IN PARTNERSHIP WITH:
Institut polytechnique de Grenoble

Université de Grenoble Alpes

Activity Report 2019

Project-Team CORSE

compiler optimization and run-time systems

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

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Architecture, Languages and Compilation

Table of contents

Project-Team CORSE

Creation of the Team: 2014 November 01, updated into Project-Team: 2016 July 01

Keywords:

Computer Science and Digital Science:

A1.1.1. - Multicore, Manycore
A1.1.3. - Memory models
A1.6. - Green Computing
A2.1.6. - Concurrent programming
A2.1.7. - Distributed 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...
A2.3.2. - Cyber-physical systems
A4.4. - Security of equipment and software
A7.1. - Algorithms

Other Research Topics and Application Domains:

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, Visitors, External Collaborators

Research Scientists

Fabrice Rastello [Team leader, Inria, Senior Researcher, HDR]
Frederic Desprez [Inria, Senior Researcher, 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, from Sep 2019]

Post-Doctoral Fellow

Manuel Selva [Univ Grenoble Alpes, Post-Doctoral Fellow, until Aug 2019]

PhD Students

Théo Barollet [Inria, PhD Student, from Oct 2019]

Georgios Christodoulis [Univ Grenoble Alpes, PhD Student, until Dec 2019]
Nicolas Derumigny [Univ Grenoble Alpes, PhD Student, from Sep 2019]
Fabian Gruber [Univ Grenoble Alpes, PhD Student, until Sep 2019]
Raphael Jakse [Univ Grenoble Alpes, PhD Student, until Sep 2019]
Auguste Olivry [Univ Grenoble Alpes, PhD Student, from Sep 2019]
Mathieu Stoffel [Bull, PhD Student]
Nicolas Tollenaere [Inria, PhD Student, from Apr 2019]

Technical staff

Ali Kassem [Inria, Engineer]
Chukri Soueidi [Inria, Engineer, from Oct 2019]

Interns and Apprentices

Théo Barollet [Inria, from Mar 2019 until Aug 2019]
Nicolas Derumigny [Ecole Normale Supérieure Lyon, from May 2019 until Aug 2019]
Auguste Olivry [Ecole Normale Supérieure Paris, from Mar 2019 until Jul 2019]
Alaa Ben Fatma [Inria, from May 2019 until Aug 2019]
Antoine Clavel [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Florian Gallay [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Lisa Girod [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Thomas Hervé [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Bruno Lehouque [Univ Grenoble Alpes, until Jun 2019]
Anthony Martinez [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Beryl Piasentin [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Fabien Portas Pittet [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]
Steven Sengchanh [Univ Grenoble Alpes, from Jun 2019 until Jul 2019]

Administrative Assistant

Maria Immaculada Presseguer [Inria, Administrative Assistant]

2. Overall Objectives

2.1. 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. New Software and Platforms

5.1. 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.

- Participants: Kevin Pouget, Ylies Falcone, Raphael Jakse and Jean-François Méhaut
- Contact: Raphael Jakse
- Publication: Interactive Runtime Verification - When Interactive Debugging meets Runtime Verification
- URL: https://gitlab.inria.fr/monitoring/verde

5.2. Mickey

KEYWORDS: Dynamic Analysis - Performance analysis - Profiling - Polyhedral compilation

FUNCTIONAL DESCRIPTION: Mickey is a set of tools for profiling based performance debugging for compiled binaries. It uses a dynamic binary translator to instrument arbitrary programs as they are being run to reconstruct the control flow and track data dependencies. This information is then fed to a polyhedral optimizer that proposes structured transformations for the original code.

Mickey can handle both inter- and intra-procedural control and data flow in a unified way, thus enabling inter-procedural structured transformations. It is based on QEMU to allow for portability, both in terms of targeted CPU architectures, but also in terms of programming environment and the use of third-party libraries for which no source code is available.

- Partner: STMicroelectronics
- Contact: Fabrice Rastello
- Publications: hal-02060796v1 - hal-01967828v2

5.3. 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.

