Activity Report 2022

Team PACAP

Pushing Architecture and Compilation for Application Performance

Joint team with Centre Inria de l'Université de Rennes

D3 – Architecture

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 . . . 6
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 . . . 9
3.2.7 Security . . . 10

4 Application domains 11
4.1 Domains . . . 11

5 Highlights of the year 11
5.1 Awards . . . 11

6 New software and platforms 11
6.1 New software . . . 11
6.1.1 ATMI . . . 11
6.1.2 HEPTANE . . . 12
6.1.3 tiptop . . . 12
6.1.4 GATO3D . . . 13
6.1.5 energy-meter . . . 13
6.1.6 Static-dynamic performance autotuner . . . 13
6.1.7 Sorry . . . 13
6.1.8 Damas . . . 14

7 New results 14
7.1 Compilation and Optimization . . . 14
7.1.1 Compilation for Intermittent Systems . . . 14
7.1.2 Dynamic Binary Analysis and Optimization . . . 15
7.1.3 Accurate 3D printing time estimation . . . 15
7.2 Processor Architecture . . . 15
7.2.1 Energy-efficient microarchitecture . . . 15
7.2.2 Automatic synthesis of multi-thread pipelines . . . 16
7.2.3 High-level synthesis for quantum circuit simulation . . . 16
7.2.4 Two-dimensional memory architecture . . . 17
7.3 WCET estimation and optimization . . . 17
7.3.1 Revisiting iterative compilation for WCET minimization . . . 17
7.3.2 Using machine learning for timing analysis of complex processors . . . 18
7.3.3 WCET estimation for multi-core targets . . . 18
7.4 Security . . . 18
7.4.1 Verification of Data Flow Integrity for Real-Time Embedded Systems . . . 18
7.4.2 Multi-nop fault injection attack . . . 19
7.4.3 Platform for adaptive dynamic protection of programs . . . 19

7.4.4 Compiler approaches to protect against ROP and JOP attacks . . . 19

8 Bilateral contracts and grants with industry 20
8.1 Bilateral contracts with industry . . . 20

9 Partnerships and cooperations 20
9.1 European initiatives . . . 20
9.1.1 H2020 projects . . . 20
9.2 National initiatives . . . 21
9.3 Regional initiatives . . . 26

10 Dissemination 26
10.1 Promoting scientific activities . . . 26
10.1.1 Scientific events: selection . . . 26
10.1.2 Journal . . . 27
10.1.3 Leadership within the scientific community . . . 27
10.1.4 Research administration . . . 27
10.2 Teaching - Supervision - Juries . . . 27
10.2.1 Teaching . . . 27
10.2.2 Supervision . . . 28
10.2.3 Juries . . . 28
10.3 Popularization . . . 29
10.3.1 Internal or external Inria responsibilities . . . 29
10.3.2 Education . . . 29
10.3.3 Interventions . . . 29

11 Scientific production 29
11.1 Major publications . . . 29
11.2 Publications of the year . . . 30
11.3 Cited publications . . . 31

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

A1.1.11. – Quantum 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, HDR]
- Caroline Collange [INRIA, Researcher]
- Pierre Michaud [INRIA, Researcher]

Faculty Members

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

PhD Students

- Abderaouf Nassim Amalou [UNIV RENNES I]
- Nicolas Bailluet [UNIV RENNES I, from Sep 2022]
- Nicolas Bellec [UNIV RENNES I, until Nov 2022]
- Antoine Gicquel [INRIA]
- Sara Sadat Hoseininasab [INRIA]
- Anis Peysieux [INRIA]
- Lucien Poirier [INRIA, from Oct 2022]
- Hugo Reymond [INRIA]

Technical Staff

- Pierre Bedell [INRIA, Engineer, from Oct 2022]
- Oussama Houidar [UNIV RENNES I, Engineer, until Feb 2022]
- Camille Le Bon [INRIA, Engineer, from Feb 2022]
- Mohammed Mehdi Merah [UNIV RENNES I, Engineer, from Sep 2022]

Interns and Apprentices

- Nicolas Bailluet [UNIV RENNES I, from Feb 2022 until Jul 2022]
- Hector Chabot [INRIA, from May 2022 until Aug 2022]
- Audrey Fauveau [INRIA, from Jun 2022 until Aug 2022]
- Brian Gentile [UNIV RENNES i, from May 2022 until Jul 2022]
- Mohammed Mehdi Merah [UNIV RENNES I, from May 2022 until Jul 2022]
- Lucien Poirier [INRIA, from Feb 2022 until Jul 2022]

Administrative Assistant

- Virginie Desroches [INRIA]

2 Overall objectives

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

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

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

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

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

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

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

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

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

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

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

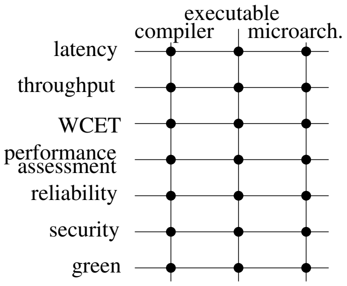

Figure 1: Perimeter of Research Objectives

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

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

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

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

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

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

3 Research program

3.1 Motivation

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

3.1.1 Technological constraints

Until recently, binary compatibility guaranteed portability of programs, while increased clock frequency and improved micro-architecture provided increased performance. However, in the last decade, advances in technology and micro-architecture started translating into more parallelism instead. Technology roadmaps even 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 map efficiently computations to compute nodes.

3.1.3 Domain constraints

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

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

Note on Applications Domains.

PACAP works on fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impact on any application domain that requires high performance execution (telecommunication, multimedia, biology, health, engineering, environment...), but also on many embedded applications that exhibit other constraints such as power consumption, code size and guaranteed response time.

