RESEARCH CENTRE

Grenoble - Rhône-Alpes

IN PARTNERSHIP WITH:

Institut polytechnique de Grenoble, Université de Grenoble Alpes

2020 ACTIVITY REPORT

Project-Team CORSE

compiler optimization and run-time systems

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Architecture, Languages and Compilation

Contents

Project-Team CORSE

Creation of the Team: 2014 November 01, updated into 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

A1.1.4. – High performance computing

A1.1.12. – Non-conventional architectures

A1.6. – Green Computing

A2.1.6. – Concurrent programming

A2.1.7. – Distributed programming

A2.1.8. – Aspect-oriented programming

A2.1.10. – Domain-specific languages

A2.2. – Compilation

A2.2.1. – Static analysis

A2.2.2. – Memory models

A2.2.3. – Memory management

A2.2.4. – Parallel architectures

A2.2.5. – Run-time systems

A2.2.6. – GPGPU, FPGA...

A2.2.8. – Code generation

A2.2.9. – Security by compilation

A2.3.2. – Cyber-physical systems

A4.4. – Security of equipment and software

A7.1. – Algorithms

Other research topics and application domains

B2.3. – Epidemiology

B2.7. – Medical devices

B4.5. – Energy consumption

B5.3. – Nanotechnology

B6.1.2. – Software evolution, maintenance

B6.6. – Embedded systems

B6.7. – Computer Industry (harware, equipments...)

B9.1. – Education

1 Team members, visitors, external collaborators

Research Scientists

- Fabrice Rastello [Team leader, Inria, Senior Researcher, HDR]
- Frederic Desprez [Inria, Senior Researcher, until Sep 2020, HDR]

Faculty Members

- Florent Bouchez - Tichadou [Univ Grenoble Alpes, Associate Professor]
- Francois Broquedis [Institut polytechnique de Grenoble, Associate Professor]
- Ylies Falcone [Univ Grenoble Alpes, Associate Professor]
- Manuel Selva [Institut polytechnique de Grenoble, Associate Professor]

Post-Doctoral Fellow

- Guillaume Iooss [Inria, from Feb 2020]

PhD Students

- Theo Barollet [Inria]
- Nicolas Derumigny [Univ Grenoble Alpes]
- Nihel Kaboubi [Orange Labs, CIFRE]
- Auguste Olivry [Univ Grenoble Alpes]
- Chukri Soueidi [Inria, from Oct 2020]
- Mathieu Stoffel [Bull]
- Nicolas Tollenaere [Inria]

Technical Staff

- Theophile Bastian [Inria, Engineer, from Oct 2020]
- Ali Kassem [Inria, Engineer]
- Chukri Soueidi [Inria, Engineer, until Sep 2020]

Interns and Apprentices

- Muhammad Ikram [Inria, from Mar 2020 until Jul 2020]

Administrative Assistant

- Imma Presseguer [Inria]

2 Overall objectives

Languages, compilers, and run-time systems are some of the most important components to bridge the gap between applications and hardware. With the continuous increasing power of computers, expectations are evolving, with more and more ambitious, *computational intensive and complex applications*. As desktop PCs are becoming a niche and servers mainstream, three categories of computing impose themselves for the next decade: mobile, cloud, and super-computing. Thus *diversity, heterogeneity* (even on a single chip) and thus also *hardware virtualization* is putting more and more pressure both on compilers and run-time systems. However, because of the energy wall, *architectures* are becoming more and more *complex* and *parallelism ubiquitous* at every level. Unfortunately, the memory-CPU gap continues to increase and energy consumption remains an important issue for future platforms. To address the challenge of *performance and energy consumption* raised by silicon companies, compilers and run-time systems must *evolve* and, in particular, interact, *taking into account the complexity of the target architecture.*

The overall objective of CORSE is to address this challenge by *combining static and dynamic compilation* techniques, with more interactive *embedding of programs and compiler environment in the run-time system.*

3 Research program

3.1 Scientific Foundations

One of the characteristics of CORSE is to base our researches on diverse advanced mathematical tools. Compiler optimization requires the usage of the several tools around discrete mathematics: combinatorial optimization, algorithmic, and graph theory. The aim of CORSE is to tackle optimization not only for general purpose but also for domain specific applications. We believe that new challenges in compiler technology design and in particular for split compilation should also take advantage of graph labeling techniques. In addition to run-time and compiler techniques for program instrumentation, hybrid analysis and compilation advances will be mainly based on polynomial and linear algebra.

The other specificity of CORSE is to address technical challenges related to compiler technology, run-time systems, and hardware characteristics. This implies mastering the details of each. This is especially important as any optimization is based on a reasonably accurate model. Compiler expertise will be used in modeling applications (e.g. through automatic analysis of memory and computational complexity); Run-time expertise will be used in modeling the concurrent activities and overhead due to contention (including memory management); Hardware expertise will be extensively used in modeling physical resources and hardware mechanisms (including synchronization, pipelines, etc.).

The core foundation of the team is related to the combination of static and dynamic techniques, of compilation, and run-time systems. We believe this to be essential in addressing high-performance and low energy challenges in the context of new important changes shown by current application, software, and architecture trends.

Our project is structured along two main directions. The first direction belongs to the area of run-time systems with the objective of developing strong relations with compilers. The second direction belongs to the area of compiler analysis and optimization with the objective of combining dynamic analysis and optimization with static techniques. The aim of CORSE is to ground those two research activities on the development of the end-to-end optimization of some specific domain applications.

4 Application domains

4.1 Transfer

The main industrial sector related to the research activities of CORSE is the one of semi-conductor (programmable architectures spanning from embedded systems to servers). Obviously any computing application which has the objective of exploiting as much as possible the resources (in terms of high-performance but also low energy consumption) of the host architecture is intended to take advantage of advances in compiler and run-time technology. These applications are based over numerical kernels (linear algebra, FFT, convolution...) that can be adapted on a large spectrum of architectures. More specifically, an important activity concerns the optimization of machine learning applications for some high-performance accelerators. Members of CORSE already maintain fruitful and strong collaborations with several companies such as STMICROELECTRONICS, Atos/Bull, Kalray.

5 Highlights of the year

5.1 Awards

- HiPEAC Paper award for paper *Automated derivation of parametric data movement lower bounds for affine programs* [8], presented at PLDI 2020.

6 New software and platforms

6.1 New software

6.1.1 Verde

Keywords: Debug, Verification

Functional Description: Interactive Debugging with a traditional debugger can be tedious. One has to manually run a program step by step and set breakpoints to track a bug.

i-RV is an approach to bug fixing that aims to help developpers during their Interactive Debbugging sessions using Runtime Verification.

