RESEARCH CENTRE

Rennes - Bretagne Atlantique

IN PARTNERSHIP WITH:

Université Rennes 1

2021

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]
- Pierre Michaud [Inria, Researcher]

Faculty Members

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

PhD Students

- Abderaouf Nassim Amalou [Univ de Rennes I]
- Nicolas Bellec [Univ de Rennes I]
- Lily Blanleuil [Univ de Rennes I, until Sep 2021]
- Antoine Gicquel [Inria, from Sep 2021]
- Sara Sadat Hoseininasab [Inria, from Nov 2021]
- Camille Le Bon [Inria]
- Anis Peysieux [Inria]
- Hugo Reymond [Inria, from Oct 2021]
- Bahram Yarahmadi [Inria, until Apr 2021]

Technical Staff

- Oussama Houidar [Univ de Rennes I, Engineer, from Sep 2021]
- Pierre Yves Péneau [Inria, Engineer, until Sep 2021]

Interns and Apprentices

- Lauric Desauw [Inria, from Feb 2021 until Jul 2021]
- Maxime Desmarais [Inria, from Mar 2021 until Aug 2021]
- Audrey Fauveau [Inria, from May 2021 until Aug 2021]
- Antoine Gicquel [Univ de Rennes I, from Feb 2021 until Aug 2021]
- Oussama Houidar [Inria, from May 2021 until Jul 2021]
- Valentin Pasquale [École Normale Supérieure de Lyon, from Oct 2021]
- Mathieu Vaudeleau [Inria, from May 2021 until Aug 2021]

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.

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.

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

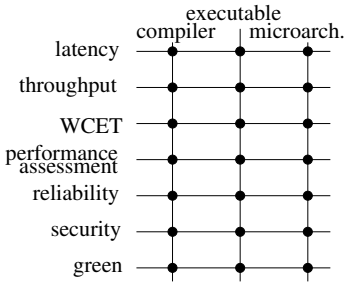

Figure 1: Perimeter of Research Objectives

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 [29] 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 technologies. These techniques expose new characteristics to the software layers. As an example, non-volatile memories typically have asymmetric read-write latencies (writes are much longer than reads) and different power consumption profiles. PACAP 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.

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

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 [29]: 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 [37] 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 [39] [35] and in cache prefetching [5] and they have revived the forgotten concept of value prediction [8][7]. 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) [5]. 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 [6]. 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 [6] 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. [38]. 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 [40].

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 [8][7] [39] [35]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [8]. 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 [7]. 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 [34] 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 [15] 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 or the Recore manycore hardware) 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;

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 [41] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [33], or thread-based control flow mangling [36]). 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 New software and platforms

5.1 New software

5.1.1 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/

Contact: Pierre Michaud

Participant: Pierre Michaud

5.1.2 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/

Contact: Isabelle Puaut

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

Partner: Université de Rennes 1

5.1.3 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. Unless the system administrator has restricted access to performance counters, 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/

Contact: Erven Rohou

Participant: Erven Rohou

5.1.4 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

Contact: Erven Rohou

5.1.5 TRAITOR tool suite

Keywords: Security, Clock glitch, Fault injection

Functional Description: TRAITOR is a low-cost evaluation platform for multifault injection. It is based on clock glitches with the capacity to inject numerous and precise bursts of faults. TRAITOR tool suite is a suite of tools that makes it easier to interact with TRAITOR to simplify its use, configure the platform and automate the characterization of TRAITOR on machine instructions.

Authors: Maxime Desmarais, Pierre-Yves Peneau, Damien Hardy

Contact: Damien Hardy

5.1.6 energy-meter

Keywords: Monitoring, Power consumption, Linux, Applications

Functional Description: Energy-meter provides a single interface to mechanisms for measuring the energy consumed by a computer running Linux/x86. Two native mecanisms are supported: perfevent and powercap. Energy-meter provides a library and a use-case in the form of a tool similar to "top".

URL: https://gitlab.inria.fr/rohou/energy-meter

Author: Erven Rohou

Contact: Erven Rohou

5.1.7 Static-dynamic performance autotuner

Keywords: Performance, Compilation, Heterogeneity, Autotuning, Power consumption

Functional Description: Performance and energy consumption of an application depend on both the processor on which it runs, and the way it has been compiled. This software automatically explores different optimization sequences and different execution cores (on heterogeneous machines) and retains the best configuration. We demonstrated it with an MPEG encoder on several processors: x86, Arm big.LITTLE, and a heptacore from Cortus.

URL: https://hal.inria.fr/hal-03375509v1

Author: Erven Rohou

Contact: Erven Rohou

5.1.8 PADRONE

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

Functional Description: Padrone is a 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

Contact: Erven Rohou

Participants: Emmanuel Riou, Erven Rohou

5.1.9 Sorry

Keywords: Dynamic Analysis, Security, High performance computing

Functional Description: Dynamic binary modification consists in rewriting a program, in binary form, directly in memory, and while it runs. This offers a number of advantages, but it requires skills and it is error-prone. Sorry is a library that facilitates dynamic binary modification. Its main features consist in 1) its ability to attach to a running program, 2) its minimal impact on the performance of the target program.