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

3.2 Research Objectives

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

Heterogeneity and diversity of processor architectures now require new techniques to guarantee that the hardware is satisfactorily exploited by the software. One of our goals is to devise new static compilation techniques (cf. Section 3.2.1), but also build upon iterative [1] and split [23] compilation to continuously adapt software to its environment (Section 3.2.2). Dynamic binary optimization will also play a key role in delivering adapting software and increased performance.

The end of Moore's law and Dennard's scaling [2] offer an exciting window of opportunity, where performance improvements will no longer derive from additional transistor budget or increased clock frequency, but rather come from breakthroughs in micro-architecture (Section 3.2.3). Reconciling CPU and GPU designs (Section 3.2.4) is one of our objectives.

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

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

3.2.1 Static Compilation

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

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

We propose to build upon iterative compilation [1] to explore how applications perform on different configurations. When possible, Pareto points are related to application characteristics. The best configuration, however, may actually depend on runtime information, such as input data, dynamic events, or properties that are available only at runtime. Unfortunately a runtime system has little time and means to determine the best configuration. For these reasons, we also leverage split-compilation [23]: 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 [30, 21] 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 [32] [28] and in cache prefetching [5] and they have revived the forgotten concept of value prediction [8] [7]. This research was supported by the ERC Advanced grant DAL (2011-2016) and also by Intel. We pursue research on achieving ultimate unicore performance. Below are several non-orthogonal directions that we have identified for mid-term research:

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

Memory design issues:

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

The *Data Prefetching Contest* held with ISCA 2015 has illustrated that achieving high prefetching efficiency is still a challenge for wide-issue superscalar processors, particularly those featuring a very large instruction window. The large instruction window enables an implicit data prefetcher. The interaction between this implicit hardware prefetcher and the explicit hardware prefetcher is still relatively mysterious as illustrated by Pierre Michaud's BO prefetcher (winner of DPC2) [5]. The first research objective is to better understand how the implicit prefetching enabled by the large instruction window interacts with the L2 prefetcher and then to understand how explicit prefetching on the L1 also interacts with the L2 prefetcher.

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

The prefetcher is not the only actor in the hierarchy that must be carefully controlled. Significant benefits can also be achieved through careful management of memory access bandwidth, particularly the management of spatial locality on memory accesses, both for reads and writes. The exploitation of this locality is traditionally handled in the memory controller. However, it could be better handled if larger temporal granularity was available. Finally, we also intend to continue to explore the promising avenue of compressed caches. In particular we proposed the skewed compressed cache [11]. It offers new possibilities for efficient compression schemes.

Ultra wide-issue superscalar.

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

The first direction that we are exploring is the use of clustered architectures [6]. Symmetric clustered organization allows to benefit from a simpler bypass network, but induce large complexity on the issue queue. One remarkable finding of our study [6] is that, when considering two large clusters (e.g. 8-wide), steering large groups of consecutive instructions (e.g. 64 μops) to the same cluster is quite efficient. This opens opportunities to limit the complexity of the issue queues (monitoring fewer buses) and register files (fewer ports and physical registers) in the clusters, since not all results have to be forwarded to the other cluster.

The second direction that we are exploring is associated with the approach that we developed with Sembrant et al. [31]. 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 [33].

In speculative execution, we have recently revisited value prediction (VP) which was a hot research topic between 1996 and 2002. However it was considered until recently that value prediction would lead to a huge increase in complexity and power consumption in every stage of the pipeline. Fortunately, we have recently shown that complexity usually introduced by value prediction in the OoO engine can be overcome [8][7] [32] [28]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [8]. Thus, both prediction validation and recovery by squashing can be done outside the out-of-order engine, at commit time. Furthermore, we propose a new pipeline organization, EOLE ({Early | Out-of-order | Late} Execution), that leverages VP with validation at commit to execute many instructions outside the OoO core, in-order [7]. With EOLE, the issue-width in OoO core can be reduced without sacrificing performance, thus benefiting the performance of VP without a significant cost in silicon area and/or energy. In the near future, we will explore new avenues related to value prediction. These directions include register equality prediction and compatibility of value prediction with weak memory models in multiprocessors.

3.2.4 Towards heterogeneous single-ISA CPU-GPU architectures

Heterogeneous single-ISA architectures have been proposed in the literature during the 2000's [27] 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 [15] opens the way for heterogeneous architectures including latency-optimized superscalar cores and throughput-optimized GPU-style cores, which all share the same instruction set. Using SIMT cores in place of superscalar cores will enable the highest energy efficiency on regular sections of applications. As with existing single-ISA heterogeneous architectures, task migration will not necessitate any software rewrite and will accelerate existing applications.

Compilers for emerging heterogeneous architectures.

Single-ISA CPU+GPU architectures will provide the necessary substrate to enable efficient heterogeneous processing. However, it will also introduce substantial challenges at the software and firmware level. Task placement and migration will require advanced policies that leverage both static information at compile time and dynamic information at run-time. We are tackling the heterogeneous task scheduling problem at the compiler level.

3.2.5 Real-time systems

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

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

1. definition of methods to estimate WCETs tightly on manycores, that smartly analyze and/or control shared resources such as buses, NoCs or caches;

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

3.2.6 Power efficiency

PACAP addresses power-efficiency at several levels. First, we design static and split compilation techniques to contribute to the race for Exascale computing (the general goal is to reach 10^{18} FLOP/s at less than 20 MW). Second, we focus on high-performance low-power embedded compute nodes. Within the ANR project Continuum, in collaboration with architecture and technology experts from LIRMM and the SME Cortus, we 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 [34] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [26], or thread-based control flow mangling [29]). Still, security is not core competence of PACAP members.

Our strategy consists in partnering with security experts who can provide intuition, know-how and expertise, in particular in defining threat models, and assessing the quality of the solutions. Our expertise in compilation and architecture helps design more efficient and less expensive protection mechanisms.