Verde is the reference implementation of i-RV.

URL: `https://gitlab.inria.fr/monitoring/verde`

Publication: hal-01592671

Contacts: Raphael Jakse, Ylies Falcone

Participants: Kevin Pouget, Ylies Falcone, Raphael Jakse, Jean-François Méhaut

6.1.2 GUS

Keywords: CPU, Microarchitecture simulation, Performance analysis, Dynamic Analysis

Functional Description: GUS' goal is to detect performance bottlenecks at the very low level on monothread applications by the use of sensitivity analysis. It is coded as a QEMU plug-in in order to collect runtime information that are later treated by the generic CPU model.

URL: `https://gitlab.inria.fr/nderumig/gus`

Contacts: Fabrice Rastello, Nicolas Derumigny, Fabian Gruber

6.1.3 Pipedream

Name: Pipedream

Keywords: Performance analysis, CPU, Reverse engineering

Scientific Description: Pipedream reverse engineers the following performance characteristics: - Instruction latency – The number of cycles an instruction requires to execute. - Peak micro-op retirement rate – How many fused micro-ops the CPU can retire per cycle. - Micro-fusion – The number of fused micro-ops an instruction decomposes into. - Micro-op decomposition and micro-op port usage – The list of unfused micro-ops every instruction decomposes into and the list of execution ports every one of these micro-ops can execute on.

The first step of the reverse engineering process consists of generating a number of microbenchmarks. Pipedream then runs these benchmark, measuring their performance using hardware counters. The latency, throughput, and micro-fusion of different instructions can then be read directly from these measurements.

The process of finding port mappings, i.e. micro-op decompositions and micro-op port usage, however, is more involved. For this purpose, we have defined a variation of the maximum flow problem which we call the "instruction flow problem". We have developed a linear program (LP) formulation of the instruction flow problem which can be used to calculate the peak IPC and micro-operations per cycle (MPC) a benchmark kernel can theoretically achieve with a given port mapping. The actual port mapping of the underlying hardware is then determined by finding the mapping for which the throughput predicted by instruction flow best matches the actual measured IPC and MPC.

Functional Description: Pipedream is a tool for measuring specific performance characteristics of CPUs It is used to build the performance model of another tool called Gus (https://gitlab.inria.fr/nderumig/gus). Pipedream finds measured performance characteristics such as the throughput and latency of instructions by running a large set of automatically generated microbenchmarks. The tool can also find port mappings, a model of part of the CPU instruction scheduler, by analysing performance measurements of specially crafted microkernels using a LP solver. We have used it to produce a port mapping for the Intel Skylake CPU architecture. Pipedream is able to find the port mappings for some instructions for which existing approaches fall back to manual analysis.

URL: https://gitlab.inria.fr/fgruber/pipedream

Contacts: Fabian Gruber, Fabrice Rastello, Nicolas Derumigny

6.1.4 IOLB

Keywords: Complexity, Polyhedral compilation, Performance analysis

Functional Description: IOLB computes a symbolic lower bound on the I/O, or data movement, complexity of a computer program, that is the amount of data that needs to be moved between cache and main memory to perform its computation. The input is a C program, and the output is a mathematical formula that depends on program parameters (array sizes...) and cache size.

URL: https://gitlab.inria.fr/CORSE/iolb

Contacts: Auguste Olivry, Guillaume Iooss, Fabrice Rastello

6.1.5 IOOpt

Keywords: I/O, Polyhedral compilation

Functional Description: IOOpt takes as input an abstract representation of a tileable program. The tool generates a tractable set of relevant permutations of the tiling loops, and a symbolic I/O cost expression for each of them. It then uses a non linear problem optimizer to find the best permutations and corresponding tile sizes for a given value of machine parameters (cache sizes and bandwidths at each level). IOOop can also be used to find an upper bound on the I/O cost of a program, for a given tiling scheme.

Contacts: Auguste Olivry, Fabrice Rastello

6.1.6 PALMED

Keywords: CPU, Performance measure, Performance analysis, Reverse engineering

Functional Description: PALMED computes a bipartite graph assembly instructions <-> abstract resources that may be use for performance prediction, targeting static analysis tools and compilers. Internally, PALMED uses PIPEDREAM as a framework for microbenchmarking code generation, and use gurobi to find a first small graph. Then, PALMED deduces from the found resources and the microbenchmarks that saturates them a complete mapping of every supported instruction.

URL: `https://gitlab.inria.fr/nderumig/gus`

Contacts: Fabrice Rastello, Nicolas Derumigny, Theophile Bastian

6.1.7 BISM

Name: BISM: Bytecode-level Instrumentation for Software Monitoring

Keywords: Java, Bytecode, Instrumentation, Control Flow

Functional Description: BISM (Bytecode-level Instrumentation for Software Monitoring) is a lightweight Java bytecode instrumentation tool which features an expressive high-level control-flow-aware instrumentation language. The language follows the aspect-oriented programming paradigm by adopting the joinpoint model, advice inlining, and separate instrumentation mechanisms. BISM provides joinpoints ranging from byte- code instruction to method execution, access to comprehensive context information, and instrumentation methods. BISM runs in two modes: build-time and load-time.

URL: `https://gitlab.inria.fr/monitoring/bism`

Publication: hal-03081265

Contacts: Chukri Soueidi, Ylies Falcone, Ali Kassem

Participants: Chukri Soueidi, Ylies Falcone, Ali Kassem

6.1.8 TTiLE

Keywords: Deep learning, Optimization, HPC

Functional Description: TTiLE is a code generation tool for tensor operators present in CNNs such as tensor contraction and convolution. It takes as input: 1. a kernel specification, that is, a fonctionnal description of the operator (iteration space, tensors, single assignment description of the computation), 2. an optimization scheme that describes the layered structure of the generated code. The scheme "language" allows to express loop permutation, loop tiling, vectorization, unrolling, packing (which consists in using temporary buffers making cache access better behaved) and branching. Indeed, as opposed to existing schemes that rely on partial tiles, TTile can create non-perfectly nested loops to combine several "micro-kernels" (micro-kernel stands for the innermost part of the loop nest that is fully unrolled, register promoted and vectorized here). Using a specialized search strategy that combines operational research on analytical performance model and native execution time measurement, TTile outperforms current highly optimized libraries such as Intel oneDNN and Intel MKL.