URL: https://gitlab.inria.fr/klebon/sorry

Author: Camille Le Bon

Contact: Erven Rohou

5.1.10 Damas

Keyword: Cybersecurity

Functional Description: Damas is a framework for Control-Data Isolation at Runtime through Dynamic Binary Modification. It attaches to a target application and rewrites its binary code to eliminate indirect branch instructions in order to make some attacks impossible. Damas is based on the Sorry library.

URL: https://gitlab.inria.fr/klebon/damas

Author: Camille Le Bon

Contact: Erven Rohou

6 New results

Participants: Abderaouf Nassim Amalou, Nicolas Bellec, Lily Blanleuil, Caroline Collange, Maxime Desmarais, Audrey Fauveau, Antoine Gicquel, Damien Hardy, Oussama Houidar, Kleovoulos Kalaitzidis, Camille Le Bon, Pierre Michaud, Valentin Pasquale, Pierre-Yves Péneau, Anis Peysieux, Isabelle Puaut, Daniel Rodrigues Carvalho, Erven Rohou, André Seznec, Bahram Yarahmadi.

6.1 Compilation and Optimization

Participants: Caroline Collange, Audrey Fauveau, Damien Hardy, Oussama Houidar, Erven Rohou, Bahram Yarahmadi

6.1.1 Optimization in the Presence of NVRAM

Participants: Erven Rohou, Bahram Yarahmadi
Context: Inria Project Lab ZEP.

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.

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 [23] 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 work concludes the PhD of Bahram Yarahmadi [26].

6.1.2 Dynamic Binary Analysis and Optimization

Participants: Erven Rohou
Context: ANR Project Continuum.

On a heterogeneous processor, the performance of an application depends on both the core on which it runs, and how it has been compiled. We proposed [28] an autotuning approach where a performance-critical region of an application is periodically recompiled at run-time, as well as migrated across cores. The performance and energy consumption is monitored. The system retains the most efficient version encountered so far.

We experimented with a MPEG encoder where the critical function is the encoding of a single frame. The metrics we chose to tune the software are time-per-frame and energy-per-frame. We varied frame sizes and image content to observe the behavior of the auto-tuning mechanism, highlighting tradeoffs between performance and consumed energy, as well as the benefit of the fixed-point representation.

The prototype is described in Section 5.1.7. Energy consumption is measured using the software described in Section 5.1.6.

6.1.3 Adapting the LLVM compiler to the Gated-SSA intermediate representation

Participants: Caroline Collange, Audrey Fauveau
External collaborators: Delphine Demange from the CELTIQUE Team

The *Gated Single-Assignment* (GSA) intermediate representation offers multiple advantages over the *Static Single Assignment* (SSA) intermediate representation which is the de facto standard in modern compilers. By making control dependencies explicit in addition to data dependencies, it facilitates backward symbolic analyses as well as divergence analysis on GPUs [10] or High-Level Synthesis [31]. In order to make GSA-based analyses easier to implement, we add support for GSA in the LLVM compiler. Our approach maintains compatibility with existing analysis and optimization passes by preserving the SSA-based intermediate representation and augmenting it with control-flow information.

6.1.4 Accurate 3D printing time estimation

Participants: Damien Hardy, Oussama Houidar
Context: Inria Exploratory Action Ofast3D

Fused deposition modeling 3D printing is a process that requires hours or even days to print a 3D model. To assess the benefits of optimizations, it is mandatory to have a fast 3D printing time estimator to avoid waste of materials and a very long validation process. Furthermore, the estimation must be accurate [32]. To reach that goal, we have modified an existing firmware called Klipper in simulation mode to determine the timing per G-code instruction. The validation is in progress but the first results are promising and reveal a very good accuracy while the simulation is fast.

6.2 Processor Architecture

Participants: Lily Blanleuil, Caroline Collange, Kleovoulos Kalaitzidis, Pierre Michaud, Anis Peysieux, Daniel Rodrigues Carvalho, André Seznec

6.2.1 Value prediction

Participants: Kleovoulos Kalaitzidis, André Seznec

Value Prediction (VP) has recently been 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. We address [16] 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.

6.2.2 Compressed caches

Participants: Daniel Rodrigues Carvalho, André Seznec

Hardware cache compression derives from software-compression research; yet, its implementation is not a straightforward translation, since it must abide by multiple restrictions to comply with area, power, and latency constraints. This study [17] sheds light on the challenges of adopting compression in cache design—from the shrinking of the data until its physical placement. The goal of this work is not to summarize proposals but to put in evidence the solutions they employ to handle those challenges. An in-depth description of the main characteristics of multiple methods is provided, as well as criteria that can be used as a basis for the assessment of such schemes. It is expected that this work will ease the understanding of decisions to be taken for the design of compressed systems and provide directions for future work.

Compressed cache layouts require adding the block's size information to the metadata array. This field can be either constrained – in which case compressed blocks must fit in predetermined sizes; thus, it reduces co-allocation opportunities but has easier management – or unconstrained – in which case compressed blocks can compress to any size; thus, it increases co-allocation opportunities, at the cost of more metadata and latency overheads. This work [21] introduces the concept of partial constraint, which explores multiple layers of constraint to reduce the overheads of unconstrained sizes, while still allowing a high co-allocation flexibility. Finally, Pairwise Space Sharing (PSS) is proposed, which leverages a special case of a partially constrained system. PSS can be applied orthogonally to compaction methods at no extra latency penalty to increase the cost-effectiveness of their metadata overhead. This concept is compression-algorithm independent, and results in an increase of the effective compression ratios achieved while making the most of the metadata bits. When normalized against compressed systems not using PSS, a compressed system extended with PSS further enhances the average cache capacity of nearly every workload.

