

RESEARCH CENTRE

**Inria Centre at the University of  
Bordeaux**

IN PARTNERSHIP WITH:

**Institut Polytechnique de Bordeaux,  
Université de Bordeaux, CNRS**

2024

ACTIVITY REPORT

Project-Team

STORM

## **Static Optimizations, Runtime Methods**

IN COLLABORATION WITH: Laboratoire Bordelais de Recherche en  
Informatique (LaBRI)

### **DOMAIN**

**Networks, Systems and Services,  
Distributed Computing**

### **THEME**

**Distributed and High Performance  
Computing**

*Inria*

# Contents

|   |           |
|---|-----------|
| <b>Project-Team STORM</b>   | <b>1</b>  |
| <b>1 Team members, visitors, external collaborators</b>   | <b>2</b>  |
| <b>2 Overall objectives</b>   | <b>3</b>  |
| <b>3 Research program</b>   | <b>5</b>  |
| 3.1 Parallel Computing and Architectures . . . . .  | 5         |
| 3.2 Scientific and Societal Stakes . . . . .  | 5         |
| 3.3 Towards More Abstraction . . . . .  | 6         |
| <b>4 Application domains</b>  | <b>7</b>  |
| 4.1 Application domains benefiting from HPC . . . . .   | 7         |
| 4.2 Application in High performance computing/Big Data . . . . .  | 7         |
| <b>5 Highlights of the year</b>   | <b>7</b>  |
| 5.1 Awards . . . . .  | 7         |
| <b>6 New software, platforms, open data</b>   | <b>7</b>  |
| 6.1 New software . . . . .  | 7         |
| 6.1.1 AFF3CT . . . . .  | 7         |
| 6.1.2 PARCOACH . . . . .  | 8         |
| 6.1.3 MIPP . . . . .  | 8         |
| 6.1.4 CERE . . . . .  | 9         |
| 6.1.5 DUF . . . . .   | 9         |
| 6.1.6 MBI . . . . .   | 9         |
| 6.1.7 EasyPAP . . . . .   | 9         |
| 6.1.8 StarPU . . . . .  | 10        |
| <b>7 New results</b>  | <b>11</b> |
| 7.1 Scheduling for Pipelined and Replicated Task Chains and Graphs for Software-Defined Radio   | 11        |
| 7.2 Optimization Space Exploration . . . . .  | 11        |
| 7.3 Task scheduling with memory constraints . . . . .   | 12        |
| 7.4 Programming Heterogeneous Architectures Using Hierarchical Tasks . . . . .  | 12        |
| 7.5 Optimal Time and Energy-Aware Client Selection Algorithms for Federated Learning on Heterogeneous Resources . . . . .                           | 13        |
| 7.6 Task scheduling to improve throughput and reduce latency for deep neural network inference  | 13        |
| 7.7 Predicting errors in parallel applications with ML . . . . .  | 13        |
| 7.8 Static-Dynamic analysis for Performance and Accuracy of Data Race Detection in MPI One-Sided Programs . . . . .                                 | 14        |
| 7.9 Leveraging private container networks for increased user isolation and flexibility on HPC clusters . . . . .                                    | 14        |
| 7.10 Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver - Best Paper Award . . . . .          | 14        |
| 7.11 MPI-BugBench: A Framework for Assessing MPI Correctness Tools . . . . .  | 15        |
| 7.12 Designing Quality MPI Correctness Benchmarks: Insights and Metrics . . . . .   | 15        |
| 7.13 Highlighting EasyPAP Improvements . . . . .  | 16        |
| 7.14 Automatic Dimensioning and Load Balancing on Heterogeneous Architectures . . . . .   | 16        |
| 7.15 Improving energy efficiency of HPC applications using unbalanced GPU power capping . .   | 16        |
| 7.16 Approximation Algorithms for Scheduling with/without Deadline Constraints where Rejection Costs are Proportional to Processing Times . . . . . | 17        |

|           |   |           |
|-----------|---|-----------|
| <b>8</b>  | <b>Bilateral contracts and grants with industry</b>                     | <b>17</b> |
| 8.1       | Bilateral contracts with industry                                       | 17        |
| 8.1.1     | Airbus  | 17        |
| 8.1.2     | ATOS / EVIDEN   | 18        |
| 8.1.3     | IFPEN   | 18        |
| 8.1.4     | Qarnot  | 18        |
| <b>9</b>  | <b>Partnerships and cooperations</b>                                    | <b>19</b> |
| 9.1       | International initiatives   | 19        |
| 9.1.1     | Inria associate team not involved in an IIL or an international program | 19        |
| 9.1.2     | Visits to international teams   | 19        |
| 9.2       | European initiatives  | 20        |
| 9.2.1     | EuroHPC   | 20        |
| 9.3       | National initiatives  | 20        |
| 9.3.1     | PEPR  | 20        |
| 9.3.2     | AID   | 21        |
| 9.3.3     | Défis Inria   | 21        |
| 9.3.4     | Inria exploratory actions   | 22        |
| <b>10</b> | <b>Dissemination</b>  | <b>22</b> |
| 10.1      | Promoting scientific activities   | 22        |
| 10.1.1    | Scientific events: organisation   | 22        |
| 10.1.2    | Scientific events: selection  | 22        |
| 10.1.3    | Journal   | 22        |
| 10.1.4    | Invited talks   | 23        |
| 10.1.5    | Leadership within the scientific community                              | 23        |
| 10.1.6    | Scientific expertise  | 23        |
| 10.1.7    | Research administration   | 24        |
| 10.2      | Teaching - Supervision - Juries   | 24        |
| 10.2.1    | Teaching  | 24        |
| 10.2.2    | Supervision   | 26        |
| 10.2.3    | Juries  | 27        |
| 10.3      | Popularization  | 27        |
| 10.3.1    | Specific official responsibilities in science outreach structures       | 27        |
| 10.3.2    | Productions (articles, videos, podcasts, serious games, ...)            | 28        |
| 10.3.3    | Participation in Live events  | 28        |
| <b>11</b> | <b>Scientific production</b>  | <b>29</b> |
| 11.1      | Major publications  | 29        |
| 11.2      | Publications of the year  | 30        |
| 11.3      | Cited publications  | 32        |

## Project-Team STORM

*Creation of the Project-Team: 2017 July 01*

### Keywords

#### Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)
- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A1.1.9. – Fault tolerant systems
- A1.1.13. – Virtualization
- A1.6. – Green Computing
- A2.1.6. – Concurrent programming
- A2.1.7. – Distributed programming
- A2.4.1. – Analysis
- A2.4.2. – Model-checking
- A4.3. – Cryptography
- A6.2.7. – High performance computing
- A6.2.8. – Computational geometry and meshes
- A9.6. – Decision support

#### Other research topics and application domains

- B2.2.1. – Cardiovascular and respiratory diseases
- B3.2. – Climate and meteorology
- B4.2. – Nuclear Energy Production
- B5.2.3. – Aviation
- B5.2.4. – Aerospace
- B6.2.2. – Radio technology
- B6.2.3. – Satellite technology
- B9.1. – Education

# 1 Team members, visitors, external collaborators

## Research Scientists

- Olivier Aumage [Team leader, INRIA, Researcher]
- Laercio Lima Pilla [CNRS, Researcher, until Nov 2024]
- Mihail Popov [INRIA, ISFP, until Sep 2024]
- Emmanuelle Saillard [INRIA, Researcher]

## Faculty Members

- Marie-Christine Counilh [UNIV BORDEAUX, Associate Professor]
- Amina Guermouche [BORDEAUX INP, Associate Professor]
- Raymond Namyst [UNIV BORDEAUX, Professor]
- Samuel Thibault [UNIV BORDEAUX, Professor]
- Pierre-André Wacrenier [UNIV BORDEAUX, Associate Professor]

## PhD Students

- Vincent Alba [UNIV BORDEAUX]
- Asia Auville [INRIA, from Oct 2024]
- Albert D Aviau De Piolant [INRIA]
- Lise Jolicoeur [CEA, CIFRE]
- Alice Lasserre [INRIA]
- Alan Lira Nunes [INRIA, from Jun 2024, Joint PhD Thesis with UF Fluminense, Brazil]
- Alan Lira Nunes [UF Fluminense, Brazil, until May 2024, Joint PhD with UF Fluminense, Brazil]
- Thomas Morin [UNIV BORDEAUX]
- Diane Orhan [UNIV BORDEAUX]
- Lana Scravaglieri [IFPEN, CIFRE]
- Radjasouria Vinayagame [ATOS, CIFRE]

## Technical Staff

- Francois Cheminade [INRIA, Engineer, from Sep 2024, AID AFF3CT]
- Guillaume Doyen [INRIA, Engineer, from Dec 2024, EUROHPC MICROCARD-2]
- Nicolas Ducarton [INRIA, Engineer, from Oct 2024, PEPR NumPEX]
- Nicolas Ducarton [UNIV BORDEAUX, Engineer, until Sep 2024, EUROHPC MICROCARD]
- Nathalie Furmento [CNRS, Engineer]
- Andrea Lesavourey [INRIA, Engineer, until Sep 2024, AID AFF3CT]
- Romain Lion [INRIA, Engineer, DGAC MAMBO]
- Joachim Rosseel [INRIA, Engineer, from Apr 2024, AID AFF3CT]
- Victor-Benjamin Villain [INRIA, Engineer, from Dec 2024, AID AFF3CT]

## Interns and Apprentices

- Asia Auville [INRIA, Intern, from Mar 2024 until Sep 2024]
- Abdelbarie El Metni [INRIA, Intern, from Mar 2024 until Aug 2024]
- Jules Evans [INRIA, Intern, from Mar 2024 until May 2024]
- Theo Grandsart [INRIA, Intern, from Jun 2024 until Aug 2024]
- Patrick Gutsche [ENS DE LYON, Intern, from Jun 2024 until Jul 2024]
- Evan Potin [INRIA, Intern, from Mar 2024 until Aug 2024]

## Administrative Assistant

- Ellie Correa Da Costa De Castro Pinto [INRIA]

## Visiting Scientist

- Mariza Ferro [UFF NITEROI BRAZIL, until Mar 2024]

## External Collaborator

- Jean-Marie Couteyen [AIRBUS]

## 2 Overall objectives

Runtime systems successfully support the complexity and heterogeneity of modern architectures thanks to their dynamic task management. Compiler optimizations and analyses are aggressive in iterative compilation frameworks, suitable for library generations or domain specific languages (DSL), in particular for linear algebra methods. To alleviate the difficulties for programming heterogeneous and parallel machines, we believe it is necessary to provide inputs with richer semantics to runtime and compiler alike, and in particular by combining both approaches.

