocated together with the requested block on a miss access. Prefetched blocks that share storage with existing blocks do not need to evict a valid existing entry; therefore, CLAP avoids cache pollution.

In order to decide the co-allocatable blocks to prefetch, we propose a compression predictor. Based on our experimental evaluations, CLAP reduces the number of cache misses by 9 % and improves performance by 3 % on average, comparing to a compressed cache. Furthermore, in order to get more improvements, we unify CLAP and other prefetchers and introduce two adaptive CLAPs which select the best prefetcher based on the application.

In addition, Daniel Rodrigues Carvalho made many improvements to the open-source processor simulator gem5. They have been contributed to the public repository. Cache replacement policies and new compression support have been discussed in a community article [31, §2.11].

7.2.3 Instruction prefetching

Participants Pierre Michaud, André Seznec.

Certain workloads have an instruction footprint so large that they generate long-latency instruction cache misses. Long-latency instruction cache misses hurt processor performance badly. Instruction prefetching is a possible solution to this problem. So far, instruction prefetching has received less attention from the academic research community than data prefetching, mainly because the benchmarks commonly used by researchers generate few long-latency instruction cache misses. The first Instruction Prefetching Championship workshop, organized by Intel in 2020, testifies that the industry is still looking for solutions to this problem. We have proposed two different prefetchers to the workshop.

When designing a prefetcher, the computer architect has to define which event should trigger a prefetch action and which blocks should be prefetched. We propose to trigger prefetch requests on I-Shadow cache misses. The I-Shadow cache is a small tag-only cache that monitors only demand misses. FNL+MMA [24] combines two prefetchers that exploit two characteristics of the I-cache usage. In many cases, the next line is used by the application in the near future. But systematic next-line prefetching leads to overfetching and cache pollution. The Footprint Next Line prefetcher, FNL, overcomes this difficulty through predicting if the next line will be used in the “not so long” future. Prefetching up to 5 next lines, FNL achieves a 16.5 % speed-up on the championship public traces. If no prefetching is used, the sequence of I-cache misses is partially predictable and in advance. That is, when block B is missing, the n^{th} next miss after the miss on block B is often on the same block B (n). This property holds for relatively large n up to 30. The Multiple Miss Ahead prefetcher, MMA, leverages the property. We predict the n^{th} next miss on the I-Shadow cache and predict if it might miss the overall I-cache. A 96 KB FNL+MMA achieves a 28.7 % speed-up and decreases the I-cache miss rate by 91.8 %.

The second prefetcher that we proposed, PIPS, is based on a new idea called Probabilistic Scouting [22]. Unlike a branch predictor, which predicts the most likely control-flow path and follows that single path, a prefetcher can follow multiple control-flow paths simultaneously. For instance, a loop branch is most frequently taken. However, eventually, the loop branch will be not taken and we will exit the loop. PIPS builds a dynamic model of the control-flow graph, learning graph edges frequencies. The graph nodes are cache block addresses. The prefetcher sends “scouts” to explore the graph. A scout moves from node to node, following graph edges, and prefetching the blocks encountered along its way. The scout selects the next node in a probabilistic fashion, where the probability to take an edge is proportional to that edge’s frequency. Several scouts can be active simultaneously. A scout “dies” when a certain ending condition is met. Several possible ending conditions are possible depending on how aggressive we want the prefetcher to be. When a scout dies, it is replaced by a new scout that starts from the current fetch address. The graph is stored in a big, relatively slow on-chip memory. A small, faster Scouting Cache boosts the scouting speed so that instructions are prefetched in time, before they are needed. When evaluated with the championship simulation infrastructure, PIPS, like FNL+MMA, provides a speedup close to that of an ideal prefetcher.

7.2.4 Dynamic thermal management

Participants Pierre Michaud.

Modern high-performance processor chips adjust their clock frequency dynamically so as to not exceed a maximum circuit temperature. This feature is sometimes called turbo mode. Such chips feature integrated thermal sensors at potentially hot circuit locations and monitor temperature continuously. Whenever temperature approaches the limit, the clock frequency is reduced. However, this implies that processor performance depends on ambient temperature. This form of performance non-determinism is a problem for certain equipment manufacturers. To mitigate performance non-determinism, some commercial processors include a calculated temperature in the turbo frequency control loop. This way, the turbo frequency is independent of the ambient temperature, with sensor-based thermal protection triggering only under atypical workloads or unusually high room temperature. For instance, the Intel Sandy Bridge uses a thermal model based on an exponentially weighted moving average (EWMA) of the chip’s power to control its turbo frequency. This explains why the Sandy Bridge’s turbo mode, while motivated by the temperature limit, is largely independent of the actual chip temperature.

Nevertheless, besides the Sandy Bridge’s EWMA, very little information has been disclosed about the turbo control of recent commercial processors. Moreover, the problem of providing a deterministic turbo frequency has been ignored by academic researchers so far. The Sandy Bridge’s EWMA is a very simple thermal model, but an inaccurate one. Implementing a deterministic turbo control based on an inaccurate thermal model means either that the turbo frequency is not completely deterministic or that the performance reduction traded for determinism is higher than necessary.