Cache compression algorithms must abide by hardware constraints; thus, their efficiency ends up being low, and most cache lines end up barely compressed. Moreover, schemes that compress relatively well often decompress slowly, and vice versa. This work [22] proposes a compression scheme achieving high (good) compaction ratio and fast decompression latency. The key observation is that by further subdividing the chunks of data being compressed one can tailor the algorithms. This concept is orthogonal to most existent compressors, and results in a reduction of their average compressed size. In particular, we leverage this concept to boost a single-cycle-decompression compressor to reach a compressibility level competitive to stateof-the-art proposals. When normalized against the best long decompression latency state-of-the-art compressors, the proposed ideas further enhance the average cache capacity by 2.7 % (geometric mean), while featuring short decompression latency.

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 [27, §2.11].

6.2.3 Energy-efficient microarchitecture

Participants: Pierre Michaud, Anis Peysieux

Since around 2005, CPU performance has kept increasing while the CPU thermal design power remained limited by the cooling capacity. Twenty years ago, it was possible to sacrifice energy efficiency for maximizing performance. However, in today's CPUs, energy efficiency is a necessary condition for high performance, even for desktop and server CPUs. This fact is manifest in the turbo clock frequency of today's CPUs being up to 50 % higher than their base frequency.

From a microarchitect's point of view, improving energy efficiency generally means simplifying the microarchitecture without hurting performance. The microarchitect's quest for energy efficiency generally entails many incremental improvements in various parts of the microarchitecture, as no single part is responsible for more than a fraction of the whole CPU energy consumption. Nevertheless, some parts of the microarchitecture are hotter than others because power density on the chip is not uniform. Improving energy-efficiency in regions of high power density is doubly rewarding as this is where hot spots are more likely to be located.

The physical integer register file (IRF) is one such region. The IRF of modern superscalar cores is read and written by multiple instructions almost every clock cycle. Moreover, the IRF has many ports, and the number of physical registers keeps increasing for exploiting more instruction-level parallelism. As a consequence, the IRF is among the most power-hungry parts of the microarchitecture.

We propose a dual-banking scheme to reduce the power consumption of the IRF: the odd bank holds odd-numbered physical registers, and the even bank holds even-numbered ones. Half of the read ports come from the odd bank, the other half from the even bank. This way the number of read ports per bank is halved, and the area and power of the IRF is roughly halved. Execution pipes with two read ports, such as ALUs, have one read port in the odd bank and the other read port in the even bank. If a 2-input micro-op happens to have its two source operands in the same bank, this is a bank conflict, and the micro-op reads its two operands sequentially. Bank conflicts hurt performance, as each conflict generates a one-cycle penalty. To minimize the performance loss, we propose a new register renaming scheme that allocates physical registers so as to reduce the number of bank conflicts. Our simulations show that, with this new scheme, very little performance is lost from banking the IRF. We are currently writing a paper describing and evaluating this idea.

6.2.4 Thread convergence prediction for general-purpose SIMT architectures

Participants: Lily Blanleuil, Caroline Collange

Thread divergence optimization in GPU architectures have long been hindered by restrictive control-flow mechanisms based on stacks of execution masks. However, GPU architectures recently began implementing more flexible hardware mechanisms, presumably based on path tables. We leverage this opportunity by proposing a hardware implementation of iteration shifting, a divergence optimization that enables lockstep execution across arbitrary iterations of a loop [24]. Although software implementations of iteration shifting have been previously proposed, implementing this scheduling technique in hardware lets us leverage dynamic information such as divergence patterns and memory stalls. Evaluation using simulation suggests that the expected performance improvements will remain modest or even nonexistent unless the organization of the memory access path is also revisited.

6.3 WCET estimation and optimization

Participants: Abderaouf Nassim Amalou, Valentin Pasquale, Isabelle Puaut

6.3.1 Revisiting iterative compilation for WCET minimization

Participants: Valentin Pasquale, Isabelle Puaut

Static Worst-Case Execution Time (WCET) estimation techniques take as input the binary code of a program, and output a conservative estimation of its execution time. While compilers and iterative compilation usually optimize for the average case, it is possible to use existing optimization and techniques to drastically lower the WCET estimates. In this work, we demonstrate that the use of a large number of compilation options allows a significant reduction of WCET estimates (up to 70 % on some benchmarks)

compared to the best compilation level applicable. These gains are far better than the state-of-art [30], which, on the same benchmarks, reduce the WCET estimation by 21 % on average.

6.3.2 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 available. We propose [18] a novel hybrid WCET estimation technique, WE-HML, 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, WE-HML (i) operates on binary code for improved precision of learning, as compared to the related techniques operating at source code or intermediate code level; (ii) trains the ML algorithms on a large set of automatically generated programs for improved quality of learning; (iii) proposes a technique to take into account data caches. Experiments on an Arm Cortex-A53 processor show that for all benchmarks, WCET estimates obtained by WE-HML are larger than all possible execution times. Moreover, the cache modeling technique of WE-HML yields an improvement of 65 % on average of WCET estimates compared to its cache-agnostic equivalent.

