RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

2025 *Activity Report*

RESEARCH CENTRE: **Inria Centre at Rennes University**

IN PARTNERSHIP WITH: **Université de Rennes**

Project-Team

PACAP

Pushing Architecture and Compilation for Application Performance

In collaboration with Institut de recherche en informatique et systèmes aléatoires (IRISA)

Project-Team PACAP

Creation of the Project-Team: 2016 July 01

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

Keywords

Computer sciences and digital sciences

A1.1.1. – Multicore, Manycore

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

A1.1.3. – Memory models

A1.1.8. – Security of architectures

A1.6. – Green Computing

A2.2.1. – Static analysis

A2.2.3. – Memory management

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

A4.4. – Security of equipment and software

A9.2. – Machine learning

Other research topics and application domains

B1. – Life sciences

B2. – Digital health

B3. – Environment and planet

B4. – Energy

B5. – Industry of the future

B6. – IT and telecom

B7. – Transport and logistics

B8. – Smart Cities and Territories

B9. – Society and Knowledge

Contents

1 Team members, visitors, external collaborators

Research Scientists

- Erven Rohou [Team leader, INRIA, Senior Researcher, HDR]
- Caroline Collange [INRIA, Researcher]
- Pierre Michaud [INRIA, Researcher]
- Thomas Rubiano [INRIA, Starting Research Position]

Faculty Members

- Damien Hardy [UNIV RENNES, Associate Professor]
- Isabelle Puaut [UNIV RENNES, Professor, HDR]

Post-Doctoral Fellows

- Xabier Legaspi Juanatey [UNIV RENNES, Post-Doctoral Fellow]
- Sébastien Michelland [INRIA, Post-Doctoral Fellow, from Nov 2025]

PhD Students

- Nicolas Bailluet [INRIA, from Sep 2025 until Nov 2025]
- Nicolas Bailluet [UNIV RENNES, until Aug 2025]
- Hector Chabot [UNIV RENNES]
- Niels Cobat [UNIV RENNES]
- Sara Sadat Hoseininasab [INRIA, until Feb 2025]
- Ariane Nicolas [INRIA, from Oct 2025]
- Aurore Poirier [INRIA]
- Matthieu Rodet [INRIA]

Technical Staff

- Pierre Bedell [UNIV RENNES, Engineer, from Oct 2025]
- Antoine Gicquel [INRIA, Engineer, from Apr 2025]
- Jean-Michel Gorius [INRIA, Engineer, from Oct 2025]
- Imane Lasri [INRIA, Engineer, until Feb 2025]
- Hugo Reymond [INRIA, Engineer, until Jun 2025]

Interns and Apprentices

- Maxime Desbans [INRIA, Intern, from May 2025 until Jun 2025]
- Vincent Michel [INRIA, Intern, from May 2025 until Aug 2025]

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 (before even trying to increase it) has become a non-trivial task which we also plan to tackle.

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

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

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

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

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

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

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

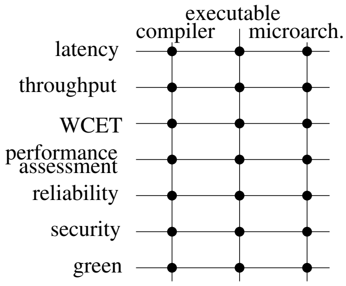


Figure 1: Perimeter of Research Objectives

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

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

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

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

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

Dealing with Attacks – Security. Computer systems are under constant attack, from young hackers trying to show their skills, to "professional" criminals stealing credit card information, and even government agencies with virtually unlimited resources. A vast amount of techniques have been proposed in the literature to circumvent attacks. Many of them cause significant slowdowns due to additional checks and countermeasures.

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

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

3 Research program

3.1 Motivation

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

3.1.1 Technological constraints

Until recently, binary compatibility guaranteed portability of programs, while increased clock frequency and improved micro-architecture provided increased performance. However, in the last decade, advances in technology and micro-architecture started translating into more parallelism instead. Technology roadmaps even predicted the feasibility of thousands of cores on a chip by the 2020's. 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 efficiently map 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 [34] compilation to continuously adapt software to its environment (Section 3.2.2). Dynamic binary optimization will also play a key role in delivering adapted 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

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

trade-offs. We study how programs can adequately exploit the specific features of the proposed heterogeneous compute nodes.

We propose to build upon iterative compilation [1] to explore how applications perform on different configurations. When possible, Pareto points are related to application characteristics. The best configuration, however, may actually depend on runtime information, such as input data, dynamic events, or properties that are available only at runtime. Unfortunately a runtime system has little time and means to determine the best configuration. For these reasons, we also leverage split-compilation [34]: 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. Just-in-Time (JIT) compilers can handle the situation to some extent, but binary deployment requires dynamic binary rewriting. Our work has shown how Single-Instruction Multiple-Data (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 software platforms [41, 38] to analyze and optimize programs at runtime, and we started working on automatic dynamic parallelization of sequential code, and dynamic specialization.

We addressed some of these challenges in previous 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 addressed these challenges from the energy perspective, while the ANR Continuum project and the Inria Challenge ZEP focused on opportunities brought by non-volatile memories. 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 25 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 [43] [39] and in cache prefetching [6] and they have revived the forgotten concept of value prediction [9][8]. This research was supported by the ERC Advanced grant DAL (2011-2016) and also by Intel. We pursue research on achieving ultimate unicore performance. Below are several non-orthogonal directions that we have identified for mid-term research:

1. management of the memory hierarchy (particularly the hardware prefetching);

2. practical design of very wide-issue execution cores;

3. speculative execution.

Memory design issues:

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

The *Data Prefetching Contest* held with ISCA 2015 has illustrated that achieving high prefetching efficiency is still a challenge for wide-issue superscalar processors, particularly those featuring a very large instruction window. The large instruction window enables an implicit data prefetcher. The interaction between this implicit hardware prefetcher and the explicit hardware prefetcher is still relatively mysterious as illustrated by Pierre Michaud's BO prefetcher (winner of DPC2) [6]. The first research objective is to better understand how the implicit prefetching enabled by the large instruction window interacts with the L2 prefetcher and then to understand how explicit prefetching on the L1 also interacts with the L2 prefetcher.

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

The prefetcher is not the only actor in the hierarchy that must be carefully controlled. Significant benefits can also be achieved through careful management of memory access bandwidth, particularly the management of spatial locality on memory accesses, both for reads and writes. The exploitation of this locality is traditionally handled in the memory controller. However, it could be better handled if larger temporal granularity was available. Finally, we also intend to continue to explore the promising avenue of compressed caches. In particular we proposed the skewed compressed cache [12]. It offers new possibilities for efficient compression schemes.

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

The first direction that we are exploring is the use of clustered architectures [7]. Symmetric clustered organization allows to benefit from a simpler bypass network, but induce large complexity on the issue queue. One remarkable finding of our study [7] is that, when considering two large clusters (e.g. 8-wide), steering large groups of consecutive instructions (e.g. 64 μops) to the same cluster is quite efficient. This opens opportunities to limit the complexity of the issue queues (monitoring fewer buses) and register files (fewer ports and physical registers) in the clusters, since not all results have to be forwarded to the other cluster.

The second direction that we are exploring is associated with the approach that we developed with Sembrant et al. [42]. 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 25 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 [44].

In speculative execution, we have recently revisited value prediction (VP) which was a hot research topic between 1996 and 2002. However it was considered until recently that value prediction would lead to a huge increase in complexity and power consumption in every stage of the pipeline. Fortunately, we have recently shown that complexity usually introduced by value prediction in the OoO engine can be overcome [9][8] [43] [39]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [9]. Thus, both prediction validation and recovery by squashing can be done outside the out-of-order engine, at commit time. Furthermore, we propose a new pipeline organization, EOLE ({Early | Out-of-order | Late} Execution), that leverages VP with validation at commit to execute many instructions outside the OoO core, in-order [8]. With EOLE, the issue-width in OoO core can be reduced without sacrificing performance, thus benefiting the performance of VP without a significant cost in silicon area and/or energy. In the near future, we will explore new avenues related to value prediction. These directions include register equality prediction and compatibility of value prediction with weak memory models in multiprocessors.

3.2.4 Towards heterogeneous single-ISA CPU-GPU architectures

Heterogeneous single-ISA architectures have been proposed in the literature during the 2000's [37] and are now widely used in the industry (Arm big.LITTLE, NVIDIA 4+1, Intel Alder Lake...) 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 [17] 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, Networks on Chip (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 researched new static and dynamic compilation techniques that fully exploit emerging memory and NoC technologies. Finally, in collaboration with the TARAN 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. We focus 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 [45] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [36], or thread-based control flow mangling [40]). Still, security is not a 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 countermeasures cannot take into account the current threat level. In collaboration with CIDRE, we proposed 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 developed a technique to detect such attacks thanks to a hardware monitor that makes sure that statically computed time information is preserved (TARAN 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 Social and environmental responsibility

5.1 Impact of research results

For a few years now, the PACAP team has been contributing to the transition from traditional IoT networks to battery-less networks. The increasing number of IoT devices led to a profileration of batteries in the environment, associated with their well-known ecological and social footprint.