Examples of collaborations so far include the following:

Compilation: We partnered with experts in security and codes to prototype a platform that demonstrates resilient software. They designed and proposed advanced masking techniques to hide sensitive data in application memory. PACAP's expertise is key to select and tune the protection mechanisms developed within the project, and to propose safe, yet cost-effective solutions from an implementation point of view.

Dynamic Binary Rewriting: Our expertise in dynamic binary rewriting combines well with the expertise of the CIDRE team in protecting application. Security has a high cost in terms of performance, and static insertion of 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 Highlights of the year

5.1 Awards

- Nicolas Bellec, Isabelle Puaut and their co-authors received an outstanding paper award for their ECRTS 2022 paper "RT-DFI: Optimizing Data-Flow Integrity for Real-Time Systems" [18].
- André Seznec received the Inria - Académie des sciences - Dassault Systèmes Innovation Award. This prize rewards his multiple research and discoveries on microprocessor architecture.

6 New software and platforms

6.1 New software

6.1.1 ATMI

Keywords: Analytic model, Chip design, Temperature

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

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

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

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

Contact: Pierre Michaud

Participant: Pierre Michaud

6.1.2 HEPTANE

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

Scientific Description: WCET estimation

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

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

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

Contact: Isabelle Puaut

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

Partner: Université de Rennes 1

6.1.3 tiptop

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

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

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display. Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.

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

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

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

Contact: Erven Rohou

Participant: Erven Rohou

6.1.4 GATO3D

Keywords: Code optimisation, 3D printing

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

Authors: Damien Hardy, Erven Rohou

Contact: Erven Rohou

6.1.5 energy-meter

Keywords: Monitoring, Power consumption, Linux, Applications

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

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

Author: Erven Rohou

Contact: Erven Rohou

6.1.6 Static-dynamic performance autotuner

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

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

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

Author: Erven Rohou

Contact: Erven Rohou

6.1.7 Sorry

Keywords: Dynamic Analysis, Security, High performance computing

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

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

Author: Camille Le Bon

Contact: Erven Rohou

6.1.8 Damas

Keyword: Cybersecurity

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

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

Author: Camille Le Bon

Contact: Erven Rohou

7 New results

Participants: Abderaouf Nassim Amalou, Nicolas Bellec, Caroline Collange, Audrey Fauveau, Antoine Gicquel, Damien Hardy, Oussama Houidar, Camille Le Bon, Pierre Michaud, Valentin Pasquale, Anis Peysieux, Isabelle Puaut, Erven Rohou.

7.1 Compilation and Optimization

Participants: Pierre Bedell, Caroline Collange, Audrey Fauveau, Damien Hardy, Oussama Houidar, Mohammed Mehdi Merah, Lucien Poirier, Isabelle Puaut, Hugo Reymond, Erven Rohou

Participants: Pierre Bedell, Damien Hardy, Oussama Houidar, Mohammed Mehdi Merah, Lucien Poirier, Isabelle Puaut, Hugo Reymond, Erven Rohou.

7.1.1 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 (e.g. photovoltaic, piezoelectric, or thermoelectric) and store it in a capacitor. 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 propose ELOISE, a compiler technique that automates checkpoint placement and memory allocation to minimize overall energy consumption. Compared to existing systems, ELOISE is safe by design, in the sense that any program will eventually terminate (*forward progress* property), adapting checkpoint placement and memory allocation to architectural parameters (size of the energy buffer, capacity of volatile memory). We tested ELOISE for different experimental settings (size of volatile memory and capacitor) and results show an average energy reduction of 46 % compared to related techniques.

7.1.2 Dynamic Binary Analysis and Optimization

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

Since the creation of JavaScript, web browsers embedding it have been relying on JIT compilers or even interpreters to execute it, a choice that seems natural because of the dynamic aspect of the language. However, researchers have wondered whether it is possible to produce ahead-of-time compiled code that allows dynamic languages to be used in contexts that do not allow the use of JIT compilers (due to limited resources, security policies, response time...) In order to implement performance measurement methods that feed back information from the processor to the high-level languages and design new ways to compile JavaScript, we explored ways to exploit low-level performance measurement from hardware counters.

7.1.3 Accurate 3D printing time estimation

Participants: Pierre Bedell, Damien Hardy, Oussama Houidar, Mohammed Mehdi Merah
Context: Inria Exploratory Action Ofast3D
External collaborators: MimeTIC 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 [25]. To reach that goal, we have modified existing 3D printer firmwares: Klipper and Marlin in simulation mode to determine the timing per G-code instruction (the language interpreted by 3D printers). The validation is in progress, the first results are promising and reveal a very good accuracy while the simulation is fast.

See also the GATO3D in Section 6.1.4.

7.2 Processor Architecture

Participants: Caroline Collange, Erven Rohou, Audrey Fauveau, Sara Sadat Hoseininasab, Pierre Michaud, Anis Peysieux.

7.2.1 Energy-efficient microarchitecture

Participants: Pierre Michaud, Anis Peysieux

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

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

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

We propose a dual-banking scheme to reduce the power consumption of the IRF: the odd bank holds odd-numbered physical registers, and the even bank holds even-numbered ones [16]. Half of the read ports come from the odd bank, the other half from the even bank. This way the number of read ports per bank is halved, and the area and power of the IRF is roughly halved. Execution pipes with two read ports, such as ALUs, have one read port in the odd bank and the other read port in the even bank. If a 2-input micro-op happens to have its two source operands in the same bank, this is a bank conflict, and the micro-op reads its two operands sequentially. Bank conflicts hurt performance, as each conflict generates a one-cycle penalty. To minimize the performance loss, we propose a new register renaming scheme that allocates physical registers so as to reduce the number of bank conflicts. Our simulations show that, with this new scheme, very little performance is lost from banking the IRF [16].