This general objective is declined into three sub-objectives, the first concerning the expression of parallelism itself, the second the optimization and adaptation of this parallelism by compilers and runtimes and the third concerning the necessary user feedback, either as debugging or simulation results, to better understand the first two steps.

1. Expressing parallelism: As shown in the following figure, we propose to work on parallelism expression through Application Programming Interfaces, C++ enhanced with libraries or pragmas, Domain Specific Languages, PGAS languages able to capture the essence of the algorithms used through usual parallel languages such as SyCL, OpenMP and through high performance libraries. The language richer semantics will be driven by applications, with the idea to capture at the algorithmic level the parallelism of the problem and perform dynamic data layout adaptation, parallel and algorithmic optimizations. The principle here is to capture a higher level of semantics, enabling users to express not only parallelism but also different algorithms.
2. Optimizing and adapting parallelism: The goal is to address the evolving hardware, by providing mechanisms to efficiently run the same code on different architectures. This implies to adapt parallelism to the architecture by either changing the granularity of the work or by adjusting the execution parameters. We rely on the use of existing parallel libraries and their composition, and more generally on the separation of concerns between the description of tasks, that represent semantic units of work, and the tasks to be executed by the different processing units. Splitting or coarsening moldable tasks, generating code for these tasks, and exploring runtime parameters (e.g., frequency, vectorization, prefetching, scheduling) is part of this work.

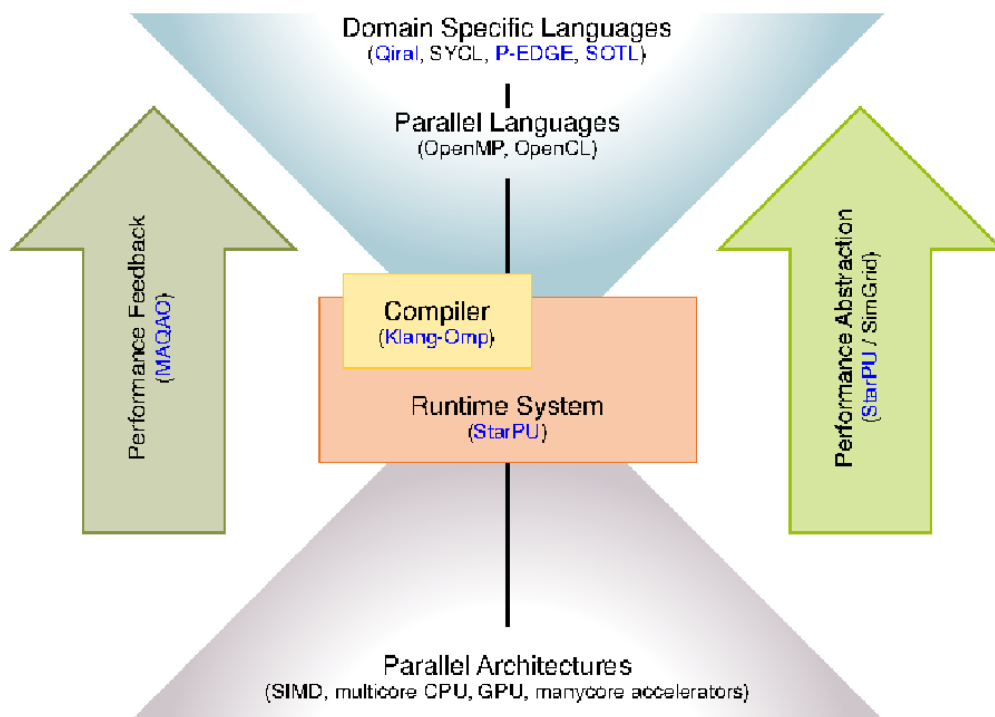


Figure 1: STORM Big Picture

3. Finally, the abstraction we advocate for requires to propose a feed back loop. This feed back has two objectives: to make users better understand their application and how to change the expression of parallelism if necessary, but also to propose an abstracted model for the machine. This allows to develop and formalize the compilation, scheduling techniques on a model, not too far from the real machine. Here, simulation techniques are a way to abstract the complexity of the architecture while preserving essential metrics.

## 3 Research program

### 3.1 Parallel Computing and Architectures

Following the current trends of the evolution of HPC systems architectures, it is expected that future Exascale systems (i.e. Sustaining  $10^{18}$  flops) will have millions of cores. Although the exact architectural details and trade-offs of such systems are still unclear, it is anticipated that an overall concurrency level of  $O(10^9)$  threads/tasks will probably be required to feed all computing units while hiding memory latencies. It will obviously be a challenge for many applications to scale to that level, making the underlying system sound like “embarrassingly parallel hardware.”

From the programming point of view, it becomes a matter of being able to expose extreme parallelism within applications to feed the underlying computing units. However, this increase in the number of cores also comes with architectural constraints that actual hardware evolution prefigures: computing units will feature extra-wide SIMD and SIMT units that will require aggressive code vectorization or “SIMDization”, systems will become hybrid by mixing traditional CPUs and accelerators units, possibly on the same chip as the AMD APU solution, the amount of memory per computing unit is constantly decreasing, new levels of memory will appear, with explicit or implicit consistency management, etc. As a result, upcoming extreme-scale system will not only require unprecedented amount of parallelism to be efficiently exploited, but they will also require that applications generate adaptive parallelism capable to map tasks over heterogeneous computing units.

The current situation is already alarming, since European HPC end-users are forced to invest in a difficult and time-consuming process of tuning and optimizing their applications to reach most of current supercomputers’ performance. It will go even worse with the emergence of new parallel architectures (tightly integrated accelerators and cores, high vectorization capabilities, etc.) featuring unprecedented degree of parallelism that only too few experts will be able to exploit efficiently. As highlighted by the ETP4HPC initiative, existing programming models and tools won’t be able to cope with such a level of heterogeneity, complexity and number of computing units, which may prevent many new application opportunities and new science advances to emerge.

The same conclusion arises from a non-HPC perspective, for single node embedded parallel architectures, combining heterogeneous multicores, such as the ARM big.LITTLE processor and accelerators such as GPUs or DSPs. The need and difficulty to write programs able to run on various parallel heterogeneous architectures has led to initiatives such as HSA, focusing on making it easier to program heterogeneous computing devices. The growing complexity of hardware is a limiting factor to the emergence of new usages relying on new technology.

### 3.2 Scientific and Societal Stakes

In the HPC context, simulation is already considered as a third pillar of science with experiments and theory. Additional computing power means more scientific results, and the possibility to open new fields of simulation requiring more performance, such as multi-scale, multi-physics simulations. Many scientific domains able to take advantage of Exascale computers, these “Grand Challenges” cover large panels of science, from seismic, climate, molecular dynamics, theoretical and astrophysics physics... Besides, more widespread compute intensive applications are also able to take advantage of the performance increase at the node level. For embedded systems, there is still an on-going trend where dedicated hardware is progressively replaced by off-the-shelf components, adding more adaptability and lowering the cost of devices. For instance, Error Correcting Codes in cell phones are still hardware chips, but new software and adaptive solutions relying on low power multicores are also explored for antenna. New usages are also appearing, relying on the fact that large computing capacities are becoming more affordable



and widespread. This is the case for instance with Deep Neural Networks where the training phase can be done on supercomputers and then used in embedded mobile systems. Even though the computing capacities required for such applications are in general a different scale from HPC infrastructures, there is still a need in the future for high performance computing applications.

However, the outcome of new scientific results and the development of new usages for these systems will be hindered by the complexity and high level of expertise required to tap the performance offered by future parallel heterogeneous architectures. Maintenance and evolution of parallel codes are also limited in the case of hand-tuned optimization for a particular machine, and this advocates for a higher and more automatic approach.

### 3.3 Towards More Abstraction

As emphasized by initiatives such as the European Exascale Software Initiative (EESI), the European Technology Platform for High Performance Computing (ETP4HPC), or the International Exascale Software Initiative (IESP), the HPC community needs new programming APIs and languages for expressing heterogeneous massive parallelism in a way that provides an abstraction of the system architecture and promotes high performance and efficiency. The same conclusion holds for mobile, embedded applications that require performance on heterogeneous systems.

This crucial challenge given by the evolution of parallel architectures therefore comes from this need to make high performance accessible to the largest number of developers, abstracting away architectural details providing some kind of performance portability, and provided a high level feed-back allowing the user to correct and tune the code. Disruptive uses of the new technology and groundbreaking new scientific results will not come from code optimization or task scheduling, but they require the design of new algorithms that require the technology to be tamed in order to reach unprecedented levels of performance.

Runtime systems and numerical libraries are part of the answer, since they may be seen as building blocks optimized by experts and used as-is by application developers. The first purpose of runtime systems is indeed to provide *abstraction*. Runtime systems offer a uniform programming interface for a specific subset of hardware or low-level software entities (e.g., POSIX-thread implementations). They are designed as thin user-level software layers that complement the basic, general purpose functions provided by the operating system calls. Applications then target these uniform programming interfaces in a portable manner. Low-level, hardware dependent details are hidden inside runtime systems. The adaptation of runtime systems is commonly handled through drivers. The abstraction provided by runtime systems thus enables portability. Abstraction alone is however not enough to provide portability of performance, as it does nothing to leverage low-level-specific features to get increased performance and does nothing to help the user tune his code. Consequently, the second role of runtime systems is to *optimize* abstract application requests by dynamically mapping them onto low-level requests and resources as efficiently as possible. This mapping process makes use of scheduling algorithms and heuristics to decide the best actions to take for a given metric and the application state at a given point in its execution time. This allows applications to readily benefit from available underlying low-level capabilities to their full extent without breaking their portability. Thus, optimization together with abstraction allows runtime systems to offer portability of performance. Numerical libraries provide sets of highly optimized kernels for a given field (dense or sparse linear algebra, tensor products, etc.) either in an autonomous fashion or using an underlying runtime system.