- Contact: Nicolas Derumigny
- URL: https://gitlab.inria.fr/nderumig/gus

5.4. 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.

- Contact: Nicolas Derumigny
- URL: https://gitlab.inria.fr/fgruber/pipedream

5.5. Platforms

5.5.1. 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.

5.5.2. 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) is also currently designed with other european partners (Spain, Cyprus, Greece, Netherland, Switzerland, Poland, ...).

6. New Results

6.1. Compiler Optimizations and Analysis

Participants: Fabrice Rastello, Manuel Selva, Fabian Grüber, Diogo Sampaio [CORSE, Inria], Christophe Guillon [STMicroelectronics], 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].

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

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. Our first result presented at SC 2019 [10] focuses on tensor contractions. 2. The second consists in developing dynamic analysis based performance debugging tools. Our first results published at PPoPP 2019 [9] and TACO 2020 [7] shows a scalable approach that compresses an execution trace obtained from binary instrumentation and analyses it using a polyhedral compiler.

6.1.1. Analytical Cache Modeling and Tilesize Optimization for Tensor Contractions

Data movement between processor and memory hierarchy is a fundamental bottleneck that limits the performance of many applications on modern computer architectures. Tiling and loop permutation are key techniques for improving data locality. However, selecting effective tile-sizes and loop permutations is particularly challenging for tensor contractions due to the large number of loops. Even state-of-the-art compilers usually produce sub-optimal tile-sizes and loop permutations, as they rely on naïve cost models. In this work we provide an analytical model based approach to multilevel tile size optimization and permutation selection for tensor contractions. Our experimental results show that this approach achieves comparable or better performance than state-of-the-art frameworks and libraries for tensor contractions.

This work is the fruit of the collaboration 8.3.1.1 with OSU. It has been presented at ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis, SC 2019 [10].

6.1.2. Profiling-based Polyhedral Optimization Feedback

This work addresses the problem of reconstructing a compact (static) representation of a binary execution, automatically detecting hot regions and enabling precise feedback about optimization opportunities potentially missed by the compiler. Our framework handles codes with irregular accesses, pointers with indirections, inter-procedural or recursive loop regions. By enabling binary execution analysis we are able to discover runtime properties (i.e., the ability to form a compact representation) as well as inter-procedural optimization opportunities that cannot be uncovered by standard static analyses. Our design choices were driven towards achieving portability, both in terms of targeted architecture, but also in terms of programming environment (e.g., being robust to arbitrary programming language, compiler, use of third-party binaries, etc.).

A compact and yet precise inter-procedural dynamic dependence graph (DDG) is first computed via: 1. a new instrumentation framework based on QEMU; 2. the use of a new concept of inter-procedural loop-nesting tree; 3. followed by new techniques we introduce for folding, clamping, and widening of the DDG to agglomerate dynamic dependence instances into polyhedra of integer points whenever possible. State-of-the-art polyhedral analysis and transformation systems we specifically modified to provide useful feedback to the user is then used. We extensively evaluate our tool on numerous benchmarks, demonstrating the pratical usefulness of our tool-chain.

This work is the fruit of the collaboration 8.3.1.1 with OSU and and the past collaboration Nano2017 with STMicroelectronics. The main contributions has been presented at the ACM conference on Principles and Practice of Parallel Programming, PPoPP 2019 [9]. The new techniques that allow to build the polyhedral representation from the instrumented execution in a scalable way lead to a separate publication in the ACM Transactions on Architecture and Code Optimization, TACO 2020 [7].