We propose a thermal model for multicore chips that is both reasonably accurate and simple enough to provide real-time temperature calculations. Based on that model, we propose a new deterministic turbo control exploiting thermal transients while guaranteeing that the power dissipation of each core converges towards a definite steady-state value [16].

7.2.5 Thread convergence prediction for general-purpose SIMT architectures

Participants Arthur Blanleuil, Caroline Collange.

GPUs group threads of SPMD programs in warps and synchronize them to execute the same instruction at the same time. This execution model, referred to as Single-Instruction, Multiple-Thread (SIMT), enables the use of energy-efficient SIMD execution units by factoring out control logic such as instruction fetch and decode pipeline stages for a whole warp. SIMT execution is the key enabler for the energy efficiency of GPUs. We seek to generalize the SIMT execution model to general-purpose superscalar cores.

As threads within a warp may follow different directions through conditional branches in the program, the warp must follow each taken path in turn, while disabling individual threads that do not participate. Following divergence, current GPU architectures attempt to restore convergence at the earliest program point following static annotations in the binary. However, this policy has been shown to be suboptimal in many cases, in which later convergence improves performance. In fact, optimal convergence points depend on dynamic program behavior, so static decisions are unable to capture them.

The goal of the thesis of Arthur Blanleuil is to design predictors that enable the microarchitecture to infer dynamic code behavior and place convergence points appropriately. Convergence predictors have analogies with branch predictors and control independence predictors studied in superscalar processor architecture, but they present one additional challenge: the thread runaway problem. Although a branch misprediction will be identified and repaired locally, a wrong thread scheduling decision may go unnoticed and delay convergence by thousands of instructions. To address the thread runaway problem, we plan to explore promise-based speculation and recovery strategies. When no information is available, we follow the traditional conservative earliest-convergence scheduling policy. Once the predictor has enough information to make a more aggressive prediction, it generates assumptions about the prediction. The microarchitecture then keeps checking dynamically whether the assumptions actually hold true in the near future. If assumptions turn out to be wrong, the prediction will be reconsidered by changing back priorities to conservative. Such promise-based speculation policies can address the thread runaway problem by fixing a bound on the worst-case performance degradation of an aggressive scheduling policy against the conservative baseline.

Accurate thread convergence policies will enable dynamic vectorization to adapt to application characteristics dynamically. They will both improve performance and simplify programming of many-core architectures by alleviating the need for advanced code tuning by expert programmers.

7.2.6 Out-of-order SIMT architectures

Participants Caroline Collange, André Seznec.

In collaboration with Anita Tino, professor at Simon Fraser University in Canada, we proposed SIMT-X, a general-purpose CPU microarchitecture which enables GPU-style SIMT execution across multiple threads of the same program for high throughput, while retaining the latency benefits of out-of-order execution, and the programming convenience of homogeneous multi-thread processors [19]. SIMT-X leverages the existing SIMD backend to provide CPU/GPU-like processing on a single core with minimal overhead. We demonstrate that although SIMT-X invokes a restricted form of OoO, the microarchitecture successfully captures majority of the benefits of aggressive OoO execution using at most two concurrent register mappings per architectural register, while addressing issues of partial dependencies and supporting a general-purpose ISA.

This research was done in cooperation with the Simon Fraser University, Canada.

7.3 WCET estimation and optimization

Participants Abderaouf Nassim Amalou, Isabelle Puaut, Stefanos Skalistis.

7.3.1 WCET estimation and run-time adaptation for many core processors

Participants Stefanos Skalistis.

In time-critical systems, run-time adaptation is required to improve the performance of time-triggered execution, derived based on Worst-Case Execution Time (WCET) of tasks. By improving performance, the systems can provide higher Quality-of-Service, in safety-critical systems, or execute other best-effort applications, in mixed-critical systems. To reduce WCET pessimism, interference-sensitive WCET (aka isWCET) estimations are used. Although they provide tighter WCET bounds, they are valid only for a specific schedule solution. Existing approaches have to maintain this isWCET schedule solution at run-time, via time-triggered execution, in order to be safe. Hence, any earlier execution of tasks, enabled by adapting the isWCET schedule solution, is not possible. In [26], we present a dynamic approach that safely adapts isWCET schedules during execution, by relaxing or completely removing isWCET schedule dependencies, depending on the progress of each core. In this way, an earlier task execution is enabled, creating time slack that can be used by safety-critical and mixed-criticality systems to provide higher Quality-of-Services or execute other best-effort applications. The Response-Time Analysis (RTA) of the proposed approach is presented, showing that although the approach is dynamic, it is fully predictable with bounded WCET. To support our contribution, we evaluate the behavior and the scalability of the proposed approach for different application types and execution configurations on the 8-core Texas Instruments TMS320C6678 platform, obtaining significant performance improvements compared to static approaches.

Note: Stefanos Skalistis was a member of PACAP at the time of this work.

7.3.2 Using machine learning for timing analysis of complex processors

Participants Abderaouf Nassim Amalou, Isabelle Puaut.