URL: `https://gitlab.inria.fr/CORSE/ttile`

Contact: Nicolas Tollenaere

6.2 New platforms

6.2.1 Grid'5000

Grid'5000 [1] is a large-scale and versatile testbed for experiment-driven research in all areas of computer science, with a focus on parallel and distributed computing including Cloud, HPC and Big Data. It provides access to a large amount of resources: 14828 cores, 829 compute-nodes grouped in homogeneous clusters located in 8 sites in France connected through a dedicated network (Renater), and featuring various technologies (GPU, SSD, NVMe, 10G and 25G Ethernet, Infiniband, Omni-Path) and advanced monitoring and measurement features for traces collection of networking and power consumption, providing a deep understanding of experiments. It is highly reconfigurable and controllable. Researchers can experiment with a fully customized software stack thanks to bare-metal deployment features, and can isolate their experiment at the networking layer advanced monitoring and measurement features for traces collection of networking and power consumption, providing a deep understanding of experiments designed to support Open Science and reproducible research, with full traceability of infrastructure and software changes on the testbed. Frédéric Desprez is director of the GRID5000 GIS.

6.2.2 SILECS/SLICES

Frédéric Desprez is co-PI with Serge Fdida (Université Sorbonne) of the SILECS [2] infrastructure ("*IR ministère*") which goal is to provide an experimental platform for experimental computer Science (Internet of things, clouds, HPC, big data, IA, wireless technologies, ...). This new infrastructure is based on two existing infrastructures, Grid'5000 and FIT. A European infrastructure (SLICES, ESFRI proposal) is also currently designed with 15 european partners (Spain, Cyprus, Greece, Netherland, Norway, Poland, Switzerland, ...).

7 New results

7.1 Compiler Optimizations and Analysis

Participants Fabrice Rastello, Manuel Selva, Fabian Grüber, P. Sadayappan *(OSU, USA)*, , Louis-Noël Pouchet *(CSU, USA)*, , Atanas Rountev *(OSU, USA)*, , Richard Veras *(LSU, USA)*, , Rui Li *(UoU, USA)*, , Aravind Sukumaran-Rajam *(OSU, USA)*, , Tse Meng Low *(CMU, USA)*.

Our current efforts with regard to code optimization follows two directions.

1. The first consists in improving compiler optimization techniques by considering pattern specific applications such as those related to machine learning. In this context we developed: 1. a tool for computing data movement complexity (both lower and upper bounds [8]) for affine programs (Section 9.1.1); 2. a new cache model for sparse matrix-matrix multiplication [7] and along with a code generator, a tool for computing caches misses for DNNs (Section 7.1.2).
2. The second consists in generating dynamic analysis based performance debugging tools. For that purpose we: 1. developed a tool for automatically characterize CPU resources [11] and extended a binary translator (QEMU) to perform an abstract simulation based sensitivity analysis (Section 7.1.3).

7.1.1 Automated derivation of parametric data movement complexity for affine programs

Participants Guillaume Iooss, Auguste Olivry, Fabrice Rastello, Louis-Noël Pouchet *(CSU, USA)*, P. Sadayappan *(OSU, USA)*.

[1]https://www.grid5000.fr/
[2]https://www.silecs.net/

Researchers and practitioners have for long worked on improving the computational complexity of algorithms, focusing on reducing the number of operations needed to perform a computation. However the hardware trend nowadays clearly shows a higher performance and energy cost for data movements than computations: quality algorithms have to minimize data movements as much as possible.

The theoretical operational complexity of an algorithm is a function of the total number of operations that must be executed, regardless of the order in which they will actually be executed. But theoretical data movement (or, I/O) complexity is fundamentally different: one must consider all possible legal schedules of the operations to determine the minimal number of data movements achievable, a major theoretical challenge. I/O complexity has been studied via complex manual proofs, e.g., refined from $\Omega(n^3/\sqrt{S})$ for matrix-multiply on a cache size S by Hong & Kung to $2n^3/\sqrt{S}$ by Smith et al. While asymptotic complexity may be sufficient to compare I/O potential between broadly different algorithms, the accuracy of the reasoning depends on the tightness of these I/O lower bounds. Precisely, exposing constants is essential to enable precise comparison between different algorithms: for example the $2n^3/\sqrt{S}$ lower bound allows to demonstrate the optimality of panel-panel tiling for matrix-multiplication.

We developed the first static analysis to automatically derive non-asymptotic parametric expressions of data movement lower bounds with scaling constants, for arbitrary affine computations. Our approach is fully automatic, assisting algorithm designers to reason about I/O complexity and make educated decisions about algorithmic alternatives. The tool allowed us to compute the lower bound for all the kernels of the Polybench suite. We extended this work by: 1. designing the first algorithm for computing a symbolic over-approximation of the data-movement for a parametric (multi-dimensional) tiled version of an affine code 3. designing the first fully automated scheme for expressing as an operational research problem the minimization of this data-movement expression; 3. integrating those techniques into a tool that computes for a class of affine computations a parametric expressions for I/O upper bounds.

This work has been done in the context of the IOcomplexity associate team (see Section 9.1.1). A paper that has been presented at PLDI 2020 [8] details the automated approach for the lower-bound complexity. The extensions to this approach have been conditionaly accepted to PLDI 2021.

7.1.2 Analytical Cache Modeling and Tilesize Optimization for Tensor Computations

Participants Fabrice Rastello, Nicolas Tollenaere, Manuel Selva, Guillaume Iooss, Auguste Olivry, Albert Cohen *(Google, France)*, Süreyya Emre Kurt *(OSU, USA)*, Aravind Sukumaran-Rajam *(WSU, USA)*, P. Sadayappan *(OSU, USA)*.

Tensor computation such as Sparse Matrix Multi-vector multiplication, Sampled Dense Dense Matrix Multiplication, Dense Matrix Multiplication, Tensor Contraction, Convolution are important kernels used in many domains like Fluid Dynamics, Data Analytics, Economic Modelling, and Machine Learning. Developing highly optimized code for such kernels requires the combination of highly tuned register/instruction level micro-kernels and appropriate multi-level tiling. In this context we developed: 1. an analytical cache model for sparse matrices; 2. an analytical cache model-based tiles size optimization along with a code generator for DNNs.