6.4 Security

Participants: Nicolas Bellec, Maxime Desmarais, Antoine Gicquel, Damien Hardy, Camille Le Bon, Pierre-Yves Péneau, Isabelle Puaut, Erven Rohou

6.4.1 Verification of Data Flow Integrity for Real-Time Embedded Systems

Participants: Nicolas Bellec, Isabelle Puaut
External collaborators: CIDRE and TARAN teams

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.

Data-flow integrity (DFI) is a safety property that aims at preventing memory-corruption attacks by ensuring that the flow of data at runtime is consistent with the statically analyzed data-flow graph (DFG) of the program. Such protection is implemented by analyzing the program to retrieve its data-flow graph (at compilation time) and then instrumenting memory operations so they cannot perform illicit operations (according to the obtained data-flow graph). DFI protects against a wide range of memory-corruption attacks from classic Return-Oriented Programming (ROP) to more subtle non-control data attacks that aim at modifying security critical data of the program to modify the behavior of the program while maintaining a correct control-flow.

In this work, we propose a compiler technique that enforces DFI for RTES. The proposed technique, in contrast to the state of the art, optimizes the worst-case execution times of DFI-protected programs, by iteratively applying optimizations along the worst-case execution path.

6.4.2 Multi-nop fault injection attack

Participants: Maxime Desmarais, Antoine Gicquel, Damien Hardy, Pierre-Yves Péneau, Erven Rohou
External collaborators: CIDRE team.

Fault injection is a well-known method to physically attack embedded systems, microcontrollers in particular. It aims to find and exploit vulnerabilities in the hardware to induce malfunction in the software and eventually bypass software security or retrieve sensitive information. We propose [19] a cheap (in the order of $100) platform called TRAITOR inducing faults with clock glitches with the capacity to inject numerous and precise bursts of faults. From an evaluation point of view, this platform allows easier and cheaper investigations over complex attacks than costly EMI benches or laser probes.

See also the TRAITOR tool suite in Section 5.1.5.

6.4.3 Compiler-based automation of side-channel countermeasures

Participants: Damien Hardy, Pierre-Yves Péneau, Erven Rohou
External collaborators: Nicolas Kiss from SED, Olivier Zendra from DiverSE, Annelie Heuser from EMSEC.

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. Practical side-channel analyses are being performed to assess the security and the performance of the resulting code.

6.4.4 Platform for adaptive dynamic protection of programs

Participants: Camille Le Bon, Erven Rohou
External collaborators: Guillaume Hiet and Frédéric Tronel, from the CIDRE team

Memory corruption attacks have been a major issue in software security for over two decades and are still one of the most dangerous and widespread types of attacks nowadays. Among these attacks, control-flow hijack attacks are the most popular and powerful, enabling the attacker to execute arbitrary code inside the target process. Many approaches have been developed to mitigate such attacks and to prevent them from happening. One of these approaches is the Control-Data Isolation (CDI) that tries to prevent such attacks by removing their trigger from the code, namely indirect branches. This approach has been previously implemented as a compiler pass that replaces every indirect branches in the program with a table that leads the control-flow to direct hard-written branches. The drawback of this approach is that it needs the recompilation of the program. We present an approach and its implementation, DAMAS [20] (see Sections 5.1.9 for Damas and 5.1.10 for the underlying library Sorry), a framework capable of deploying protections on a running software and use runtime information to optimize them during the process execution. We implemented a coarse-grain CDI protection using our framework and evaluated its impact on performance.

7 Bilateral contracts and grants with industry

Participants: Pierre Michaud.

7.1 Bilateral contracts with industry

Ampere Computing:

- Duration: 2021-2022
- Local coordinator: Pierre Michaud
- Collaboration between the PACAP team and Ampere Computing on features of the microarchitecture of next generation CPUs.

8 Partnerships and cooperations

Participants: Lily Blanleuil, Caroline Collange, Antoine Gicquel, Damien Hardy, Sara Hoseininasab, Camille Le Bon, Pierre Michaud, Isabelle Puaut, Hugo Reymond, Erven Rohou, Bahram Yarahmadi.

8.1 European initiatives

8.1.1 Other European programs/initiatives

HPCQS: High Performance Computer and Quantum Simulator hybrid

- Funding: EuroHPC - European High-Performance Computing Joint Undertaking
- Duration: 2021-2025
- Local coordinator: Caroline Collange
- The aim of HPCQS is to prepare European research, industry and society for the use and federal operation of quantum computers and simulatiors. These are future computing technologies that are promising to overcome the most difficult computational challenges. HPCQS is developing the programming platform for the quantum simulator, which is based on the European ATOS Quantum Learning Machine (QLM), and the deep, low-latency integration into modular HPC systems based on ParTec's European modular supercomputing concept. HPCQS develops the connection between the classical supercomputer and the quantum simulator by deep integration in the modular supercomputing architecture and will provide cloud access and middleware for programming and execution of applications on the quantum simulator through the QLM, as well as a Jupyter-Hub platform with safe access guarantee through the European UNICORE system to its ecosystem of quantum programming facilities and application libraries.
- website: www.hpcqs.eu

CERCICAS: Connecting Education and Research Communities for an Innovative Resource Aware Society

- Type of action: COST Action
- Duration: 2020-2024
- Local coordinator: Isabelle Puaut
- Parallel computing platforms have revolutionized the hardware landscape by providing high-performance, low-energy, and specialized (viz. heterogeneous) processing capabilities to a variety of application domains, including mobile, embedded, data-center and high-performance computing. However, to leverage their potential, system designers must strike a difficult balance in the apportionment of resources to the application components, striving to avoid under- or over-provisions against worst-case utilization profiles. The entanglement of hardware components in the emerging platforms and the complex behavior of parallel applications raise conflicting resource requirements, more so in smart, (self-)adaptive and autonomous systems. This scenario presents the hard challenge of understanding and controlling, statically and dynamically, the trade-offs in the usage of system resources, (time, space, energy, and data), also from the perspective of the development and maintenance efforts.