Modern processors raise a challenge for WCET estimation, since detailed knowledge of the processor microarchitecture is not or at least not fully available. This work, stated in 2020, proposes a novel hybrid WCET estimation technique, in which the longest path is estimated using static techniques, whereas machine learning (ML) is used to determine the WCET of basic blocks. In contrast to existing literature using ML techniques for WCET estimation, the proposed technique operates on binary code for improved precision of learning, as compared to the related techniques operating at source code or intermediate code level. Moreover, the ML algorithms are trained on a large set of automatically generated programs for improved quality of learning. Finally, the proposed technique includes a way to account for data caches. Experiments on an ARM Cortex-A53 processor show that for all benchmarks, WCET estimates obtained by WE-HML are larger than or equal to all possible execution times. Moreover, the cache modeling technique of WE-HML allows an improvement of 43 % on average of WCET estimates compared to its cache-agnostic equivalent.

This work is done in collaboration with the LACODAM and Whisper teams.

7.4 Security

Participants Romain Belafia, Nicolas Bellec, Antoine Geimer, Damien Hardy, Kévin Le Bon, Pierre-Yves Péneau, Isabelle Puaut, Erven Rohou.

Attack detection co-processor for real-time systems

Participants Nicolas Bellec, Isabelle Puaut.

Real-time embedded systems (RTES) are required to interact more and more with their environment, thereby increasing their attack surface. Recent security breaches on car brakes and other critical components have already proven the feasibility of attacks on RTES. Such attacks may change the control-flow of the programs, which may lead to violations of the system's timing constraints. In [21], we present a technique to detect attacks in RTES based on timing information. Our technique, designed for single-core processors, is based on a monitor implemented in hardware to preserve the predictability of instrumented programs. The monitor uses timing information (Worst-Case Execution Time - WCET) of code regions to detect attacks. The proposed technique guarantees that attacks that delay the run-time of any region beyond its WCET are detected. Since the number of regions in programs impacts the memory resources consumed by the hardware monitor, our method includes a region selection algorithm that limits the amount of memory consumed by the monitor. An implementation of the hardware monitor and its simulation demonstrates the practicality of our approach. In particular, an experimental study evaluates the attack detection latency.

This work is done in collaboration with the CIDRE and CAIRN teams.

Multi-nop fault injection attack

Participants Romain Belafia, Damien Hardy, Pierre-Yves Péneau, Erven Rohou.

Many fault injection techniques have been proposed in the recent years to attack computing systems, as well as the corresponding countermeasures. Most of published attacks are limited to one or a few faults. We provide a theoretical analysis of instruction skip attacks to show how an attacker can modify an application behavior at run-time when thousands of instruction skips are possible. Our main result is that instruction skip is Turing-complete under our theoretical model while requiring the presence of only common instructions in the binary. As a consequence, we show [23] that *current* software-based countermeasures are fragile. In addition, we release a modification of gem5 that implements a classical instruction skip fault model that we used for our experiments. We believe this kind of simulation tools are useful to help the community explore attacks and hardware and software countermeasures.

This work is done in collaboration with the CIDRE team.

Compiler-based automation of side-channel countermeasures

Participants Antoine Geimer, Damien Hardy, Pierre-Yves Péneau, Erven Rohou.

Masking is a popular protection against side-channel analysis exploiting the power consumption or electromagnetic radiations. Besides the many schemes based on simple Boolean encoding, some alternative schemes such as Orthogonal Direct Sum Masking (ODSM) or Inner Product Masking (IP) aim to provide more security, reduce the entropy or combine masking with fault detection. The practical implementation of those schemes is done manually at assembly or source-code level, some of them even stay purely theoretical. We proposed a compiler extension to automatically apply different masking schemes for block cipher algorithms. We introduced a generic approach to describe the schemes and we inserted three of them at compile-time on an AES implementation. Currently, a practical side-channel analysis is performed in collaboration with TAMIS to assess the correctness and the performance of the code inserted.

This work is done in collaboration with the TAMIS team.

Platform for adaptive dynamic protection of programs

Participants Kévin Le Bon, Erven Rohou.

Memory corruption attacks are a serious threat for system integrity. Many techniques have been developed in order to protect systems from these attacks. However, the deployment of heavy protections often degrades the performance of programs. We propose [37] a dynamic approach that injects protections of the control-flow of a running target process and then adapts and refines the protection codes during its execution depending on the observed behaviour. Our approach replaces indirect jumps in the process' code with direct jumps to jump tables, forcing a jump to only reach valid target addresses.

This work is done in collaboration with the CIDRE team.

8 Bilateral contracts and grants with industry

8.1 Bilateral grants with industry

Intel research grant INTEL2016-11174

Participants Niloofar Charmchi, Kleovoulos Kalaitzidis, Anis Peysieux, André Seznec.

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

Intel research grant INTEL2019-14458 Intel is supporting the research of the PACAP project-team on “Towards simple and highly efficient execution cores”.

Participants André Seznec.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Inria international partners

Informal international partners Caroline Collange is collaborating with Anita Tino, professor at Simon Fraser University, Canada [19].

Erven Rohou is collaborating with Prof. Ahmed El-Mahdy (Egypt-Japan University of Science and Technology, Alexandria, Egypt) and his group [18, 17, 17].

9.2 European initiatives

9.2.1 FP7 & H2020 Projects

HiPEAC4 NoE

Participants Damien Hardy, Erven Rohou, André Seznec.

D. Hardy, 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.

Eurolab-4-HPC

Participants Erven Rohou.