In an effort to reduce this footprint, PACAP provides compiler building blocks to support intermittent computing, i.e. the execution of programs on battery-less devices, powered by energy harvesting. This supports allow the devices to endure frequent power failures.

This work has been presented and discussed in events on sustainable development such as an international conference [24] and a local event [26].

The team also makes contributions to extend the life of legacy computing systems by enabling the reverse-engineering and re-creation of obsolete components using reconfigurable circuits [25].

6 Highlights of the year

6.1 Awards

André Seznec received the 2025 ACM-IEEE CS Eckert-Mauchly Award. The award recognizes contributions to computer and digital systems architecture. According to the ACM, he “is recognized for his extensive impact on computing, most notably pioneering contributions to branch prediction and cache memories”.

7 Latest software developments, platforms, open data

7.1 Latest software developments

7.1.1 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

7.1.2 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: Damien Hardy, Isabelle Puaut, 4 anonymous participants

Partner: Université de Rennes 1

7.1.3 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 2.3.2, released December 2023. 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

7.1.4 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.

Contact: Erven Rohou

7.1.5 OptiPrint

Keywords: 3D printing, Planning, Optimization

Functional Description: OptiPrint is a software library dedicated to print time optimization for fused filament deposition (FDM) printers. This library is integrated to the Gato3D compiler. Its role is to allow the optimization of the printing time by reordering / filtering the G-code sent to a 3D printer. The optimization is fully configurable. It adapts to the characteristics of the printers (type of nozzle, speed of movement of the nozzle). It also allows to describe scheduling constraints allowing to make a compromise between printing quality and optimization.

Contact: Fabrice Lamarche

7.1.6 SAMVA

Keywords: Static analysis, Fault injection

Functional Description: SAMVA is a software package for determining attack paths in the context of precise, multiple fault injection attacks. It is a framework for efficiently searching vulnerabilities of applications in presence of multiple instruction-skip faults with various widths. SAMVA relies solely on static analysis to determine attack paths in a binary code. It is configurable with the fault injection capacity of the attacker and the attacker's objective

Contact: Erven Rohou

Participants: Antoine Gicquel, Erven Rohou, Damien Hardy

7.1.7 TimeKlip

Keywords: Simulator, 3D printing

Functional Description: 3D printing simulator calculating the printing time of a G-code file. It is able to give timing information for each instruction in the file. The simulator does not require a printer to run, only configuration files. It is also slicer agnostic.

The simulator takes the form of a module integrated into the Klipper firmware.

Contact: Damien Hardy

7.1.8 HARCOM

Name: Hardware Complexity Model

Keywords: Microarchitecture simulation, Transistor, Energy, Hardware complexity

Scientific Description: Research in processor microarchitecture is essentially based on simulation. Microarchitecture simulators evaluate mainly the performance of processors, not their hardware complexity. This allows a certain level of abstraction in simulators, which are generally written with general-purpose programming languages such as C++. These simulators are fast and easy to modify, two essential qualities for research in microarchitecture. Hardware complexity, however, is generally evaluated with CAD tools (RTL and hardware synthesis), which is too time consuming for research in microarchitecture. Yet, it is important that microarchitects be able to estimate the hardware complexity of the mechanisms they study. HARCOM fills this gap. HARCOM is a C++ library, compatible with microarchitecture simulators, allowing a fast functional simulation of microarchitectural mechanisms while providing directly an estimate of their hardware complexity.

Functional Description: C++ library for writing processor microarchitecture performance simulators, providing estimates of hardware complexity (silicon area, transistors, energy, latencies).

URL: `https://gitlab.inria.fr/pmichaud/harcom`

Contact: Pierre Michaud

Participant: Pierre Michaud

7.2 New platforms

7.2.1 Ofast3D

Participants: Pierre Bedell, Damien Hardy.

The objective of the Inria exploratory action Ofast3D was to optimize programs in G-code representations. As opposed to the more traditional programs PACAP considers (which run on general purpose computers), these programs run on 3D printers. Testing requires a 3D printing platform for research experiments, which is under construction. At this stage, it is composed of 11 printers and 4 test benches. This allows to evaluate optimizations and time prediction on different kinematics and configurations as well as different firmwares. Furthermore, air quality sensors are under deployment to evaluate the impact of 3D printing materials.

This platform is used by other teams in particular: ComBO, Rainbow, and MALT.

7.2.2 Arsene evaluation environment

Participants: Herinomena Andrianatrehina, Ronan Lashermes, Thomas Rubiano.

With TARAN team, in the context of ARSENE PEPR, an evaluation platform for RISCV new extension is developed and shared with other ARSENE members in a form of Inria Gitlab repositories and Nix derivations.

The platform can be described with the diagram shown in Figure 2.

It is composed of:

- LLVM custom for RISCV new extension;
- GCC toolchain custom for RISCV new extension;
- NaxRISCV with different implementations for new extension;
- Verilator custom to generate custom traces;

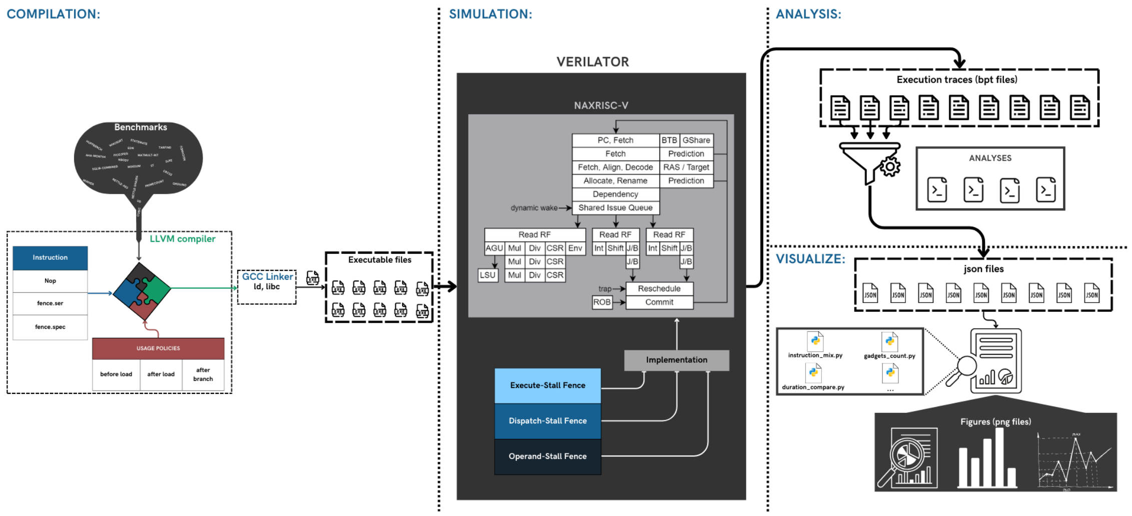


Figure 2: Arsene evaluation environment

- analyzer of traces;
- scripts to manage the platform and generate vizualisations.

7.2.3 Arsene “LLVM CSR” Secret Flag companion

Participants: Thomas Rubiano, Sébastien Michelland.

This tool is an other customized LLVM for manipulating secrets and communicating what values are secrets to the microarchitecture through a specific register class. It is composed of taint analysis, new register allocation and CSR insertion. This tool works within the environment described above. The TARAN team built a specific NaxRISCV core to work in tandem with this LLVM.

7.3 Open data

Digitalized material from the Bull company public archives

Contributors: Caroline Collange

Description We digitalized and made available online about 1000 pages of documentation about the CII Mitra, SEA CAB 500 and Bull Gamma 60 French computer architectures from the 1950s to the 1970s, from the Bull company collection, reference 2012.007, in *Archives Nationales du Monde du Travail* in Roubaix, France.

Dataset PID: DOI 10.34847/nkl.fc6e2857

Project link: https://nakala.fr/collection/10.34847/nkl.fc6e2857

8 New results

Participants: Nicolas Bailluet, Pierre Bedell, Hector Chabot, Niels Cobat, Caroline Collange, Antoine Gicquel, Damien Hardy, Sara Sadat Hoseininasab, Imane Lasri, Xabier Legaspi Juanatey, Pierre Michaud, Sébastien Michelland, Aurore Poirier, Isabelle Puaut, Hugo Reymond, Matthieu Rodet, Erven Rohou, Thomas Rubiano.

