UMR IRISA

Activity Report 2024

Team PACAP

Pushing Architecture and Compilation for Application Performance

Joint team with Centre Inria de l'Université de Rennes

D3 – Architecture

Université de Rennes | ENS rennes | INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES | cnrs | Inria | Inserm | CentraleSupélec | IMT Atlantique Bretagne-Pays de la Loire École Mines-Télécom | Université Bretagne Sud ubs

Contents

Project-Team PACAP 1

1 Team members, visitors, external collaborators 2

2 Overall objectives 3

3 Research program 5
3.1 Motivation . . . 5
3.1.1 Technological constraints . . . 5
3.1.2 Evolving community . . . 5
3.1.3 Domain constraints . . . 6
3.2 Research Objectives . . . 6
3.2.1 Static Compilation . . . 7
3.2.2 Software Adaptation . . . 7
3.2.3 Research directions in uniprocessor micro-architecture . . . 7
3.2.4 Towards heterogeneous single-ISA CPU-GPU architectures . . . 9
3.2.5 Real-time systems . . . 9
3.2.6 Power efficiency . . . 10
3.2.7 Security . . . 10

4 Application domains 11
4.1 Domains . . . 11

5 Social and environmental responsibility 11
5.1 Impact of research results . . . 11

6 Highlights of the year 11
6.1 Awards . . . 11

7 New software, platforms, open data 12
7.1 New software . . . 12
7.1.1 ATMI . . . 12
7.1.2 HEPTANE . . . 12
7.1.3 tiptop . . . 12
7.1.4 GATO3D . . . 13
7.1.5 OptiPrint . . . 13
7.1.6 SAMVA . . . 13
7.1.7 TimeKlip . . . 14
7.1.8 ExtLib . . . 14
7.2 New platforms . . . 14
7.2.1 Ofast3D . . . 14
7.2.2 Arsene evaluation environment . . . 14
7.3 Open data . . . 15

8 New results 15
8.1 Compilation and Optimization . . . 15
8.1.1 Compilation for Intermittent Systems . . . 15
8.1.2 Dynamic Binary Analysis and Optimization . . . 16
8.1.3 3D printing time estimation and optimization . . . 17
8.2 Processor Architecture . . . 17
8.2.1 TAGE: an engineering cookbook . . . 17
8.2.2 Automatic synthesis of multi-thread pipelines . . . 17
8.2.3 Two-dimensional memory architecture . . . 18
8.3 WCET estimation and optimization . . . 18
8.3.1 Using machine learning for timing analysis of complex processors . . . 18

8.3.2 Static estimation of memory access profiles for real-time multi-core systems 19
8.3.3 Estimation of interference delays in real-time multi-core systems 19
8.4 Security 19
8.4.1 Multi-nop fault injection attack 19
8.4.2 Gadget chains synthesis driven by SMT Solving for Code-Reuse Attacks 20

9 Bilateral contracts and grants with industry **20**
9.1 Bilateral contracts with industry 20

10 Partnerships and cooperations **21**
10.1 International initiatives 21
10.1.1 Inria associate team not involved in an IIL or an international program 21
10.2 European initiatives 22
10.2.1 Other European programs/initiatives 22
10.3 National initiatives 22
10.4 Regional initiatives 28

11 Dissemination **28**
11.1 Promoting scientific activities 28
11.1.1 Scientific events: selection 28
11.1.2 Journal 29
11.1.3 Invited talks 29
11.1.4 Leadership within the scientific community 29
11.1.5 Research administration 29
11.2 Teaching - Supervision - Juries 30
11.2.1 Teaching 30
11.2.2 Supervision 30
11.2.3 Juries 31
11.3 Popularization 31
11.3.1 Productions (articles, videos, podcasts, serious games, ...) 31
11.3.2 Participation in Live events 32

12 Scientific production **32**
12.1 Major publications 32
12.2 Publications of the year 33
12.3 Cited publications 34

Project-Team PACAP

Creation of the Project-Team: 2016 July 01

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. – Embedded and cyber-physical systems
A2.3.1. – Embedded systems
A2.3.2. – Cyber-physical systems
A2.3.3. – Real-time systems
A4.4. – Security of equipment and software
A5.10.3. – Planning
A5.10.5. – Robot interaction (with the environment, humans, other robots)
A9.2. – Machine learning

Other research topics and application domains

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

1 Team members, visitors, external collaborators

Research Scientists

- Erven Rohou [Team leader, INRIA, Senior Researcher]
- Caroline Collange [INRIA, Researcher]
- Pierre Michaud [INRIA, Researcher]
- Thomas Rubiano [INRIA, Starting Research Position]
- André Seznec [INRIA, Senior Researcher, from Mar 2024 until Nov 2024]

Faculty Members

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

Post-Doctoral Fellow

- Xabier Legaspi Juanatey [UNIV RENNES, Post-Doctoral Fellow, from Oct 2024]