- Title: EuroLab-4-HPC: Foundations of a European Research Center of Excellence in High Performance Computing Systems
- Program: H2020
- Duration: September 2018 – September 2020
- Coordinator: Chalmers Tekniska Högskola AB (Sweden)
- Partners:
  - Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)
  - Chalmers Tekniska Högskola (Sweden)
  - Foundation for Research and Technology Hellas (Greece)
  - Universität Stuttgart (Germany)
  - The University of Manchester (United Kingdom)
  - Inria (France)
  - Universität Augsburg (Germany)
  - ETH Zürich (Switzerland)
  - École Polytechnique Federale de Lausanne (Switzerland)
  - Technion - Israel Institute of Technology (Israel)
  - The University of Edinburgh (United Kingdom)
  - Rheinisch-Westfälische Technische Hochschule Aachen (Germany)
  - Universiteit Gent (Belgium)
- 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 long-term 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.

9.2.2 Collaborations in European programs, except FP7 and H2020

Participants Isabelle Puaut.

Isabelle Puaut is member of the COST action Cerciras: Connecting Education and Research Communities for an Innovative Resource Aware Society (`https://www.cost.eu/actions/CA19135/`). The COST action was approved on March 2020, started on September 2020 and will end on September 2024.

9.3 National initiatives

Zero Power Computing Systems (ZEP): Inria Project Lab (2017–2020)

Participants Erven Rohou, Bahram Yarahmadi.

This proposal addresses the issue of designing tiny wireless, batteryless, computing objects, harvesting energy in the environment. The energy level harvested being very low, very frequent energy shortages are expected. In order for the new system to maintain a consistent state, it will be based on a new architecture embedding non-volatile RAM (NVRAM). In order to benefit from the hardware innovations related to energy harvesting and NVRAM, software mechanisms will be designed. On the one hand, a compilation pass will compute a worst-case energy consumption. On the other hand, dedicated runtime mechanisms will allow:

1. to manage efficiently and correctly the NVRAM-based hardware architecture;
2. to use energy intelligently, by computing the worst-case energy consumption.

The ZEP project gathers four Inria teams that have a scientific background in architecture, compilation, operating systems together with the CEA Lialp and Lisan laboratories of CEA LETI & LIST [32]. The main application target is Internet of Things (IoT).

EQIP: Inria Challenge project, 2021 – 2024

Participants Caroline Collange.

Building a functional quantum computer is one of the grand scientific challenges of the 21st century. This formidable task is the object of Quantum Engineering, a new and very active field of research at the interface between physics, computer science and mathematics. EQIP brings together all the competences already present in the institute, to turn Inria into a major international actor in quantum engineering, including both software and hardware aspects of quantum computing.

DGA/PEC ARMOUR (2018–2021)

Participants Kévin Le Bon, Erven Rohou.

ARMOUR (dynAmic binaRy optiMizatiOn cyber-secURity) aims at improving the security of computing systems at the software level. Our contribution will be twofold: (1) identify vulnerabilities in existing software, and (2) develop adaptive countermeasure mechanisms against attacks. We will rely on dynamic binary rewriting (DBR) which consists in observing a program and modifying its binary representation in memory while it runs. DBR does not require the source code of the programs it manipulates, making it convenient for commercial and legacy applications. We will study the feasibility of an adaptive security agent that monitors target applications and deploys (or removes) countermeasures based on dynamic conditions. Lightweight monitoring is appropriate when the threat condition is low, heavy countermeasures will be dynamically woven into the code when an attack is detected. Vulnerability analysis will be based on advanced fuzzing. DBR makes it possible to monitor and modify deeply embedded variables, inaccessible to traditional monitoring systems, and also to detect unexpected/suspicious values taken by variables and act before the application crashes.

ARMOUR is funded by DGA (*Direction Générale de l'Armement*) and PEC (*Pôle d'Excellence Cyber*).

ANR DYVE (31/03/2020 – 30/09/2023)

Participants Arthur Blanleuil, Caroline Collange, Pierre-Yves Péneau.

Most of today's computer systems have CPU cores and GPU cores on the same chip. Though both are general-purpose, CPUs and GPUs still have fundamentally different software stacks and programming models, starting from the instruction set architecture. Indeed, GPUs rely on static vectorization of parallel applications, which demands vector instruction sets instead of CPU scalar instruction sets. In the DYVE project, we advocate a disruptive change in both CPU and GPU architecture by introducing Dynamic Vectorization at the hardware level.

Dynamic Vectorization will combine the efficiency of GPUs with the programmability and compatibility of CPUs by bringing them together into heterogeneous general-purpose multicores. It will enable processor architectures of the next decades to provide (1) high performance on sequential program sections thanks to latency-optimized cores, (2) energy-efficiency on parallel sections thanks to throughput-optimized cores, (3) programmability, binary compatibility and portability.

DYVE is funded by the ANR through the JCJC funding instrument.

9.4 Regional initiatives

Participants Nicolas Bellec, Niloofar Charmchi.

The Brittany Region is partially funding the PhD fellowship for Niloofar Charmchi on the topic "Hardware prefetching and related issues" and Nicolas Bellec on the topic "Security in real-time embedded systems".