Efficient Tiled Sparse Matrix Multiplication through Matrix Signatures Tiling is a key technique to reduce data movement in matrix computations. While tiling is well understood and widely used for dense matrix/tensor computations, effective tiling of sparse matrix computations remains a challenging problem. This work proposes a novel method to efficiently summarize the impact of the sparsity structure of a matrix on achievable data reuse as a one-dimensional *signature*, which is then used to build an analytical cost model for tile size optimization for sparse matrix computations. The proposed model-driven approach to sparse tiling is evaluated on two key sparse matrix kernels: Sparse Matrix - Dense Matrix Multiplication (SpMM) and Sampled Dense-Dense Matrix Multiplication (SDDMM). Experimental results demonstrate that model-based tiled SpMM and SDDMM achieve high performance relative to the state-of-the-art.

Code Generation and Optimization for Tensors Addressing the problem of automatic generation of optimized operators raises two challenges: The first is associated to the design of a domain specific code generation framework able to output high-quality binary code. The second is to carefully bound the search space and choose an optimizing objective function that neither leads to yet another combinatorial optimizing problem, nor leads to a too approximate performance objective. This work tackles those two challenges by: 1. revisiting the usual belief that packing should enable stride-1 accesses at every level allowing to make packing optional; 2. highlighting the importance of considering the packing decision and shape as being part of the optimization problem; 3. revisiting the usual belief that register spilling should be avoided if possible allowing to consider other (more packing friendly) micro-kernels as good candidates; 4. revisiting the misleading intuition that convolution dimensions should be brought at the innermost level allowing more freedom for memory reuse at outer-dimensions; 5. showing that the optimization problem can be decoupled into: finding a small set of good micro-kernels candidates using an exhaustive search; finding a good schedule (loop tiling/permutation) and associated packing using operational research; finding the best tiles sizes using auto-tuning; 6. designing a single-pass micro-kernel generation algorithm, to emit code for any choice of register blocking dimensions, unrolling factor, and packing decisions; 7. designing a lowering scheme for abstract iterators, compatible with diverse packing and tiling strategies thrifty with integer arithmetic and loop control usage; 8. designing a packing algorithm compatible with various choices of transposition and subviews; 9. implementing a code generator based on these algorithms, driven by a simple and modular configuration language.

This work has been done in the context of the IOcomplexity associate team (see Section 9.1.1) and the PBI project ES3CAP (see Section 8.2.1). A paper that has been presented at SC 2020 that details the tiling for sparse matrix computation.

7.1.3 Automatic Resource Characterization and Performance Feedback

Participants Fabrice Rastello, Nicolas Derumigny, Théophile Bastian, Guillaume Iooss, Louis-Noël Pouchet.

Performance modeling is a critical component for program optimizations, assisting compilers as well as developers in predicting the performance of code variations ahead of time. Performance models can be obtained through different approaches that span from precise and complex simulation of a hardware description (Zesto, GEM5, PTLSim) to application level analytical formulations. An interesting approach for modeling the CPU of modern pipelined, super-scalar, out-of-order processors trades simulation time with accuracy by separately characterizing both latency and throughput of instructions. This approach is suitable both for optimizing compilers, but also for hand-tuning critical kernels written in assembler (see Section 7.1.2). It is used by performance-analysis tools such as CQA, Intel IACA, OSACA, MIAMI or llvm-mca. Cycle-approximate simulators such as ZSim or MCsimA can also take advantage of such an instruction characterization. In this context, we developed two tools: PALMED and GUS (see Section 6).

PALMED: Throughput Characterization for Any Architecture PALMED is to a tool that automatically builds a resource mapping, a performance model for pipelined, super-scalar, out-of-order CPU architectures. Resource mappings describe the execution of a program by assigning instructions in the program to abstract resources. They can be used to predict the throughput of basic blocks or as a machine model for the backend of an optimizing compiler.

PALMED does not require hardware performance counters, and relies solely on runtime measurements to construct resource mappings. This allows it to model not only execution port usage, but also other limiting resources, such as the frontend or the reorder buffer. Also, thanks to a dual representation of resource mappings, our algorithm for constructing mappings scales to large instruction sets, like that of x86.

GUS: Practical Performance Debugging For Out-of-Order Processors GUS is the first framework for performance debugging of complex, realistic pipelined out-of-order processors executing arbitrary programs, via abstract simulation for sensitivity analysis. Abstract simulation aims at providing significant speed improvement versus cycle-accurate simulation. This allows to perform quickly multiple runs with several value of latency, throughput and count of the modeled resources: this is sensitivity analysis. To determine if a resource is a bottleneck, its capacity is artificially increased to determine whether it affects the (simulated) program execution time accordingly. Such approach is not feasible with cycle-accurate simulator, yet to be realistic it requires an accurate modeling of the execution time.

This work has been done in the context of the associate team IOComplexity (see Section 9.1.1) and the European project CPS4EU (see Section 9.2.1).

7.1.4 Extraction of Periodic Patterns of Scientific Applications to Identify DVFS Opportunities

Participants Mathieu Stoffel, François Broquedis, Frederic Desprez, Abdelhafid Mazouz *(Atos/Bull)*, Philippe Rols *(Atos/Bull)*.

Mathieu Stoffel started his PhD in February 2018 on a CIFRE contract with Atos/Bull. The purpose of this work is to enhance the energy consumption of HPC applications on large-scale platforms by monitoring and predicting the behavior of executed applications to optimize resources utilisation, notably on power consumption. In this context, Mathieu developed Phase-TA, an offline tool which detects and characterizes the inherent periodicities of iterative HPC applications, with no prior knowledge of the latter. To do so, it analyses the evolution of several performance counters at the scale of the compute node, and infers patterns representing the identified periodicities. As a result, Phase-TA offers a non-intrusive mean to gain insights on the processor use associated with an application, and paves the way to predicting its behavior. Phase-TA was tested on a panel of 3 applications and benchmarks from the supercomputing field: HPCG, NEMO, and OpenFoam. For all of them, periodicities, accountable for on average 78% of their execution time, were detected and represented by accurate patterns. Furthermore, it was demonstrated that there is no need to analyse the whole profile of an application to precisely characterize its periodic behaviors. Indeed, an extract of the aforementioned profile is enough for Phase-TA to infer representative patterns on-the-fly, opening the way to energy-efficiency optimization through Dynamic Voltage-Frequency Scaling (DVFS). This work has been accepted at the HPCS2020 conference (postponed in 2021). The next step will be to use Phase-TA to identify DVFS opportunities and dynamically adapt the cores frequencies at runtime.

7.2 Runtime Monitoring, Verification, and Enforcement

Participants Mohamad Jaber *(Google Zurich)*, , Antoine El-Hokayem *(Univ. Grenoble Alpes, Verimag)*, Yliès Falcone, Thierry Jéron *(Inria Rennes)*, Ali Kassem, Matthieu Renard *(Foxi)*, Antoine Rollet *(Université de Bordeaux)*.