6.2. 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. The first phase of the thesis project consists in an in-depth study of the evolution of the metrics characterizing the state of the supercomputer during the execution of a highly parallel application. Indeed, the utilization rates of the different components of the HPC system may demonstrate extreme variations during the execution of the aforementioned application. These variations are sometimes subject to repeat themselves on a regular basis during the application execution. We refer to this phenomena as application "phases". In this context, we developed a tool suite resorting to fine-grain profiling and periodicity analysis to identify optimization opportunities for both performance and power-efficiency. It leverages the fact that a large share of HPC parallel applications are constituted of a restrained set of compute kernels executed a huge number of times to extract periodic patterns representative of the aforementioned kernels. By doing so, our tool offers a simple and condensed proxy to analyze and predict the behavior of complex parallel applications. For instance, we were able to identify and extract periodic patterns for a panel of reference HPC applications such as NAMD and NEMO. Then, as an example of the many ways to exploit the aforementioned extracted periodic patterns, we evaluated the impact of the CPU frequency on the latter. As a result, we were able to identify DVFS opportunities we plan to exploit in a future work.

6.3. Runtime Monitoring, Verification, and Enforcement

Participants: Antoine El-Hokayem [Univ. Grenoble Alpes, Verimag], Yliès Falcone, Thierry Jéron [Inria Rennes], Ali Kassem, Hervé Marchand [Inria Rennes], Srinivas Pinisetty [IIT Bhubaneswar], Matthieu Renard [Foxi], Antoine Rollet [Université de Bordeaux], César Sànchez [IMDEA Madrid], Gerardo Schneider [University of Gothenborg].

Our contributions in the domain of runtime monitoring, verification, and enforcement are threefold. First, we contributed to the publication of general papers aimed to structure the community by publishing a tutorial on runtime enforcement of timed properties [16], a review of the first five years of the competition on runtime verification [15] and a survey of future challenges of runtime verification [6]. We also concluded some other previous work by realizing journal publications on the topics of decentralized runtime verification [3] and on runtime enforcement of timed properties [5]. We started a new activity on monitoring for security properties, and more particularly on the detection of fault-injection attacks [12].

6.3.1. On the Runtime Enforcement of Timed Properties

This work [16] is concerned with runtime enforcement which refers to the theories, techniques, and tools for enforcing correct behavior of systems at runtime. We are interested in such behaviors described by specifications that feature timing constraints formalized in what is generally referred to as timed properties. This tutorial presents a gentle introduction to runtime enforcement (of timed properties). First, we present a taxonomy of the main principles and concepts involved in runtime enforcement. Then, we give a brief overview of a line of research on theoretical runtime enforcement where timed properties are described by timed automata and feature uncontrollable events. Then, we mention some tools capable of runtime enforcement, and we present the TiPEX tool dedicated to timed properties. Finally, we present some open challenges and avenues for future work.

6.3.2. Detecting Fault Injection Attacks with Runtime Verification

This work [12] is concerned with fault injections which are increasingly used to attack/test secure applications. In this paper, we define formal models of runtime monitors that can detect fault injections that result in test inversion attacks and arbitrary jumps in the control flow. Runtime verification monitors offer several advantages. The code implementing a monitor is small compared to the entire application code. Monitors have a formal semantics; and we prove that they effectively detect attacks. Each monitor is a module dedicated to detecting an attack and can be deployed as needed to secure the application. A monitor can run separately from the application or it can be weaved inside the application. Our monitors have been validated by detecting simulated attacks on a program that verifies a user PIN.

6.3.3. International Competition on Runtime Verification (CRV)

In this work [15], we review the first five years of the international Competition on Runtime Verification (CRV), which began in 2014. Runtime verification focuses on verifying system executions directly and is a useful lightweight technique to complement static verification techniques. The competition has gone through a number of changes since its introduction, which we highlight in this paper.

6.3.4. A Survey of Challenges for Runtime Verification from Advanced Application Domains (beyond software)

In this work [6], we survey the future challenges for runtime verification. Typically, the two main activities in runtime verification efforts are the process of creating monitors from specifications, and the algorithms for the evaluation of traces against the generated monitors. Other activities involve the instrumentation of the system to generate the trace and the communication between the system under analysis and the monitor. Most of the applications in runtime verification have been focused on the dynamic analysis of software, even though there are many more potential applications to other computational devices and target systems. In this paper we present a collection of challenges for runtime verification extracted from concrete application domains, focusing on the difficulties that must be overcome to tackle these specific challenges. The computational models that characterize these domains require to devise new techniques beyond the current state of the art in runtime verification.