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: selection

Member of the conference program committees

- Isabelle Puaut was a member of program committee of the Euromicro Conference on Real Time Systems (ECRTS) 2020
- Isabelle Puaut was a member of program committee of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2020
- Isabelle Puaut was a member of program committee of the Real Time Systems Symposium (RTSS), 2020
- Isabelle Puaut was a member of program committee of the Conference on Real-Time Networks and Systems (RTNS), 2020
- André Seznec was a member of program committee of the ACM/IEEE ISCA 2020 conference

- André Seznec was a member of program committee of the IPDPS 2020 conference
- André Seznec was a member of program committee of the ICCD 2020 conference
- André Seznec was a member of program committee of the HiPC 2020 conference
- Pierre Michaud was a member of the Industry Session committee of the HPCA 2021 conference
- Caroline Collange was a member of the program committee of Design Automation and Test in Europe (DATE) 2020
- Caroline Collange was a program committee member of Supercomputing Conference (SC) 2020

Reviewer Members of PACAP routinely review submissions to numerous international conferences and events.

10.1.2 Journal

Member of the editorial boards

- Isabelle Puaut was Associate Editor of IEEE Transactions on Computers (IEEE TC) up to April 2020
- Isabelle Puaut is associate editor of the Springer International Journal of Time-Critical Computing Systems since February 2020.
- André Seznec is a member of the editorial board of ACM Transactions on Architecture and Compiler Optimization.

Reviewer - reviewing activities Members of PACAP routinely review submissions to numerous international conferences and events.

10.1.3 Leadership within the scientific community

- Isabelle Puaut is member of the IEEE technical committee on real-time systems (TCRTS), since February 2020
- 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 advisory board of the Euromicro Conference on Real Time Systems (ECRTS).
- Caroline Collange is member of the steering committee of *Conférence francophone d'informatique en Parallélisme, Architecture et Système* (Compas).

10.1.4 Research administration

- Isabelle Puaut is member of the Research Council (Commission Recherche) of the Université de Rennes 1, till Jul 2020. She was member of the working group “Habilitation à Diriger des Recherches” till July 2020.
- Isabelle Puaut is member of the board of directors (Conseil d'Administration) of ISTIC (computer science and electrical engineering departement of Université de Rennes 1).
- Erven Rohou is “correspondant scientifique des relations internationales” for Inria Rennes Bretagne Atlantique. As such he is a member of the Inria COST GTRI (Groupe de Travail “Relations Internationales”).
- Erven Rohou is a member of the steering committee of the high security research laboratory (LHS).

- Erven Rohou is a member of the steering committee of the Inria Rennes computer grid “igrida”.
- Erven Rohou is a member of the “Comité de Centre” of the Inria Rennes Research Center.
- André Seznec is an elected member of the Administration Council of Inria.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Master: N. Bellec, ITR (Informatique Temps Réel), 16h, M1, Université Rennes 1, France.
- Master: A. Blanleuil, NOY (Systèmes d’exploitation – implémentation de noyaux de systèmes), 22 hours, M1, Université Rennes 1, France
- Master: C. Collange, GPU programming, 20 hours, M1, Université de Rennes 1, France
- Licence: D. Hardy, Real-time systems, 68 hours, L3, Université de Rennes 1, France
- Master: D. Hardy, Operating systems, 59 hours, M1, Université de Rennes 1, France
- Master: K. Le Bon, Architecture à Objet Canonique, 12 hours, M2, Université de Rennes 1, France
- Master: I. Puaut, Operating systems: concepts and system programming under Linux (SEL), 77 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Operating systems kernels (NOY), 54 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Real-time systems, 40 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Optimizing and Parallelizing Compilers (OPC), 9 hours, M2, Université de Rennes 1, France
- Master: I. Puaut, Writing of scientific publications, 9 hours, M2 and PhD students, Université de Rennes 1, France
- Master: A. Seznec and C. Collange, Advanced Design and Architectures, 16 hours, M2 SIF, Université de Rennes 1, France.
- Master: C. Collange, High-Performance Computing, 8 hours, M2 SFPN, Sorbonne Université, France.

10.2.2 Supervision

- PhD: Niloofar Charmchi, *Hardware prefetching and related issue* [28], Université de Rennes 1, 10 Jun 2020, advisors A. Seznec and C. Collange
- PhD: Kleovoulos Kalaitzidis, *Ultrawide Issue Superscalar Processors* [29], Université de Rennes 1, 6 Mar 2020, advisor A. Seznec
- PhD in progress : Nicolas Bellec, *Security in real-time embedded systems*, started Dec 2019, advisors I. Puaut (50 %), G. Hiet from CIDRE (25 %), F. Tronel from CIDRE (25 %)
- PhD in progress: Arthur Blanleuil, *Thread convergence prediction for SIMT architectures*, Université de Rennes 1, started Oct 2018, advisor C. Collange and A. Seznec
- PhD in progress : Kévin Le Bon, *Dynamic Binary Analysis and Optimization for Cyber-Security*, started Dec 2018, advisors E. Rohou (30 %), G. Hiet from CIDRE (35 %), F. Tronel from CIDRE (35 %)
- PhD in progress: Daniel Rodrigues Carvalho, *Towards a compressed memory hierarchy*, Université de Rennes 1, started Oct 2017, advisor A. Seznec

- PhD in progress : Bahram Yarahmadi, *Compiler Optimizations and Worst-Case Energy Consumption*, started Feb 2018, advisor E. Rohou
- PhD in progress: Abderaouf Nassim Amalou, *Security in real-time embedded systems*, started Oct 2020, advisors Isabelle Puaut (75 %), Elisa Fromont (25 %, LACODAM group).
- PhD in progress: Anis Peysieux, *Towards simple and highly efficient execution cores*, started Jan 2020, advisor E. Seznec.