During the period, our new results and contributions can be categorized as follows. First, our main efforts were related to the verification and validation of applications in the domain of the Internet of Things in the context of the CLAPS project. We had two main contributions, namely the development of the BISM tool and the design and implementation of Runtime Enforcement Monitors. BISM is a tool for expressive and efficient instrumentation of Java applications at the Bytecode level. For our verification purposes, source-level instrumentation (which we previously used) was not sufficiently expressive nor sufficiently fine grain (e.g., changes in the control flow); Second, we consolidated work that was started a few years ago through the realisation and revision of journal submission, some of which got published this year.

Moreover, we coedited a special issue on the use of formal methods to help in the software development process [2].

Finally, we realised a teaching book [10] on the introduction of automata theory and regular languages. This is as part of our effort to contribute to the evolution of teaching practice so as to provide better quality material to students and make them more autonomous.

7.2.1 Runtime enforcement of timed properties using games

In this work [4] we deal with runtime enforcement of timed properties with uncontrollable events. Runtime enforcement consists in defining and using an enforcement mechanism that modifies the executions of a running system to ensure their correctness with respect to the desired property. Uncontrollable events cannot be modified by the enforcement mechanisms and thus have to be released immediately. We present a complete theoretical framework for synthesizing such mechanism, modeling the runtime enforcement problem as a Büchi game. It permits to pre-compute the decisions of the enforcement mechanism, thus avoiding to explore the whole execution tree at runtime. The obtained enforcement mechanism is sound, compliant and optimal, meaning that it should output as soon as possible correct executions that are as close as possible to the input execution. This framework takes as input any timed regular property modelled by a timed automaton. We present GREP, a tool implementing this approach. We provide algorithms and implementation details of the different modules of GREP, and evaluate its performance. The results are compared with another state of the art runtime enforcement tool.

7.2.2 From global choreographies to verifiable efficient distributed implementations

In this work [3], we define a method to automatically synthesize efficient distributed implementations from high-level global choreographies. A global choreography describes the execution and communication logic between a set of provided processes which are described by their interfaces. At the choreography level, the operations include multiparty communications, choice, loop, and branching. A choreography is master triggered: it has one master to trigger its execution. This allows us to automatically generate conflict-free distributed implementations without controllers. The behavior of the synthesized implementations follows the behavior of choreographies. In addition, the absence of controllers ensures the efficiency of the implementation and reduces the communication needed at runtime. Moreover, we define a translation of the distributed implementations to equivalent Promela versions. The translation allows verifying the distributed system against behavioral properties. We implemented a Java prototype to validate the approach and applied it to automatically synthesize micro-service architectures. We also illustrate our method on the automatic synthesis of a verified distributed buying system.

7.2.3 On the Monitoring of Decentralized Specifications: Semantics, Properties, Analysis, and Simulation

In this work, we introduce two complementary approaches to monitor decentralized systems. The first approach relies on systems with a centralized specification, i.e., when the specification is written for the behavior of the entire system. To do so, our approach introduces a data structure that (i) keeps track of the execution of an automaton (ii) has predictable parameters and size, and (iii) guarantees strong eventual consistency. The second approach defines decentralized specifications wherein multiple specifications are provided for separate parts of the system. We study two properties of decentralized specifications pertaining to monitorability and compatibility between specification and architecture. We also present a general algorithm for monitoring decentralized specifications. We map three existing algorithms to our approaches and provide a framework for analyzing their behavior. Furthermore, we present THEMIS, a framework for designing such decentralized algorithms and simulating their behavior. We demonstrate the usage of THEMIS to compare multiple algorithms and validate the trends predicted by the analysis in two scenarios: a synthetic benchmark and the Chiron user interface.

7.2.4 BISM: Bytecode-Level Instrumentation for Software Monitoring

In this work [9], we introduce BISM. BISM (Bytecode-level Instrumentation for Software Monitoring) is a lightweight Java bytecode instrumentation tool which features an expressive high-level control-flow-aware instrumentation language. The language follows the aspect-oriented programming paradigm by adopting the joinpoint model, advice inlining, and separate instrumentation mechanisms. BISM provides joinpoints ranging from bytecode instruction to method execution, access to comprehensive context information, and instrumentation methods. BISM runs in two modes: build-time and load-time. We demonstrate BISM effectiveness using two experiments: a security scenario and a general runtime verification case. The results show that BISM instrumentation incurs low runtime and memory overheads.

7.3 Teaching of Algorithms, Programming, Debugging, and Automata

Participants Florent Bouchez Tichadou, Yliès Falcone, Théo Barollet.

This domain is a new axis of the Corse team. Our goal here is to combine our expertise in compilation and teaching to help teachers and learners in computer science fields such as programming, algorithms, data strucures, automata, or more generally computing litteracy. The most important project in this regard is the automated generation and recommendation of exercises using artificial intelligence, a thesis that started last year. Other projects focus on tools to help learning through visualization (data structures, debugger, automata) or gamification (AppoLab), and are the source of many internships that give younger students experience in a research team.

7.3.1 AI4HI: Artificial Intelligence for Human Intelligence

In an ideal educative world, each learner would have access to individual pedagogical help, tailored to its needs. For instance, a tutor who could rapidly react to the questions, and propose pedagogical contents that match the learner's kills, and who could identify and work on his or her weaknesses. However, the real world imposes constraints that make this individual pedagogical help hard to achieve.

The goal of the AI4HI project is to combine the new advances in artificial intelligence with the team's skills in compilation and teaching to aid teaching through the automated generation and recommendation of exercises to learners. In particular, we target the teaching of programming and debugging to novices. This system will propose exercises that match the learners' needs and hence improve the learning, progression, and self-confidence of learners.

This projet has received an "Action Exploratoire" funding from Inria and Théo Barollet started his PhD in September 2019. There is a collaboration ongoing with the startup Toxicode, who provided data from one of their educational games. Our current goal is to be able to predict reliably how much time or tries a student will take to successfully pass an exercice, for which we also use publicly avaible data from the educational data mining community. So far, we had mitigated results using CNN neural networks but now promising results using matrix factorization.

7.3.2 AppoLab

Classical teaching of algorithms and low-level data structures is often tedious and unappealing to students. AppoLab is an online platform to engage students in their learning by including gamification in Problem-Based Learning. In its core, it is a server with scripted "exercises". Students can communicate with the server manually, but ultimately they need to script the communication also from their side, since the server will gradually impose constraints on the problems such as timeouts or large input sizes. This platform receives gradual improvements. This year we provided a full "game" on the platform to teach basic data structure complexity, and how to understand and modify basic algorithms workings on lists over a three-week period.