  Making resource-usage trade-offs at specification, design, implementation, and run time requires profound awareness of the local and global impact caused by parallel threads of applications on individual resources. Such awareness is crucial for academic researchers and industrial practitioners across all European and COST member countries, and, therefore, a strategic priority. Reaching this goal requires acting at two levels: (1) networking otherwise fragmented research efforts towards more holistic views of the problem and the solution; (2) leveraging appropriate educational and technology assets to improve the understanding and management of resources by the academia and industry of underperforming economies, in order to promote cooperation inside Europe and achieve economical and societal benefits.

- website: www.cost.eu/actions/CA19135/

8.2 National initiatives

ZEP: Zero Power Computing Systems

- Funding: Inria Project Lab
- Duration: 2017-2021
- Local coordinator: Erven Rohou
- Participants: Bahram Yarahmadi
- ZEP 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 is 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 are designed. On the one hand, a compilation pass computes a worst-case energy consumption. On the other hand, dedicated runtime mechanisms 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 main application target is Internet of Things (IoT).
- Partners: CAIRN (TARAN), CORSE, SOCRATE, CEA Lialp and Lisan laboratories of CEA LETI & LIST

EQIP: Engineering for Quantum Information Processors

- Funding: Inria Challenge project
- Duration: 2021-2024
- Local coordinator: Caroline Collange
- Partners: COSMIQ, CAGE, CASCADE, DEDUCTEAM, GRACE, HIEPACS, MATHERIALS, MOCQUA, PACAP, PARSYS, QUANTIC, STORM, and ATOS Quantum
- 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.
- website: project.inria.fr/eqip

ARMOUR: Dynamic Binary Optimization Cyber-security

- Funding: DGA (*Direction Générale de l'Armement*) and PEC (*Pôle d'Excellence Cyber*)
- Duration: 2018-2021
- Local coordinator: Erven Rohou
- Participants: Camille Le Bon

- ARMOUR aims at improving the security of computing systems at the software level. Our contribution is 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 studied 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.

DYVE: Dynamic vectorization for heterogeneous multi-core processors with single instruction set

- Funding: ANR, JCJC
- Duration: 2020-2023
- Local coordinator: Caroline Collange
- Participants: Lily Blanleuil, Sara Hoseininasab
- 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 aims to 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.

NOP: Safe and Efficient Intermittent Computing for a Batteryless IoT

- Funding: LabEx CominLabs (50 %)
- Duration: 2021-2024
- Local coordinator: Erven Rohou
- Participants: Isabelle Puaut, Hugo Reymond
- Partners: IRISA/Granit Lannion, LS2N/STR Nantes, IETR/Syscom Nantes
- Intermittent computing is an emerging paradigm for batteryless IoT nodes powered by harvesting ambient energy. It intends to provide transparent support for power losses so that complex computations can be distributed over several power cycles. It aims at significantly increasing the complexity of software running on these nodes, and thus at reducing the volume of outgoing data, which improves the overall energy efficiency of the whole processing chain, reduces reaction latencies, and, by limiting data movements, preserves anonymity and privacy.

  NOP aims at improving the efficiency and usability of intermittent computing, based on consolidated theoretical foundations and a detailed understanding of energy flows within systems. For this, it brings together specialists in system architecture, energy-harvesting IoT systems, compilation, and real-time computing, to address the following scientific challenges:

  1. develop sound formal foundations for intermittent systems,
  2. develop precise predictive energy models of a whole node (including both harvesting and consumption) usable for online decision making,

3. significantly improve the energy efficiency of run-time support for intermittency,
4. develop techniques to provide formal guarantee through static analysis of the systems behavior (forward progress),
5. develop a proof of concept: an intermittent system for bird recognition by their songs, to assess the costs and benefits of the proposed solutions.

- website: project.inria.fr/nopcl/

Maplurinum (Machinæ pluribus unum): (make) one machine out of many

- Funding: ANR, PRC
- Duration: 2021-2024
- Local coordinator: Pierre Michaud
- Participants: Erven Rohou
- Partners: Télécom Sud Paris/PDS, CEA List, Université Grenoble Alpes/TIMA
- Cloud and high-performance architectures are increasingly heteregenous and often incorporate specialized hardware. We have first seen the generalization of GPUs in the most powerful machines, followed a few years later by the introduction of FPGAs. More recently we have seen nascence of many other accelerators such as tensor processor units (TPUs) for DNNs or variable precision FPUs. Recent hardware manufacturing trends make it very likely that specialization will not only persist, but increase in future supercomputers. Because manually managing this heterogeneity in each application is complex and not maintainable, we propose in this project to revisit how we design both hardware and operating systems in order to better hide the heterogeneity to supercomputer users.
- website: project.inria.fr/maplurinum/