PhD Students

- Nicolas Bailluet [UNIV RENNES]
- Hector Chabot [UNIV RENNES]
- Niels Cobat [UNIV RENNES, from Oct 2024]
- Antoine Gicquel [INRIA, until Nov 2024]
- Sara Sadat Hoseininasab [INRIA]
- Aurore Poirier [INRIA]
- Hugo Reymond [INRIA, until Sep 2024]
- Matthieu Rodet [INRIA, from Oct 2024]

Technical Staff

- Pierre Bedell [INRIA, Engineer, until Aug 2024]
- Imane Lasri [INRIA, Engineer, from Mar 2024]
- Camille Le Bon [INRIA, Engineer]
- Hugo Reymond [INRIA, Engineer, from Oct 2024]

Interns and Apprentices

- Killian Callac [CNRS, Intern, from May 2024 until Jul 2024]
- Hugo Hamon [UNIV RENNES, Intern, from Jun 2024 until Jul 2024]
- Xabier Legaspi Juanatey [UNIV RENNES, Intern, from Mar 2024 until Aug 2024]
- Louis-Quentin Noe [UNIV RENNES, Intern, from May 2024 until Jul 2024]
- Matthieu Rodet [INRIA, Intern, until Aug 2024]
- Erwan Tanguy-Legac [ENS RENNES, Intern, from May 2024 until Jul 2024]

Administrative Assistants

- Nathalie Denis [INRIA]
- Virginie Desroches [INRIA]
- Sophie Maupile [CNRS]

2 Overall objectives

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

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

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

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

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

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

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

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

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

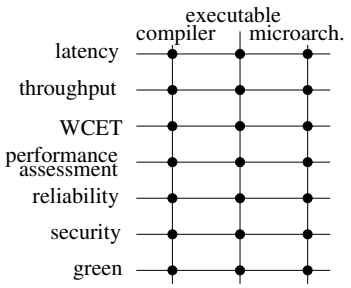


Figure 1: Perimeter of Research Objectives

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

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

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 [32] 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).

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

3.2.1 Static Compilation

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

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 [32]: the idea consists in pre-computing alternatives, and embedding in the program enough information to assist and drive a runtime system towards to the best solution.

3.2.2 Software Adaptation

More than ever, software needs to adapt to its environment. In most cases, this environment remains unknown until runtime. This is already the case when one deploys an application to a cloud, or an “app” to mobile devices. The dilemma is the following: for maximum portability, developers should target the most general device; but for performance they would like to exploit the most recent and advanced hardware features. JIT compilers can handle the situation to some extent, but binary deployment requires dynamic binary rewriting. Our work has shown how SIMD instructions can be upgraded from SSE to AVX transparently [2]. Many more opportunities will appear with diverse and heterogeneous processors, featuring various kinds of accelerators.

On shared hardware, the environment is also defined by other applications competing for the same computational resources. It becomes increasingly important to adapt to changing runtime conditions, such as the contention of the cache memories, available bandwidth, or hardware faults. Fortunately, optimizing at runtime is also an opportunity, because this is the first time the program is visible as a whole: executable and libraries (including library versions). Optimizers may also rely on dynamic information, such as actual input data, parameter values, etc. We have already developed software platforms [39, 36] 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 [41] [37] 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. [40]. 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 [42].

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] [41] [37]. 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 [35] 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 [43] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [34], or thread-based control flow mangling [38]). 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 [22].

6 Highlights of the year

6.1 Awards

H. Reymond, I. Puaut, E. Rohou and their co-authors received a best paper award for their RTCSA 2024 paper "EarlyBird:Energy belongs to those who wake up early" [21].

7 New software, platforms, open data

7.1 New software

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

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 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

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 ExtLib

Keyword: Algorithm

Functional Description: ExtLib is a library written in C++ 20. It contains a number of fairly common functionalities (classes and algorithms) standard template library, geometry, mathematics, graphs and search algorithms, mathematics, graphs and search algorithms. The aim of this library is to provide a set of tools to accelerate the development of various software projects.

Contact: Fabrice Lamarche

7.2 New platforms

7.2.1 Ofast3D

Participants: Pierre Bedell, Damien Hardy.

In the context of the Inria exploratory action Ofast3D, a 3D printing platform for research experiments 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: MimeTIC, Rainbow, LACODAM, TARAN and LogicA.