8.1 Compilation and Optimization

Participants: Pierre Bedell, Niels Cobat, Damien Hardy, Imane Lasri, Xabier Legaspi Juanatey, Aurore Poirier, Isabelle Puaut, Matthieu Rodet, Hugo Reymond, Erven Rohou.

8.1.1 Compilation for Intermittent Systems

Participants: Isabelle Puaut, Matthieu Rodet, Hugo Reymond, Erven Rohou
Context: ANR project OWL
External collaborators: Sébastien Faucou, Mikaël Briday, Jean-Luc Béchennec, LS2N Nantes

Battery-less embedded systems powered by energy harvesting eliminate the need for battery maintenance and enable their deployment in remote environments. However, their intermittent execution, disrupted by unpredictable power failures, complicates data processing. Solutions for intermittency management gravitate around one key technique: checkpointing volatile data before power failures, and retrieving data at system reboot. Moreover, since data transmission is a major source of energy consumption, performing computations directly on-device is essential. Initially used for simple tasks such as goods identifications, battery-less systems are now being applied to more energy-intensive tasks such as image recognition leveraging machine learning algorithms such as Convolutional Neural Networks (CNNs). We introduce Circadia [24], a checkpointing strategy dedicated to CNN inference in battery-less systems. By leveraging the structured dataflow and control flow of CNNs, Circadia strategically places checkpoints within the CNN code to ensure task termination, data consistency, and low energy consumption. By design, Circadia has a linear complexity relative to model size, a significant improvement over the closest state-of-the-art checkpointing method, which has cubic complexity. This enables Circadia to handle much larger CNNs. Experimental results, on both generated and state-of-the-art embedded CNNs, show that its checkpoint placement time is several orders of magnitude lower than existing approaches, while its energy consumption at runtime remains nearly identical.

Circadia has been made publically available as a conference artifact. It has been presented at a summer school poster session [33].

This study is part of the PhD work of Matthieu Rodet, who is co-supervized by Sébastien Faucou, Jean-Luc Béchennec and Mikaël Briday from LS2N.

8.1.2 Dynamic Binary Analysis and Optimization

Participants: Aurore Poirier, Erven Rohou
Context: Exploratory Action AoT.js
External collaborators: Manuel Serrano, SPLiTS team (Sophia)

Just-in-Time (JIT) compilers are able to specialize the code they generate according to a continuous profiling of the running programs. This gives them an advantage when compared to Ahead-of-Time (AoT) compilers that must choose the code to generate once for all. Is it possible to improve the performance of AoT compilers by adding Dynamic Binary Modification (DBM) to the executions? We added to the Hopc AoT JavaScript compiler a new optimization based on DBM to the inline cache, a classical optimization dynamic languages use to implement object property accesses efficiently. Reducing the number of memory accesses – as the new optimization does – does not shorten execution times on contemporary architectures. The DBM optimization we have implemented is fully operational on `x86_64` architectures. We have conducted several experiments to evaluate its impact on performance and to study the reasons of the lack of acceleration. This (negative) result [19] sheds new light on the best strategy to be used to implement dynamic languages. It tells that the old days where removing instructions or removing memory reads always yielded speedups is over. Nowadays, implementing sophisticated compiler optimizations is only worth the effort if the processor is not able by itself to accelerate the code. This result applies to AoT compilers as well as JIT compilers.

8.1.3 3D printing time estimation and optimization

Participants: Pierre Bedell, Niels Cobat, Damien Hardy, Imane Lasri, Xabier Legaspi Juanatey
Context: Inria Exploratory Action Ofast3D, SCI3D
External collaborators: ComBo, MALT and MFX (Nancy) teams.

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 [35].

To reach that goal, we have modified the existing 3D printer firmware Klipper in simulation mode to determine the timing per G-code instruction (the language interpreted by 3D printers) as well as the trapezoid time and speed information. This extension named TimeKlip (cf. Section 7.1.7) is printer- and slicer-agnostic. We conduct an extensive study to highlight the precision and versatility of our simulator on 3D printers with different kinematics, using different slicers. We show that our simulator can be up to 2000 times faster than an actual print. Its average error, without requiring any calibration, is 0.04 % on a total of 66 printed models representing more than 133 hours of print. A data set based on TimeKlip is under construction to study the applicability of machine learning models to predict accurately the print duration of 3D models.

Concerning G-code optimization, we have developed OptiPrint (cf. Section 7.1.5) in collaboration with ComBo team. It is an optimizer focusing on trajectories to reduce air-time and retract. Our experiments show that the printing time can be reduced by 13 % on average and up to 25 % depending on the 3D model geometry. Another optimization accounting for the 3D printer kinematics is under evaluation. The first results show that it can reduce the print time by 10 % on average and up to 18 % depending on the 3D model.

See also GATO3D (Section 7.1.4).

8.1.4 Compilation Challenges Related to the Aging of Computing Systems

Participants: Erven Rohou

Extending the lifetime of High-Performance Computing (HPC) machines is becoming an important concern for a variety of reasons. These include the environmental and human costs associated with chip manufacturing, the rising demands by AI workloads, the soaring prices of accelerator chips, political blocks, and delays in the delivery of next-generation supercomputers. We advocate that traditional HPC paradigm must be reconsidered and we propose to explore new strategies for making existing HPC infrastructure viable for longer periods. In collaboration with TARAN and KERDATA, we started studying [30] the current barriers related to prolonging HPC machines lifespan and, in particular, we discuss key technical and operational challenges related to compilation techniques.

8.2 Processor Architecture

Participants: Caroline Collange, Erven Rohou, Sara Sadat Hoseininasab, Pierre Michaud.

8.2.1 Hardware complexity model for microarchitecture exploration

Participants: Pierre Michaud
Context: collaboration with Ampere Computing

Microarchitecture exploration is generally conducted with performance simulators written in general-purpose programming languages, often C++. A performance simulator does not need to simulate all the details of the hardware implementation. It is often sufficient to simulate the events that can impact performance significantly, such as cache misses, branch mispredictions, data dependences, etc. Performance simulators often use approximations and abstractions. This is what allows them to simulate the execution of many instructions in a short amount of time, which is important for estimating millisecond-scale performance and for design space exploration.

In general, microarchitects try to simulate realistic mechanisms. However, assessing the hardware complexity of a mechanism which only exists as a piece of C++ code in a performance simulator can be difficult. Hardware complexity is a multidimensional quantity including silicon area, energy consumption and delay. A simple, oft-used estimate of hardware complexity is the amount of storage used by a mechanism. Nevertheless, there is more to hardware complexity than storage. For instance, the delay of a branch predictor depends not only on its storage but also on the logic circuits processing the stored information. On the one hand, some hardware complexity models are available for microarchitecture research, such as CACTI and McPAT. However, their applicability is limited to cache-like structures (CACTI) or fixed microarchitectures (McPAT). On the other hand, electronic design automation tools can be used to implement the hardware. However, this requires too much time and effort for microarchitecture exploration.

We have developed a C++ library, called HARCOM, for estimating approximately the hardware complexity of microarchitectural parts, such as caches, branch predictors, hardware prefetcher, etc. [27] HARCOM is compatible with existing performance simulators that are written in C++ (gem5, ChampSim, ...). HARCOM tries to find a useful middle ground between several contradictory objectives: the accuracy of the hardware complexity model, simulation speed, flexibility and ease of use. The microarchitectural part under study is modeled with HARCOM values instead of C++ integers. HARCOM simulates the functional behavior and, simultaneously, provides estimates of the silicon area, number of transistors, dissipated energy and circuits delays.

8.2.2 Automatic synthesis of multi-thread pipelines

Participants: Sara Sadat Hoseininasab, Caroline Collange, Erven Rohou
Context: ANR Project DYVE
External collaborator: Steven Derrien, TARAN team.

Register-Transfer Level (RTL) design has been a traditional approach in hardware design for several decades. However, with the growing complexity of designs and the need for fast time-to-market, the design and verification process at the RTL level can become impractical. This has motivated for raising the abstraction level in hardware design. High-Level Synthesis (HLS) provides higher-level abstraction by automatically transforming a behavioral specification of a circuit into a low-level RTL, making it easier to design, simulate and verify complex digital systems. HLS relies on statically scheduled data paths which can limit its effectiveness. This limitation makes it difficult to design the micro-architectural features of processors from an Instruction Set Architecture described in high-level languages.

The PhD of Sara Sadat Hoseininasab, defended in February 2025, has demonstrated how the available features of HLS can be deployed in designing various pipelined processors micro-architecture. The approach takes advantage of the capabilities of HLS and employs multi-threading and dynamic scheduling techniques to overcome the limitation of HLS in pipelining a processor from an Instruction Set Simulator written in C. [29]

8.2.3 Reverse-engineering historical and legacy computer circuits

Participants: Caroline Collange
Context: CNRS INS2I project JuraSTIC
In order to re-create and repair computer systems from the 1970s and 1980s, we propose a hardware and software tooling named *Méduse* to assist in the reverse-engineering and replication of printed circuit boards implementing digital logic. From series of multiple electric continuity measurements between points in the circuit, *Méduse* produces a netlist that can be exported as Verilog code for analysis, simulation or synthesis on FPGA. Its use is illustrated with the reverse-engineering of several boards of a Mitra 125 mini-computer from 1978 [25].