After the IRF, we have focused our attention on another hot spot: the scheduler. In a superscalar microarchitecture, the scheduler is the circuit orchestrating the out-of-order execution of instructions. It consists of an instruction queue (IQ) and a picker. Instructions to be executed are inserted into the IQ and, when they are ready to execute, they are selected by the picker and issued for execution. At the same time, signals are broadcasted to the IQ to wake up the instructions depending on the just-issued ones so that they can themselves be issued in the next clock cycle. The schedule loop cannot be pipelined, as this would hurt processor performance. Reducing the clock frequency or the IQ size would also hurt processor performance. Therefore, the scheduler employs aggressive circuit design techniques rising its power density and temperature.

To loosen the schedule loop and reduce the scheduler energy consumption, we are exploring the possibility to pipeline the wake-up partially. We partition the IQ into two halves. The instructions in the first partition are woken up immediately, as in a conventional scheduler. However, the instructions in the second partition are woken up one cycle later. The problem is then to find about 50 % of instructions that can tolerate the extra delay.

Our first idea was to reuse the method proposed by Sembrant et al. to classify instructions into *urgent* and *non-urgent* instructions [31], the urgent ones being defined as the instructions on the backward slice of load instructions missing the last-level cache. However, our simulations show that the instructions classified as non-urgent do not always tolerate the extra cycle, so using this classification on our partitioned IQ hurts performance on some workloads. We are currently exploring a new approach. We start from instructions that can obviously tolerate the extra cycle, in particular store instructions that do not forward their value to a contemporaneous load and branch instructions that the branch predictor predicts correctly most of the time. We are currently searching a systematic way to identify more instructions that can tolerate the extra delay.

7.2.2 Automatic synthesis of multi-thread pipelines

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

Capitalizing on early results at the intersection of hardware synthesis, compilers, and micro-architecture, the goal of this project is to extend high-level synthesis to generate automatically the micro-architectural features of modern processors and GPUs, such as hardware multi-threading. The outcome of the PhD will ease the development of custom embedded processors by synthesising them directly from a high-level description rather than require manual hardware design at the RTL level.

7.2.3 High-level synthesis for quantum circuit simulation

Participants: Audrey Fauveau, Caroline Collange
Context: PEPR EPIQ
External collaborator: Gildas Ménier, UBS.

Simulation of quantum computing circuits on classical hardware is a performance and memory-intensive application that could take advantage of FPGA acceleration. We take advantage of High-level synthesis tools to develop generic quantum circuit simulation acceleration tools on FPGAs, whose fast turnaround time enables efficient design space exploration from a single source code, and let us identify the most advantageous configurations in terms of area and throughput.

7.2.4 Two-dimensional memory architecture

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

The Maplurinum project revisits the foundations of computer architecture and operating systems for cloud computing and high-performance computing, to better manage the growing heterogeneity of hardware components. A 128-bit address space is considered as one of the possible solutions to achieve this goal. As a participant of the Maplurinum project, the PACAP team explores some of the architectural implications of a 128-bit address space.

Instead of representing an address as a single value like existing instruction-set architectures (ISAs) do, we consider the possibility to represent an address as two 64-bit values: an X coordinate and a Y coordinate. Our intuition is that the data structures used by programmers are often multi-dimensional, and that mapping a multi-dimensional data structure onto a one-dimensional address space degrades the spatial locality of memory accesses. Spatial locality is an empirical, qualitative property of programs stating that memory addresses that are spatially close to each other are likely to be accessed during the same period of time. This property is heavily exploited by high-performance microarchitectures. Nevertheless, there exists workloads processing a large amount of data with poor spatial locality. Performance-wise, these workloads suffer from long-latency memory accesses. Our objective is to answer the following question: can a two-dimensional (2D) address space improve the spatial locality of memory accesses?

To study a 2D address space, ideally, one would need to create a new ISA, adapt a compiler and an operating system to this ISA, invent a new microarchitecture and write a simulator to evaluate performance. This would be a daunting task. Our goal is precisely to bring qualitative and quantitative arguments that would justify such endeavor.

To get around this chicken-and-egg problem, we wrote a C++ library called MXY allowing to write algorithms in a language resembling the C programming language. Executing an algorithm written with MXY generates an address trace which can then be analyzed. The current, early version of MXY emulates a conventional one-dimensional memory. On the algorithms that we have tested so far (matrix multiplication, quicksort, FFT, LU factorization), the addresses traces generated with MXY reproduce accurately the main (i.e., independent of compiler optimizations) spatiotemporal characteristics of the address traces obtained from instrumenting binaries generated by compiling C programs with the gcc compiler. We are currently looking for more algorithms and data structures to use as benchmarks. Once we have sufficient confidence that MXY is flexible enough to reproduce the memory behavior of any C program, we will use MXY to explore 2D memory architectures.

7.3 WCET estimation and optimization

Participants: Abderaouf Nassim Amalou, Valentin Pasquale, Isabelle Puaut.

7.3.1 Revisiting iterative compilation for WCET minimization

Participants: Valentin Pasquale, Isabelle Puaut

Static Worst-Case Execution Time (WCET) estimation techniques take as input the binary code of a program and output a conservative estimate of its execution time. While compilers, and iterative compilation, usually optimize for the average-case, previous work such as [24] has shown that it is also possible to use existing optimization and iterative compilation techniques to lower the WCET estimates drastically. In this work [20] we revisit the use of iterative compilation for WCET minimization and show that previous work can be improved both in terms of complexity and reduction of WCET estimates. In particular, we found that the use of long chains of compilation flags, from a few hundred to a few thousand, allows a significant reduction of WCET estimates, of 35 % on average, and up to 70 % on some benchmarks, compared to the best compilation level (-O0 .. -O3) applicable. These gains are significantly better than the reductions of WCET estimates obtained by Dardaillon et al. [24], which, on the same benchmarks and experimental conditions, reduce the WCET estimates by 20 % on average.