Ofast3D

- Funding: Inria Exploratory Action
- Duration: 2021-2024
- Local coordinator: Damien Hardy
- Participants: Erven Rohou
- Partners: MimeTIC (Rennes) and MFX (Nancy)
- The goal of Ofast3D is to increase the production capacity of fused deposition modeling 3D printing, without requiring any modification of existing production infrastructures. Ofast3D aims to reduce printing time without impacting the print quality by optimizing the code interpreted by 3D printers during its generation by taking into account the geometry of 3D models. Ofast3D is complementary to methods aiming either at improving printers or at optimizing 3D models.
- website: project.inria.fr/ofast3d

8.3 Regional initiatives

PluriNOP

- Funding: Région Bretagne (43 %), EUR CyberSchool (50 %)
- Duration: 2021-2024
- Local coordinator: Erven Rohou

- Participants: Damien Hardy, Antoine Gicquel
- Partners: Sorbonne Université
- In a world where computer systems control large parts of our societies and lives on a daily basis, the stakes of computer security are high. Many types of attacks exist, we focus here in fault-based attacks on embedded systems. These attacks are becoming a threat to systems that were previously spared: certain injection methods are now available to a large community and software means of fault injection are being developed. The literature mainly deals with a single fault, but more and more works refer to the possibility of injecting multiple faults. The objectives of PluriNOP are:

  1. to propose an automatic approach, based on static analysis, to determine possible exploits for an attacker model and a target, and then perform them (in simulation or experimentation) on a binary code, in order to reach a given objective (write a data to a memory location, call a function with given function with given parameters, extract a secret...);
  2. propose a method for quantifying the level of vulnerability of a binary code, for example on the basis of the minimum number of faults necessary for an exploit to be performed, the nature of the faults, etc.;
  3. propose countermeasures to these attacks, software or hardware, and an automation of their deployment.

9 Dissemination

9.1 Promoting scientific activities

9.1.1 Scientific events: organisation

Member of the organizing committees

- Caroline Collange was in the organizing committee of the Technoférence national event of *Pôle Image & Réseau* on quantum computing.
- Caroline Collange was in the steering committee of the Compas conference.
- Isabelle Puaut is member of the Advisory board of the Euromicro Conference on Real Time Systems (ECRTS).
- Isabelle Puaut is member of the steering committee of the international Conference on Real-Time Networks and Systems (RTNS).

9.1.2 Scientific events: selection

Member of the conference program committees

- Pierre-Yves Péneau was a member of the artifact evaluation committee of 30th ACM SIGPLAN International Conference on Compiler Construction (CC-2021).
- Pierre Michaud was a member of the program committees of the ISCA 2021 and HPCA 2022 conferences.
- Caroline Collange was a member of the program committees of Supercomputing (SC) 2021, Compiler Construction (CC) 2021, High Performance Computer Architectures (HPCA) 2022, Design Automation and Test in Europe (DATE) 2021, as well as the external review committees of ISCA 2021 and 2022.
- Isabelle Puaut was member of the program committee of the following conferences in 2021: Euromicro Conference on Real Time Systems (ECRTS), IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Conference on Real-Time Networks and Systems (RTNS).

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

9.1.3 Journal

Member of the editorial boards

- Isabelle Puaut is associate editor of the Springer International Journal of Time-Critical Computing Systems (RTSJ).

Reviewer - reviewing activities: Members of PACAP routinely review submissions to international journals.

9.1.4 Invited talks

- Pierre Michaud was invited to give a presentation of his ACM TACO 2016 paper on optimal cache replacement at the Workshop on Principles of Memory Hierarchy Optimization collocated with the PPoPP 2021 conference.
- Caroline Collange gave an invited talk on out-of-order SIMT architectures at ENS Lyon on Nov 2021.

9.1.5 Leadership within the scientific community

- Isabelle Puaut is a member of the Technical Committee on Real-Time Systems (TCRTS) of the IEEE Computer Society, conference planning sub-committee. She participated in 2021 to the selection of Test-of-Time papers.

9.1.6 Scientific expertise

- Isabelle Puaut was a member of Paul Caspi PhD thesis prize 2021, awarded by the ACM Special Interest Group on Embedded Systems (ACM SIGBED).
- Caroline Collange was a member of the Gilles Kahn PhD Award committee in 2021.
- Caroline Collange is a co-author of the *référentiel cyber*.

9.1.7 Research administration

- Erven Rohou is the contact for international relations for Inria Rennes Bretagne Atlantique (for scientific matters).
- 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”.
- Caroline Collange is a member of the Inria Rennes information technology users committee (CUMIR).