Application domains cannot resort to libraries for all codes however, computation patterns such as stencils are a representative example of such difficulty. The compiler technology plays here a central role, in managing high level semantics, either through templates, domain specific languages or annotations. Compiler optimizations, and the same applies for runtime optimizations, are limited by the level of semantics they manage and the optimization space they explore. Providing part of the algorithmic knowledge of an application, and finding ways to explore a larger space of optimization would lead to more opportunities to adapt parallelism, memory structures, and is a way to leverage the evolving hardware. Compilers and runtime play a crucial role in the future of high performance applications, by defining the input language for users, and optimizing/transforming it into high performance code. Adapting the parallelism and its orchestration according to the inputs, to energy, to faults, managing heterogeneous memory, better define and select appropriate dynamic scheduling methods, are among

the current works of the STORM team.

## 4 Application domains

### 4.1 Application domains benefiting from HPC

The application domains of this research are the following:

- Health and heart disease analysis (see MICROCARD and MICROCARD-2 projects [9.2.1](#))
- Software infrastructures for Telecommunications (see AFF3CT [9.3.2](#))
- Aeronautics (collaboration with Airbus, J.-M. Couteyen, MAMBO project [8.1.1](#))
- CO2 storage (collaboration with IFPEN, see [8.1.3](#))

### 4.2 Application in High performance computing/Big Data

Most of the research of the team has application in the domain of software infrastructure for HPC and compute intensive applications.

## 5 Highlights of the year

### 5.1 Awards

The Inria - Académie des Sciences - Dassault Systèmes innovation price was awarded to Samuel THIBAUT (STORM) and Brice GOGLIN (TADaaM) for the hwloc software.

## 6 New software, platforms, open data

### 6.1 New software

#### 6.1.1 AFF3CT

**Name:** A Fast Forward Error Correction Toolbox

**Keywords:** High-Performance Computing, Signal processing, Error Correction Code

**Functional Description:** AFF3CT proposes high performance Error Correction algorithms for Polar, Turbo, LDPC, RSC (Recursive Systematic Convolutional), Repetition and RA (Repeat and Accumulate) codes. These signal processing codes can be parameterized in order to optimize some given metrics, such as Bit Error Rate, Bandwidth, Latency, ...using simulation. For the designers of such signal processing chain, AFF3CT proposes also high performance building blocks so to develop new algorithms. AFF3CT compiles with many compilers and runs on Windows, Mac OS X, Linux environments and has been optimized for x86 (SSE, AVX instruction sets) and ARM architectures (NEON instruction set).

**URL:** <https://aff3ct.github.io/>

**Publications:** [hal-02358306](#), [hal-01965629](#), [hal-01977885](#), [hal-01203105](#), [hal-01363980](#), [hal-01363975](#), [hal-01987848](#), [hal-01965633](#)

**Contact:** Olivier Aumage

**Partners:** IMS, LIP6

### 6.1.2 PARCOACH

**Name:** PARallel Control flow Anomaly CHecker

**Keywords:** Verification, HPC

**Scientific Description:** PARCOACH verifies programs in two steps. First, it statically verifies applications with a data- and control-flow analysis and outlines execution paths leading to potential deadlocks. The code is then instrumented, displaying an error and synchronously interrupting all processes if the actual scheduling leads to a deadlock situation.

**Functional Description:** Supercomputing plays an important role in several innovative fields, speeding up prototyping or validating scientific theories. However, supercomputers are evolving rapidly with now millions of processing units, posing the questions of their programmability. Despite the emergence of more widespread and functional parallel programming models, developing correct and effective parallel applications still remains a complex task. As current scientific applications mainly rely on the Message Passing Interface (MPI) parallel programming model, new hardware designed for Exascale with higher node-level parallelism clearly advocate for an MPI+X solutions with X a thread-based model such as OpenMP. But integrating two different programming models inside the same application can be error-prone leading to complex bugs - mostly detected unfortunately at runtime. PARallel COnTrol flow Anomaly CHecker aims at helping developers in their debugging phase.

**URL:** <https://parcoach.github.io/index.html>

**Publications:** [hal-03882459](#), [hal-03374614](#), [hal-00920901](#), [hal-01078762](#), [hal-01078759](#), [hal-01252321](#), [hal-01253204](#), [hal-01199718](#), [hal-01420655](#), [hal-01937316](#), [hal-02390025](#)

**Contact:** Emmanuelle Saillard

**Participants:** Emmanuelle Saillard, Denis Barthou, Philippe Virouleau, Tassadit Ait Kaci

### 6.1.3 MIPP

**Name:** MyIntrinsics++

**Keywords:** SIMD, Vectorization, Instruction-level parallelism, C++, Portability, HPC, Embedded

**Scientific Description:** MIPP is a portable and Open-source wrapper (MIT license) for vector intrinsic functions (SIMD) written in C++11. It works for SSE, AVX, AVX-512 and ARM NEON (32-bit and 64-bit) instructions.

**Functional Description:** MIPP enables writing portable and yet highly optimized kernels to exploit the vector processing capabilities of modern processors. It encapsulates architecture specific SIMD intrinsics routine into a header-only abstract C++ API.

**Release Contributions:** ARM SVE support

**URL:** <https://github.com/aff3ct/MIPP>

**Publications:** [hal-01888010](#), [tel-03118420](#)

**Contact:** Olivier Aumage

**Participants:** Adrien Cassagne, Denis Barthou, Edgar Baucher, Olivier Aumage

**Partner:** LIP6

#### 6.1.4 CERE

**Name:** Codelet Extractor and REplayer

**Keywords:** Checkpointing, Profiling

**Functional Description:** CERE finds and extracts the hotspots of an application as isolated fragments of code, called codelets. Codelets can be modified, compiled, run, and measured independently from the original application. Code isolation reduces benchmarking cost and allows piecewise optimization of an application.

**Contact:** Mihail Popov

**Partners:** Université de Versailles St-Quentin-en-Yvelines, Exascale Computing Research

#### 6.1.5 DUF

**Name:** Dynamic Uncore Frequency Scaling

**Keywords:** Power consumption, Energy efficiency, Power capping, Frequency Domain

**Functional Description:** Just as core frequency, uncore frequency usage depends on the target application. As a matter of fact, the uncore frequency is the frequency of the L3 cache and the memory controllers. However, it is not well managed by default. DUF manages to reach power and energy saving by dynamically adapting the uncore frequency to the application needs while respecting a user-defined tolerated slowdown. Based on the same idea, it is also able to dynamically adapt the power cap.

**Contact:** Amina Guermouche

#### 6.1.6 MBI

**Name:** MPI Bugs Initiative

**Keywords:** MPI, Verification, Benchmarking, Tools

**Functional Description:** Ensuring the correctness of MPI programs becomes as challenging and important as achieving the best performance. Many tools have been proposed in the literature to detect incorrect usages of MPI in a given program. However, the limited set of code samples each tool provides and the lack of metadata stating the intent of each test make it difficult to assess the strengths and limitations of these tools. We have developed the MPI BUGS INITIATIVE, a complete collection of MPI codes to assess the status of MPI verification tools. We introduce a classification of MPI errors and provide correct and incorrect codes covering many MPI features and our categorization of errors.

**Publication:** [hal-03474762](https://hal.archives-ouvertes.fr/hal-03474762)

**Contact:** Emmanuelle Saillard

**Participants:** Emmanuelle Saillard, Martin Quinson

#### 6.1.7 EasyPAP

**Name:** easyPAP

**Functional Description:** EasyPAP provides students with a simple and attractive programming environment to facilitate their discovery of the main concepts of parallel programming.

EasyPAP is a framework providing interactive visualization, real-time monitoring facilities, and off-line trace exploration utilities. Students focus on parallelizing 2D computation kernels using Pthreads, OpenMP, OpenCL, MPI, SIMD intrinsics, or a mix of them.

EasyPAP was designed to make it easy to implement multiple variants of a given kernel, and to experiment with and understand the influence of many parameters related to the scheduling policy or the data decomposition.

**URL:** <https://gforgeron.gitlab.io/easypap/>

**Contact:** Raymond Namyst

### 6.1.8 StarPU

**Name:** The StarPU Runtime System

**Keywords:** Runtime system, High performance computing

**Scientific Description:** Traditional processors have reached architectural limits which heterogeneous multicore designs and hardware specialization (eg. coprocessors, accelerators, ...) intend to address. However, exploiting such machines introduces numerous challenging issues at all levels, ranging from programming models and compilers to the design of scalable hardware solutions. The design of efficient runtime systems for these architectures is a critical issue. StarPU typically makes it much easier for high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors: rather than handling low-level issues, programmers may concentrate on algorithmic concerns. Portability is obtained by the means of a unified abstraction of the machine. StarPU offers a unified offloadable task abstraction named "codelet". Rather than rewriting the entire code, programmers can encapsulate existing functions within codelets. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (eg. one function for CUDA and one function for CPUs). StarPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. In order to relieve programmers from the burden of explicit data transfers, a high-level data management library enforces memory coherency over the machine: before a codelet starts (eg. on an accelerator), all its data are transparently made available on the compute resource. Given its expressive interface and portable scheduling policies, StarPU obtains portable performances by efficiently (and easily) using all computing resources at the same time. StarPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.

StarPU is a task programming library for hybrid architectures.

The application provides algorithms and constraints: - CPU/GPU implementations of tasks, - A graph of tasks, using StarPU's rich C API.

StarPU handles run-time concerns: - Task dependencies, - Optimized heterogeneous scheduling, - Optimized data transfers and replication between main memory and discrete memories, - Optimized cluster communications.

Rather than handling low-level scheduling and optimizing issues, programmers can concentrate on algorithmic concerns!

**Functional Description:** StarPU is a runtime system that offers support for heterogeneous multicore machines. While many efforts are devoted to design efficient computation kernels for those architectures (e.g. to implement BLAS kernels on GPUs), StarPU not only takes care of offloading such kernels (and implementing data coherency across the machine), but it also makes sure the kernels are executed as efficiently as possible.