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

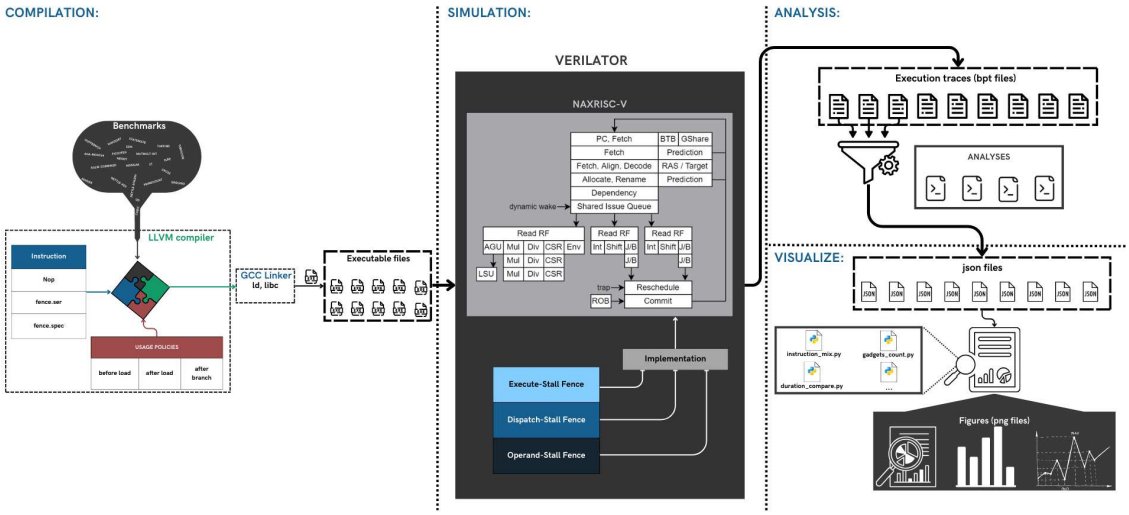


Figure 2: Arsene evaluation environment

7.3 Open data

One dataset has been published on the Zenodo platform to accompany our article on the use of machine learning for performance prediction:

- [30] Reymond, H., Chabot, H., Amalou, A. N., & Puaut, I. (2024). MSP430FR5969 Basic Block Worst Case Energy Consumption (WCEC) and Worst Case Execution Time (WCET) dataset (1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.11066623

8 New results

Participants: Pierre Bedell, Nicolas Bellec, Niels Cobat, Caroline Collange, Antoine Gicquel, Damien Hardy, Imane Lasri, Camille Le Bon, Xabier Legaspi Juanatey, Pierre Michaud, Valentin Pasquale, Anis Peysieux, Isabelle Puaut, Erven Rohou.

8.1 Compilation and Optimization

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

8.1.1 Compilation for Intermittent Systems

Participants: Isabelle Puaut, Hugo Reymond, Erven Rohou
Context: CominLabs project NOP
External collaborators: Sébastien Faucou, Mikaël Briday, Jean-Luc Béchennec, LS2N Nantes

Battery-free devices enable sensing in hard-to-access locations, opening up new opportunities in various fields such as healthcare, space, or civil engineering. Such devices harvest ambient energy and store it in a capacitor, and thus progress in an *intermittent* manner since they operate only when the capacitor has enough energy.

Two research papers were produced in 2024. The first focuses on data checkpointing to address energy failures and was published in CGO'24 [22]. Due to the unpredictable nature of the harvested energy, a power failure can occur at any time, resulting in a loss of all non-persistent information (e.g., processor registers, data stored in volatile memory). Checkpointing volatile data in non-volatile memory allows the system to recover after a power failure, but raises two issues: (i) spatial and temporal placement of checkpoints; (ii) memory allocation of variables between volatile and non-volatile memory, with the overall objective of using energy as efficiently as possible. While many techniques rely on the developer to address these issues, we have presented SCHEMATIC, a compiler technique that automates checkpoint placement and memory allocation to minimize the overall energy consumption. SCHEMATIC ensures that programs will eventually terminate (forward progress property). Moreover, checkpoint placement and memory allocation adapt to the size of the energy buffer and the capacity of volatile memory. SCHEMATIC takes advantage of volatile memory (VM) to reduce the energy consumed, by automatically placing the most used variables in VM. We tested SCHEMATIC for different experimental settings (size of volatile memory and capacitor) and results show an average energy reduction of 51 % compared to related techniques.

SCHEMATIC has been made publically available [31].

Our second study explores early wake-up mechanisms in intermittent systems and was published in RTCSA'24 [21], where it received the *Best Paper Award*. In most existing techniques, the device resumes execution only when the capacitor is full. However, we argue that doing so is sub-optimal. Instead, we advocate that waking-up the device sooner may yield better performance since the microcontroller consumes less power when operating at lower voltage. To this extent, we introduce EarlyBird, a technique that automatically computes a fine-tuned wake-up voltage for each resume point. EarlyBird leverages static analysis to determine how much energy is needed before resuming from a given program location, and provides a runtime library to enforce the early wake-up strategy. We evaluated how EarlyBird improves existing checkpointing techniques and results show an increase in the number of benchmarks executed per minute of up to 5.65×.