6.3.5. On the Monitoring of Decentralized Specifications Semantics, Properties, Analysis, and Simulation

In this work [3], we define two complementary approaches to monitor decentralized systems. The first relies on those 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 introduce THEMIS, a framework for designing such decentralized algorithms and simulating their behavior. We show the usage of THEMIS to compare multiple algorithms and verify the trends predicted by the analysis by studying two scenarios: a synthetic benchmark and a real example.

6.3.6. Optimal Enforcement of (timed) Properties with Uncontrollable Events

This work deals with runtime enforcement of untimed and timed properties with uncontrollable events [5]. Runtime enforcement consists in defining and using mechanisms that modify the executions of a running system to ensure their correctness with respect to a desired property. We introduce a framework that takes as input any regular (timed) property described by a deterministic automaton over an alphabet of events, with some of these events being uncontrollable. An uncontrollable event cannot be delayed nor intercepted by an enforcement mechanism. Enforcement mechanisms should satisfy important properties, namely soundness, compliance and optimality – meaning that enforcement mechanisms should output as soon as possible correct executions that are as close as possible to the input execution. We define the conditions for a property to be enforceable with uncontrollable events. Moreover, we synthesise sound, compliant and optimal descriptions of runtime enforcement mechanisms at two levels of abstraction to facilitate their design and implementation.

6.4. Teaching of Algorithms, Programming, Debugging, and Automata

Participants: Florent Bouchez Tichadou, Yliès Falcone, Théo Barollet, Antoine Clavel, Thomas Hervé, Anthony Martinez, Beryl Piasentin, Steven Sengchanh.

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 this 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.

6.4.1. 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 this September so is still in its early stages.

6.4.2. 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.

6.4.3. Data Structures and Program Visualization at Runtime

Debuggers are powerful tools to observe a program behaviour 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.

6.4.4. Aude

Aude is a pedagogical software for manipulating, learning, and teaching finite state automata and the automata theory. It is used by the students in the second year of the bachelor in computer science at Univ. Grenoble Alpes. It allows students to get acquainted and autonomously work on the concepts involved in the theory of regular languages and automata.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

7.1.1. 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.

7.2. Bilateral Grants with Industry

7.2.1. ES3CAP

- Title: Embedded Smart Safe Secure Computing Autonomous Platform
- CORSE participants: Fabrice Rastello, Nicolas Tolenaere
- Duration: July 2018 - August 2021
- 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.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. HEAVEN Persyval Project

- Title: HEterogenous Architectures: Versatile Exploitation and programmiNg
- HEAVEN leaders: François Broquedis, Olivier Muller [TIMA lab]
- CORSE participants: François Broquedis, Frédéric Desprez, Georgios Christodoulis, Manuel Selva
- Duration: September 2015 - December 2019
- Abstract: The main objective of this project was to improve the accessibility of heterogeneous architectures comprising FPGA accelerators with portability and real experimentation in mind. The portability criterion allows application programmers to benefit from FPGA devices with only small modifications to their applications. It was achieved by extending a standard parallel programming environment already targeting heterogeneous architectures comprising CPUs and GPUs. During the project, we developed an operational prototype targeting Xilinx FPGAs. Experiments have been conducted using both matrix multiplication and Cholesky decomposition kernels. These experiments have shown the usability of the framework and its very low overhead. This framework opens the path for challenging questions regarding the scheduling of heterogeneous applications targeting FPGAs.

8.2. National Initiatives

8.2.1. IPL ZEP

- Title: Zero-Power computing systems
- Coordinator: Kevin Marquet (INRIA Socrate)
- CORSE participants: Fabrice Rastello
- Other INRIA Partners: Cairn, Pacap
- Duration: from Apr. 2017 to Sept. 2019
- Abstract: The ZEP project addresses the issue of designing tiny computing objects with no battery by combining non-volatile memory (NVRAM), energy harvesting, micro-architecture innovations, compiler optimizations, and static analysis. The main application target is Internet of Things (IoT) where small communicating objects will be composed of this computing part associated to a low-power wake-up radio system. The ZEP project gathers four Inria teams that have a scientific background in architecture, compilation, operating system and low power together with the CEA Lialp and Lisan laboratories of CEA LETI & LIST. The major outcomes of the project will be a prototype harvesting board including NVRAM and the design of a new microprocessor associated with its optimizing compiler and operating system.