8.3 WCET estimation and optimization

Participants: Hector Chabot, Isabelle Puaut.

8.3.1 Using machine learning for timing analysis of complex processors

Participants: Isabelle Puaut
External collaborators: Abderaouf Nassim Amalou, LS2N, Nantes

Real-time and energy-constrained systems rely heavily on accurate estimates of worst-case execution time (WCET) and worst-case energy consumption (WCEC) to ensure trustworthy operation. Designing architecture-specific analytical models for execution time and energy is often challenging and time-consuming. When such analytical models are unavailable or incomplete, machine learning (ML) techniques emerge as a promising alternative for building WCET/WCEC models.

Primarily in the context of the PhD thesis of Abderaouf Nassim Amalou, defended in 2023, we have conducted a series of research efforts investigating the use of ML to predict WCET and WCEC for small code snippets on single-core platforms. We summarize this body of work [18], highlight the key observations derived from our studies, and advocate for further exploration of this research direction.

8.3.2 Static estimation of memory access profiles for real-time multi-core systems

Participants: Hector Chabot, Isabelle Puaut
External collaborators: Hugues Cassé, Thomas Carle, IRIT Toulouse

In multi-core systems, shared-resource usage leads to *interference* between tasks running on parallel cores, resulting in additional delays in the execution time of tasks. Schedulability analysis techniques rely on *Interference-Aware WCET* of tasks (IA-WCET, WCET integrating delays resulting from interference) to safely consider these delays. Calculation of IA-WCET requires knowledge about the worst-case shared-resource usage of tasks, in the form of a *memory access profile* as far as shared memory accesses are concerned.

State-of-the-art memory profiles only provide coarse-grain information (at the level of an entire task), resulting in pessimism in IA-WCET computation. More recent solutions propose to refine the information available in memory profiles, but are still limited: they lack information about shared-resource usage of code inside loops and are unable to use contextual information, which leads to over-approximation. Recently we proposed Marmot, a technique that extends recent memory access profile extraction solutions for real-time software. In Marmot, tasks are split in successive *intervals*, with the worst-case resource usage of each interval described as a *distribution* instead of a single value. Our current work investigates the extent to which these profiles improve off-line schedules, in term of makespan and/or total amount of interference.

This work is part of the PhD thesis of Hector Chabot, who is co-supervized by Hugues Cassé and Thomas Carle from IRIT, Toulouse. Work is funded by the ANR project CAOTIC.

8.3.3 Estimation of interference delays in real-time multi-core systems

Participant: Isabelle Puaut

Identifying interference delays when using multi-core architectures in real-time systems requires knownledge on the shared resources (bus, memory controller, interconnect), which might not be available due to intellectual property constraints or complex hardware. This study, as a follow-up to our work on ML for timing analysis for single-core systems, aims at using AI for quantification of interference.

This work is done in collaboration with Thomas Carle from IRIT, Toulouse within the AIxIA project.

8.3.4 Design of predictable processors using High-Level Synthesis (HLS)

Participants: Isabelle Puaut
External collaborators: Thomas Feuilletin, Dylan Léothaud, Simon Rokicki (Inria, TARAN group), Steven Derrien (Université de Bretagne Occidentale)

This direction of research is part of the ANR project LOTR, aiming at designing processors that are area efficient [23], secure and predictable, all this using High-Level Synthesis (HLS).

Regarding timing predictability, real-time, domain-specific processors require faithful timing models for WCET analysis. However, existing models are typically hand-crafted from sparse documentation, making them error-prone and difficult to maintain. Our work [22] aims to automatically extract WCET timing models from single-issue in-order processor pipelines generated by High-Level Synthesis (HLS). By deriving timing models directly from the SpecHLS intermediate representation, the models are faithful by construction. Experimental results show that our timing-model extraction process generalizes across diverse RISC-V core variants and yields WCET estimates within 0.48 % on average of those from a handcrafted model, on the Mälardalen WCET benchmarks.

8.4 Security

Participants: Nicolas Bailluet, Antoine Gicquel, Damien Hardy, Sébastien Michelland, Isabelle Puaut, Erven Rohou, Thomas Rubiano.

8.4.1 Speculative fences as a countermeasure to Spectre-like attacks

Participants: Damien Hardy, Thomas Rubiano, Erven Rohou

External collaborators: TARAN team, SED.

Speculative execution poses significant security risks to modern out-of-order cores, exemplified by attacks such as Spectre. Numerous countermeasures, including selective speculation in both software and hardware, have been proposed. This approach allows enabling or disabling speculative behavior based on circumstances. However, challenges such as evolving attack methods and the complexity of simulating outof-order cores make these solutions difficult to reproduce and compare. We investigated [20] the use of RISC-V speculation fences to achieve selective speculation in a realistic scenario where the microarchitecture cannot distinguish between confidential and non-confidential data. We examine three aspects: the semantics of speculation fences (ranging from broad to selective constraints), the placement of fences in programs by compilers, and their hardware implementation in a modified NaxRiscv RISC-V out-of-order core. Using a new security metric, we compare configurations within a unified framework. Our findings highlight that speculative execution of load instructions is critical for out-of-order core performance. Furthermore, we demonstrate that selective speculation without confidentiality-tagged data fails to achieve a meaningful security-performance trade-off.

8.4.2 Multi-nop fault injection

Participants: Antoine Gicquel, Damien Hardy, Sébastien Michelland, Erven Rohou

External collaborators: TARAN team.

Multi-fault injections are powerful since they allow to bypass software security mechanisms of embedded devices. Assessing the vulnerability of an application while considering multiple faults with various effects is an open problem due to the size of the fault space to explore. We previously proposed SAMVA (see Section 7.1.6), a framework for efficiently searching vulnerabilities of applications in presence of multiple instruction-skip faults with various widths. SAMVA relies solely on static analysis to determine attack paths in a binary code.

However, these analyses did not take into account the physical constraints inherent in the realization of the faults inducing the models. As a result, the attack paths identified are not always feasible in practice for a given injection platform and target. We addressed this issue by proposing CHAPATI, a comprehensive approach comprising three main elements: 1) an extensible static analysis, based on SAMVA, capable of taking into account, during the attack path search phase, the attacker's capabilities as well as the specific conditions required to perform an instruction jump at ISA level; 2) the conversion of these attack paths into time parameters for fault injection; and 3) the automated execution of attacks using these parameters, combined with other injection parameters derived from a prior calibration of the fault injection bench. This work is currently under submission.

8.4.3 Gadget chains synthesis driven by SMT Solving for Code-Reuse Attacks

Participants: Nicolas Bailluet, Isabelle Puaut, Erven Rohou
External collaborators: Emmanuel Fleury, LaBRI Bordeaux.

Automating gadget chaining is a challenge that has attracted significant attention since the introduction of code-reuse attacks. Influenced by the primitives offered by stack-overflow vulnerabilities, several approaches were proposed that required the attacker to control the stack. Since then, most proposed approaches have had strong requirements on the capabilities of the attacker. However, during the last decade, a plethora of new attack primitives have emerged, e.g. use-after-free, heap-overflow, often breaking the requirements of existing approaches – e.g. controlling the stack.

This line of work aims at synthesizing code-reuse gadget chains that supports arbitrary exploitation primitives and layouts. In this work [21], we present ARCANIST, a technique, based on SMT solving and tainting, to chain gadgets for arbitrary exploitation primitives. We thoroughly compare the performance of our approach to the state-of-the-art. We show its ability to outperform its competitors by supporting intricate exploitation primitives and layouts that other approaches cannot. Especially, we demonstrate its real-world applicability by synthesizing gadget chains for ten real-world vulnerabilities with diverse exploitation primitives that competing tools struggle with. Among them is our case study (CVE-2022-46152) which targets a widely used trusted execution environment. We further developed an evaluation framework, based on SAT model counting, to prove whether a synthesized chain generated by ARCANIST, is valid across other contexts, and quantify the proportion of contexts in which it works.

These two studies were part of the PhD work of Nicolas Bailluet, who defended in November 2025 [28].

9 Bilateral contracts and grants with industry

Participants: Pierre Bedell, Damien Hardy, Imane Lasri, Xabier Legaspi Juanatey, Pierre Michaud, Erven Rohou.

9.1 Bilateral contracts with industry

Ampere Computing:

Participants: Pierre Michaud.