These two studies were part of the PhD work of Hugo Reymond, who was co-supervized by Sébastien Faucou, Jean-Luc Bechennec and Mikaël Briday from LS2N [24]. We recently started specializing our work on intermittent systems to systems with embedded AI, through the PhD thesis of Matthieu Rodet, who started in October 2024.

8.1.2 Dynamic Binary Analysis and Optimization

Participants: Aurore Poirier, Erven Rohou
Context: Exploratory Action AoT.js
External collaborators: Manuel Serrano, INDES/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 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 to speed up 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.

This result yielded to an article accepted for publication at the International Conference on the Art, Science, and Engineering of Programming in 2025 [18].

8.1.3 3D printing time estimation and optimization

Participants: Pierre Bedell, Niels Cobat, Damien Hardy, Imane Lasri, Camille Le Bon, Xabier Legaspi Juanatey
Context: Inria Exploratory Action Ofast3D, SCI3D
External collaborators: MimeTIC, LACODAM 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 [33].

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 145 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 MimeTIC 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) and ExtLib (Section 7.1.8).

8.2 Processor Architecture

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

8.2.1 TAGE: an engineering cookbook

Participants: André Seznec

CBP2016 TAGE-SC-L is generally considered as the state-of-the-art of branch predictors that have been been proposed in the academic world. This proposition suffers from several drawbacks, that forbids its direct implementation in hardware. However TAGE has been implemented by industry in many processor cores. We present [28] a set of tradeoffs that could be used in an effective hardware implementation of TAGE-SC. This proposed implementation would still achieve state-of-the-art branch prediction accuracy.

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 aims to demonstrate how the available features of HLS can be deployed in designing various pipelined processors micro-architecture. Our 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.

8.2.3 Two-dimensional memory architecture

Participants: Pierre Michaud, Erven Rohou
Context: ANR Project Maplurinum.

Performance-aware programming is generally done with low-level programming languages such as C or C++ exposing the address space to the programmer. The address space seen by the programmer is the virtual address space defined in the instruction-set architecture (ISA), which is linear, i.e., one-dimensional. However, many programs manipulate multi-dimensional data, which must be flattened in linear memory. Such flattening makes performance-aware programming more difficult. For example, fast matrix multiplication algorithms for modern CPUs copy submatrices into contiguous memory regions to reduce TLB and cache misses, an operation called *packing*. The need for packing stems from the underlying linear virtual memory and from the way caches and TLBs are implemented in hardware.

To ease performance-aware programming, we propose an ISA, called XYA, offering a two-dimensional (2D) address space to the programmer. An address in XYA is a pair of 64-bit coordinates that can be manipulated independently from each other. The physical address space remains one-dimensional, as usual. XYA allows mapping 2D data in virtual memory without flattening. We propose a programming language called XYC giving programmers control over the 2D address space. XYC is mostly similar to C except that it distinguishes two sorts of arrays, x-arrays and y-arrays, and two sorts of pointers, full pointers and ground pointers. The address space of XYA is partitioned into huge regions called books. Each book corresponds to a distinct 2D page aspect ratio. Book selection offers a new degree of freedom to programmers for performance optimization. We show that, by selecting the right book, matrix multiplication on a XYA machine does not need packing [27].

8.3 WCET estimation and optimization

Participants: Hector Chabot, Isabelle Puaut, Hugo Reymond.

8.3.1 Using machine learning for timing analysis of complex processors

Participants: Abderaouf Nassim Amalou, Isabelle Puaut
External collaborators: Elisa Fromont, LACODAM team

Real-time and energy-constrained systems heavily rely on estimates of the worst-case execution time (WCET) and worst-case energy consumption (WCEC) of code snippets to ensure trustworthy operation. Designing architecture-specific analytical models for time and energy is often challenging and time-consuming. In situations where analytical models are unavailable or incomplete, machine learning (ML) techniques emerge as a promising solution to build WCET/WCEC models.

As a follow-up to our research on the use of machine learning for WCET estimation, we have introduced WORTEX, a toolkit for WCET and WCEC estimation of basic blocks based on explanable AI techniques [20]. To ensure the real-world applicability of its models, WORTEX extracts large datasets of basic blocks from real programs and precisely measures their energy consumption/execution time on the physical target platform. The dataset is used to train various WCET/WCEC models using different ML techniques. Experimental results on simple and time-predictable hardware show that even the most basic ML techniques provide accurate results, that never underestimate actual values. We also discuss the use of explainability techniques to gain trustworthiness for the models.

As far as average-case execution time is concerned, we have designed ORXESTRA, a context-aware execution time prediction model based on Transformers XL, specifically designed to accurately estimate performance in embedded system applications [19]. Unlike traditional machine learning models that often overlook contextual information, resulting in biased predictions for individual basic blocks, ORXESTRA overcomes this limitation by incorporating execution context awareness. By doing so, ORXESTRA effectively accounts for the processor micro-architecture without explicitly modeling micro-architectural elements such as caches, pipelines, and branch predictors. Our evaluations demonstrate ORXESTRA's ability to provide precise timing estimations for different Arm targets (Cortex M4, M7, A53, and A72), surpassing existing machine learning-based approaches in both prediction accuracy and prediction speed.