8.3. International Initiatives

8.3.1. Inria Associate Teams Not Involved in an Inria International Labs

8.3.1.1. IOComplexity

Title: Automatic characterization of data movement complexity

International Partner (Institution - Laboratory - Researcher):

Ohio State University (United States). P. Sadayappan

Colorado State University (United States). Louis-Noël Pouchet

Start year: 2018

See also: https://team.inria.fr/corse/iocomplexity/

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

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

8.4. International Research Visitors

8.4.1. Visits to International Teams

8.4.1.1. Research Stays Abroad

- Fabrice Rastello visited the University of Utah to work with P. Sadayappan during the month of November. He worked on abstract simulation, and optimization of pattern specific programs.
- Nicolas Derumigny visited the University of Utah to work with P. Sadayappan during the month of November. He worked on abstract simulation.
- Nicolas Tollenaere visited the University of Utah to work with P. Sadayappan during the month of November. He worked on abstract simulation, and optimization of convolutions
- Theo Barollet visited the Colorado State University to work with Steve Kommrusch during the month of October. He worked on graph neural networks.
- Nicolas Tollenaere visited the university of Utah to work with P. Sadayappan during the month of August. He worked on optimizing packing and transposition of tensors.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events: Organisation

9.1.1.1. Member of the Organizing Committees

- Fabrice Rastello: Steering Committee ACM/IEEE CGO; Steering Committee “Journées française de la compilation”

9.1.2. Scientific Events: Selection

9.1.2.1. Member of the Conference Program Committees

- Frédéric Desprez: IPDPS, HPC, CCGRID, ParCo, ICA3PP, CloudCom
- François Broquedis: IPDPS, COMPAS
- Fabrice Rastello: ACM SIGPLAN/SIGBED LCTES 2019; IEEE/ACM SIGPLAN CGO 2020; IEEE/ACM SIGARCH PACT 2019; CADO 2019
- Yliès Falcone: AFADL’19, SHPCS’19, IFIP-ICTSS’19, RV’19, FDL’19, 4PAD’19, TASE’19, SAC-SVT’19
- Manuel Selva: ReCoSoC’19

9.1.3. Journal

9.1.3.1. Member of the Editorial Boards

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

9.1.3.2. Reviewer - Reviewing Activities

- Frédéric Desprez: ACM Transactions on Internet of Things, IEEE Access
- Fabrice Rastello: ACM TACO, ACM TOPLAS

9.1.4. 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

9.1.5. Scientific Expertise

- Frédéric Desprez: Genci: attribution heures de calcul CT6
- Frédéric Desprez: Groupe de travail “Cloud pour l’IA” d’Allistène
- Frédéric Desprez: Comité des sages IRIT
- Yliès Falcone: Review for ANR call for projects

9.1.6. Research Administration

- 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

9.2. Teaching - Supervision - Juries

9.2.1. 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, 80 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 responsible of the first year of Ensimag

Master: Florent Bouchez Tichadou, Algorithmic Problem Solving, 41 hours, M1 MoSIG

Licence: Florent Bouchez Tichadou, Algorithms languages and programming, 160 hours, L2 UGA

Licence: Florent Bouchez Tichadou was responsible of the second year (L2) of INF (informatique) and MIN (mathématiques et informatique) students at UGA from January to June, eq. 42 hours

Licence: Florent Bouchez Tichadou, Software Project, 10 hours, L3 UGA

Data Asperger, Florent Bouchez Tichadou, Formation d'autistes Asperger aux métiers du développement informatique et analyse des données. Bloc Algorithmique. 20 hours, GEM & Le Campus Numérique.