9.2 Teaching - Supervision - Juries

9.2.1 Teaching

- Master: N. Bellec, ITR (Informatique Temps Réel), 16h, M1, Université Rennes 1, France
- Master: L. 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, 95 hours, L3, Université de Rennes 1, France

- Master: D. Hardy, Operating systems, 59 hours, M1, Université de Rennes 1, France
- Master: D. Hardy, Students project, 8 hours, M1, Université de Rennes 1, France
- Master: C. 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

9.2.2 Supervision

- PhD: Daniel Rodrigues Carvalho, *Towards Compression At All Levels In The Memory Hierarchy* [25], Université de Rennes 1, Apr 2021, advisor A. Seznec
- PhD: Bahram Yarahmadi, *Static and dynamic compiler support for intermittently powered computer systems* [26], Jul 2021, advisor E. Rohou
- 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: Lily Blanleuil, *Thread convergence prediction for SIMT architectures*, Université de Rennes 1, started Oct 2018, advisor C. Collange and A. Seznec
- PhD in progress : Camille 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: Anis Peysieux, *Towards simple and highly efficient execution cores*, started Jan 2020, advisor since Jan 2021: Pierre Michaud (André Seznec was advisor from Jan 2020 until Dec 2020)
- PhD in progress: Abderaouf Nassim Amalou, *Security in real-time embedded systems*, started Oct 2020, advisors Isabelle Puaut (75 %), Elisa Fromont (25 %, LACODAM)
- PhD in progress: Hugo Reymond, *Energy-aware execution model in intermittent systems*, started Oct 2021, avdisors Isabelle Puaut, Erven Rohou, Sébastien Faucou (LS2N Nantes), Jean-Luc Béchennec (LS2N Nantes)
- PhD in progress: Antoine Gicquel, *Étude de vulnérabilité d'un programme au format binaire en présence de fautes précises et nombreuses : métriques et contremesures*, started Sep 2021, advisors Damien Hardy, Erven Rohou, Karine Heydemann (Sorbonne Université)
- PhD in progress: Sara Hoseininasab, *Automatic synthesis of multi-thread pipelines*, started Nov 2021, advisors C. Collange (70 %) and Steven Derrien (30 %, TARAN)

9.2.3 Juries

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

- Guillaume Devic, *Étude d'architectures dédiées aux systèmes embarqués intelligents et efficaces en énergie*, Université de Montpellier, Dec 2021 (reviewer)
- Étienne Louboutin, *Sensibilité de logiciels au détournement de flot de contrôle*, IMT Atlantique, Brest, Jan 2021 (examiner)
- Son Tuan Vu, *Optimizing Property-Preserving Compilation*, Sorbonne University, Apr 2021 (reviewer)
- Tiago Trevisan Jost, *Compilation and optimizations for Variable Precision Floating-Point Arithmetic*, Université de Grenoble, Jun 2021 (reviewer)

Isabelle Puaut was member of the following habilitation and PhD thesis committees:

- Quentin Dufour (PhD), *High-throughput real-time onion networks to protect everyone's privacy*, Université de Rennes 1, Feb 2021 (examiner, president of the committee)
- Marc Boyer (HdR), *Garantir les temps de réponse des réseaux embarqués à l'aide du calcul réseau (Guaranteing response times in embedded networks using network calculus)*, Habilitation à diriger des recherches, Université de Toulouse, Mar 2021 (reviewer)
- Gautier Berthou (PhD), *Operating system dedicated to NVRAM-based low power embedded systems*, Université de Lyon, Mar 2021 (examiner, president of the committee)
- Nathanael Sensfelder (PhD), *Analyse et contrôle des interférences liées à la cohérence de cache dans les multi-cœurs COTS (Analysis and control of interferences due to cache coherence in COTS multi-cores)*, Université de Toulouse, Mar 2021, (reviewer)
- Jean Guyomarc'h (PhD), *Analyse de systèmes temps-réels de sûreté et mitigation de leurs interférences temporelles (analysis of safety-critical systems and mitigation of their temporal interference)*, Université de Paris Saclay, Oct 2021 (examiner, president of the committee)
- Hugo Martin (PhD), *Machine learning for performance modelling on colossal software configuration spaces*, Dec 2021 (examiner, president of the committee)

Erven Rohou was a member of the CSID of Anis Peysieux, Guillaume Didier, Antoine Bernabeu, Quentin Ducasse, Davide Pala.

Isabelle Puaut is member of the CSID commitee of Zineb Boukili and Jean-Michel Gorius.

Caroline Collange was a member of the PhD thesis committee of Titouan Carette, *Wielding the ZX-Calculus*, LORIA, Nov 2021. She was a member of the CSID committee of Corentin Ferry and Louis Narmour.

Isabelle Puaut was member of the following hiring committees:

- Associate professor position, Université de Lille, on topic "systems, security and architecture"
- Professor position, Université de Rennes 1, on topic "software security"

Caroline Collange was member of the following hiring committee:

- Associate professor position, University Paris 6, on topic "architecture and systems"

9.3 Popularization

9.3.1 Internal or external Inria responsibilities

Caroline Collange is a member of the Inria Committee on Gender Equality and Equal Opportunities.

9.3.2 Education

Isabelle Puaut taught *Basics of computer architecture* a training of high school teachers as part of the opening of the new computer science option in the two final years before Baccalauréat, 6 hours.

10 Scientific production

10.1 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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] D. Sampaio, R. M. De Souza, C. Collange and F. M. Quintão Pereira. 'Divergence Analysis'. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 35.4 (Nov. 2013), 13:1–13:36. DOI: 10.1145/2523815. URL: https://hal.inria.fr/hal-00909072.

[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] M. Y. Siraichi, V. F. d. Santos, C. Collange and F. M. Quintão Pereira. 'Qubit allocation as a combination of subgraph isomorphism and token swapping'. In: OOPSLA. Vol. 3. Athens, Greece, 10th Oct. 2019, pp. 1–29. DOI: 10.1145/3360546. URL: https://hal.inria.fr/hal-02316820.