7.3.2 Using machine learning for timing analysis of complex processors

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

Modern processors raise a challenge for timing estimation in general, and WCET estimation in particular, since detailed knowledge of the processor microarchitecture is not available. In this work [17], we present CATREEN, a recurrent neural network able to predict the steady-state execution time of each basic block in a program. Contrarily to other models, CATREEN can take into account the *execution context* formed by the previously executed basic blocks which allows accounting for the processor micro-architecture without explicit modeling of micro-architectural elements (caches, pipelines, branch predictors, etc.). The evaluations conducted with synthetic programs and real ones (programs from Mibench and Polybench) show that CATREEN can provide accurate prediction for execution time with 11.4 % and 16.5 % error on average, respectively and that we obtained an improvement of 18 % and 27.6 % respectively when comparing our tool estimations to the state-of-the-art LSTM-based model. Our ongoing work consists in enlarging the scope of application of CATREEN to real-time systems, where WCET are needed (CATREEN currently provides average-case timing values and not worst-case timing values).

This work is done in collaboration with Elisa Fromont, from the LACODAM team, who co-supervises the PhD thesis of Abderaouf Nassim Amalou.

7.3.3 WCET estimation for multi-core targets

Participants: Isabelle Puaut

Accesses to shared resources in multi-core systems raise predictability issues. The delay in accessing a resource for a task executing on a core depends on concurrent resource sharing from tasks executing on the other cores. In this work, we describe StAMP [19], a compiler technique that splits the code of tasks into a sequence of code intervals intervals, each with a distinct worst-case memory access profile. The intervals identified by StAMP can serve as inputs to scheduling techniques for a tight calculation of worst-case delays of memory accesses. The provided information can also ease the design of mechanisms that avoid and/or control interference between tasks at run-time. An important feature of StAMP compared to related work lies in its ability to link back time intervals to unique locations in the code of tasks, allowing easy implementation of elaborate run-time decisions related to interference management.

7.4 Security

Participants: Nicolas Bailluet, Nicolas Bellec, Antoine Gicquel, Damien Hardy, Camille Le Bon, Isabelle Puaut, Erven Rohou.

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

Participants: Nicolas Bellec, Isabelle Puaut
External collaborators: Guillaume Hiet, Frédéric Tronel, CIDRE team and Simon Rokicki, TARAN team

The emergence of Real-Time Systems with increased connections to their environment has led to a greater demand in security for these systems. Memory corruption attacks, which modify the memory to trigger unexpected executions, are a significant threat against applications written in low-level languages. Data-Flow Integrity (DFI) is a protection that verifies that only a trusted source has written any loaded data. The overhead of such a security mechanism remains a major issue that limits its adoption. This work [18] presents RT-DFI, a new approach that optimizes Data-Flow Integrity to reduce its overhead on the Worst-Case Execution Time. We model the number and order of the checks and use an Integer Linear Programming solver to optimize the protection on the Worst-Case Execution Path. Our approach protects the program against many memory-corruption attacks, including Return-Oriented Programming and Data-Only attacks. Moreover, our experimental results show that our optimization reduces the overhead by 7 % on average compared to a state-of-the-art implementation.

This work is done in collaboration with the CIDRE and TARAN teams, and received an outstanding paper award at ECRTS 2022 [18].

7.4.2 Multi-nop fault injection attack

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

Fault injection is a well-known method to physically attack embedded systems, microcontrollers in particular. It aims to find and exploit vulnerabilities in the hardware to induce malfunction in the software and eventually bypass software security or retrieve sensitive information. We propose [22] a cheap (in the order of $100) platform called TRAITOR inducing faults with clock glitches with the capacity to inject numerous and precise bursts of faults. From an evaluation point of view, this platform allows easier and cheaper investigations over complex attacks than costly EMI benches or laser probes.

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 propose SAMVA, a framework for efficiently assessing the vulnerability 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. We evaluate the proposed approach on eight PIN verification programs containing various software countermeasures. We show that our framework is able to find numerous attack paths, even for the most hardened version, in a very limited amount of time with an average of less than half of a second per fault parameter.

7.4.3 Platform for adaptive dynamic protection of programs

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

Memory corruption attacks have been a major issue in software security for over two decades and are still one of the most dangerous and widespread types of attacks nowadays. Among these attacks, control-flow hijack attacks are the most popular and powerful, enabling the attacker to execute arbitrary code inside the target process. Many approaches have been developed to mitigate such attacks and to prevent them from happening. One of these approaches is the Control-Data Isolation (CDI) that tries to prevent such attacks by removing their trigger from the code, namely indirect branches. This approach has been previously implemented as a compiler pass that replaces every indirect branches in the program with a table that leads the control-flow to direct hard-written branches. The drawback of this approach is that it needs the recompilation of the program. We present an approach and its implementation, DAMAS (see Sections 6.1.7 for Damas and 6.1.8 for the underlying library Sorry), a framework capable of deploying protections on a running software and use runtime information to optimize them during the process execution. We implemented a coarse-grain CDI protection using our framework and evaluated its impact on performance.

This concludes the PhD of Camille Le Bon [21].

7.4.4 Compiler approaches to protect against ROP and JOP attacks

Participants: Nicolas Bailluet, Isabelle Puaut, Erven Rohou
External collaborators: Emmanuel Fleury from the EPICURE team.

The 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) of Jump-Oriented Programming (JOP).*

Code-reuse attacks such as ROP consist in reusing the code from the program under attack to manipulate registers and memory. ROP relies on chaining small pieces of code that ends with a return instruction – known as gadgets. In order to get the desired effects on memory and registers, one has to find the appropriate gadgets and properly chain them. Chaining gadgets is a complicated task, they are bound to specific assembly instructions and registers, and often have undesired side-effects.