**Release Contributions:** StarPU is a runtime system that offers support for heterogeneous multicore machines. While many efforts are devoted to design efficient computation kernels for those architectures (e.g. to implement BLAS kernels on GPUs), StarPU not only takes care of offloading such kernels (and implementing data coherency across the machine), but it also makes sure the kernels are executed as efficiently as possible.

**URL:** <https://starpu.gitlabpages.inria.fr/>

**Publications:** tel-04213186, inria-00326917, inria-00378705, inria-00384363, inria-00411581, inria-00421333, inria-00467677, inria-00523937, inria-00547614, inria-00547616, inria-00547847, inria-00550877, inria-00590670, inria-00606195, inria-00606200, inria-00619654, hal-00643257, hal-00648480, hal-00654193, hal-00661320, hal-00697020, hal-00714858, hal-00725477, hal-00772742, hal-00773114, hal-00773571, hal-00773610, hal-00776610, tel-00777154, hal-00803304, hal-00807033, hal-00824514, hal-00851122, hal-00853423, hal-00858350, hal-00911856, hal-00920915, hal-00925017, hal-00926144, tel-00948309, hal-00966862, hal-00978364, hal-00978602, hal-00987094, hal-00992208, hal-01005765, hal-01011633, hal-01081974, hal-01101045, hal-01101054, hal-01120507, hal-01147997, tel-01162975, hal-01180272, hal-01181135, hal-01182746, hal-01223573, tel-01230876, hal-01283949, hal-01284004, hal-01284136, hal-01284235, hal-01316982, hal-01332774, hal-01353962, hal-01355385, hal-01361992, hal-01372022, hal-01386174, hal-01387482, hal-01409965, hal-01410103, hal-01473475, hal-01474556, tel-01483666, hal-01502749, hal-01507613, hal-01517153, tel-01538516, hal-01616632, hal-01618526, hal-01718280, tel-01816341, hal-01842038, tel-01959127, hal-02120736, hal-02275363, hal-02296118, hal-02403109, hal-02421327, hal-02872765, hal-02914793, hal-02933803, hal-02943753, hal-02970529, hal-02985721, hal-03144290, hal-03273509, hal-03290998, hal-03298021, hal-03318644, hal-03348787, hal-03552243, hal-03609275, hal-03623220, hal-03773486, hal-03773985, hal-03789625, hal-03936659, tel-03989856, hal-04005071, hal-04088833, hal-04115280, hal-04146714, hal-04236246, tel-04260094, tel-04316145, hal-04548787, hal-04646530, hal-04668550, hal-04690154

**Contact:** Nathalie Furmento

**Participants:** Cedric Augonnet, Olivier Aumage, Nathalie Furmento, Samuel Thibault, Simon Archipoff, Bérenger Bramas, Alfredo Buttari, Jérôme Clet-Ortega, Terry Cojean, Nicolas Collin, Camille Coti, Ludovic Courtes, Alexandre Denis, Lionel Eyraud Dubois, Maxime Gonthier, Amina Guermouche, Kun He, Sylvain Henry, Andra Hugo, Antoine Jego, Loïc Jouans, Mehdi Juhoor, Yanis Khorsi, Xavier Lacoste, Romain Lion, Benoit Lize, Gwenole Lucas, Mariem Makni, Thomas Morin, Raymond Namyst, Cyril Roelandt, Corentin Salingue, Lucas Schnorr, Marc Sergeant, Luka Stanisic, Ludovic Stordeur, Philippe Swartvagher, François Tessier, Leo Villeveygoux, Philippe Virouleau, Pierre Wacrenier

## 7 New results

### 7.1 Scheduling for Pipelined and Replicated Task Chains and Graphs for Software-Defined Radio

**Participants:** Olivier Aumage, Denis Barthou, Laércio Lima Pilla, Diane Orhan.

Software-Defined Radio (SDR) represents a move from dedicated hardware to software implementations of digital communication standards. This approach offers flexibility, shorter time to market, maintainability, and lower costs, but it requires an optimized distribution of SDR tasks in order to meet performance requirements. In this context, we study the problem of scheduling SDR linear stateless and stateful tasks. Following OTAC, an algorithm that we previously proposed that provides optimal throughput while also minimizing the number of allocated hardware resources for the pipelined workflow scheduling problem (based on pipelined and replicated parallelism on homogeneous resources), we have studied how to schedule multiple task chains over a shared pool of homogeneous resources, and how to apply these ideas to task graphs composed of multiple internal task chains. Our approach combines the solutions for multiple-choice knapsack problems, graph algorithms, and graph partitioners to achieve high throughput while avoiding the use of unnecessary resources.

### 7.2 Optimization Space Exploration

**Participants:** Olivier Aumage, Mihail Popov, Lana Scravaglieri.

HPC systems expose configuration options that help users optimize their applications' execution. Questions related to the best thread and data mapping, number of threads, or cache prefetching have been posed for different applications, yet they have been mostly limited to a single optimization objective (e.g., performance) and a fixed application problem size. Unfortunately, optimization strategies that work well in one scenario may generalize poorly when applied in new contexts.

In previous work [37], we investigated the impact of configuration options and different problem sizes over both performance and energy: NUMA-related options and cache prefetchers provide significantly more gains for energy (5.9x) than performance (1.85x) over a standard baseline configuration.

In the context of Lana Scravaglieri Ph.D. thesis and in collaboration with IFP Energies nouvelles (IFPEN), we further carry this research by focusing on the exploration of SIMD transformations over carbon storage applications. To do so, we are designing a more general exploration infrastructure, CORHPEX, that can easily incorporate more diverse optimization knobs and applications. This work is under review.

### 7.3 Task scheduling with memory constraints

**Participants:** Maxime Gonthier, Samuel Thibault.

When dealing with larger and larger datasets processed by task-based applications, the amount of system memory may become too small to fit the working set, depending on the task scheduling order. We had previously introduced a dynamic strategy with a locality-aware principle, and we had observed that the obtained behavior is actually very close to the proven-optimal behavior. We have submitted the results to JPDC, a RR of the draft is available [27].

We have also tackled the same type of problem, but with a different situation, in collaboration with the University of Uppsala. On their production cluster, various jobs use large files as input for their computations. The current job scheduler does not take into account the fact that an input data can be re-used between job executions, when they happen to need the same file, thus saving the time to transfer the file. We have devised a heuristic that orders jobs according to input file affinity, thus improving the rate of input data re-use, and leading to better overall usage of the platform over all jobs. This was published at the APDCM workshop [14]

### 7.4 Programming Heterogeneous Architectures Using Hierarchical Tasks

**Participants:** Mathieu Faverge, Nathalie Furmento, Abdou Guermouche, Thomas Morin, Raymond Namyst, Samuel Thibault, Pierre-André Wacrenier.

The efficiency of heterogeneous parallel systems can be significantly improved by using task-based programming models. Among these models, the Sequential Task Flow (STF) model is widely embraced since it efficiently handles task graphs while offering ample optimization perspectives. However, STF is limited to task graphs with task sizes that are fixed at submission, posing a challenge in determining the optimal task granularity. For instance, in heterogeneous systems, the optimal task size varies across different processing units. StarPU's recursive tasks allow graphs with several task granularities by turning some tasks into subgraphs dynamically at runtime. The decision to transform these tasks into subgraphs is decided by a StarPU component called the Splitter [13, 24]. We propose a new policy for the Splitter, which is designed for heterogeneous platforms, that relies on linear programming aimed at minimising execution time and maximising resource utilization. This results in a dynamic well-balanced set comprising both small tasks to fill multiple CPU cores, and large tasks for efficient execution on accelerators



like GPU devices. Experimental evaluations show that just-in-time adaptations of the task graph lead to improved performance across various dense linear algebra algorithms. This is pending submission to the JPDC journal.

## 7.5 Optimal Time and Energy-Aware Client Selection Algorithms for Federated Learning on Heterogeneous Resources

**Participants:** Laércio Lima Pilla, Alan Lira Nunez.

In [20], we study the effects of scheduling decisions over the performance and energy consumption of Federated Learning (FL) models. FL systems allow training machine learning models distributed across multiple clients, each one using private local data. Iteratively, the clients send their training contributions to a server, which performs a merge to produce an enhanced global model. Due to resource and data heterogeneity, client selection is crucial to optimize the system efficiency and improve the global model generalization. Selecting more clients is likely to increase the overall energy consumption, while a small number of clients may decline the performance of the trained model or require longer training time. We propose two time- and energy-aware client selection algorithms, MEC and ECMTTC, which are proven regarding their optimality and evaluated against state-of-the-art algorithms on an extensive series of experiments in both simulation and HPC platform scenarios. The results indicate the benefits of jointly optimizing the time and energy consumption metrics using our proposals.

## 7.6 Task scheduling to improve throughput and reduce latency for deep neural network inference

**Participants:** Jean-François David, Samuel Thibault.

Graphics Processing Units (GPUs) are widely used for training and inference of DNNs. However, this exclusive use can quickly lead to saturation of GPU resources while CPU resources remain underutilized. We proposed a performance evaluation of a solution that exploits processor heterogeneity by combining the computational power of GPUs and CPUs. A solution was proposed for distributing the computational load across the different processors to optimize their utilization and achieve better performance. A solution for partitioning a DNN model with different computational resources was also proposed. This solution transfers part of the load from the GPUs to the CPUs when necessary to reduce latency and increase throughput. The partitioning of DNN models is performed using METIS to balance the computational load to be distributed among the different resources while minimizing communications. The experimental results show that latency and throughput are improved for a number of DNN models [22, 11, 12]

## 7.7 Predicting errors in parallel applications with ML

**Participants:** Asia Auville, Mihail Popov, Emmanuelle Saillard.

Investigating if parallel applications are correct is a very challenging task. Yet, recent progress in ML and text embedding show promising results in characterizing source code or the compiler intermediate representation to identify optimizations. We propose to transpose such characterization methods to the context of verification. In particular, we train ML models that take as labels the code correctness along with intermediate representations embeddings as features. Results over small MPI verification benchmarks including MBI and DataRaceBench demonstrate that we can train models that detect if a



code is correct with 90% accuracy and up to 75% over new unseen errors. This work, published at IPDPS 2024 [6], is a collaboration with the Iowa State University.