- Duration: 2025
- Local coordinator: Pierre Michaud
- Collaboration between the PACAP team and Ampere Computing on features of the microarchitecture of next generation CPUs.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Inria associate team not involved in an IIL or an international program

COLD

Participants: Aurore Poirier, Erven Rohou.

Title: Compilation and Optimization of Dynamic Programming Languages

Duration: 2024 – 2026

Coordinator: Erven Rohou

Partners:

- Université de Montréal, Montréal (Canada)

Inria contact: Erven Rohou

Summary: Dynamic programming languages offer flexibility and generally allow rapid software development. Programs written using dynamic languages are typically slower, consume more memory, and are less energy efficient. This is especially concerning, considering that dynamic languages such as Python and JavaScript are extensively used. JavaScript is the main language for implementing web applications, while Python is the most used language for software development today and in particular in the very active field of Machine Learning and Artificial Intelligence.

To improve the efficiency of Python implementations, the proposed COLD team will study optimizing compilation techniques for dynamic languages. These techniques will generate optimized code when translating a program from its source code to machine code. This provides better performance without having to sacrifice the flexibility of dynamic languages. Furthermore, since novel optimizing techniques can be integrated into existing compilers, they can improve current programs with no additional effort by the application programmers.

10.2 International research visitors

10.2.1 Visits of international scientists

Other international visits to the team

Joel Emer

Status Professor

Institution of origin: MIT

Country: USA

Dates: 26-28 May 2025

Context of the visit: Invited seminar on the occasion of the celebration of 50 years of IRISA

Mobility program/type of mobility: lecture

Moinuddin K. Qureshi

Status Professor

Institution of origin: Georgia Institute of Technology

Country: USA

Dates: 26-28 May 2025

Context of the visit: Invited seminar on the occasion of the celebration of 50 years of IRISA

Mobility program/type of mobility: lecture

10.3 National initiatives

ARSENE: Secure architectures for embedded digital systems (ARchitectures SEcurisées pour le Numérique Embarqué)

Participants: Damien Hardy, Erven Rohou, Thomas Rubiano.

- Funding: PEPR
- Duration: 2022-2027
- Local coordinator: Ronan Lashermes, Thomas Rubiano
- Partners: CNRS, Inria, CEA, UGA, IMT
- The security of communicating objects and the components they integrate is of growing importance in the cybersecurity arena. To address those challenges, the already-rich French research community in embedded systems security is joining forces within the ARSENE project in order to accelerate research & development in this field in a coordinated and structured way to achieve secure solutions. The main objectives of the project are to allow the French community to make significant advances in the field to strengthen the community's expertise and visibility on the international stage. The first part of the ARSENE project is on the study and implementation of two families of RISC-V processors: 32-bit RISC-V for low power secure circuits against physical attacks for IoT applications and 64-bit RISC-V secure circuits against micro-architectural attacks for rich applications. The second aspect of the project pertains to the secure integration of such new generations of secure processors into System of Chips, to the research and development of secure building blocks for such SoCs like secure and robust Random Number Generators, memory blocks secured against physical attacks, memories instrumented for security and agile hardware accelerators for next generation of cryptography. This work on hardware security is completed by studies on software tools for dynamic annotation of code for next generation of secure embedded software, by the implementation of a secure kernel for an embedded OS and by research work on the dynamic embedded supervision of the system. A last, but very significant, aspect of this project is the implementation of FPGA and ASIC demonstrators integrating the components developed in this project. Those demonstrators shall offer a unique opportunity to showcase the results of the project. This ambitious project will result in increasing the scientific visibility of the research teams involved on the international level, but also in the regional, national and international ecosystems. This project shall trigger a durable, lifelong, cooperation among the main French research teams of the field, not only in terms of scientific achievements, but also for building new collaborative projects on the EU level or other national projects involving industrial partners.

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

Participants: Caroline Collange, Sara Sadat Hoseininasab.

- Funding: ANR, JCJC
- Duration: 2020-2025
- Local coordinator: Caroline Collange
- 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.

CAOTIC: Collaborative Action on Timing Interference

Participants: Hector Chabot, Isabelle Puaut.

- Funding: ANR
- Duration: 2022-2026
- Local coordinator: Isabelle Puaut
- Partners: CEA List, Inria, Univ Rennes/IRISA, IRIT, IRT Saint Exupery, LS2N, LTCI, Verimag (Project Coordinator)
- Project CAOTIC is an ambitious initiative aimed at pooling and coordinating the efforts of major French research teams working on the timing analysis of multicore real-time systems, with a focus on interference due to shared resources. The objective is to enable the efficient use of multicore in critical systems. Based on a better understanding of timing anomalies and interference, taking into account the specificities of applications (structural properties and execution model), and revisiting the links between timing analysis and synthesis processes (code generation, mapping, scheduling), significant progress is targeted in timing analysis models and techniques for critical systems, as well as in methodologies for their application in industry.

  In this context, the originality and strength of the CAOTIC project resides in the complementarity of the approaches proposed by the project members to address the same set of scientific challenges: (i) build a consistent and comprehensive set of methods to quantify and control the timing interferences and their impact on the execution time of programs; (ii) define interference-aware timing analysis and real-time scheduling techniques suitable for modern multi-core real-time systems; (iii) consolidate these methods and techniques in order to facilitate their transfer to industry.
- website: anr-caotic.imag.fr/

OWL: Operating Within Limits

Participants: Erven Rohou, Isabelle Puaut.

- Funding: ANR
- Duration: 2023-2027
- Local coordinator: Erven Rohou
- Partners: IRISA/Granit Lannion, LS2N/STR Nantes (Project Coordinator), LS2N/SIMS Nantes
- Project OWL proposes a new model of computation for more frugal intelligent autonomous sensors: circadian artificial intelligence (AI). The targeted applications are in the field of environmental monitoring, especially bioacoustic and its application to conservation ecology. This model is particularly well suited for sensors without batteries that are intermittently powered by ambient energy. The great promises of these systems is the extension of their lifetime without the need for human intervention allowing for long-term biostatistics observation missions, and a lower impact on the environment thanks to the absence of battery.

Circadian AI is interested in observing phenomena that have a period of one day, such as the activity of birds or the pollution associated with traffic in a metropolis. It exploits the fact that this period is shared with the availability of solar energy, which is used to power the sensors. This correlation allows the systems to temporally shift the costly computations required to perform the AI functions to times when the observed phenomenon is at rest and energy is abundant.

The project proposes two main contributions. The first is to design new algorithms for circadian AI that allow for this temporal shift in computation. The second is to provide the software and hardware infrastructure necessary to run circadian AI on intermittently powered sensors.

The work done in the project will be based as much as possible on open source / open hardware technologies. Those built during the project (dataset, software, hardware design) will all be freely distributed.

FAIR: Fault Attack Injection Resilience

Participants: Erven Rohou, Isabelle Puaut.

- Funding: ANR
- Duration: 2025-2030
- Local coordinator: Erven Rohou
- Partners: IMT-Atlantique, Université de Bretagne Sud
- The FAIR project aims to develop a secure and efficient processor, along with its accompanying tools, to counter fault injection attacks targeting embedded systems (smartcards, smartphones, etc.). The goal is to overcome the limitations of "lockstep" processors and current Instruction Set Randomization (ISR) schemes, which are often inefficient in terms of performance and energy consumption. In the state of the art, proposed solutions attempt to adapt existing tools (cryptographic primitives, instruction sets) to this problem. We argue, on the contrary, for the need to develop new tools specifically for this use case. First, current cryptographic schemes for ISR suffer from primitives and modes with excessive latency, as they were designed for other purposes. Our first focus is therefore the development of a specific primitive and mode to ensure cryptographic integrity with low latency. Second, the resilience and integrity of the microarchitecture must scale to larger cores. We are therefore targeting a CVA6 core. Finally, we must acknowledge that modifying the instruction set can yield security gains. To this end, we propose modifying the RISC-V instruction set to remove the possibility of forward indirect jumps, enabling a simpler cryptographic scheme and allowing the compiler to efficiently and accurately determine the control flow graph of our application.

  This work is carried out in collaboration with an industrial partner, particularly to validate the realism of our designs.

  PACAP is in particular involved in creating a dedicated compiler capable of leveraging this architecture without resorting to indirect jumps.

LOTR: Lord Of The RISCs

Participants: Isabelle Puaut.

- Funding: ANR
- Duration: 2023-2027
- Local coordinator: Simon Rokicki (Univ Rennes/IRISA)

- Partners: CEA List, Univ. Rennes/IRISA (coordinator)
- Lord Of The RISCs (LOTR) is a novel flow for designing highly customized RISC-V processor microarchitectures for embedded and IoT platforms. The LOTR flow operates on a description of the processor Instruction Set Architecture (ISA). It can automatically infer synthesizable Register Transfer Level (RTL) descriptions of a large number of microarchitecture variants with different performance/cost trade-offs. In addition, the flow integrates two domain-specific toolboxes dedicated to the support of timing predictability (for safety-critical systems) and security (through hardware protection mechanisms)

AIxIA (Artificial Intelligence for Interference Analysis)

Participants: Isabelle Puaut.