[15] 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.

10.2 Publications of the year

International journals

[16] K. Kalaitzidis and A. Seznec. 'Leveraging Value Equality Prediction for Value Speculation'. In: *ACM Transactions on Architecture and Code Optimization* 18.1 (21st Jan. 2021), pp. 1–20. DOI: 10.1145/3436821. URL: https://hal.inria.fr/hal-03097413.

[17] D. Rodrigues Carvalho and A. Seznec. 'Understanding Cache Compression'. In: *ACM Transactions on Architecture and Code Optimization* 18.3 (June 2021), pp. 1–27. DOI: 10.1145/3457207. URL: https://hal.inria.fr/hal-03285041.

International peer-reviewed conferences

[18] A. N. Amalou, I. Puaut and G. Muller. 'WE-HML: hybrid WCET estimation using machine learning for architectures with caches'. In: RTCSA 2021 - 27th IEEE International Conference on Embedded Real-Time Computing Systems and Applications. Online Virtual Conference, France: IEEE, 18th Aug. 2021, pp. 1–10. URL: https://hal.inria.fr/hal-03280177.

[19] L. Claudepierre, P.-Y. Péneau, D. Hardy and E. Rohou. 'TRAITOR: A Low-Cost Evaluation Platform for Multifault Injection'. In: ASSS '21: Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems. Virtual Event Hong Kong, Hong Kong SAR China: ACM, 24th May 2021, pp. 51–56. DOI: 10.1145/3457340.3458303. URL: https://hal.inria.fr/hal-03266561.

[20] C. Le Bon, E. Rohou, F. Tronel and G. Hiet. 'DAMAS: Control-Data Isolation at Runtime through Dynamic Binary Modification'. In: SILM 2021 - Workshop on the Security of Software / Hardware Interfaces. 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). digital event, Austria, 6th Sept. 2021, pp. 86–95. DOI: 10.1109/EuroSPW54576.2021.00016. URL: https://hal.archives-ouvertes.fr/hal-03340008.

[21] D. Rodrigues Carvalho and A. Seznec. 'A Case for Partial Co-Allocation Constraints in Compressed Caches'. In: SAMOS XXI 2021 - International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation. Samos, Greece, 5th July 2021, pp. 1–13. URL: https://hal.inria.fr/hal-03284824.

[22] D. Rodrigues Carvalho and A. Seznec. 'Conciliating Speed and Efficiency on Cache Compressors'. In: ICCD 2021 - 39th IEEE International Conference on Computer Design. Virtual, United States, 24th Oct. 2021, pp. 1–5. URL: https://hal.inria.fr/hal-03354883.

[23] B. Yarahmadi and E. Rohou. 'So Far So Good: Self-Adaptive Dynamic Checkpointing for Intermittent Computation based on Self-Modifying Code'. In: SCOPES 2021 - 24th International Workshop on Software and Compilers for Embedded Systems. Eindhoven (virtual), Netherlands, 1st Nov. 2021, pp. 1–7. URL: https://hal.inria.fr/hal-03410647.

Conferences without proceedings

[24] L. Blanleuil and C. Collange. 'Scheduling paths leveraging dynamic information in SIMT architectures'. In: COMPAS 2021 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Lyon / Virtual, France, 6th July 2021, pp. 1–6. URL: https://hal.inria.fr/hal-03269966.

Doctoral dissertations and habilitation theses

[25] D. Rodrigues Carvalho. 'Towards Compression At All Levels In The Memory Hierarchy'. Université de Rennes 1, 9th Apr. 2021. URL: https://hal.inria.fr/tel-03454941.

[26] B. Yarahmadi. 'Static and dynamic compiler support for intermittently powered computer systems'. Université Rennes 1, 1st July 2021. URL: https://hal.inria.fr/tel-03280004.

Reports & preprints

[27] 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.

[28] E. Rohou. *Deliverable D2.3 - Illustration of system reconfiguration due to varying conditions: same-island, and migration*. Inria Rennes Bretagne Atlantique, 10th Sept. 2021, pp. 1–13. URL: https://hal.inria.fr/hal-03372263.

10.3 Cited publications

[29] 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.

[30] M. Dardaillon, S. Skalistis, I. Puaut and S. Derrien. 'Reconciling Compiler Optimizations and WCET Estimation Using Iterative Compilation'. In: *IEEE Real-Time Systems Symposium, RTSS 2019, Hong Kong, SAR, China, December 3-6, 2019*. IEEE, 2019, pp. 133–145. DOI: 10.1109/RTSS46320.2019.00022. URL: https://doi.org/10.1109/RTSS46320.2019.00022.

[31] S. Derrien, T. Marty, S. Rokicki and T. Yuki. 'Toward Speculative Loop Pipelining for High-Level Synthesis'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 4229–4239. DOI: 10.1109/TCAD.2020.3012866. URL: https://hal.archives-ouvertes.fr/hal-02949516.

[32] D. Hardy. *Ofast3D - Étude de faisabilité*. Technical Report RT-0511. Inria Rennes - Bretagne Atlantique ; IRISA, Dec. 2020, p. 18. URL: https://hal.inria.fr/hal-03093905.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.