In the context of Asia Auville Ph.D. thesis, we are currently investigating the prediction capabilities of ML models to detected errors beyond simple errors, by considering more complicated errors through github repositories crawling. We are also planning to use LLMs models to not only detect errors, but also to propose fixes. This work is done in collaboration with the University of Versailles and Intel.

## 7.8 Static-Dynamic analysis for Performance and Accuracy of Data Race Detection in MPI One-Sided Programs

**Participants:** Emmanuelle Saillard, Samuel Thibault, Radjasouria Vinayagame.

To take advantage of asynchronous communication mechanisms provided by the recent platforms, the Message Passing Interface (MPI) proposes operations based on one-sided communications. These operations enable a better overlap of communications with computations. However, programmers must manage data consistency and synchronization to avoid data races, which may be a daunting task. This work proposes three solutions to improve the performance and the accuracy of the data race detection in MPI one-sided programs. First, we extend the node-merging algorithm based of a Binary Search Tree (BST) presented in a previous work that keeps track of memory accesses during execution to take into account non-adjacent memory accesses. Then, we use an alias analysis to reduce the number of load/store instrumented. Finally, we extend our analyses to manage synchronization routines. Our solutions have been implemented in PARCOACH, a MPI verification tool. Experiments on real-life applications show that our contributions lead to a better accuracy, a reduction of the memory usage by a factor up to 4 of the dynamic analysis and a reduction of the overhead at runtime at larger scale [21].

## 7.9 Leveraging private container networks for increased user isolation and flexibility on HPC clusters

**Participants:** Lise Jolicoeur, Raymond Namyst.

To address the increasing complexity of modern scientific computing workflows, HPC clusters must be able to accommodate a wider range of workloads without compromising their efficiency in processing batches of highly parallel jobs. Cloud computing providers have a long history of leveraging all forms of virtualization to let their clients easily and securely deploy complex distributed applications and similar capabilities are now expected from HPC facilities. In recent years, containers have been progressively adopted by HPC practitioners to facilitate the installation of applications along with their software dependencies. However little attention has been given to the use of containers with virtualized networks to securely orchestrate distributed applications on HPC resources. We describe a way to leverage network virtualization to benefit from the flexibility and isolation typically found in a cloud environment while being as transparent and as easy to use as possible for people familiar with HPC clusters. Users are automatically isolated in their own private network which prevents unwanted network accesses and allows them to easily define network addresses so that components of a distributed workflow can reliably reach each other. We describe the implementation of this approach in the pcooc (private cloud on a compute cluster) container runtime. We evaluate both its overhead as well as its benefits for representative use-cases on a Slurm based cluster.

## 7.10 Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver - Best Paper Award

**Participants:** Alice Lasserre, Raymond Namyst.

The aerospace industry is one of the largest users of numerical simulation, which is an essential tool in the field of aerodynamic engineering, where many fluid dynamics simulations are involved. In order to obtain the most accurate solutions, some of these simulations use unstructured finite volume solvers that cope with irregular meshes by using explicit time-adaptive integration methods. Modern parallel implementations of these solvers rely on task-based runtime systems to perform fine-grained load balancing and to avoid unnecessary synchronizations. Although such implementations greatly improve performance compared to a classical fork-join MPI+OpenMP variants, it remains a challenge to keep all cores busy throughout the simulation loop. In this article, we first investigate the origins of this lack of parallelism. We emphasize that the irregular structure of the task graph plays a major role in the inefficiency of the computation distribution. Our main contribution is to improve the shape of the task graph by using a new mesh partitioning strategy. The originality of our approach is to take the temporal level of mesh cells into account during the mesh partitioning phase. We evaluate our approach by integrating our solution in an ArianeGroup production code used by Airbus. We show that our partitioning method leads to a more balanced task graph. The resulting task scheduling is up to two times faster for meshes ranging from 200,000 to 12,000,000 components.

### 7.11 MPI-BugBench: A Framework for Assessing MPI Correctness Tools

**Participants:** Emmanuelle Saillard, Radjasouria Vinayagame.

MPI's low-level interface is prone to errors, leading to bugs that can remain dormant for years. MPI correctness tools can aid in writing correct code but lack a standardized benchmark for comparison. This makes it difficult for users to choose the best tool and difficult for developers to gauge their tools' effectiveness. MPI correctness benchmarks, MPI-CorrBench, the MPI Bugs Initiative, and RMA RaceBench have emerged to address this problem. However, comparability is hindered by having separate benchmarks, and none fully reflects real-world MPI usage patterns. Hence, we have developed MPI-BugBench, a unified MPI correctness benchmark replacing previous efforts. It addresses the shortcomings of its predecessors by providing a single, standardized test harness for assessing tools and incorporates a broader range of real-world MPI usage scenarios. This work, published at EuroMPI 2024, is a collaboration with the Technical University Darmstadt and the RWTH Aachen University.

### 7.12 Designing Quality MPI Correctness Benchmarks: Insights and Metrics

**Participants:** Emmanuelle Saillard, Radjasouria Vinayagame.

Several MPI correctness benchmarks have been proposed to evaluate the quality of MPI correctness tools. The design of such a benchmark comes with different challenges, which we address in this paper. First, an imbalance in the proportion of correct and erroneous codes in the benchmarks requires careful metric interpretation (recall, accuracy, F1 score). Second, tools that detect errors but do not report additional information, like the affected source line or class of error, are less valuable. We extend the typical notion of a true positive with stricter variants that consider a tool's helpfulness. We introduce a new noise metric to consider the amount of distracting error reports. We evaluate those new metrics with MPI-BugBench, on the MPI correctness tools ITAC, MUST, and PARCOACH. This work also discusses the complexities of hand-crafted and automatically generated benchmark codes and the additional challenges of non-deterministic errors.

### 7.13 Highlighting EasyPAP Improvements

**Participants:** Alice Lasserre, Raymond Namyst, Pierre-André Wacrenier.

We have integrated 3D meshing capabilities into EasyPAP, allowing for visual monitoring of the progression of computation tasks on the mesh. This also facilitates the observation of domain decomposition performed by partitioning tools such as Scotch. This marks a major advancement, as EasyPAP was previously limited to processing 2D images. Additionally, the visualization and monitoring libraries have been extracted from EasyPAP and are now standalone, enabling collaborations with organizations such as Airbus, Eviden, and others.

### 7.14 Automatic Dimensioning and Load Balancing on Heterogeneous Architectures

**Participants:** Vincent Alba, Olivier Aumage, Denis Barthou, Marie-Christine Counilh, Amina Guermouche.

Electrophysiology simulation applications, such as the community-developed OPENCARP framework for in-silico experiments, involve applying a broad range of ionic model kernels with different computational weights and arithmetic intensity characteristics. Efficiently processing such kernels on modern heterogeneous architectures necessitates to accurately dimension the set of computing resources to use and to actively balance the load on the available computing units, to account for discrepancies in kernel duration and distinct computing unit speeds.

We thus propose the following contributions [25]: 1) the adaptation of an existing load-balancing algorithm to transparently manage the mapping of these ionic model kernels onto the heterogeneous units of a computing node; 2) a resource dimensioning heuristic that constraints the number of devices that should be used to maximize efficiency, according to the selected ionic models' computational weight; 3) the integration of these mechanisms in OPENCARP, building on prior work that took advantage of LLVM's MLIR framework to generate multiple device-specialized variants of kernels from ionic models expressed in OPENCARP's high-level DSL; 4) a thorough experimentation of the mechanisms on a comprehensive series of 30 ionic models provided by OPENCARP.

The experiments show that when using the combination of the load-balancing algorithm and the resource dimensioning heuristic to compute each ionic model, the geometric mean of speedup is  $9.97\times$  with respect to the original multi-threaded code on an architecture with two A100 GPUs and  $2\times$  32-cores AMD Zen3 CPUs.

### 7.15 Improving energy efficiency of HPC applications using unbalanced GPU power capping

**Participants:** Albert D'Aviau De Piolant, Hayfa Tayeb, Berenger Bramas, Mathieu Faverge, Abdou Guermouche, Amina Guermouche.

Energy efficiency represents a significant challenge in the domain of high-performance computing (HPC). One potential key parameter to improve energy efficiency is the use of power capping, a technique for controlling the power limits of a device, such as a CPU or GPU. In this paper, we propose to examine the impact of GPU power capping in the context of HPC applications using heterogeneous computing systems. As the environmental cost of electrical consumption increases, it is imperative that we make greater use of the energy efficiency provided. Our goal is to optimize energy efficiency using static GPU power capping. To this end, we first conduct an extensive study of the impact of GPU power capping on a compute intensive kernel, namely matrix multiplication kernel (GEMM), on different Nvidia GPU architectures. Interestingly, such compute-intensive kernels are up to 30% more energy efficient when the GPU is set to 55-70% of its Thermal Design Power (TDP). Using the best power capping configuration

provided by this study, we investigate how setting different power caps for GPU devices of a heterogeneous computing node can improve the energy efficiency of the running application. We consider dense linear algebra task-based operations, namely matrix multiplication and Cholesky Factorization. We show how the underlying runtime system scheduler can then automatically adapt its decisions to take advantage of the heterogeneous performance capability of each GPU. The obtained results show that, for a given platform equipped with 4 GPU devices, applying a power cap on all GPUs improves the energy efficiency for matrix multiplication up to 24.3% (resp. 33.78%) for double (resp. simple) precision [29].

## 7.16 Approximation Algorithms for Scheduling with/without Deadline Constraints where Rejection Costs are Proportional to Processing Times

**Participants:** Laércio Lima Pilla.

We address two offline job scheduling problems, where jobs can either be processed on a limited supply of energy-efficient machines on the edge, or offloaded to an unlimited supply of energy-inefficient machines on the cloud (called rejected in our context). The goal is to minimize the total energy consumed in processing all tasks. We consider a first scheduling problem with no due date (or deadline) constraints, and we formulate it as a scheduling problem with rejection, where the cost of rejecting a job is directly proportional to its processing time. In [10] (code in [36]), we introduce a novel  $5/4(1 + \epsilon)$  approximation algorithm BEKP by associating it with a Multiple Subset Sum problem for this version. Our algorithm is an improvement over the existing literature, which provides a  $(3/2 - 1/2m)$  approximation for scenarios with arbitrary rejection costs. In [26], we also cover a second scheduling problem, where jobs have due date (or deadline) constraints, and the goal is to minimize the weighted number of late jobs. In this context, if a job is late, it is offloaded (rejected) to an energy-inefficient machine on the cloud, which incurs a cost directly proportional to its processing time of the job. We position this problem in the literature, and introduce a novel  $(1 - (m - 1)^m / m^m)$ -approximation algorithm MDP for this version, where we got our inspiration from an algorithm for the interval selection problem with a  $(1 - m^m / (m + 1)^m)$  approximation ratio for arbitrary rejection costs. We evaluate and discuss the effectiveness of our approaches through a series of experiments, comparing them to existing algorithms.