Master 1: Yliès Falcone, Programming Language Semantics and Compiler Design, MoSIG and Master informatique, 96 hours

License: Yliès Falcone, Languages and Automata, Univ. Grenoble Alpes, 105 hours

Master: Yliès Falcone, was co-responsible of the first year of the International Master of Computer Science (Univ. Grenoble Alpes and INP ENSIMAG) until August 2019.

License 3: Manuel Selva, Imperative programming using python, 60 hours, Grenoble Institute of Technology (Ensimag)

License 3: Manuel Selva, Introduction to UNIX, 15 hours, Grenoble Institute of Technology (Ensimag)

Master 1: Manuel Selva, Operating systems and concurrent programming, 15 hours, Grenoble Institute of Technology (Ensimag)

9.2.2. Supervision

PhD: Georgios Christodoulis, Adaptation of a heterogeneous run-time system to efficiently exploit FPGA, December 5, 2019, advised by Frederic Desprez, Olivier Muller (TIMA/SLS), and François Broquedis

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: Fabian Grüber, Performance Debugging Toolbox for Binaries: Sensitivity Analysis and Dependence Profiling, December 17 2019, advised by Fabrice Rastello

PhD: Raphaël Jakse, Interactive Runtime Verification, December 18 2019, advised by Jean-François Méhaut and Yliès Falcone.

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.

9.2.3. Juries

9.2.3.1. Fabrice Rastello

Fabian Grüber, advisor, *Performance Debugging Toolbox for Binaries: Sensitivity Analysis and Dependence Profiling*, PhD, Université Grenoble Alpes, December 17 2019
Maxime Schmitt, reviewer, *Automatic Generation of Adaptative Codes*, PhD, Université de Strasbourg, September 30 2019

9.2.3.2. Frédéric Desprez

Michael Mercier, examiner, *Contribution to High Performance Computing and Big Data Infrastructure Convergence*, PhD, Université Grenoble Alpes, January 07 2019
Alexandre Veith, examiner, *Quality of Service Aware Mechanisms for (Re)Configuring Data Stream Processing Applications on Highly Distributed Infrastructure*, PhD, Lyon, September 23 2019
Nathanaël Cheriere, reviewer, *Towards Malleable Distributed Storage Systems From Models to Practice*, PhD, Ecole normale supérieure de Rennes, November 5 2019
Ayham Kassab, reviewer/president, *Optimisation de l'ordonnancement de calculs parallèles et de l'engagement de sources d'énergie renouvelable pour l'alimentation des centres de calcul*, PhD, Université de Bourgogne Franche Comté, November 14 2019
Georgios Christodoulis, advisor, *Adaptation of a heterogeneous run-time system to efficiently exploit FPGA*, PhD, Université Grenoble Alpes, December 5, 2019

9.2.3.3. François Broquedis

- Georgios Christodoulis, advisor, *Adaptation of a heterogeneous run-time system to efficiently exploit FPGA*, PhD, Université Grenoble Alpes, December 5, 2019

9.2.3.4. Yliès Falcone

- Raphaël Jakse, advisor, *Interactive Runtime Verification*, PhD, Université Grenoble Alpes, December 18 2019.

9.3. Popularization

9.3.1. 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). Bloc algorithmique par Apprentissage Par Problèmes (APP). Académie de Grenoble ainsi que les enseignants dans les lycées français à l'étranger.

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] R. JAKSE. *Interactive runtime verification*, CORSE - Compiler Optimization and Run-time Systems ; Université Grenoble - Alpes ; LIG (Laboratoire informatique de Grenoble) ; Inria Grenoble Rhône-Alpes, December 2019, https://hal.inria.fr/tel-02460734

Articles in International Peer-Reviewed Journals

[2] E. CRUZ, M. DIENER, L. LIMA PILLA, P. NAVAUX. *EagerMap: A Task Mapping Algorithm to Improve Communication and Load Balancing in Clusters of Multicore Systems*, in "ACM Transactions on Parallel Computing", 2019, vol. 5, n° 4, 17 p. [*DOI :* 10.1145/3309711], https://hal.archives-ouvertes.fr/hal-02062952