As a first step toward protecting against ROP and JOP attacks, we have investigated an approach for automatically generating chains of gadgets using program synthesis techniques. The proposed approach generates gadget chains based on program synthesis techniques and SMT solvers.

8 Bilateral contracts and grants with industry

Participants: Pierre Michaud.

8.1 Bilateral contracts with industry

Ampere Computing:

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

9 Partnerships and cooperations

Participants: Pierre Bedell, Caroline Collange, Audrey Fauveau, Antoine Gicquel, Damien Hardy, Sara Hoseininasab, Camille Le Bon, Pierre Michaud, Lucien Poirier, Isabelle Puaut, Hugo Reymond, Erven Rohou.

9.1 European initiatives

9.1.1 H2020 projects

HPCQS

Participants: Caroline Collange.

HPCQS project on cordis.europa.eu

Title: High Performance Computer and Quantum Simulator hybrid

Duration: From December 1, 2021 to November 30, 2025

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (GENCI), France
- NATIONAL UNIVERSITY OF IRELAND GALWAY (NUI GALWAY), Ireland
- FORSCHUNGSZENTRUM JULICH GMBH (FZJ), Germany
- PARITY QUANTUM COMPUTING GMBH (ParityQC), Austria
- FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV (FHG), Germany

- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), France
- EURICE EUROPEAN RESEARCH AND PROJECT OFFICE GMBH, Germany
- CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), Italy
- BULL SAS (BULL), France
- FLYSIGHT SRL, Italy
- PARTEC AG (PARTEC), Germany
- UNIVERSITAET INNSBRUCK (UIBK), Austria
- CINECA CONSORZIO INTERUNIVERSITARIO (CINECA), Italy
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (CNRS), France
- CENTRALESUPELEC (CentraleSupélec), France
- BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION (BSC CNS), Spain
- SORBONNE UNIVERSITE, France

Inria contact: Luc Giraud

Coordinator: Forschungszentrum Jülich GmbH

Summary: The aim of HPCQS is to prepare European research, industry and society for the use and federal operation of quantum computers and simulators. These are future computing technologies that are promising to overcome the most difficult computational challenges. HPCQS is developing the programming platform for the quantum simulator, which is based on the European ATOS Quantum Learning Machine (QLM), and the deep, low-latency integration into modular HPC systems based on ParTec's European modular supercomputing concept. A twin pilot system, developed as a prototype by the European company Pasqal, will be implemented and integrated at CEA/TGCC (France) and FZJ/JSC (Germany), both hosts of European Tier-0 HPC systems. The pre-exascale sites BSC (Spain) and CINECA (Italy) as well as ICECH (Ireland) will be connected to the TGCC and JSC via the European data infrastructure FENIX. It is planned to offer quantum HPC hybrid resources to the public via the access channels of PRACE. To achieve these goals, HPCQS brings together leading quantum and supercomputer experts from science and industry, thus creating an incubator for practical quantum HPC hybrid computing that is unique in the world. The HPC-QS technology will be developed in a co-design process together with selected exemplary use cases from chemistry, physics, optimization and machine learning suitable for quantum HPC hybrid calculations. HPCQS fits squarely to the challenges and scope of the call by acquiring a quantum device with two times 100+ neutral atoms. HPCQS develops the connection between the classical supercomputer and the quantum simulator by deep integration in the modular supercomputing architecture and will provide cloud access and middleware for programming and execution of applications on the quantum simulator through the QLM, as well as a Jupyter-Hub platform with safe access guarantee through the European UNICORE system to its ecosystem of quantum programming facilities and application libraries.

9.2 National initiatives

EPIQ: Study of the quantum stack: Algorithm, models, and simulation for quantum computing

Participants: Caroline Collange, Audrey Fauveau.

- Funding: PEPR
- Duration: 2022-2027
- Local coordinator: Caroline Collange

- Partners: CNRS, Inria, CEA
- The EPIQ project aims at developing algorithmic techniques for both noisy quantum machines (NISQ) and fault-tolerant ones so as to facilate their practical implementation. To this end, a first Work Package (WP) is dedicated to algorithmic techniques, a second one focuses on computational models and languages so as to facilitate the programming of quantum machines and to optimize the code execution steps. Lastly, the third WP aims at developing the simulation techniques of quantum computers.

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

Participants: Damien Hardy, Erven Rohou.

- 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.

EQIP: Engineering for Quantum Information Processors

Participants: Caroline Collange.

- Funding: Inria Challenge project
- Duration: 2021-2024

- Local coordinator: Caroline Collange
- Partners: COSMIQ, CAGE, CASCADE, DEDUCTEAM, GRACE, HIEPACS, MATHERIALS, MOCQUA, PACAP, PARSYS, QUANTIC, STORM, and ATOS Quantum
- Building a functional quantum computer is one of the grand scientific challenges of the 21st century. This formidable task is the object of Quantum Engineering, a new and very active field of research at the interface between physics, computer science and mathematics. EQIP brings together all the competences already present in the institute, to turn Inria into a major international actor in quantum engineering, including both software and hardware aspects of quantum computing.
- website: project.inria.fr/eqip

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

Participants: Caroline Collange, Sara Sadat Hoseininasab.

- Funding: ANR, JCJC
- Duration: 2020-2023
- 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 (50 %)
- 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/

CAOTIC: Collaborative Action on Timing Interference

Participants: 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/

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

Participants: Pierre Michaud, Erven Rohou.

- 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, Camille Le Bon, Erven Rohou.

- Funding: Inria Exploratory Action
- Duration: 2021-2024
- 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

AoT.js

Participants: Lucien Poirier, Erven Rohou.

- Funding: Inria Exploratory Action
- Duration: 2022-2025
- Local coordinator: Erven Rohou
- Partners: INDES (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.

9.3 Regional initiatives

PluriNOP

Participants: Antoine Gicquel, Damien Hardy, Erven Rohou.