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. This paper presents Marmot [26], 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. Experimental results show that IA-WCET computation and schedulability analysis can take advantage of the fine-grain intervals produced by Marmot to obtain more precise IA-WCET and therefore higher schedulability than coarser-grain profiles.

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

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 machine learning 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.

8.4 Security

Participants: Nicolas Bailluet, Antoine Gicquel, Damien Hardy, Isabelle Puaut, Erven Rohou.

8.4.1 Multi-nop fault injection attack

Participants: Antoine Gicquel, Damien Hardy, Erven Rohou
External collaborators: CIDRE and TARAN team.

Multi-fault injection attacks 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 aimed to address this issue by proposing SAMPLAI, 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 concludes the PhD of Antoine Gicquel [23].

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

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

The final objective of this work is to develop compiler approaches, based on binary code modifications, to protect programs against attacks such as *Return-Oriented Programming* (ROP) or *Jump-Oriented Programming (JOP)*.

As a first step towards this ambitious goal, we have designed a new technique for the automatic chaining of gadgets. Performing complex code-reuse attacks require discovering small code snippets (gadgets) and chaining them together to reach the attacker's goal. We have designed a new method to synthesize gadget chains using SMT solving. The proposed method addresses three challenges, yet not solved by other related works: (i) it is able to build gadget chains for arbitrary exploitation contexts (whether the stack is controlled or not); (ii) it guarantees that data that are out of attacker's control do not interfere with the generated gadget chains (robust reachability property); (iii) it is able to leverage multi-path gadgets to synthesize multi-behavior chains that contain conditions and can behave differently based on their execution context. Experiments show the benefit of the robust reachability property and thoroughly evaluate the quality of the approach in terms of expressivity and performance compared to other related chain synthesis techniques.

This work has been accepted for publication at the 34th USENIX Security Symposium in 2025.

9 Bilateral contracts and grants with industry

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

9.1 Bilateral contracts with industry

Ampere Computing:

Participants: Pierre Michaud.

- Duration: 2024
- Local coordinator: Pierre Michaud

- Collaboration between the PACAP team and Ampere Computing on features of the microarchitecture of next generation CPUs.

Ofast3D:

Participants: Pierre Bedell, Damien Hardy, Imane Lasri, Camille Le Bon, Erven Rohou.

- Duration: 2022-2024
- Local coordinator: Damien Hardy
- Collaboration between the PACAP team in the context of the Ofast3D Inria exploratory action and the following companies: 3D News Tech, Cosmyx and Pollen AM.

SCI3D:

Participants: Damien Hardy, Xabier Legaspi Juanatey.

- Duration: 2024-2026
- Local coordinator: Damien Hardy
- Collaboration between the PACAP team in the context of the SCI3D project and the following companies: Cosmyx and Vistory.

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: Marc Feeley (feeley@iro.umontreal.ca)

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 European initiatives

10.2.1 Other European programs/initiatives

Participants: Isabelle Puaut.

CERCICAS (COST Action): The entanglement of hardware components in emerging platforms and the complex behavior of parallel applications raise conflicting resource requirements, more so in smart, (self-)adaptive and autonomous systems. This scenario presents the hard challenge of understanding and controlling, statically and dynamically, the trade-offs in the usage of system resources, (time, space, energy, and data), also from the perspective of the development and maintenance efforts. CERCICAS aims at (1) networking otherwise fragmented research efforts; (2) leveraging appropriate educational and technology assets to improve the understanding and management of resources by the academia and industry of underperforming economies, in order to promote cooperation inside Europe and achieve economical and societal benefits.

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
- 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: 2020-2024
- 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.

NOP: Safe and Efficient Intermittent Computing for a Batteryless IoT

Participants: Isabelle Puaut, Hugo Reymond, Erven Rohou.

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

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

1. develop sound formal foundations for intermittent systems,
2. develop precise predictive energy models of a whole node (including both harvesting and consumption) usable for online decision making,
3. significantly improve the energy efficiency of run-time support for intermittency,
4. develop techniques to provide formal guarantee through static analysis of the systems behavior (forward progress),
5. develop a proof of concept: an intermittent system for bird recognition by their songs, to assess the costs and benefits of the proposed solutions.

- website: project.inria.fr/nopcl/

NOPAL: Characterization of energy-aware design for battery-less devices

Participants: Isabelle Puaut, Hugo Reymond, Erven Rohou.

- Funding: PIA LabEx CominLabs
- Duration: 2024–2025
- Local coordinator: Erven Rohou
- Partners: IRISA/Granit Lannion, LS2N/STR Nantes, IETR/Syscom Nantes
- NOPAL is a follow-up action to NOP. Its goal is to assess the benefits of the designed platform over its entire lifecycle, from design phase, through operation phase, up to late operation phase.