10.2.3 Juries

Isabelle Puaut was a member of the following PhD thesis committees:

- Cédric Courtaud, *Characterization of memory interferences in embedded real-time multi-core platform*, Sorbonne Université, Jan 2020
- Stephan Plassart, *Online energy optimisation for real-time systems*, Université de Grenoble, Jun 2020
- Khaoula Boukir, *Implementation of proved multiprocessor scheduling policies*, Université de Nantes, Dec 2020 (president)

Isabelle Puaut was a member of the following hiring committees :

- Inria CRCN and ISFP hiring committee, 2020
- Assistant professor Université de Rennes 1/ESIR, "software engineering"

Erven Rohou was a member of the following PhD thesis committee:

- Pierre Graux, *Challenges of Native Android Applications: Obfuscation and Vulnerabilities*, Dec 2020 (president).

Caroline Collange was a member of the following PhD thesis committee:

- Darius Mercadier, *Usuba, Optimizing Bitslicing Compiler*, Nov 2020

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

Isabelle Puaut is member of Paul Caspi PhD thesis prize 2020, awarded by the ACM Special Interest Group on Embedded Systems (ACM SIGBED)

Caroline Collange is a committee member of the Gilles Kahn PhD thesis prize 2020, awarded by Société Informatique de France (SIF)

10.3.2 Articles and contents

An article introducing the ANR/JCJC project of C. Collange, titled "Unifier la programmation sur les processeurs multi-cœurs hétérogènes", was published in *Émergences*. [6]

[6]https://project.inria.fr/emergences/unifier-la-programmation-sur-les-processeurs-multicoeurs-heterogenes-le-defi-de-caroline-collange/

11 Scientific production

11.1 Major publications

[1] F. Bodin, T. Kisuki, P. M. W. Knijnenburg, M. F. P. O'Boyle and 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*. Paris, France, Oct. 1998.

[2] N. Hallou, E. Rohou, P. Clauss and A. Ketterlin. 'Dynamic Re-Vectorization of Binary Code'. In: *SAMOS*. July 2015. URL: https://hal.inria.fr/hal-01155207.

[3] D. Hardy and 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, Oct. 2013. DOI: 10.1145/2516821.2516842. URL: https://hal.inria.fr/hal-00862604.

[4] D. Hardy, I. Sideris, N. Ladas and Y. Sazeides. 'The performance vulnerability of architectural and non-architectural arrays to permanent faults'. In: *MICRO 45*. Vancouver, Canada, Dec. 2012. URL: https://hal.inria.fr/hal-00747488.

[5] S. Kalathingal, S. Collange, B. Swamy and A. Seznec. 'DITVA: Dynamic Inter-Thread Vectorization Architecture'. In: *Journal of Parallel and Distributed Computing* (Oct. 2018), pp. 1–32. DOI: 10.1016/j.jpdc.2017.11.006. URL: https://hal.archives-ouvertes.fr/hal-01655904.

[6] P. Michaud. 'Best-Offset Hardware Prefetching'. In: *International Symposium on High-Performance Computer Architecture*. Barcelona, Spain, Mar. 2016. DOI: 10.1109/HPCA.2016.7446087. URL: https://hal.inria.fr/hal-01254863.

[7] P. Michaud, A. Mondelli and A. Seznec. 'Revisiting Clustered Microarchitecture for Future Superscalar Cores: A Case for Wide Issue Clusters'. In: *ACM Transactions on Architecture and Code Optimization (TACO)* 13.3 (Aug. 2015), p. 22. DOI: 10.1145/2800787. URL: https://hal.inria.fr/hal-01193178.

[8] A. Perais and A. Seznec. 'EOLE: Paving the Way for an Effective Implementation of Value Prediction'. In: *International Symposium on Computer Architecture*. Vol. 42. ACM/IEEE. Minneapolis, MN, United States, June 2014, pp. 481–492. DOI: 10.1109/ISCA.2014.6853205. URL: https://hal.inria.fr/hal-01088130.

[9] A. Perais and A. Seznec. 'Practical data value speculation for future high-end processors'. In: *International Symposium on High Performance Computer Architecture*. IEEE. Orlando, FL, United States, Feb. 2014, pp. 428–439. DOI: 10.1109/HPCA.2014.6835952. URL: https://hal.inria.fr/hal-01088116.

[10] E. Rohou, B. Narasimha Swamy and A. Seznec. 'Branch Prediction and the Performance of Interpreters - Don't Trust Folklore'. In: *International Symposium on Code Generation and Optimization*. Burlingame, United States, Feb. 2015. URL: https://hal.inria.fr/hal-01100647.