## 8 Bilateral contracts and grants with industry

### 8.1 Bilateral contracts with industry

#### 8.1.1 Airbus

**Participants:** Jean-Marie Couteyen, Nathalie Furmento, Alice Lasserre, Romain Lion, Raymond Namyst, Pierre-André Wacrenier.

MAMBO is a 4 years collaboration project funded by Civil Aviation Direction (DGAC) gathering more than twenty industrial and academic partners to develop advanced methods for modelling Aircrafts' Engines acoustic Noise. Inria and Airbus are actively contributing to the subtask devoted to high performance simulation of acoustic waves interferences. Our work is focusing on extensions to the FLUSEPA CFD simulator to enable:

- efficient parallel intersections of multiple meshes, using task-based parallelism ;
- optimized mesh partitioning techniques to maintain load balance when using local time stepping computing schemes ;
- efficient task-based implementation to optimize granularity of tasks and communications.

### 8.1.2 ATOS / EVIDEN

**Participants:** Mihail Popov, Emmanuelle Saillard, Samuel Thibault, Radjasouria Vinayagame, Philippe Virouleau.

#### **Contract with Atos/Eviden for the PhD CIFRE of Radjasouria VINAYAGAME (2022-2025)**

Exascale machines are more and more powerful and have more nodes and cores. This trend makes the task of programming these machines and using them efficiently much more complicated. To tackle this issue, programming models are evolving from models that make an abstraction of the machine into PGAS models. Unlike MPI two-sided communications, where the sender and the receiver explicitly call the send and receive functions, one-sided communications decouple data movement from synchronization. While MPI-RMA allows efficient data movement between processes with less synchronizations, its programming is error-prone as it is the user responsibility to ensure memory consistency. It thus poses programming challenges to use as few synchronizations as possible, while preventing data race and unsafe accesses without tampering with the performance. As part of Celia Ait Kaci Tassadit PhD, we have developed a tool called RMA-Analyzer that detects memory consistency errors (also known as data races) during MPI-RMA program executions. The goal of the PhD is to push further the RMA-Analyzer with performance debugging and support to notified RMA developed by Atos. The tool will help to transform a program using point-to-point communications into a MPI-RMA program. This will lead to specific work on scalability and efficiency. The goal is to (1) evaluate the benefit of the transformation and (2) develop tools to help in this process.

#### **Contract "Plan de relance" to develop statistical learning methods for failures detection**

Exascale systems are not only more powerful but also more prone to hardware errors or malfunction. Users or sysadmins must anticipate such failures to avoid waisting compute ressources. To detect such scenarios, a "Plan de relance" is focusing on detecting hardware errors in clusters. We monitor a set of hardware counters that reflect the behavior of the system, and train auto-encodes to detect anomalies. The main challenge lies in detecting real world failures and connecting them to the monitoring counters.

### 8.1.3 IFPEN

**Participants:** Olivier Aumage, Mihail Popov, Lana Scravaglieri.

Numerical simulation is a strategic tool for IFPEN, useful for guiding research. The performance of simulators has a direct impact on the quality of simulation results. Faster modeling enable to explore a wider range of scientific hypotheses by carrying out more simulations. Similarly, more efficient models can analyze fine-grained behaviors.

Such simulations are executed on HPC systems. Such systems expose parallelism, complex out-of-order execution and cache hierarchies, and Single Instruction, Multiple Data (SIMD) units. Different architectures rely on different instructions (e.g., avx, avx-2, neon) that make portable performance a challenge.

This Ph.D. studies and designs models to optimize numerical simulations by adjusting the programs to the underline HPC systems. This involves exploring and carefully setting the different paramters (e.g., degree of parallelism, simd instructions, compiler optimizations) during an execution.

### 8.1.4 Qarnot

**Participants:** Laércio Lima Pilla.

Among the different HPC centers, data centers, and Cloud providers, Qarnot distinguishes itself by proposing a decentralized and geo-distributed solution with an aim of promoting a more virtuous approach to the emissions generated by the execution of compute-intensive tasks. With their compute

clusters, Qarnot focuses on capturing the heat released by the processors that carry out computing tasks. This heat is then used to power third-party systems (boilers, heaters, etc.). By reusing the energy from computing as heating, Qarnot provides a low-carbon infrastructure to its compute and heating users.

In the joint project PULSE (PUshing Low-carbon Services towards the Edge), Inria teams work together with Qarnot on the holistic analysis of the environmental impact of its computing infrastructure and on implementing green services on the Edge. In this context, researchers from the STORM team are working on the optimized scheduling of computing tasks based on aspects of time, cost and carbon footprint.

## 9 Partnerships and cooperations

### 9.1 International initiatives

#### 9.1.1 Inria associate team not involved in an IIL or an international program

##### MAELSTROM

**Participants:** Olivier Aumage, Abdelbarie El Metni.

**Partner:** Simula Research Laboratory, Norway

**Summary:** Scientific simulations are a prominent means for academic and industrial research and development efforts nowadays. Such simulations are extremely computing intensive due to the process involved in expressing modelled phenomena in a computer-enabled form. Exploiting supercomputer resources is essential to compute the high quality simulations in an affordable time. However, the complexity of supercomputer architectures makes it difficult to exploit them efficiently. SIMULA's HPC Dept. is the major contributor of the FEniCS computing platform. FEniCS is a popular open-source (LGPLv3) computing platform for solving partial differential equations. FEniCS enables users to quickly translate scientific models into efficient finite element code, using a formalism close to their mathematical expression.

The purpose of the Maelstrom associate team proposal started in 2022 is to build on the potential for synergy between STORM and SIMULA to extend the effectiveness of FEniCS on heterogeneous, accelerated supercomputers, while preserving its friendliness for scientific programmers, and to readily make the broad range of applications on top of FEniCS benefit from Maelstrom's results.

#### 9.1.2 Visits to international teams

##### Research stays abroad

**Participants:** Olivier Aumage, Abdelbarie El Metni.

**Visited institution:** Simula Research Laboratory

**Country:** Norway

**Dates:** June 6–10, 2024

**Context of the visit:** MAELSTROM Associate Team

**Mobility program/type of mobility:** research stay

## 9.2 European initiatives

### 9.2.1 EuroHPC

#### MICROCARD-2

**Participants:** Olivier Aumage, Guillaume Doyen.

#### MICROCARD-2 on EuroHPC-Ju

**Title:** MICROCARD-2: numerical modeling of cardiac electrophysiology at the cellular scale

**Duration:** from November 1, 2024 to April 30, 2027

**Partners:**

- Inria, France
- Karlsruher Institut Für Technologie, Germany
- Megware, Germany
- Simula Research Laboratory (Simula), Norway
- Technical University München (TUM), Germany
- Università degli Studi di Pavia, Italy
- Università di Trento (UTrento), Italy
- Université de Bordeaux, France
- Université de Strasbourg, France

**Inria contact:** Olivier AUMAGE (Storm)

**Coordinator:** Mark POTSE, Université de Bordeaux

**Summary:** The MICROCARD-2 project is coordinated by Université de Bordeaux and involves the Inria teams CARMEN, STORM, and TADAAM in Bordeaux and CAMUS in Strasbourg, among a total of ten partner institutions in France, Germany, Italy, and Norway. This Centre of Excellence for numerical modeling of cardiac electrophysiology at the cellular scale builds on the MICROCARD project (2021–2024) and has [the same website](#).

The modelling of cardiac electrophysiology at the cellular scale requires thousands of model elements per cell, of which there are billions in a human heart. Even for small tissue samples such models require at least exascale supercomputers. In addition the production of meshes of the complex tissue structure is extremely challenging, even more so at this scale. MICROCARD-2 works, in concert, on every aspect of this problem: tailored numerical schemes, linear-system solvers, and preconditioners; dedicated compilers to produce efficient system code for different CPU and GPU architectures (including the EPI and other ARM architectures); mitigation of energy usage; mesh production and partitioning; simulation workflows; and benchmarking.

The contribution of STORM concerns the energy consumption management and optimization in the openCARP simulation code, to reduce the impact of the large simulation runs required to simulate cardiac electrophysiology at a sufficient grain.

## 9.3 National initiatives

### 9.3.1 PEPR



### PEPR NumPEX / Exa-Soft focused project

**Participants:** Albert D'Aviau De Piolant, Nicolas Ducarton, Nathalie Furmento, Amina Guermouche, Thomas Morin, Raymond Namyst, Samuel Thibault, Pierre-André Wacrenier.

- 2023 - 2028 (60 months)
- Coordinator: Raymond Namyst
- Other partners: CEA, CNRS, Univ. Paris-Saclay, Telecom SudParis, Univ. of Bordeaux, Bordeaux INP, Univ. Rennes, Univ. Strasbourg, Univ. Toulouse 3, Univ. Grenoble Alpes.
- Abstract: The NumPEX project (High Performance numerics for Exascale) aims to design and develop the software components and tools that will equip future exascale machines and to prepare the major application domains to fully exploit the capabilities of these machines. It is composed of 5 scientific focused project. The Exa-Soft project aims at consolidating the exascale software ecosystem by providing a coherent, exascale-ready software stack featuring breakthrough research advances enabled by multidisciplinary collaborations between researchers. Meeting the needs of complex parallel applications and the requirements of exascale architectures raises numerous challenges which are still left unaddressed. As a result, several parts of the software stack must evolve to better support these architectures. More importantly, the links between these parts must be strengthened to form a coherent, tightly integrated software suite. The main scientific challenges we intend to address are: productivity, performance portability, heterogeneity, scalability and resilience, performance and energy efficiency.