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.

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: 2021-2024
- 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/

Ofast3D

Participants: Pierre Bedell, Damien Hardy, Imane Lasri, Camille Le Bon, Erven Rohou.

- Funding: Inria Exploratory Action
- Duration: 2022-2024
- Local coordinator: Damien Hardy

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

SCI3D

Participants: 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.

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: 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: Damien Hardy, 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 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.

10.4 Regional initiatives

PluriNOP: Région Bretagne + Cyberschool

PluriNOP focuses on assessing vulnerabilities in program binaries when fault injection means allow for frequent and precise fault. Assuming two different fault models (instruction skip and instruction replay), we show that many hardening techniques can be defeated.

11 Dissemination

Participants: Nicolas Bailluet, Pierre Bedell, Hector Chabot, Niels Cobat, Caroline Collange, Antoine Gicquel, Damien Hardy, Sara Sadat Hoseininasab, Camille Le Bon, Pierre Michaud, Aurore Poirier, Isabelle Puaut, Hugo Reymond, Matthieu Rodet, Erven Rohou, Thomas Rubiano, André Seznec.

11.1 Promoting scientific activities

11.1.1 Scientific events: selection

Member of the conference program committees

- T. Rubiano was a PC member of the 19th Conference on Logical and Semantic Frameworks with Applications: LSFA.
- I. Puaut was a PC member of the Euromicro Conference on Real Time Systems (ECRTS) 2024 and 2025.

- I. Puaut was a PC member of the Workshop on Worst-Case Execution Time Estimation (WCET) 2024.
- I. Puaut was a PC member of the Code Generation and Optimization (CGO) 2025.
- P. Michaud was a member of the Program Committee of the PACT 2024 conference.

Reviewer

- P. Michaud was a member of the External Program Committee of the ISCA 2024, MICRO 2024, and HPCA 2025 conferences.

11.1.2 Journal

Member of the editorial boards

- I. Puaut is associate editor of the Springer International Journal of Time-Critical Computing Systems.
- A. Seznec is an associate editor of the ACM Journal on Transactions on Architecture and Code Optimization.

Reviewer - reviewing activities

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

11.1.3 Invited talks

- C. Collange gave an invited talk at *Collège de France* in Mar 2024.
- I. Puaut gave a keynote presentation entitled "Machine learning for timing estimation: the good, the bad and the ugly", at the WCET Workshop, Jul 2024 [29].

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).
- I. Puaut is member of the steering committee of the WCET workshop, satellite workshop to ECRTS

11.1.5 Research administration

- 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.
- I. Puaut was member of the best paper selection committee for RTAS 2024 and ECRTS 2024.
- E. Rohou is the contact for international relations for Inria Rennes Bretagne Atlantique (for scientific matters).

11.2 Teaching - Supervision - Juries

11.2.1 Teaching

- Master: C. Collange, GPU programming, 20 hours, M1, Université de Rennes, France
- Master: D. Hardy, Operating systems, 59 hours, M1, Université de Rennes, France
- Master: D. Hardy, Students project, 30 hours, M1, Université de Rennes, France
- Licence: D. Hardy, Real-time systems, 95 hours, L3, Université de 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
- Licence: I. Puaut, Language-Machine Interface (RISC-V programming + basic computer micro-architecture), L3, 20 hours
- Licence: N. Bailluet, C and network programming, 20 hours, L3, ENS Rennes, France
- Master: N. Bailluet, Software Exploitation (SE), 20 hours, M1-cybersecurity, Université de Rennes, France
- Licence: A. Poirier, ALG2 (Algorithmique 2), 18 hours, L2, Université de Rennes, France
- Master: A. Poirier, NFS (Network For Security), 15 hours, M1, Université de Rennes, France
- Licence: H. Chabot, CMPL (Compilation et programmation par la syntaxe), 20.5 hours, L3, Université de Rennes, France
- Licence: H. Chabot, BAD (Base de données), 9 hours, L2, Université de Rennes, France
- Licence: H. Chabot, PO (Programmation Objet), 15 hours, L2, Université de Rennes, France
- Licence: H. Chabot, SYRE (Base de système et réseaux), 18.5 hours, L3, Université de Rennes, France
- Master: M. Rodet, Labs of Programming, 44.5 hours, M2, ENS Rennes, France
- Master: M. Rodet, Low Level Programming, 19.5 hours, M1, Université de Rennes, France
- Master: E. Rohou, Veille Technologique, 12h, M2 Cyber, Université de Rennes, France

11.2.2 Supervision

- PhD: Hugo Reymond, *Energy-aware execution model in intermittent systems* [24], Nov 2024, advisors I. Puaut, E. Rohou, S. Faucou (LS2N Nantes), J.-L. Béchennec (LS2N Nantes). Funding: CominLabs project NOP.
- PhD: Antoine Gicquel, *Étude de vulnérabilité d'un programme au format binaire en présence de fautes précises et nombreuses : métriques et contremesures* [23], Dec 2024, advisors D. Hardy, E. Rohou, K. Heydemann (Sorbonne Université). Funding: grant from Région Bretagne + Cyberschool.
- 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: Sara Hoseininasab, *Automatic synthesis of multi-thread pipelines*, started Nov 2021, advisors C. Collange (70 %) and S. Derrien (30 %, TARAN). Funding: ANR project DYVE.