- Funding: Région Bretagne (43 %), EUR CyberSchool (50 %)
- Duration: 2021-2024
- Local coordinator: Erven Rohou
- Partners: Sorbonne Université
- In a world where computer systems control large parts of our societies and lives on a daily basis, the stakes of computer security are high. Many types of attacks exist, we focus here in fault-based attacks on embedded systems. These attacks are becoming a threat to systems that were previously spared: certain injection methods are now available to a large community and software means of fault injection are being developed. The literature mainly deals with a single fault, but more and more works refer to the possibility of injecting multiple faults. The objectives of PluriNOP are:
    1. to propose an automatic approach, based on static analysis, to determine possible exploits for an attacker model and a target, and then perform them (in simulation or experimentation) on a binary code, in order to reach a given objective (write a data to a memory location, call a function with given function with given parameters, extract a secret...);
    2. propose a method for quantifying the level of vulnerability of a binary code, for example on the basis of the minimum number of faults necessary for an exploit to be performed, the nature of the faults, etc.;
    3. propose countermeasures to these attacks, software or hardware, and an automation of their deployment.

10 Dissemination

Participants: Abderaouf Nassim Amalou, Nicolas Bailluet, Nicolas Bellec, Caroline Collange, Antoine Gicquel, Damien Hardy, Sara Sadat Hoseininasab, Camille Le Bon, Pierre Michaud, Anis Peysieux, Lucien Poirier, Isabelle Puaut, Hugo Reymond, Erven Rohou.

10.1 Promoting scientific activities

10.1.1 Scientific events: selection

Member of the conference program committees

- E. Rohou was a member of the program committees of the CGO 2023 conference.
- I. Puaut was member of the program committee of the following conferences: Euromicro Conference on Real Time Systems (ECRTS) 2022, IEEE Real-Time Systems Symposium (RTSS) 2022, IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2022, Conference on Real-Time Networks and Systems (RTNS) 2023.
- P. Michaud was a member of the program committees of the HPCA 2023 conference.
- D. Hardy was a member of the program committees of the WCET 2022 workshop.
- C. Collange was an external review committee member of the ISCA 2022 and ASPLOS 2023 conferences.

Reviewer

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

10.1.2 Journal

Member of the editorial boards

- I. Puaut is associate editor of the Springer International Journal of Time-Critical Computing Systems.

Reviewer - reviewing activities

Members of PACAP routinely review submissions to international journals.

10.1.3 Leadership within the scientific community

E. Rohou is a member of the organization committee of focused days within the GdR SoC2.

10.1.4 Research administration

- E. Rohou is the contact for international relations for Inria Rennes Bretagne Atlantique (for scientific matters).
- E. Rohou is a member of the steering committee of the high security research laboratory (LHS).
- E. Rohou is a member of the steering committee of the Inria Rennes computer grid “igrida”.
- C. Collange is a member of the Inria Rennes information technology users committee (CUMIR).
- 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

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Master: A. Amalou, ITR (Informatique Temps Réel), 16h, M1, Université Rennes 1, France
- Master: C. Collange, GPU programming, 20 hours, M1, Université de Rennes 1, France
- Master: C. Collange, Advanced Design and Architectures, 16 hours, M2 SIF, Université de Rennes 1, France
- Licence: D. Hardy, Real-time systems, 95 hours, L3, Université de Rennes 1, France
- Master: D. Hardy, Operating systems, 59 hours, M1, Université de Rennes 1, France
- Master: D. Hardy, Students project, 30 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Operating systems: concepts and system programming under Linux (SEL), 77 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Operating systems kernels (NOY), 54 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Real-time systems, 48 hours, M1, Université de Rennes 1, France
- Master: I. Puaut, Optimizing and Parallelizing Compilers (OPC), 9 hours, M2, Université de Rennes 1, France
- Master: I. Puaut, Writing of scientific publications, 9 hours, M2 and PhD students, Université de Rennes 1, France

- Master: H. Reymond, Operating systems, 32 hours, M1, Université de Rennes 1, France
- Licence: H. Reymond, Architecture & OS, 30 hours, L3, Université de Rennes 1, France
- Licence: L. Poirier, Object Programming, 40 hours, L2, Université de Rennes 1, France
- Licence: N. Bailluet, C Programming and Network, 20 hours, L3, ENS Rennes, France
- Licence: A. Gicquel, Unix and C programming, 40 hours, L3, Université de Rennes 1, France

10.2.2 Supervision

- PhD: Camille Le Bon, *Dynamic Binary Analysis and Optimization for Cyber-Security*, Université de Rennes 1, Jul 2022, advisors E. Rohou (30 %), G. Hiet from CIDRE (35 %), F. Tronel from CIDRE (35 %)
- PhD in progress : Nicolas Bellec, *Security in real-time embedded systems*, started Dec 2019, advisors I. Puaut (50 %), G. Hiet from CIDRE (25 %), F. Tronel from CIDRE (25 %)
- PhD in progress: Anis Peysieux, *Towards simple and highly efficient execution cores*, started Jan 2020, advisor since Jan 2021: P. Michaud (A. Seznec was advisor from Jan 2020 until Dec 2020)
- PhD in progress: Abderaouf Nassim Amalou, *Machine learning for performance prediction*, started Oct 2020, advisors I. Puaut (75 %), E. Fromont (25 %, LACODAM)
- PhD in progress: Hugo Reymond, *Energy-aware execution model in intermittent systems*, started Oct 2021, advisors I. Puaut, E. Rohou, S. Faucou (LS2N Nantes), J.-L. Béchennec (LS2N Nantes)
- PhD in progress: Antoine Gicquel, *Étude de vulnérabilité d'un programme au format binaire en présence de fautes précises et nombreuses : métriques et contremesures*, started Sep 2021, advisors D. Hardy, E. Rohou, K. Heydemann (Sorbonne Université)
- PhD in progress: Sara Hoseininasab, *Automatic synthesis of multi-thread pipelines*, started Nov 2021, advisors C. Collange (70 %) and S. Derrien (30 %, TARAN)
- PhD in progress: Lucien Poirier, *Profile-Guided optimization for Dynamic Languages*, started Oct 2022, advisors E. Rohou (50 %) and M. Serrano (50 %, Inria Sophia)
- 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 %)