7.3.3 Data Structures and Program Visualization at Runtime

Debuggers are powerful tools to observe a program behavior and find bugs but they have a hard learning curve. They provide information on low level data but are not able to analyze higher level elements such as data structures. This work tries to provide a more intuitive representation of the program execution to ease debugging and algorithms understanding. We developed a prototype, Moly, a GDB extension that explores a program runtime memory and analyze its data structures. It provides an interface with an external visualizer, Lotos, through a formatted output. Work has also started to include a tutorial on how to use GDB and these extensions but was put on hold due to the difficulties in recruiting interns during the Covid crisis.

7.4 COVID related activities

Participants Guillaume Iooss, Auguste Olivry, Fabrice Rastello, Nicolas Terzi *(CHU Grenoble)*, Christophe Déhan *(MinMaxMedical)*, Florian Sigaud *(CHU Grenoble)*, Guillaume Rigault *(CHU Grenoble)*, Stan Borkowski, Cyril Fromentin *(FineHeart)*, Adrien Farrugia *(SteadXP)*, Claude Guérin *(HCL Lyon)*, Emeline Lagrange *(CHU Grenoble)*.

7.4.1 Mechanized Assisted Ventilation

The COVID-19 is a pneumonia that may culminate in the acute respiratory distress syndrome (ARDS). With the pandemic, intensive care unit (ICU) beds and ventilators have become resources of the utmost value. While France and many other countries were able to rapidly increase the number of ICU beds by upgrading step-down units and post-operative recovery rooms, a major shortage of ventilators became a worldwide critical concern. Therefore, COVID-19 pandemic stressed the need for emergency ventilator systems that can be rapidly deployed to try to avoid that the demand for ventilators overcame their supply. Various scenarios, such as regional emergencies, global pandemic, or situation in low-resource ICUs, require a ventilator sharing strategy that maximizes the number of patients able to receive potentially life-saving treatment. To address this issue of ventilator shortage, our group (eSpiro Network) developed a rapidly deployable and open-source ventilator. Based on frugal innovation, we expect it to allow performing better care with fewer resources, and for more critically ill patients.

This work has been done in the context of the Recovid project supported by Inria.

7.4.2 Epidemiological Modeling

Epidemiological modeling is an important tool that allows to predict the evolution of a pandemic (predictive) and understand the parameters that affect it (mechanistic). There exist several approaches usually opposed: agent-based versus compartmental; stochastic versus deterministic; analytical versus computational. The different approaches trade expressiveness with complexity. In this project we started the development of an open-source computational model which objective is to provide the expressiveness of a stochastic agent-based model but with a computational complexity orders of magnitude more efficient. Because of health issues (long-COVID) of the leader of the project, the developments have been interrupted but should start again in 2021.

7.4.3 Privacy-by-Design Survey Application

Most of health database used for COVID suffer from several problems: lack of coordination (double counting); reliability (manual gathering, obfuscation); representativeness (no quota method); lack of longitudinal data... The objective of the project is the development of an open-source application survey that would provide longitudinal data. As opposed to Zoe (Covid Symptom Study App – `https://covid.joinzoe.com`) that has been deployed in the UK, the ambition is to provide privacy by design, but also agility with special attention not to weary the cohort with too long surveys. We first considered extending Zoe to our usage in France, but several design choices in this application lead us not to choose this direction. We are at the stage of evaluating if REDCap `https://www.project-redcap.org/` can be adapted for our purpose. As for the previous project (epidemiological modeling), health issues interrupted the project. The hope is to be able to restart it again in 2021.

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

8.1.1 Atos/Bull

- Title: Static and dynamic approaches for the optimization of the energy consumption associated with applications of the High Performance Computing (HPC) field

- CORSE participants: François Broquedis, Frédéric Desprez, Mathieu Stoffel
- Partner: Atos/Bull
- Duration: February 2018 - February 2021
- Abstract: The purpose of this project is to dynamically improve the energy consumption of HPC applications on large-scale platforms. It relies on an adaptation of the CPU frequency at runtime, based on the analysis of hardware-related metrics to determine an *application profile*. This profile is then split into different *phases*, each of which being associated to a best CPU frequency, depending on its nature (CPU bound, memory bound, ...). This project is funding the PhD of Mathieu Stoffel, and the corresponding development is to be integrated into *Bull Dynamic Power Optimizer*, a software suite developed by Atos/Bull.

8.2 Bilateral grants with industry

8.2.1 ES3CAP

- Title: Embedded Smart Safe Secure Computing Autonomous Platform
- CORSE participants: Fabrice Rastello, Nicolas Tolenaere
- Duration: July 2018 - February 2022
- INRIA Partners: AOSTE, PARKAS, CHROMA
- Other Partners: Renault-Nissan, EasyMile, Safran E&D, MBDA, ANSYS/ESterel Technologies, Kronno-Safe, Prove & Run, Kalray, Prophesee, CEA
- Abstract: The objective of ES3CAP is to develop a tool-chain that targets multi- and many-core architectures for critical systems. In particular it should address the different challenges related to making existing critical systems solutions (heterogeneous, decentralized, single-core, single-task) match the industrial constraints targeted by Kalray's MPPA (MPPA, high-performance, real-time, safety, security). Considered applications are autonmous driving, drones, avionics, and defense. CORSE is involved in the optimization of machine learning algorithms for many-core architectures.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Inria associate team not involved in an IIL

IOComplexity

Title: IOComplexity

Duration: 2018 - 2020

Coordinator: Fabrice Rastello

Partners:

- Department of computer science, University of Utah (United States)

Inria contact: Fabrice Rastello

Summary: The goal of this project is to extend techniques for automatic characterisation of data movement of an application to the design of performance estimation.

The EA has three main objectives: 1. broader applicability of IO complexity analysis; 2. Hardware characterisation; 3. Performance model.