[3] A. EL-HOKAYEM, Y. FALCONE. *On the Monitoring of Decentralized Specifications: Semantics, Properties, Analysis, and Simulation*, in "ACM Transactions on Software Engineering and Methodology", September 2019, pp. 1-57, https://hal.archives-ouvertes.fr/hal-02283429

[4] P. H. PENNA, A. T. A. GOMES, M. CASTRO, P. PLENTZ, H. C. D. FREITAS, F. BROQUEDIS, J.-F. MÉHAUT. *A Comprehensive Performance Evaluation of the BinLPT Workload-Aware Loop Scheduler*, in "Concurrency and Computation: Practice and Experience", February 2019, vol. 31, n° 18, pp. 1-22 [*DOI :* 10.1002/CPE.5170], https://hal.archives-ouvertes.fr/hal-01986361

[5] M. RENARD, Y. FALCONE, A. ROLLET, T. JÉRON, H. MARCHAND. *Optimal Enforcement of (Timed) Properties with Uncontrollable Events*, in "Mathematical Structures in Computer Science", 2019, vol. 29, n° 1, pp. 169-214 [*DOI :* 10.1017/S0960129517000123], https://hal.archives-ouvertes.fr/hal-01262444

[6] C. SANCHEZ, G. SCHNEIDER, W. AHRENDT, E. BARTOCCI, D. BIANCULLI, C. COLOMBO, Y. FALCONE, A. FRANCALANZA, S. KRSTIĆ, J. LOURENÇO, D. NICKOVIC, G. PACE, J. RUFINO, J. SIGNOLES, D. TRAYTEL, A. WEISS. *A survey of challenges for runtime verification from advanced application domains (beyond software)*, in "Formal Methods in System Design", November 2019, vol. 54, n° 3, pp. 279-335 [*DOI :* 10.1007/S10703-019-00337-W], https://hal.archives-ouvertes.fr/hal-02433558

[7] M. SELVA, F. GRUBER, D. SAMPAIO, C. GUILLON, L.-N. POUCHET, F. RASTELLO. *Building a Polyhedral Representation from an Instrumented Execution: Making Dynamic Analyses of Non-Affine Programs Scalable*, in "ACM Transactions on Architecture and Code Optimization", December 2019, vol. 16, n° 4, pp. 1-26 [*DOI :* 10.1145/3363785], https://hal.inria.fr/hal-02418987

International Conferences with Proceedings

[8] F. AIT SALAHT, F. DESPREZ, A. LEBRE, C. PRUD'HOMME, M. ABDERRAHIM. *Service Placement in Fog Computing Using Constraint Programming*, in "SCC 2019 : IEEE International Conference on Services Computing", Milan, Italy, IEEE, July 2019, pp. 19-27 [*DOI :* 10.1109/SCC.2019.00017], https://hal.archives-ouvertes.fr/hal-02108806

[9] F. GRUBER, M. SELVA, D. SAMPAIO, C. GUILLON, A. MOYNAULT, L.-N. POUCHET, F. RASTELLO. *Data-Flow/Dependence Profiling for Structured Transformations*, in "PPoPP 2019 - 24th Symposium on Principles and Practice of Parallel Programming", Washington, D.C., United States, ACM, February 2019, pp. 173-185 [*DOI :* 10.1145/3293883.3295737], https://hal.inria.fr/hal-02060796

[10] R. LI, A. SUKUMARAN-RAJAM, R. VERAS, T. M. LOW, F. RASTELLO, A. ROUNTEV, P. SADAYAPPAN. *Analytical Cache Modeling and Tilesize Optimization for Tensor Contractions*, in "SC 2019 - International Conference for High Performance Computing, Networking, Storage and Analysis", Denver, United States, ACM Press, November 2019, pp. 1-13 [*DOI :* 10.1145/3295500.3356218], https://hal.inria.fr/hal-02418875