- 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: Nicolas Bailluet, *Approches par modification de code machine pour la défense contre les attaques ROP et JOP*, started Sep 2022, advisors I. Puaut (50 %) and E. Rohou (50 %). Funding: grant from ENS.
- 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, *Analysis and optimization of 3D printing files with machine learning tools*, started Oct 2024, advisors D. Hardy (50 %), R. Gaudel (50 %, MALT). Funding: university grant (*Contrat Doctoral*).

11.2.3 Juries

I. Puaut was in the following PhD theses and habilitation juries:

- Jean-Michel Gorius. *Synthèse de haut niveau de processeurs à jeu d'instructions.* Université de Rennes, Dec 2024 (member and president of jury)
- Florian Brandner. *Towards formal predictability and analyzability in real-time systems.* Habilitation à diriger des recherches, Institut Polytechnique de Paris, Oct 2024 (reviewer)
- Emad Jakob Maroun. *Compiling for time-predictability and performance.* TU Wien, Austria, Sep 2024. (external reviewer)
- Thomas Carle, *Parallelism and timing predictability in real-time systems.* Habilitation à diriger des recherches, Université de Toulouse, Jun 2024 (reviewer)
- Iryna de Albuquerque Silva. *Implémentation certifiable et efficace de réseaux de neurones sur des systems temps réel embarqués critiques.* Université de Toulouse, Jul 2024 (member and president of jury)

I. Puaut is member of the CSID commitee of Jean-Michel Gorius and Constance Bocquillon.
E. Rohou was in the following PhD theses juries:

- Lorenzo Casalino, *(On) The Impact of the Micro-architecture on Countermeasures against Side-Channel Attacks*, Sorbonne Université, Jan 2024 (member)
- Marie Badaroux, *Dynamic Binary Translation speed and accuracy trade-offs : investigating parallel scalability and cache simulation*, Université Grenoble Alpes, Oct 2024 (reviewer)
- Gianpietro Consolaro, *Configurable Polyhedral Scheduling for All-Scenario Deep Learning Compilers*, Université Paris Sciences et Lettres, Sep 2024. (reviewer)

E. Rohou is member of the CSID commitee of Ikram Dendani, Jean-Loup Hatchikian-Houdot, Bruno Mateu.

11.3 Popularization

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

- Blog article related to PACAP's research: Featured Work: Microarchitecture Security: The Spectre Affair, RISC-V Blog, Aug 22nd, 2024
- E. Rohou was a member of the Inria Rennes working group on sustainable development and contributed to the report on GHG emissions caused by travels [25].

11.3.2 Participation in Live events

- E. Rohou was invited to present the job of a researcher to secondary-school students (*classe de 4e*) at Collège de Bourgchevreuil, Cesson-Sévigné.
- P. Bedell, D. Hardy and C. Le Bon presented and demonstrated the Ofast3D Inria exploratory action in the Tech Inn'Vitré Exhibition in Vitré (Feb 2024).

red-

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. 6, 7).

[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. 7).

[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. 7, 8).

[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. 8).

[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. 7, 9).

[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. 7, 9).

[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. 8).

[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. 9).

12.2 Publications of the year

International journals

[18] 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. 17).

International peer-reviewed conferences

[19] A. N. Amalou, E. Fromont and I. Puaut. 'Fast and Accurate Context-Aware Basic Block Timing Prediction using Transformers'. In: *Proceedings of the ACM SIGPLAN 2024 International Conference on Compiler Construction*. CC 2024 - ACM SIGPLAN 33rd International Conference on Compiler Construction. Edimbourg, United Kingdom: ACM; ACM, 2nd Mar. 2024, pp. 227–237. DOI: 10.1145/3640537.3641572. URL: https://hal.science/hal-04406073 (cit. on p. 19).

[20] H. Reymond, A. N. Amalou and I. Puaut. 'WORTEX: Worst-Case Execution Time and Energy Estimation in Low-Power Microprocessors using Explainable ML'. In: *Open Access Series in Informatics*. WCET 2024 - 22nd International Workshop on Worst-Case Execution Time Analysis. Vol. 121. 1. Lille, France: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 1:1–1:14. DOI: 10.4230/OASIcs.WCET.2024.1. URL: https://hal.science/hal-04646477 (cit. on p. 18).