- Funding: FRAE (Fondation de Recherche pour l'Aéronautique et l'Espace) AIRSTRIP (*L'intelligence Artificielle au service de l'IngénieRie des SysTèmes aéRonautIques et sPatiaux*) project
- Duration: 2024-2026
- Local coordinator: Isabelle Puaut
- Partners: IRT Saint Exupéry, INRIA Bordeaux, IRIT, Univ Rennes/IRISA
- Demonstrating the satisfaction of temporal performance in an embedded software with the required level of confidence is a difficult and costly task. One of the main issues is accounting for temporal interference phenomena that occur between software applications sharing elements of the execution structure (e.g., cores, GPU, etc.). In this context, the AIxIA project aims to study the contribution of artificial intelligence techniques to identifying these interferences and analyzing their effects. The project will apply artificial intelligence techniques to three dimensions of the problem: (i) identifying sources of interference, (ii) quantifying and predicting their effects, and (iii) avoidance.

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

Participants: Pierre Michaud.

- Funding: ANR, PRC
- Duration: 2021-2026
- Local coordinator: Pierre Michaud
- 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/

AoT.js

Participants: Aurore Poirier, Erven Rohou.

- Funding: Inria Exploratory Action
- Duration: 2022-2025
- Local coordinator: Erven Rohou
- Partners: SPLiTS (Sophia)
- JavaScript programs are typically executed by a JIT compiler, able to handle efficiently the dynamic aspects of the language. However, JIT compilers are not always viable or sensible (e.g., on constrained IoT systems, due to secured read-only memory (W⊕X), or because of the energy spent recompiling again and again). We propose to rely on ahead-of-time compilation, and achieve performance thanks to optimistic compilation, and detailed analysis of the behavior of the processor, thus requiring a wide range of expertise from high-level dynamic languages to microarchitecture.

CocoRISCo

Participants: Jean-Michel Gorius, Erven Rohou.

- Funding: Inria Challenge
- Duration: 2024-2028
- Local coordinator: Olivier Sentieys
- Partners: BENAGIL, CORSE, SUSHI, TARAN, the SLS team of the TIMA laboratory and the DSCIN of laboratory CEA List
- CocoRISCo focuses on the hardware and low-level software aspects of computer systems. Within this project, we aim at exploring the use of binary rewriting to ensure compatibility of modern software on less capable hardware (older, or relying on different ISA extensions).

FORWARD: Formal Verification and Physical Attacks Resilience of HW countermeasures

Participants: Antoine Gicquel, Damien Hardy, Sébastien Michelland, Erven Rohou.

- Funding: *Programme de Transfert du Campus Cyber* (PTCC)
- Duration: 2024-2027
- Local coordinator: Erven Rohou
- Partners: BENAGIL, CORSE, SUSHI, TARAN, the SLS team of the TIMA laboratory and the DSCIN of laboratory CEA List
- Forward targets formal verification of hardware. The goals are to 1) evolve formal analysis tools for hardware towards more realistic attack models and more complex architectures; and 2) make progress in security standards by analyzing the complementarity of formal and experimental methods. We will extend SAMVA (see Section 7.1.6) along two directions: a new attack model based on laser injection, as well as data flow analysis to widen the range of successful attack paths.

JuraSTIC: Hardware and software historical collection for research in Computer Science

Participants: Caroline Collange, Erven Rohou, Damien Hardy.

- Funding: *Appel Unique* CNRS INS2I
- Duration: 2024-2025
- Local coordinator: Caroline Collange
- Partners: EPICURE, TARAN, SED
- The JuraSTIC aims at constituting and curating a historical software and hardware collection. It will foster research in computer science, including reuse of legacy computer systems, reverse-engineering and replication, reproducibility, avoiding obsolescence, and cybersecurity.

10.4 Regional initiatives

SCI3D

Participants: Pierre Bedell, Damien Hardy, Xabier Legaspi Juanatey.

- Funding: CREACH LABS
- Duration: 2024-2026
- Local coordinator: Damien Hardy
- SCI3D addresses the security of the 3D-printing toolchain. We will study and characterize the attack vectors on 3D printer farms, with a focus on 3D printers, particularly the hardware and firmware, in a decentralized framework for distributed manufacturing. Countermeasures will be proposed to secure the printer's control by utilizing hardened hardware equipped with cryptographic accelerators, with the aim of securing the firmware and protecting the communication channel with actuator control.

11 Dissemination

Participants: Nicolas Bailluet, Hector Chabot, Niels Cobat, Caroline Collange, Damien Hardy, Sara Hoseininasab, Pierre Michaud, Ariane Nicolas, Aurore Poirier, Isabelle Puaut, Hugo Reymond, Matthieu Rodet, Erven Rohou.

11.1 Promoting scientific activities

11.1.1 Scientific events: selection

Member of the conference program committees

- E. Rohou was a PC member of the International Symposium on Code Generation and Optimization (CGO) 2026.
- P. Michaud is a member of the program committees of the International Symposium on Computer Architecture (ISCA) 2026 and of the 4th Data Prefetching Championship (DPC4) 2026.
- I. Puaut was a PC member of the following conferences:

  - Euromicro Conference on Real Time Systems (ECRTS) 2025 and 2026;
  - International Conference on Real-Time Systems and Networks (RTNS 2026), Nov 2026;
  - Real-Time and Embedded Technology and Applications Symposium (RTAS) 2026;
  - Compiler Construction (CC) 2026;
  - Embedded Real Time Systems (ERTS) 2026;
  - Real-Time Systems Symposium (RTSS) 2025;
  - Code Generation and Optimization (CGO) 2025.

Reviewer

Members of PACAP routinely review submissions to international conferences and journals.

11.1.2 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 conferences and journals.

11.1.3 Invited talks

E. Rohou was invited to present the activities of the team at the Cyber Founder Tour, an event dedicated to the creation of startups in cybersecurity, in link with research.

11.1.4 Leadership within the scientific community

I. Puaut is member of the Advisory board of the Euromicro Conference on Real Time Systems (ECRTS).

11.1.5 Scientific expertise

I. Puaut was member of the best paper selection committee for RTAS 2025 and the Test of Time of the IEEE TC RTS in 2025 and 2026.

11.1.6 Research administration

- E. Rohou is the contact for international relations for the Inria Centre at the University of Rennes (for scientific matters).
- I. Puaut is elected member of section 27 of CNU (*Conseil National des Universités* – National Council of Universities). The CNU is a national consultative and decision-making body. It makes decisions regarding the career progression of assistant professors and professors in institutions under the jurisdiction of the Ministry of Higher Education and Research (MESR).
- I. Puaut is member of the thesis committee (*comité des thèses*) at the Matisse doctoral school. The committee is responsible for reviewing thesis registration applications and forming juries. The thesis committee oversees the 250 doctoral students hosted at IRISA.

11.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

11.2.1 Teaching

- Master: A. Nicolas, Théorie du Langage et de la Compilation, 48 hours, M1, ESIR, France
- Bachelor: N. Cobat, Algorithmic in Java, 27 hours, L1, Université de Rennes, France
- Bachelor: N. Cobat, Programmation in Python, 18 hours, L1, Université de Rennes, France
- Bachelor: N. Cobat, Data Base, 18 hours, L2, Université de Rennes, France
- Master: D. Hardy, Operating systems, 33 hours, M1, Université de Rennes, France
- Master: D. Hardy, Students project, 33 hours, M1, Université de Rennes, France
- Bachelor: D. Hardy, Additive manufacturing, 16 hours, L2, Université de Rennes, France
- Bachelor: D. Hardy, Electronics, 14 hours, L1, Université de Rennes, France
- Master: M. Rodet, Low Level Programming, 19.5 hours, M1, Université de Rennes, France
- Master: M. Rodet, Travaux Pratiques, 15.5 hours, M2, ENS Rennes, France
- Master: M. Rodet, Projets, 6 hours, M2, ENS Rennes, France
- Master: M. Rodet, *Oraux blancs de Travaux Pratiques*, 8 hours, M2, ENS Rennes, France
- Master: N. Bailluet, Software Exploitation, 24 hours, M1 Cyber, Université de Rennes, France
- Master: C. Collange, Advanced Computer Architectures, 6 hours, M2, ENS Rennes, France
- Master: I. Puaut, Advanced Operating Systems (SEA), 100 hours, M1, Université de Rennes
- Master: I. Puaut, Low Level Programming (LLP), 40 hours, Université de Rennes
- Master: I. Puaut, Writing of scientific publications, 9 hours, M2 and PhD students, Université de Rennes
- Master: I. Puaut, Optimizing and Parallelizing Compilers, 6 hours, Université de Rennes
- Bachelor: I. Puaut, Computer Architecture, 25 hours, Université de Rennes