9.2 European initiatives

9.2.1 FP7 & H2020 Projects

CPS4EU

Title: Cyber Physical Systems for Europe

Duration: July 2019 – June 2022

Coordinator: Philippe Gougeon, VALEO VISION SAS

Partners:

- ABENGOA INNOVACION SOCIEDAD ANONIMA (Spain)
- ANSYS FRANCE SAS (France)
- BUDAPESTI MUSZAKI ES GAZDASAGTUDOMANYI EGYETEM (Hungary)
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (France)
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (France)
- EMMTRIX TECHNOLOGIES GMBH (Germany)
- ETH LAB SRL (Italy)
- EUROTECH SPA (Italy)
- FUNDACION CENTRO DE TECNOLOGIAS DE INTERACCION VISUAL Y COMUNICACIONES VICOMTECH (Spain)
- GREENWAVES TECHNOLOGIES (France)
- INSTITUTO TECNOLOGICO DE INFORMATICA (Spain)
- KALRAY SA (France)
- LEONARDO - SOCIETA PER AZIONI (Italy)
- M3 SYSTEMS SAS (France)
- PROVE & RUN (France)
- SCHNEIDER ELECTRIC FRANCE SAS (France)
- SEQUANS COMMUNICATIONS SA (France)
- SHERPA ENGINEERING SA (France)
- SPINSPLIT MUSZAKI KUTATO FEJLESZTOKFT (Hungary)
- TECHNISCHE UNIVERSITAT CLAUSTHAL (Germany)
- TECNOLOGIAS SERVICIOS TELEMATICOS YSISTEMAS SA (Spain)
- THALES (France)
- UNIVERSITAET AUGSBURG (Germany)
- UNIVERSITE DE LORRAINE (France)
- UNIVERSITE GRENOBLE ALPES (France)
- VALEO COMFORT AND DRIVING ASSISTANCE (France)
- VALEO VISION SAS (France)

Inria contact: Fabrice Rastello

Summary: Cyber Physical Systems (CPS) represent key drivers for the innovation capacity of European industries, large and small, generating sustainable economic growth and supporting meaningful jobs for citizens. The ultimate objective of CPS4EU is to strengthen the CPS value chain by creating world class European SMEs and by providing CPS technologies that in turn will sustain the leadership of the large European groups in key economy sectors and, in this way will stimulate innovative products to support the massive digitization increasingly integrated into our everyday environment.

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: selection

Member of the Conference Steering Committees

- Yliès Falcone: Member of the Steering Committee of the Runtime Verification conference 2020.
- Yliès Falcone: Member of the Steering Committee of Software Verification and Testing track at the Symposium on Applied Computing.
- Fabrice Rastello: Steering Committee chair of IEEE/ACM CGO

Member of the conference program committees

- Yliès Falcone: Member of the Program Committee of the Runtime Verificaiton conference 2020.
- Yliès Falcone: Member of the Program Committee of Software Verification and Testing track at the Symposium on Applied Computing 2021.
- François Broquedis: Member of the Program Committee of the Conférence francophone d'informatique en Parallélisme, Architecture et Système (COMPAS 2020)
- Frédéric Desprez: Member of the Program Committee of the Conférence CLOSER (9th International Conference on Cloud Computing and Services Science)
- Frédéric Desprez: Member of the Program Committee of the Conférence CloudCom 2020 (12th IEEE International Conference on Cloud Computing Technology and Science

Reviewer

- Fabrice Rastello: ACM Compiler Construction 2021

10.1.2 Journal

Member of the Editorial Boards

- Frédéric Desprez: IEEE Transactions on Cloud Computing (associate editor)

Reviewer - reviewing activities

- Yliès Falcone: External reviewer for FOSSACS 2021.
- Yliès Falcone: Science of Computer Programming Journal. Fabrice Rastello: ACM Transactions on Programming Languages and Systems

10.1.3 Leadership within the scientific community

- Frédéric Desprez: co-présidence du prix de thèse annuel du GDR Réseaux et Systèmes Distribués (RSD) en collaboration avec l'association ACM SIGOPS France (ASF)
- Frédéric Desprez: Scientific committee of ORAP
- Frédéric Desprez: Technical Committee of GENCI

10.1.4 Scientific expertise

- Frédéric Desprez: Genci: attribution heures de calcul (CT6)

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- License 3: François Broquedis, Imperative programming using python, 40 hours, Grenoble Institute of Technology (Ensimag)
- License 3: François Broquedis, Introduction to UNIX, 20 hours, Grenoble Institute of Technology (Ensimag)
- License 3: François Broquedis, Computer architecture, 40 hours, Grenoble Institute of Technology (Ensimag)
- License 3: François Broquedis, C programming, 100 hours, Grenoble Institute of Technology (Ensimag)
- Master 1: François Broquedis, Operating systems and concurrent programming, 30 hours, Grenoble Institute of Technology (Ensimag)
- Master 1: François Broquedis, Operating Systems Development Project - Fundamentals, 20 hours, Grenoble Institute of Technology (Ensimag)
- Master 1: François Broquedis, Object-Oriented Programming, 20 hours, Grenoble Institute of Technology (Ensimag)
- François Broquedis is in charge of the first year study at Ensimag
- Master: Florent Bouchez Tichadou, Algorithmic Problem Solving, 41 hours, M1 MoSIG
- Licence: Florent Bouchez Tichadou, Algorithms languages and programming, 106 hours, L2 UGA
- Data Asperger, Florent Bouchez Tichadou, Formation d'autistes Asperger aux métiers du développement informatique et analyse des données. Bloc Algorithmique. 30 hours, GEM & Le Campus Numérique.
- Master 1: Yliès Falcone, Programming Language Semantics and Compiler Design, MoSIG and Master informatique, 45 hours
- License: Yliès Falcone, Languages and Automata, Univ. Grenoble Alpes, 45 hours
- Master: Yliès Falcone, is responsible for the two above courses.
- License 3: Manuel Selva, Imperative programming using python, 144 hours, Grenoble Institute of Technology (Ensimag)
- License 3: Manuel Selva is responsible for the above course.
- License 3: Manuel Selva, C programming, 30 hours, Grenoble Institute of Technology (Ensimag)
- License 3: Manuel Selva, Algorithms and data structures, 18 hours, Grenoble Institute of Technology (Ensimag)

10.2.2 Supervision

- PhD in progress: Mathieu Stoffel, Static and dynamic approaches for the optimization of the energy consumption associated with applications of the High Performance Computing (HPC) field, February 2018, advised by François Broquedis, Frédéric Desprez, Abdelhafid Mazouz (Atos/Bull) and Philippe Rols (Atos/Bull)
- PhD in progress: Auguste Olivry, Data Locality and Parallelism Optimization for Linear and Multilinear Algebra, September 2019, advised by Fabrice Rastello.