[21] H. Reymond, J.-L. Béchennec, M. Briday, S. Faucou, I. Puaut and E. Rohou. 'EarlyBird: Energy belongs to those who wake up early'. In: *Proceedings of the 30th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2024)*. RTCSA 2024 - 30th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. Sokcho, South Korea: IEEE, 2024, pp. 1–10. DOI: 10.1109/RTCSA62462.2024.00011. URL: https://hal.science/hal-04663862 (cit. on pp. 11, 16).

[22] H. Reymond, J.-L. Béchennec, M. Briday, S. Faucou, I. Puaut and E. Rohou. 'SCHEMATIC: Compile-time checkpoint placement and memory allocation for intermittent systems'. In: *Proceedings of the IEEE/ACM International Symposium on Code Generation and Optimization (CGO'24)*. IEEE/ACM International Symposium on Code Generation and Optimization (CGO'24). Edinburgh, United Kingdom: IEEE, 2nd Mar. 2024, pp. 258–269. DOI: 10.1109/CGO57630.2024.10444789. URL: https://hal.science/hal-04345348 (cit. on pp. 11, 16).

Doctoral dissertations and habilitation theses

[23] A. Gicquel. 'Vulnerability assessment of a binary program in the presence of numerous and precise faults'. Université de Rennes, 12th Dec. 2024. URL: https://hal.science/tel-04842415 (cit. on pp. 20, 30).

[24] H. Reymond. 'Energy-aware execution model for intermittent systems'. Université de Rennes, 8th Nov. 2024. URL: https://hal.science/tel-04778581 (cit. on pp. 16, 30).

Reports & preprints

[25] E. Bannier, S. Castellan, S. Derrien, F. Galassi, L. Garnier, L. Hoyet, A. l'Azou, N. Lahaye, M. J.-M. Macé, O. Martineau, A. Masson, T. Maugey, B. Ninassi, E. Rohou, M. Simonin and F. Taïani. *Reducing GHG emissions from business travel: A collaborative approach at IRISA/Inria*. Groupe de travail « missions » IRISA / Centre Inria de l'Université de Rennes, Mar. 2024, pp. 1–16. URL: https://univ-rennes.hal.science/hal-04506138 (cit. on p. 31).

[26] H. Chabot, I. Puaut, T. Carle and H. Cassé. *Marmot: Extraction of Fine-Grain Memory Access Profiles for real-time software*. 2024. URL: https://hal.science/hal-04782265 (cit. on p. 19).

[27] P. Michaud. *What about a two-dimensional virtual address space?* RR-9563. Inria, Dec. 2024, p. 27. URL: https://inria.hal.science/hal-04816363 (cit. on p. 18).

[28] A. Seznec. *TAGE: an engineering cookbook*. 9561. Inria, Nov. 2024, pp. 1–73. URL: https://hal.science/hal-04804900 (cit. on p. 17).

Other scientific publications

[29] I. Puaut. *Machine Learning for Timing Analysis: The Good, the Bad and the Ugly*. Lille, France, 2024. DOI: 10.4230/OASIcs.WCET.2024.7. URL: https://inria.hal.science/hal-04672502 (cit. on p. 29).

[30] H. Reymond, H. Chabot, A. N. Amalou and I. Puaut. *MSP430FR5969 Basic Block Worst Case Energy Consumption (WCEC) and Worst Case Execution Time (WCET) dataset*. 2024. DOI: 10.5281/zenodo.11066623. URL: https://hal.science/hal-04767698 (cit. on p. 15).

Software

[31] [SW] H. Reymond, *SCHEMATIC* version 1.0, 29th Nov. 2024. LIC: MIT License. HAL: ⟨hal-04581100⟩, URL: https://hal.science/hal-04581100, VCS: https://gitlab.inria.fr/hreymond/schematic, SWHID: ⟨swh:1:dir:34080da109472c507b22c9320efef679af7a3a19;origin=https://hal.archives-ouvertes.fr/hal-04581100;visit=swh:1:snp:9d795ab4d12c55b6a805222a3846c2ea70196f40;anchor=swh:1:rel:87983351a70daa392b67729ebf0d64d1d1170868;path=/⟩ (cit. on p. 16).

12.3 Cited publications

[32] 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. 6, 7).

[33] 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. 17).

[34] M. Hataba, A. El-Mahdy and E. Rohou. 'OJIT: A Novel Obfuscation Approach Using Standard Just-In-Time Compiler Transformations'. In: *International Workshop on Dynamic Compilation Everywhere*. Jan. 2015 (cit. on p. 10).

[35] 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. 9).

[36] 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. 7).

[37] 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. 7, 9).

[38] 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. 10).

[39] 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. 7).

[40] 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. 8).

[41] 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. 7, 9).

[42] 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. 8).

[43] 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. 10).