11.2.2 Supervision

- PhD: Sara Hoseininasab, *Automatic synthesis of multi-thread pipelines* [29], Feb 2025, advisors C. Collange (70 %) and S. Derrien (30 %, TARAN). Funding: ANR project DYVE.
- PhD: Nicolas Bailluet, *Attaques par réutilisation de code : synthèse automatique et évaluation automatique de possibilité d'exploitation* [28], Nov 2025, advisors I. Puaut (50 %) and E. Rohou (50 %). Funding: grant from ENS Rennes.
- PhD in progress: Hector Chabot, *Fine grain software modeling and analysis for interference management in multi-core real-time systems*, started Sep 2023, advisors I. Puaut (50 %), H. Cassé and T. Carle (IRIT, Toulouse, 25 % each). Funding: ANR project CAOTIC.
- PhD in progress: Aurore Poirier, *Profile-Guided optimization for Dynamic Languages*, started Oct 2022, advisors E. Rohou (50 %) and M. Serrano (50 %, Inria Sophia). Funding: Inria Exploratory Action AoT.js.
- PhD in progress: Matthieu Rodet, *Software support for running Circadian AI on next generation intermittent systems*, started Oct 2024, advisors I. Puaut, E. Rohou, S. Faucou (LS2N Nantes), M. Briday (LS2N Nantes). Funding: ANR project OWL.

- PhD in progress: Niels Cobat, *Analyse et optimisation des fichiers d'impression 3D à l'aide de méthodes d'apprentissage automatique*, started Oct 2024, advisors D. Hardy (50 %) and R. Gaudel (50 %, MALT). Funding: grant from Université de Rennes (*contrat doctoral*).
- PhD in progress: Maël Coatanhay, *Évaluation par injection de fautes laser et photoémission de modèles de fautes sur un jeu d'instruction RISC-V*, started Oct 2024, advisors L. Le Brizoual (25 % IETR), L. Pichon (25 % IETR), D. Hardy (25 %), T. Rubiano (25 %). Funding: cyberschool + Cyberskills4all.
- PhD in progress: Ariane Nicolas, *CDIFC : Compilation Durcie pour l'Intégrité du Flot de Contrôle*, started Oct 2025, advisors R. Lashermes (34 %), I. Puaut (33 %), E. Rohou (33 %). Funding: ANR FAIR.
- PhD in progress: Louis Savary, *Sécurité dans les processeurs basés sur la traduction dynamique de binaire*, started Sep 2022, advisors: E. Rohou (34 %), S. Derrien (Université de Bretagne Occidentale, 33 %), S. Rokicki (TARAN, 33 %). Funding: PEPR ARSENE.
- PhD in progress: Alix Tremodeux, *Étude des conséquences du vieillissement sur les machines HPC*, started Sep 2025, advisors G. Pallez (KERDATA, 75 %), E. Rohou (25 %). Funding: grant from ENS Lyon.
- PhD in progress: Dylan Léothaud, *High-Level Synthesis of Processors for IoT*, started Oct 2024, co-directed by Isabelle Puaut (50%), mainly supervized by Steven Derrien (Université de Bretagne Occidentale) and Simon Rokicki (TARAN). Funding: grant from ENS Rennes.
- PhD in progress: Thomas Feuilletin, *High-Level Synthesis of Deterministic Micro-Architectures*, started Oct 2025, co-supervized by Steven Derrien (Université de Bretagne Occidentale, 34 %), Simon Rokicki (TARAN, 33 %), Isabelle Puaut (33 %). Funding: ANR LOTR.
- Master thesis. Thomas Feuilletin, *Automatic Extraction of Temporal Models of Micro-architecture From a High-Level Synthesis Flow of RISC-V Processors*, Master thesis, Université de Rennes, Feb to Jun 2025, co-supervized by Simon Rokicki and Steven Derrien.

11.2.3 Juries

I. Puaut was member of the following hiring committees:

- Professor, topic "Artificial Intelligence", Spring 2025. Deputy president, University of Rennes
- Assistant professor "Embedded IA", IUT de Lannion, Spring 2025, University of Rennes

Members of PACAP participated to the following PhD and HdR committees:

- P. Michaud was a member of the jury of Pierre Ravenel's PhD at Université de Grenoble Alpes, entitled *Improving the performance of in-order processors under hardware complexity constraints*.
- C. Collange was a member of the committee of Orégane Desrentes's PhD at INSA Lyon titled *Hardware Arithmetic Acceleration for Machine Learning and Scientific Computing*.
- I. Puaut was of member of the following PhD thesis of HdR committes:
  - Clément Rosetti, *Algebraic Tiling: Volume-guided Tiling of Parallel Loops for Near-Perfect Load Balancing*. PhD thesis, Université de Strasbourg, Dec 2025 (reviewer)
  - Pierrick Philippe: *Secrets in Compiler: Detection of Secret-related Weaknesses in GCC Static Analyzer*, PhD thesis, Université de Rennes, Dec 2025 (examiner, president of the jury)
  - Sébastien Michelland: *Compilation pour la sécurité matérielle : au delà de la sémantique (compiling for hardware security: beyond semantics)*. Université de Grenoble Alpes, Oct 2025 (examiner, president of the jury)
  - Ronan Lashermes. *Micro-architecture security, future-proof designs*, HdR, Université de Rennes, May 2025 (examiner, president of the jury)

E. Rohou is member of the CSID commitee of Ikram Dendani, Georges Aaron Randrianaina, Jean-Loup Hatchikian-Houdot, Arthur Branchu-Harel. I. Puaut is member of the CSID commitee of Constance Bocquillon, Cédric Cazanove and Valentin Septier.

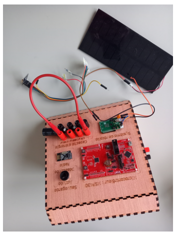
11.2.4 Educational and pedagogical outreach

- E. Rohou was invited to present the job of a researcher to secondary-school students (classe de 4e) at Collège de Bourgchevreuil, Cesson-Sévigné.
- E. Rohou contributed to the program "1 scientifique, 1 classe : Chiche !" with three interventions at Cité Scolaire Beaumont, Redon.

11.3 Popularization

11.3.1 Productions (articles, videos, podcasts, serious games, ...)

We built a prototype [32] of our intermittent computing system designed within the framework of the OWL ANR project. The prototype was demonstrated by Hugo Reymond and Matthieu Rodet on June 4, 2025 on the occasion of the institutional day dedicated to the IRISA laboratory's 50th anniversary.



11.3.2 Participation in Live events

Participants: Caroline Collange, Erven Rohou, Thomas Rubiano, Niels Cobat, Antoine Gicquel.

The JuraSTIC computer history exhibit showcased the JuraSTIC collection of computing artefacts, as part of for the annual national Science fair (*Fête de la Science*) and the 50th anniversary of the IRISA computing laboratory. The exhibit was open to the public from October 9 to November 12, 2025 on the *Diapason* exhibition center on the Rennes Beaulieu campus. It was lead by Caroline Collange and organized by team of 14 members from IRISA and University of Rennes Cultural Affairs staff, including 5 PACAP team members. The exhibit showcased a dozen historically significant computing artifacts from IRISA's collections, organized in five themes, each associated with an explanatory poster. The themes were: human-computer interfaces, data processing in servers, computer graphics and image processing, supercomputers, and communication networks.

Throughout the exhibit, we carried out commented visits and demonstrations including the operation of a Mitra 125 mini-computer and its punched card reader from the 1970s, as well as tutorials where visitors could operate working computers systems from the 1970s and 1980s. We organized three tutorials: drawing with a light pen on Thomson micro-computers, programming computer graphics on a Tektronix 4006 vector graphics terminal, and retro-gaming on early Nintendo gaming consoles and a TI-99 micro-computer.

The exhibit was attended by high-school students (4 classes of *Seconde* grade) and the general public (220 people on the opening day). Throughout the exhibit, we carried out 15 commented visits and demonstrations for about 150 people in total. It was covered by the local press (*Ouest-France*, *Ici Rennes*).

12 Scientific production

12.1 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 (cit. on pp. 9, 10).

[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 (cit. on p. 10).

[3] D. Hardy and I. Puaut. 'Static probabilistic Worst Case Execution Time Estimation for architectures with Faulty Instruction Caches'. In: *21st International Conference on Real-Time Networks and Systems*. Sophia Antipolis, France, Oct. 2013. DOI: 10.1145/2516821.2516842. URL: https://hal.inria.fr/hal-00862604.