10.2.3 Juries

C. Collange was a member of the *Jury d'Agrégation* in computer science.
I. Puaut was member of the following hiring committees of assistant professors/professors:

- Associate Professor, Scuola Superiore Sant'Anna (SSSA), Pisa, Italy
- Associate Professor, Université de Rennes 1/ISTIC, topic "Software security"
- Professor, ENSIMAG, Grenoble, topic "Infrastructures, systems and large-scale networks"

C. Collange was member of the Associate Professor hiring committee of ENS Lyon, topic "Networks, compilers, computer architecture, systems".
E. Rohou was a member of the following PhD thesis committees:

- Davide Pala, *Microarchitectures for Robust and Efficient Incremental Backup in Intermittently Powered Systems*, Nov 2022 (invited)
- Vincent Werner, *Optimiser l'identification et l'exploitation de vulnérabilités à l'injection de faute sur microcontrôleurs*, Jan 2022 (examiner)

I. Puaut was member of the following thesis committes:

- Imane Haur, *Autosar compliant multi-core RTOS formal modeling and verification*, École centrale de Nantes, Nov 2022 (examiner)
- Michael Platzer, *Predictable and performant computer architectures for time-critical systems*, Technische Universität Wien, Vienna, Austria, Dec 2022 (reviewer)
- Matheus Schuh, *Safe implementation of hard real applications on many-core platforms*, Université Grenoble Alpes, May 2022 (emaminer)

C. Collange was a member of the following PhD thesis committees:

- Arthur Hennequin, *Performance optimization for the LHCb experiment*, CERN / Sorbonne Université, Jan 2022
- Thibaut Marty, *Timing speculation for hardware accelerators*, Université Rennes 1, Mar 2022
- Paul Iannetta, *Compiling trees: combining data layouts and the polyhedral model*, ENS Lyon, May 2022

E. Rohou was a member of the CSI of Anis Peysieux, Bruno Mateu, Jean-Loup Hatchikian-Houdot, Antoine Bernabeu, Fatima-Zahra Bouchana, Aaron Randrianaina, Quentin Ducasse. I. Puaut is member of the CSI of Zineb Boukili and Jean-Michel Gorius. C. Collange was a member of the CSI of Thibaut Marty, Corentin Ferry, Louis Narmour.

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

E. Rohou is a member of the Inria Rennes working group on sustainable development.

10.3.2 Education

E. Rohou was invited to present to job of researcher to secondary-school students (classe de 4e) at Collège de Bourgchevreuil, Cesson-Sévigné.
C. Collange advised a TIPE student research project at Lycée Chateaubriand, Rennes.

10.3.3 Interventions

E. Rohou was a “witness” during the event dedicated to PhD students organized at Inria Rennes, focusing on postdocs and industrial experience.
C. Collange was invited to present her research at the ENS Rennes student seminar.

11 Scientific production

11.1 Major publications

[1] F. Bodin, T. Kisuki, P. M. W. Knijnenburg, M. F. P. O’Boyle and E. Rohou. ‘Iterative Compilation in a Non-Linear Optimisation Space’. In: *Workshop on Profile and Feedback-Directed Compilation (FDO-1), in conjunction with PACT ’98*. Paris, France, Oct. 1998.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.2 Publications of the year

International journals

[16] P. Michaud and A. Peysieux. 'HAIR: Halving the Area of the Integer Register File with Odd/Even Banking'. In: *ACM Transactions on Architecture and Code Optimization* 19.4 (Dec. 2022), pp. 1–26. DOI: 10.1145/3544838. URL: https://hal.inria.fr/hal-03740496.

International peer-reviewed conferences

[17] A. N. Amalou, E. Fromont and I. Puaut. 'CATREEN : Context-Aware Code Timing Estimation with Stacked Recurrent Networks'. In: ICTAI 2022 - 34th IEEE International Conference on Tools with Artificial Intelligence. Virtually, China: IEEE, 31st Oct. 2022, pp. 1–6. URL: https://hal.archives-ouvertes.fr/hal-03890057.

[18] N. Bellec, G. Hiet, S. Rokicki, F. Tronel and I. Puaut. 'RT-DFI: Optimizing Data-Flow Integrity for Real-Time Systems'. In: ECRTS 2022 - 34th Euromicro Conference on Real-Time Systems. 34. Modène, Italy, 28th June 2022, pp. 1–24. DOI: 10.4230/LIPIcs.ECRTS.2022.18. URL: https://hal.inria.fr/hal-03641576.

[19] T. Degioanni and I. Puaut. 'StAMP: Static Analysis of Memory access Profiles for real-time tasks'. In: WCET 2022 - 20th International Workshop on Worst-Case Execution Time Analysis. Modena, Italy, 5th July 2022. DOI: 10.4230/OASIcs.WCET.2022.1. URL: https://hal.inria.fr/hal-03723457.

[20] V. Pasquale and I. Puaut. 'Winston: Revisiting iterative compilation for WCET minimization'. In: RTNS 2022 - 30th International Conference on Real-Time Networks and Systems. Paris, France, 7th June 2022, pp. 1–11. DOI: 10.1145/3534879.3534899. URL: https://hal.inria.fr/hal-03673668.

Doctoral dissertations and habilitation theses

[21] C. Le Bon. 'Dynamic binary analysis and optimization for cybersecurity'. Université Rennes 1, 5th July 2022. URL: https://theses.hal.science/tel-03906421.

11.3 Cited publications

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

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.