### 9.3.2 AID

#### AID AFF3CT

**Participants:** Olivier Aumage, François Cheminade, Andrea Lesavourey, Laercio Lima Pilla, Diane Orhan, Joachim Rosseel, Victor-Benjamin Vilain.

- 2023 - 2025 (24 months)
- Coordinator: Laercio Lima Pilla
- Other partners: Inria CANARI, IMS, LIP6
- Abstract: This project focuses on the development of new components and functionalities to AFF3CT with the objective of improving its performance and usability. It includes the implementation of 5G and cryptography modules, an integration with the Julia programming language, and the inclusion of new components to help profile and visualize the performance of different modules and digital communication standards.

### 9.3.3 Défis Inria

#### Défi PULSE

**Participants:** Laercio Lima Pilla.

- 2022 - 2026 (48 months)
- Coordinator: Romain Rouvoy (Inria SPIRALS), Rémi Bouzel (Qarnot)
- Other partners: Qarnot, ADEME, Inria: SPIRALS, AVALON, STACK, TOPAL, STORM, CTRL+A
- Abstract: In the joint project PULSE (PUshing Low-carbon Services towards the Edge), Inria teams work together with Qarnot and ADEME on the holistic analysis of the environmental impact of its computing infrastructure and on implementing green services on the Edge.



### 9.3.4 Inria exploratory actions

#### LLM4DiCE

**Participants:** Asia Auville, Mihail Popov, Emmanuelle Saillard.

- 2024 - 2027 (36 months)
- Coordinator: Emmanuelle Saillard and Mihail Popov
- Abstract: Large Language Models (LLMs) are a hot and rapidly evolving research topic. In particular, their recent successes in summarization, question-answering, and code generation with AI pair programming make them attractive candidates in the field of error verification. We propose to harness these LLMs capabilities with fine-tuning on carefully generated datasets through a novel clustering strategy based on Natural Language Processing (NLP) techniques and code embedding to assist bug detection and correction, targeting hard domains such as parallel program verification.

## 10 Dissemination

### 10.1 Promoting scientific activities

#### 10.1.1 Scientific events: organisation

**Participants:** Olivier Aumage, Mihail Popov, Emmanuelle Saillard.

#### General chair, scientific chair

- Emmanuelle Saillard and Mihail Popov participated in the organisation of the second HPC Bugs Fest during the "Correctness" Workshop of the SuperComputing SC24 Conference in Atlanta.
- Emmanuelle Saillard was general chair of the C3PO'24 workshop and co-general chair of the ProTools'24 workshop.
- Olivier Aumage organized the 3rd AFF3CT User Day at LIP6 Laboratory, in Paris, in Nov. 2024.

#### 10.1.2 Scientific events: selection

**Participants:** Amina Guermouche, Emmanuelle Saillard, Alice Lasserre.

#### Member of the conference program committees

- Emmanuelle Saillard: COMPAS 2024, Correctness 2024, Cluster 2024
- Amina Guermouche: Cluster 2024, Super Computing 2024
- Alice Lasserre: Reproducibility 2024, Super Computing 2024

#### 10.1.3 Journal

**Participants:** Olivier Aumage, Amina Guermouche, Laércio Lima Pilla, Samuel Thibault.

### Member of the editorial boards

- Samuel Thibault: JPDC Associate Editor

### Reviewer - reviewing activities

- Olivier Aumage: JPDC
- Amina Guermouche: TPDS
- Laércio Lima Pilla: JPDC, TPDS Reproducibility, FGCS
- Samuel Thibault: JPDC, TOPC

#### 10.1.4 Invited talks

**Participants:** Olivier Aumage, Lana Scravaglieri, Samuel Thibault.

- Olivier Aumage
  - SIAM-PP 24 Conference, Baltimore, US, March 2024
  - DGA-MI Developers'day, Bruz, France, October 2024.
- Samuel Thibault
  - JLESC Workshop, Kobe, JP, April 2024
  - Compas Keynote, Nantes, FR, July 2024
  - Exposé MCIA, Bordeaux, FR, October 2024
- Lana Scravaglieri
  - IFPEN-Inrai joint laboratory Workshop, Paris, FR, December 2024
  - Teaching seminars, Bordeaux, FR, May 2024

#### 10.1.5 Leadership within the scientific community

**Participants:** Olivier Aumage.

- Olivier Aumage: Contribution to the Strategic Research Agenda 6th edition (SRA6) of the European Technology Platform for HPC (ETP4HPC).[31]

#### 10.1.6 Scientific expertise

**Participants:** Emmanuelle Saillard, Marie-Christine Counilh, Nathalie Furmento, Amina Guermouche, Laércio Lima Pilla, Samuel Thibault.

- Emmanuelle Saillard participated to the CRCN / ISFP Inria Researcher selection jury for the Inria Research Center at Lyon.
- Marie-Christine Counilh was a member of a selection committee for an Associate Professor position at the Robert Schuman University Institute of Technology (IUT), Strasbourg, May 2024.
- Nathalie Furmento was a member of recruiting committees for engineer positions.

- Amina Guermouche was a member of a selection committee for Toulouse INP, May 2024.
- Laércio Lima Pilla participated to the CRCN / ISFP Inria Researcher selection jury for the Inria Research Center at Rennes University, May 2024.
- Samuel Thibault was a member of selection committee for an Associate Professor position at the University of Bordeaux, May 2024.

### 10.1.7 Research administration

**Participants:** Olivier Aumage, Nathalie Furmento, Laércio Lima Pilla, Emmanuelle Saillard, Samuel Thibault.

- Olivier Aumage is an Elected Member of LaBRI's Scientific Council and head of LaBRI's STORM Team.
- Nathalie Furmento
  - member of the CDT (commission développement technologique) for the Inria Research Center at the University of Bordeaux.
  - selected member of the council of the LaBRI.
  - member of the societal challenges commission at the LaBRI.
  - member of the committee on gender equality and equal opportunities of the Inria Research center at the University of Bordeaux.
- Laercio Lima Pilla
  - member of the societal challenges commission at the LaBRI.
  - member of the committee on gender equality and equal opportunities of the Inria Research center at the University of Bordeaux.
- Emmanuelle Saillard is a member of the Commission de délégation at Inria Research Centre of the University of Bordeaux.
- Samuel Thibault is a selected member of the council of the LaBRI.

## 10.2 Teaching - Supervision - Juries

### 10.2.1 Teaching

**Participants:** Vincent Alba, Olivier Aumage, Asia Auville, Albert D'Aviau De Piolant, Marie-Christine Counilh, Nathalie Furmento, Amina Guermouche, Lise Jolicoeur, Alice Lasserre, Laércio Lima Pilla, Thomas Morin, Raymond Namyst, Diane Orhan, Mihail Popov, Emmanuelle Saillard, Lana Scravaglieri, Samuel Thibault.

- Training Management
  - Raymond Namyst is vice chair of the Computer Science Training Department of University of Bordeaux.
- Management
  - Samuel Thibault is responsible for the 1st year Computer Science students at the University of Bordeaux, and responsible for a professional curriculum

- Pierre-André Wacrenier is responsible for the 3rd year Computer Science students at the University of Bordeaux and director of the Resource Center for Mathematics and Computer Science Practical Work (3000+ users, 500+ computers).
- Academic Teaching
  - Engineering School + Master: Olivier Aumage, Multicore Architecture Programming, 24HeTD, M2, ENSEIRB-MATMECA + University of Bordeaux.
  - Engineering School: Emmanuelle Saillard, Languages of parallelism, 12HeC, M2, ENSEIRB-MATMECA.
  - Master: Laércio Lima Pilla, Algorithms for High-Performance Computing Platforms, 17HeTD, M2, ENSEIRB-MATMECA and University of Bordeaux.
  - Master: Laércio Lima Pilla, Scheduling and Runtime Systems, 27.75 HeTD, M2, University of Paris-Saclay.
  - Engineering School: Mihail Popov, Project C, 25HeC, L3, ENSEIRB-MATMECA.
  - Engineering School: Mihail Popov, Cryptography, 33HeC, M1, ENSEIRB-MATMECA.
  - Amina Guermouche is responsible for the computer science first year at ENSEIRB-MATMECA
  - 1st year : Amina Guermouche, Linux Environment, 24HeTD, ENSEIRB-MATMECA
  - 1st year : Amina Guermouche, Computer architecture, 36 HeTD, ENSEIRB-MATMECA
  - 1st year : Amina Guermouche, Programming project, 25HeTD, ENSEIRB-MATMECA
  - 2nd year : Amina Guermouche, System Programming, 18HeTD, ENSEIRB-MATMECA
  - 2nd year : Amina Guermouche, Operating systems, 36HeTDHeTD, ENSEIRB-MATMECA
  - 3rd year : Amina Guermouche, GPU Programming, 39HeTD, ENSEIRB-MATMECA + University of Bordeaux
  - 1st year : Albert d’Aviau de Piolant, Computer architecture, 20HeTD, ENSEIRB-MATMECA
  - 2nd year : Albert d’Aviau de Piolant, C++, 10HeTD, ENSEIRB-MATMECA
  - Licence: Samuel Thibault is responsible for the Licence Pro ADSILLH (Administration et Développeur de Systèmes Informatiques à base de Logiciels Libres et Hybrides).
  - Licence: Samuel Thibault is responsible for the 1st year of the computer science Licence.
  - Licence: Samuel Thibault, Networking, 51HeTD, Licence Pro, University of Bordeaux.
  - Licence: Samuel Thibault, Free Software contribution projects, 8HeTD, University of Bordeaux.
  - Master: Samuel Thibault, Operating Systems, 24HeTD, M1, University of Bordeaux.
  - Master: Alice Lasserre, Operating Systems, 24HeTD, M1, University of Bordeaux.
  - Licence: Alice Lasserre, Methods and tools for using computer systems, 8HeTD, L1/L2/L3, University of Bordeaux.
  - Licence: Alice Lasserre, Research discovery, L3, University of Bordeaux.
  - Master: Samuel Thibault, System Security, 20HeTD, M2, University of Bordeaux.
  - Master: Nathalie Furmento, Operating Systems, 24HeTD, M1, University of Bordeaux.
  - Licence: Marie-Christine Counilh, Introduction to Computer Science, 56HeTD, L1, University of Bordeaux.
  - Licence: Marie-Christine Counilh, Introduction to C programming, 38HeTD, L1, University of Bordeaux. Co-responsible for this teaching.
  - Licence: Marie-Christine Counilh, Object oriented programming in Java, 32HeTD, L2, University of Bordeaux.
  - Master MIAGE : Marie-Christine Counilh, Object oriented programming in Java, 30HeTD, M1, University of Bordeaux.