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

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

[6] P. Michaud. 'Best-Offset Hardware Prefetching'. In: *International Symposium on High-Performance Computer Architecture*. Barcelona, Spain, Mar. 2016. DOI: 10.1109/HPCA.2016.7446087. URL: https://hal.inria.fr/hal-01254863 (cit. on pp. 10, 11).

[7] P. Michaud, A. Mondelli and A. Seznec. 'Revisiting Clustered Microarchitecture for Future Superscalar Cores: A Case for Wide Issue Clusters'. In: *ACM Transactions on Architecture and Code Optimization (TACO)* 13.3 (Aug. 2015), p. 22. DOI: 10.1145/2800787. URL: https://hal.inria.fr/hal-01193178 (cit. on p. 11).

[8] A. Perais and A. Seznec. 'EOLE: Paving the Way for an Effective Implementation of Value Prediction'. In: *International Symposium on Computer Architecture*. Vol. 42. ACM/IEEE. Minneapolis, MN, United States, June 2014, pp. 481–492. DOI: 10.1109/ISCA.2014.6853205. URL: https://hal.inria.fr/hal-01088130 (cit. on pp. 10, 11).

[9] A. Perais and A. Seznec. 'Practical data value speculation for future high-end processors'. In: *International Symposium on High Performance Computer Architecture*. IEEE. Orlando, FL, United States, Feb. 2014, pp. 428–439. DOI: 10.1109/HPCA.2014.6835952. URL: https://hal.inria.fr/hal-01088116 (cit. on pp. 10, 11).

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

[11] 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.

[12] 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 (cit. on p. 11).

[13] 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.

[14] 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.

[15] 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.

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

[17] 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 (cit. on p. 12).

12.2 Publications of the year

International journals

[18] A. N. Amalou and I. Puaut. 'Using machine learning for timing analysis: where do we stand?' In: *Real-Time Systems* 61.2 (June 2025), pp. 300–305. DOI: 10.1007/s11241-025-09442-y. URL: https://hal.science/hal-05092607 (cit. on p. 22).

[19] A. Poirier, E. Rohou and M. Serrano. 'An Attempt to Catch Up with JIT Compilers: The False Lead of Optimizing Inline Caches'. In: *The Art, Science, and Engineering of Programming* 10.6 (15th Feb. 2025). DOI: 10.22152/programming-journal.org/2025/10/6. URL: https://hal.science/hal-04904428 (cit. on p. 20).

International peer-reviewed conferences

[20] H. Andrianatrehina, R. Lashermes, J. Paturel, S. Rokicki and T. Rubiano. 'Exploring speculation barriers for RISC-V selective speculation'. In: ARES 2025 - 20th International Conference on Availability, Reliability and Security. Ghent, Belgium, 11th Aug. 2025, pp. 1–23. URL: https://hal.science/hal-05061555 (cit. on p. 23).

[21] N. Bailluet, E. Fleury, I. Puaut and E. Rohou. 'Nothing is Unreachable: Automated Synthesis of Robust Code-Reuse Gadget Chains for Arbitrary Exploitation Primitives'. In: *Proceedings of the 34th USENIX Security Symposium*. USENIX Security 2025 - 34th USENIX Security Symposium. Seattle, WA, United States, 2025, pp. 1–18. URL: https://hal.science/hal-05193528 (cit. on p. 24).

[22] T. Feuilletin, D. Leothaud, S. Rokicki, S. Derrien and I. Puaut. 'Automatic Extraction of Timing Models for WCET Estimation From a High-Level Synthesis Flow'. In: DATE 2026 - Design, Automation and Test in Europe Conference. Verona, Italy, Apr. 2026. URL: https://hal.science/hal-05365718 (cit. on p. 23).

[23] D. Leothaud, S. Rokicki, S. Derrien and I. Puaut. 'Area Efficient Speculative Loop Pipelining for High-Level Synthesis'. In: DATE 2026 - Design, Automation and Test in Europe Conference. Verona, Italy, Apr. 2026. URL: https://hal.science/hal-05365717 (cit. on p. 22).

[24] M. Rodet, J.-L. Béchennec, M. Briday, S. Faucou, I. Puaut and E. Rohou. 'Circadia: Checkpointing for Intermittent Computing in AI Driven Applications'. In: *28th Euromicro Conference on Digital System Design (DSD)*. DSD 2025 - 28th Euromicro Conference on Digital System Design. Salerno, Italy, 12th Sept. 2025, pp. 1–10. DOI: 10.1109/DSD67783.2025.00091. URL: https://hal.science/hal-05340384 (cit. on pp. 14, 19).

National peer-reviewed Conferences

[25] C. Collange. 'Méduse : reproducing obsolete circuits'. In: *Compas 2025*. COMPAS 2025 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Bordeaux, France, 2025, pp. 1–6. URL: https://inria.hal.science/hal-05062179 (cit. on pp. 14, 21).

Conferences without proceedings

[26] H. Reymond. 'Capteurs sans batterie ou le mythe de l'autonomie infinie: Comment la variabilité et le vieillissement des composants impacte l'exécution de programmes ?' In: Greendays 2025 - Au-delà de l'efficacité, comment imaginer un numérique plus sobre ? Rennes, France, 2025, pp. 1–13. URL: https://hal.science/hal-05009459 (cit. on p. 14).

Scientific books

[27] P. Michaud. *HARCOM: Hardware complexity model for microarchitecture exploration*. 2026. URL: https://inria.hal.science/hal-05457441. In press (cit. on p. 21).

Doctoral dissertations and habilitation theses

[28] N. Bailluet. 'Code-Reuse Attacks: Automated Synthesis and Assessment of Exploitation Feasibility'. Université de Rennes, 27th Nov. 2025. URL: https://inria.hal.science/tel-05451149 (cit. on pp. 24, 33).

[29] S. S. Hoseininasab. 'Using HLS to raise the design abstraction level for faster exploration of different CPU Micro-architectures'. Université de Rennes, 18th Feb. 2025. URL: https://theses.hal.science/tel-05100590 (cit. on pp. 21, 33).

Reports & preprints

[30] R. Boëzennec, F. Fernandes dos Santos, B. Goglin, A. Kritikakou, G. Pallez, E. Rohou, O. Sentieys and M. Traiola. *Increasing the Lifetime of HPC Machines: Issues, Implications, and Open Challenges*. 2025. URL: https://hal.science/hal-05312072 (cit. on p. 20).

[31] C. Maiza, L. Rieg, M. Asavoae, J.-L. Béchennec, D. Blouin, F. Brandner, T. Carle, H. Cassé, S. Faucou, B. Ferres, P.-E. Hladik, E. Jahier, M. Jan, É. Jenn, L. Jezequel, D. Potop-Butucaru, I. Puaut, P. Raymond, C. Rochange, P. Sotin, C. Parent-Vigouroux, H.-E. Zahaf, H. Chabot, M. Essabyr, L. Jeanmougin and H. Rebhi. *Collaborative Action on Timing InterferenCes: Summary and Perspectives at Mid-term*. 19th Jan. 2026. URL: https://hal.science/hal-05465745.

Other scientific publications

[32] H. Reymond and M. Rodet. 'Démonstration : Exécution intermittente sur capteurs sans batterie: Comment exécuter un programme malgré de fréquentes pertes d'alimentation ?' In: 2025 - Journée institutionnelle des 50 ans de l'IRISA. Rennes, France, 2025, pp. 1–1. URL: https://hal.science/hal-05100824 (cit. on p. 35).

[33] M. Rodet, J.-L. Béchennec, M. Briday, S. Faucou, I. Puaut and E. Rohou. 'Circadia: Checkpointing for Intermittent Computing in AI Driven Applications'. In: ACACES 2025 - 21st Internationnal Summer School on Advanced Computer Architecture and Compilation for High-performance Embedded Systems. Salerno, Italy, 2025. URL: https://hal.science/hal-05340610 (cit. on p. 19).

12.3 Cited publications

[34] 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 (cit. on pp. 9, 10).

[35] 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 (cit. on p. 20).

[36] 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 (cit. on p. 13).

[37] R. Kumar, D. M. Tullsen, N. P. Jouppi and P. Ranganathan. 'Heterogeneous chip multiprocessors'. In: *IEEE Computer* 38.11 (Nov. 2005), pp. 32–38 (cit. on p. 12).

[38] C. Le Bon. 'Analyse et optimisation dynamiques de programmes au format binaire pour la cybersécurité'. Theses. Université Rennes 1, July 2022. URL: https://theses.hal.science/tel-03906421 (cit. on p. 10).

[39] 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 (cit. on pp. 10, 11).

[40] 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 (cit. on p. 13).

[41] 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 (cit. on p. 10).

[42] 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 (cit. on p. 11).

[43] 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 (cit. on pp. 10, 11).

[44] 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 (cit. on p. 11).

[45] 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 (cit. on p. 13).