[11] S. Sardashti, A. Seznec and D. A. Wood. 'Skewed Compressed Caches'. In: *47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014*. Minneapolis, United States, Dec. 2014. URL: https://hal.inria.fr/hal-01088050.

[12] S. Sardashti, A. Seznec and D. A. Wood. 'Yet Another Compressed Cache: a Low Cost Yet Effective Compressed Cache'. In: *ACM Transactions on Architecture and Code Optimization* (Sept. 2016), p. 25. URL: https://hal.inria.fr/hal-01354248.

[13] A. Seznec and P. Michaud. 'A case for (partially)-tagged geometric history length branch prediction'. In: *Journal of Instruction Level Parallelism* (Feb. 2006). URL: http://www.jilp.org/vol8.

[14] D. D. C. Teixeira, S. Collange and F. M. Quintão Pereira. 'Fusion of calling sites'. In: *International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD)*. Florianópolis, Santa Catarina, Brazil, Oct. 2015. DOI: 10.1109/SBAC-PAD.2015.16. URL: https://hal.archives-ouvertes.fr/hal-01410221.

11.2 Publications of the year

International journals

[15] K. Kalaitzidis and A. Seznec. 'Leveraging Value Equality Prediction for Value Speculation'. In: *ACM Transactions on Architecture and Code Optimization* 18.1 (29th Dec. 2020), pp. 1–20. DOI: 10.1145/3436821. URL: https://hal.inria.fr/hal-03097413.

[16] P. Michaud. 'Exploiting Thermal Transients With Deterministic Turbo Clock Frequency'. In: *IEEE Computer Architecture Letters* 19.1 (30th Mar. 2020), pp. 43–46. DOI: 10.1109/LCA.2020.2983920. URL: https://hal.inria.fr/hal-02562105.

[17] R. Omar, M. Abbas, A. El-Mahdy and E. Rohou. 'Binary-level data dependence analysis of hot execution regions using abstract interpretation at runtime'. In: *PLoS ONE* 15.4 (9th Apr. 2020), pp. 1–20. DOI: 10.1371/journal.pone.0230904. URL: https://hal.inria.fr/hal-02913722.

[18] R. Omar, A. El-Mahdy and E. Rohou. 'IR-Level Dynamic Data Dependence Using Abstract Interpretation Towards Speculative Parallelization'. In: *IEEE Access* 8 (26th May 2020), pp. 99910–99921. DOI: 10.1109/ACCESS.2020.2997715. URL: https://hal.inria.fr/hal-02913838.

[19] A. Tino, C. Collange and A. Seznec. 'SIMT-X: Extending Single-Instruction Multi-Threading to Out-of-Order Cores'. In: *ACM Transactions on Architecture and Code Optimization* 17.2 (May 2020), p. 15. DOI: 10.1145/3392032. URL: https://hal.inria.fr/hal-02542333.

International peer-reviewed conferences

[20] M. Abbas, R. Omar, A. El-Mahdy and E. Rohou. 'Approximate Data Dependence Profiling based on Abstract Interval and Congruent Domains'. In: ARCS 2020 - 33rd International Conference on Architecture of Computing Systems. Aachen (virtual), Germany, 9th July 2020, pp. 3–16. DOI: 10.1007/978-3-030-52794-5_1. URL: https://hal.inria.fr/hal-02914569.

[21] N. Bellec, S. Rokicki and I. Puaut. 'Attack detection through monitoring of timing deviations in embedded real-time systems'. In: ECRTS 2020 - 32nd Euromicro Conference on Real-Time Systems. Modena, Italy, 7th July 2020, pp. 1–22. DOI: 10.4230/LIPIcs.ECRTS.2020.8. URL: https://hal.inria.fr/hal-02559549.

[22] P. Michaud. 'PIPS: Prefetching Instructions with Probabilistic Scouts'. In: *First Instruction Prefetching Championship (IPC-1)*. IPC-1 - First Instruction Prefetching Championship. Valencia, Spain: https://research.ece.ncsu.edu/ipc/, 1st June 2020, pp. 1–4. URL: https://hal.inria.fr/hal-02861614.

[23] P.-Y. Péneau, L. Claudepierre, D. Hardy and E. Rohou. 'NOP-Oriented Programming: Should we Care?' In: Sécurité des Interfaces Logiciel/Matériel. Genoa (virtual), Italy, 11th Sept. 2020. DOI: 10.1109/EuroSPW51379.2020.00100. URL: https://hal.inria.fr/hal-02912301.

[24] A. Seznec. 'The FNL+MMA Instruction Cache Prefetcher'. In: *First Instruction Prefetching Championship (IPC-1)*. IPC-1 - First Instruction Prefetching Championship. Valence, Spain: https://research.ece.ncsu.edu/ipc/, 1st June 2020, pp. 1–5. URL: https://hal.inria.fr/hal-02884880.

[25] A. Seznec and K. Kalaitzidis. 'Exploring value prediction limits'. In: CVP 2020 - Championship Value Prediction. Los-Angeles, United States: https://www.microarch.org/cvp1/index.html, 15th Feb. 2020, pp. 1–5. URL: https://hal.inria.fr/hal-02884853.