- PhD in progress: Nicolas Tollenaere, Optimizing ML algorithms for MPPA Asics, April 2019, advised by Fabrice Rastello.
- PhD in progress: Nicolas Derumigny, Automatic generation of performance models for heterogeneous architectures, September 2019, advised by Fabrice Rastello.
- PhD in progress: Théo Barollet, Problem-based learning: automatic generation and recommendation of programming exercises, September 2019, advised by Florent Bouchez Tichadou and Fabrice Rastello.
- PhD in progress: Chukri Soueidi, Instrumentation, Runtime Verification and Enforcement for Multithreaded Programs, started in October 2020, advised by Yliès Falcone.
- PhD in progress: Nihel Kaboubi, distributed and collaborative AI for the IoT in the Fog Computing Environment, started in January 2020, advised by Frédéric Desprez, Thierry Coupaye and Loic Letondeur (Orange Labs)

10.2.3 Juries

Frédéric Desprez

- Arif Ahmed - Rennes, 20/01/20 "Efficient Cloud Application Deployment in Distributed Fog Infrastructures"
- Nikos Parlavantzas - Rennes, HDR, Rapporteur, 15/06/20 "Automated Application and Resource Management in the Cloud"
- Arthur Chevalier - Lyon, Rapporteur, 24/11/20 "Optimisation du placement des licences logicielles dans le Cloud pour un déploiement économique et efficient"
- Hrachya Astsatryan - Toulouse, HDR, Rapporteur, 01/12/20 "Service Tradeoff for HPC and Big Data Infrastructures"
- Olivier Aumage - Bordeaux, HDR, Examinateur, 14/12/20 "Instruments of Productivity for High Performance Computing"

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

- Frédéric Desprez: Deputy Scientific Director at INRIA
- Frédéric Desprez: Director of the GIS GRID5000
- Frédéric Desprez: Conseil Scientifique ESIEE Paris
- Frédéric Desprez: Groupe de travail “Infrastructures" Inria
- Frédéric Desprez: Director of the Grenoble Rhône-Alpes research center since October 1st, 2020.

10.3.2 Education

- DIU EIL, Florent Bouchez Tichadou, Formation des enseignants des lycées suite à la réforme du Bac et l’introduction de l’option informatique en 1ère et Terminale (NSI). Apprentissage Par Problèmes. Algorithmique, programmation, debug. Académie de Grenoble.

10.3.3 Book in french: Automates à états finis et langages réguliers - Rappels des notions essentielles et plus de 170 exercices corrigés

In this work, we provide a complete support reference manual for Bachelor students involved in a course on automata theory and regular languages. The book covers the major topics: languages, finite-state automate in their various form, regular expressions, grammars. It provides intuitive summaries, lecture notes, and exercises with their solutions.

11 Scientific production

11.1 Publications of the year

International journals

[1] F. Ait Salaht, F. Desprez and A. Lebre. ‘An overview of service placement problem in Fog and Edge Computing’. In: *ACM Computing Surveys* 53.3 (June 2020), Article 65, 35 pages. DOI: 10.1145/3391196. URL: https://hal.inria.fr/hal-02596419.

[2] Y. Falcone and L. Mariani. ‘Preface to the Special Section Issue on Improving Software Quality through Formal Methods’. In: *Software Quality Journal* (17th Feb. 2020), pp. 1–2. DOI: 10.1007/s11219-020-09508-z. URL: https://hal.inria.fr/hal-02548907.

[3] M. Jaber, Y. Falcone, P. Attie, A.-A. Khalil, R. Hallal and A. El-Hokayem. ‘From global choreographies to verifiable efficient distributed implementations’. In: *Journal of Logical and Algebraic Methods in Programming* 115 (Oct. 2020), p. 100577. DOI: 10.1016/j.jlamp.2020.100577. URL: https://hal.inria.fr/hal-03113398.

[4] M. Renard, A. Rollet and Y. Falcone. ‘Runtime enforcement of timed properties using games’. In: *Formal Aspects of Computing* 32.2-3 (July 2020), pp. 315–360. DOI: 10.1007/s00165-020-00515-2. URL: https://hal.archives-ouvertes.fr/hal-02920384.

International peer-reviewed conferences

[5] M. Á. Abella-González, P. Carollo-Fernández, L.-N. Pouchet, F. Rastello and G. Rodríguez. ‘PolyBench/Python: benchmarking Python environments with polyhedral optimizations’. In: CC ’21: 30th ACM SIGPLAN International Conference on Compiler Construction. Seoul, South Korea, Mar. 2021, pp. 59–70. DOI: 10.1145/3446804.3446842. URL: https://hal.inria.fr/hal-03153351.

[6] C. Alias, G. Iooss and S. Rajopadhye. ‘On the Verification of Polyhedral Program Transformations’. In: 18th International Conference on High Performance Computing & Simulation (HPCS 2020), 3rd Special Session on Compiler Architecture, Design and Optimization (CADO 2020). Barcelona, Spain, 25th Jan. 2020. URL: https://hal.archives-ouvertes.fr/hal-03106070.

[7] S. Emre, A. Sukumaran-Rajam, F. Rastello and P. Sadayyapan. ‘Efficient Tiled Sparse Matrix Multiplication through Matrix Signatures’. In: SC 20 - International Conference for High Performance Computing, Networking, Storage and Analysis. virtual, United States, 16th Nov. 2020. URL: https://hal.inria.fr/hal-03117491.

[8] A. Olivry, J. Langou, L.-N. Pouchet, P. Sadayappan and F. Rastello. ‘Automated derivation of parametric data movement lower bounds for affine programs’. In: PLDI ’20: 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation. London, United Kingdom, 17th June 2020, pp. 808–822. DOI: 10.1145/3385412.3385989. URL: https://hal.inria.fr/hal-02910961.

[9] C. Soueidi, A. Kassem and Y. Falcone. ‘BISM: Bytecode-Level Instrumentation for Software Monitoring’. In: International Conference on Runtime Verification. Los Angeles, United States, 6th Oct. 2020. URL: https://hal.inria.fr/hal-03081265.

Scientific books

[10] Y. Falcone and J.-C. Fernandez. *Automates à états finis et langages réguliers.* 1st July 2020. URL: https://hal.inria.fr/hal-03113401.

Reports & preprints

[11] N. Derumigny, F. Gruber, T. Bastian, G. Iooss, C. Guillon, L.-N. Pouchet and F. Rastello. *From micro-OPs to abstract resources: constructing a simpler CPU performance model through microbenchmarking.* 19th Jan. 2021. URL: https://hal.inria.fr/hal-03114933.

[12] N. Tollenaere, A. Olivry, G. Iooss, H. Brunie, A. Cohen, P. Sadayappan and F. Rastello. *Efficient convolution optimisation by composing micro-kernels*. 23rd Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03149553.