[11] F. ZANON BOITO, J.-F. MÉHAUT, T. DEUTSCH, B. VIDEAU, F. DESPREZ. *Instrumental Data Management and Scientific Workflow Execution: the CEA case study*, in "IPDPSW 2019 - International Parallel and Distributed Processing Symposium Workshops", Rio de Janeiro, Brazil, IEEE, May 2019, pp. 850-857 [*DOI :* 10.1109/IPDPSW.2019.00139], https://hal.inria.fr/hal-02076963

Conferences without Proceedings

[12] A. KASSEM, Y. FALCONE. *Detecting Fault Injection Attacks with Runtime Verification*, in "SPRO 2019 - 3rd International Workshop on Software PROtection", Londres, United Kingdom, ACM, November 2019, pp. 65-76 [*DOI :* 10.1145/3338503.3357724], https://hal.archives-ouvertes.fr/hal-02283434

[13] P. H. PENNA, J. SOUTO, D. F. LIMA, M. CASTRO, F. BROQUEDIS, H. H. FREITAS, J.-F. MÉHAUT. *On the Performance and Isolation of Asymmetric Microkernel Design for Lightweight Manycores*, in "SBESC 2019 - IX Brazilian Symposium on Computing Systems Engineering", Natal, Brazil, November 2019, pp. 1-31, https://hal.archives-ouvertes.fr/hal-02297637

[14] P. H. PENNA, M. SOUZA, E. P. JUNIOR, J. SOUTO, M. CASTRO, F. BROQUEDIS, H. COTA DE FREITAS, J.-F. MÉHAUT. *RMem: An OS Service for Transparent Remote Memory Access in Lightweight Manycores*, in "MultiProg 2019 - 25th International Workshop on Programmability and Architectures for Heterogeneous Multicores", Valencia, Spain, High-Performance and Embedded Architectures and Compilers Workshops (HiPEAC Workshops), January 2019, pp. 1-16, https://hal.archives-ouvertes.fr/hal-01986366

Scientific Books (or Scientific Book chapters)

[15] E. BARTOCCI, Y. FALCONE, G. REGER. *International Competition on Runtime Verification (CRV)*, in "Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2019", Springer, April 2019, pp. 41-49 [*DOI :* 10.1007/978-3-030-17502-3_3], https://hal.archives-ouvertes.fr/hal-02433552

[16] Y. FALCONE, S. PINISETTY. *On the Runtime Enforcement of Timed Properties*, in "Proceedings of the Runtime Verification 2019 conference", Springer, October 2019, pp. 48-69 [*DOI :* 10.1007/978-3-030-32079-9_4], https://hal.archives-ouvertes.fr/hal-02433521

Research Reports

[17] F. AIT SALAHT, F. DESPREZ, A. LEBRE. *An overview of service placement problem in Fog and Edge Computing*, Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, LYON, France, October 2019, n° RR-9295, pp. 1-43, https://hal.inria.fr/hal-02313711

[18] F. GRUBER, M. SELVA, D. SAMPAIO, C. GUILLON, L.-N. POUCHET, F. RASTELLO. *Building of a Polyhedral Representation from an Instrumented Execution: Making Dynamic Analyses of non-Affine Programs Scalable*, CORSE - Compiler Optimization and Run-time Systems, January 2019, n° RR-9244, pp. 1-24, https://hal.inria.fr/hal-01967828

[19] R. JAKSE, Y. FALCONE, J.-F. MÉHAUT. *Interactive Runtime Verification: Formal Models, Algorithms, and Implementation*, UGA (Université Grenoble Alpes) ; LIG (Laboratoire informatique de Grenoble) ; Inria Grenoble Rhône-Alpes, Université de Grenoble, July 2019, https://hal.inria.fr/hal-02190656

Other Publications

[20] A. OLIVRY, J. LANGOU, L.-N. POUCHET, P. SADAYAPPAN, F. RASTELLO. *Automated Derivation of Parametric Data Movement Lower Bounds for Affine Programs*, December 2019, https://arxiv.org/abs/1911.06664 - working paper or preprint, https://hal.inria.fr/hal-02421026