[26] S. Skalistis and A. Kritikakou. 'Dynamic Interference-Sensitive Run-time Adaptation of Time-Triggered Schedules'. In: ECRTS 2020 - 32nd Euromicro Conference on Real-Time Systems. ECRTS 2020 - 32nd Euromicro Conference on Real-Time Systems. Virtual, France, 7th July 2020, pp. 1–22. DOI: 10.4230/LIPIcs.ECRTS.2020.4. URL: https://hal.archives-ouvertes.fr/hal-02927451.

[27] B. Yarahmadi and E. Rohou. 'Compiler Optimizations for Safe Insertion of Checkpoints in Intermittently Powered Systems'. In: SAMOS 2020 - International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation. Springer's Lecture Notes in Computer Science – LNCS. Virtual, Greece, 6th July 2020, pp. 1–16. DOI: 10.1007/978-3-030-60939-9_12. URL: https://hal.inria.fr/hal-02914953.

Doctoral dissertations and habilitation theses

[28] N. Charmchi. 'Compressed cache layout aware prefetching'. Université Rennes 1, 10th July 2020. URL: https://tel.archives-ouvertes.fr/tel-03099313.

[29] K. Kalaitzidis. 'Advanced speculation to increase the performance of superscalar processors'. Université Rennes 1, 6th Mar. 2020. URL: https://tel.archives-ouvertes.fr/tel-03033709.

Reports & preprints

[30] D. Hardy. *Ofast3D – Feasibility Study*. Inria Rennes - Bretagne Atlantique; IRISA, Dec. 2020, p. 18. URL: https://hal.inria.fr/hal-03093905.

[31] J. Lowe-Power, A. M. Ahmad, A. Armejach, A. Herrera, A. Roelke, A. Farmahini-Farahani, A. Mondelli, A. Hansson, A. Sandberg, A. Gutierrez et al. *The gem5 Simulator: Version 20.0+*. 6th Jan. 2021. URL: https://hal.inria.fr/hal-03100818.

11.3 Cited publications

[32] G. Berthou, A. Carer, H.-P. Charles, S. Derrien, K. Marquet, I. Miro-Panades, D. Pala, I. Puaut, F. Rastello, T. Risset, E. Rohou, G. Salagnac, O. Sentieys and B. Yarahmadi. *The INRIA ZEP project: NVRAM and Harvesting for Zero Power Computations*. NVMW 2018 - 10th Annual Non-Volatile Memories Workshop. Poster. Mar. 2018. URL: https://hal.inria.fr/hal-01941766.

[33] A. Cohen and E. Rohou. 'Processor Virtualization and Split Compilation for Heterogeneous Multicore Embedded Systems'. In: *DAC*. Anaheim, CA, USA, June 2010, pp. 102–107.

[34] M. Hataba, A. El-Mahdy and E. Rohou. 'OJIT: A Novel Obfuscation Approach Using Standard Just-In-Time Compiler Transformations'. In: *International Workshop on Dynamic Compilation Everywhere*. Jan. 2015.

[35] S. Kalathingal, S. Collange, B. Narasimha Swamy and 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, Oct. 2016. URL: https://hal.inria.fr/hal-01356202.

[36] R. Kumar, D. M. Tullsen, N. P. Jouppi and P. Ranganathan. 'Heterogeneous chip multiprocessors'. In: *IEEE Computer* 38.11 (Nov. 2005), pp. 32–38.

[37] K. Le Bon, B. Hawkins, E. Rohou, G. Hiet and F. Tronel. 'Plateforme de protection de binaires configurable et dynamiquement adaptative'. In: *RESSI 2019 - Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*. Erquy, France, May 2019, pp. 1–3. URL: https://hal.inria.fr/hal-02385216.

[38] P. Michaud and 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. URL: https://hal.archives-ouvertes.fr/hal-01087719.

[39] R. Omar, A. El-Mahdy and 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.

[40] E. Riou, E. Rohou, P. Clauss, N. Hallou and A. Ketterlin. 'PADRONE: a Platform for Online Profiling, Analysis, and Optimization'. In: *Dynamic Compilation Everywhere*. Vienna, Austria, Jan. 2014.

[41] A. Sembrant, T. Carlson, E. Hagersten, D. Black-Shaffer, A. Perais, A. Seznec and P. Michaud. ‘Long Term Parking (LTP): Criticality-aware Resource Allocation in OOO Processors’. In: *International Symposium on Microarchitecture, Micro 2015*. Proceeding of the International Symposium on Microarchitecture, Micro 2015. Honolulu, United States: ACM, Dec. 2015. URL: https://hal.inria.fr/hal-01225019.

[42] A. Seznec. ‘TAGE-SC-L Branch Predictors: **Champion in 32Kbits and 256 Kbits category**’. In: *JILP - Championship Branch Prediction*. Minneapolis, United States, June 2014. URL: https://hal.inria.fr/hal-01086920.

[43] A. Seznec, J. San Miguel and J. Albericio. ‘The Inner Most Loop Iteration counter: a new dimension in branch history ’. In: *48th International Symposium On Microarchitecture*. Honolulu, United States: ACM, Dec. 2015, p. 11. URL: https://hal.inria.fr/hal-01208347.

[44] A. Seznec and N. Sendrier. ‘HAVEGE: A user-level software heuristic for generating empirically strong random numbers’. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 13.4 (2003), pp. 334–346.