- Licence: Marie-Christine Counilh is responsible for computer science tutoring for undergraduate students in the College of Science and Technology at the University of Bordeaux.
- 1st year : Diane Orhan, Computer architecture, 16HeTD, ENSEIRB-MATMECA
- 2nd year : Diane Orhan, C++, 10HeTD, ENSEIRB-MATMECA
- 3rd year : Diane Orhan, Algorithms for HPC, 2HeTD, ENSEIRB-MATMECA
- 1st year: Asia Auville, Logic and proof of program, 14HeTD, ENSEIRB-MATMECA
- 1st year: Vincent Alba, Logic and proof of program, 14HeTD, ENSEIRB-MATMECA
- Master: Thomas Morin, Computability and complexity, 24HeTD, M1, University of Bordeaux.
- Engineering School: Lise Jolicoeur, Software for HPC Clusters (Logiciels Cluster), 16HeTD, M1, ENSIIE.
- 1st year : Nicolas Ducarton, Functional programming, 20HeTD, ENSEIRB-MATMECA
- 1st year : Nicolas Ducarton, Functional programming project, 25HeTD, ENSEIRB-MATMECA
- BUT : Nicolas Ducarton, Virtualization, 14HeTD, S4, University of Bordeaux.
- 2nd year : Joachim Rosseel, Channel coding, 16HeTD, ENSEIRB-MATMECA
- 3rd year : Joachim Rosseel, S9 project (cyber-security and channel coding), 10HeTD, ENSEIRB-MATMECA
- 1st year : Lana Scravaglieri, Imperative programming and tools, 24HeTD, ENSEIRB-MATMECA
- 1st year : Lana Scravaglieri, Imperative programming project, 25HeTD, ENSEIRB-MATMECA
- Licence: Pierre-André Wacrenier, Programming Project, 48HeTD, M1, University of Bordeaux.
- Licence: Pierre-André Wacrenier, System Programming, 64HeTD, M1, University of Bordeaux.
- Master: Pierre-André Wacrenier, Parallel Programming, 40HeTD, M1, University of Bordeaux.
- Tutorials
  - Nathalie Furmento, Samuel Thibault: StarPU Tutorial, Saclay, FR, May 2024
- Summer school
  - Emmanuelle Saillard: Ecole jeunes chercheurs EJCP24 - Argelès-sur-Mer (Pyrénées Orientales), June 2024

### 10.2.2 Supervision

- PhD in progress: Lana Scravaglieri, Portable vectorization with numerical accuracy control for multi-precision simulation codes. Advisors: Olivier Aumage, Mihail Popov, Thomas Guignon (IFPEN) and Ani Anciaux-Sedrakian (IFPEN).
- PhD in progress: Asia Auville, Large Language Models for Detection and Correction of Errors in HPC Applications. Advisors: Emmanuelle Saillard and Mihail Popov.
- PhD in progress: Radjasouria Vinayagame, Optimization of porting and performance of HPC applications with distributed and globally addressed memory. Advisors: Emmanuelle Saillard and Samuel Thibault.
- PhD in progress: Albert D'Aviau de Piolant, Energy aware scheduling for exascale architectures. Advisors: Abdou Guermouche and Amina Guermouche.
- PhD in progress: Vincent Alba, "Task scheduling for exascale". Advisor: Denis Barthou.
- PhD in progress: Jules Risse, Fine-grain energy consumption measurement of HPC task-based programs. Advisors: Amina Guermouche and François Trahay.
- PhD in progress: Thomas Morin, Scheduling recursive task graphs. Advisors: Abdou Guermouche, Samuel Thibault, Pierre-André Wacrenier.

- Internship: Asia Auville, Feb. - Sept. 2024, Emmanuelle Saillard, Mihail Popov
- Internship: Abdelbarie El Metni, Feb. - Aug. 2024, Olivier Aumage.
- Internship: Evan Potin, Mar. - Aug. 2024, Olivier Aumage.

### 10.2.3 Juries

**Participants:** Emmanuelle Saillard, Amina Guermouche, Samuel Thibault.

- Emmanuelle Saillard
  - Guest for the PhD of Richard Satori, Optimal Parameters Determination for the Execution of MPI Applications on Parallel Architectures, Eviden.
  - Reviewer for the PhD of Tim Jammer, Modernization and Optimization of MPI Codes, Technical University Darmstadt, DE.
- Amina Guermouche
  - PhD of Jules Pénuchot, Techniques avancées de génération de code pour la performance, University Paris Saclay.
- Samuel Thibault
  - PhD of Kevin Sala Penadés, University of Catalunya, ES.
  - Reviewer for PhD of Pierre-Etienne Polet, ENS of Lyon
  - President for the PhD of Alexis Bandet, University of Bordeaux

## 10.3 Popularization

### 10.3.1 Specific official responsibilities in science outreach structures

**Participants:** Emmanuelle Saillard, Raymond Namyst, Alice Lasserre.

- Emmanuelle Saillard, Raymond Namyst: Organization of Moi Informaticienne - Moi Mathématicienne, April 2024.
- Emmanuelle Saillard
  - Responsible of popularization activities for Inria Research Centre of the University of Bordeaux.
  - Member of the scientific committee of the Blaise Pascal Fondation
  - Member of the executive board of SIF (Société Informatique de France)
- Alice Lasserre: Co-organization of JCAD 2024 (Journées Calcul et Données : Rencontres scientifiques et techniques autour du calcul et des données), November 2024.

### 10.3.2 Productions (articles, videos, podcasts, serious games, ...)

**Participants:** Emmanuelle Saillard.

- Emmanuelle Saillard:
  - Video for Numérixplore: 1 minute to talk about numeric to schoolchildren, Sept. 2024, Futuroscope
  - Article for the TIPE 2024-2025 - transition, transformation, conversion: [tipe-2024/blog/2024/Emmanuelle-Saillard/](https://tipe-2024/blog/2024/Emmanuelle-Saillard/)

### 10.3.3 Participation in Live events

**Participants:** Asia Auville, Albert d'Aviau de Piolant, Marie-Christine Counilh, Nathalie Furmento, Emmanuelle Saillard, Radjasouria Vinayagame, Lana Scravaglieri, Mihail Popov, Pierre-André Wacrenier, Diane Orhan, Alice Lasserre, Raymond Namyst.

- Nathalie Furmento, Emmanuelle Saillard
  - Participation of AI4Industry, January 2024.
  - Participation in "Têtes chercheuses" (speed-searching and demonstration), April 2024
  - Organization of the welcoming ENS-Lyon undergraduate students, December 2024.
- Lana Scravaglieri, Asia Auville, Albert d'Aviau de Piolant, Mihail Popov, Emmanuelle Saillard: Presentation to ENS-Lyon undergraduate students, December 2024.
- Asia Auville: Participation of the student speed-meeting during Moi Informaticienne - Moi Mathématicienne, University of Bordeaux, April 2024.
- Diane Orhan: Participation of the student speed-meeting during Moi Informaticienne - Moi Mathématicienne, University of Bordeaux, April 2024.
- Radjasouria Vinayagame
  - Participation of a panel organised by the ENSEIRB-MATMECA engineering school about how to become a Ph.D student, March 2024
  - Interview by L3 students of the university of Bordeaux, April 2024
- Emmanuelle Saillard
  - Co-organization and participation of the SNT days at Inria Research Centre of the University of Bordeaux, January 2024
  - Participation at the launch of the week of maths, Périgueux, March 2024
  - Participation at the "Circuit scientifique Bordelais" (2 days), Inria, Oct. 2024: ("La grande muraille d'Egypte", "Etre un citoyen numérique")
  - Organization of a workshop during Moi Informaticienne - Moi Mathématicienne, University of Bordeaux, April 2024.
  - Chiche!: Victor Louis high school (December 2024)
  - Jury member of the CGenial contest, April 2024
  - Participation at the "Nuit européenne de la recherche" (speedsearching)

- Co-organisation of doctoral training of outreach activities, Inria Research Centre of the University of Bordeaux, November 2024
- Olivier Aumage
  - "Circuit Scientifique Aquitain", 4 groups, Oloron-Sainte-Marie, October 2024
  - Chiche!: Val de Garonne high school, 2 groups, Marmande, December 2024
- Marie-Christine Counilh, Mihail Popov, Pierre-André Wacrenier: Half-day supervision of high-school students during a practical session on HPC, June 2024
- Raymond Namyst, Alice Lasserre: presentation of Easypp to Atos, June 2024

## 11 Scientific production

### 11.1 Major publications

- [1] O. Beaumont, R. Bouzel, L. Eyraud-Dubois, E. Korkmaz, L. Lima Pilla and A. van Kempen. 'A  $1.25(1+\epsilon)$ -Approximation Algorithm for Scheduling with Rejection Costs Proportional to Processing Times'. In: International European Conference on Parallel and Distributed Computing (Euro-Par). Vol. 14801. Lecture Notes in Computer Science. Madrid, Spain: Springer Nature Switzerland, 26th Aug. 2024, pp. 225–238. DOI: [10.1007/978-3-031-69577-3\\_16](https://doi.org/10.1007/978-3-031-69577-3_16). URL: <https://hal.science/hal-04670834>.
- [2] O. Beaumont, J.-F. David, L. Eyraud-Dubois and S. Thibault. 'Exploiting Processor Heterogeneity to Improve Throughput and Reduce Latency for Deep Neural Network Inference'. In: SBAC-PAD 2024 - IEEE 36th International Symposium on Computer Architecture and High Performance Computing. Hilo, Hawaii, United States, 13th Nov. 2024. URL: <https://hal.science/hal-04690154>.
- [3] N. Furmento, A. Guermouche, G. Lucas, T. Morin, S. Thibault and P.-A. Wacrenier. 'Optimizing Parallel System Efficiency: Dynamic Task Graph Adaptation with Recursive Tasks'. In: WAMTA 2024 - Workshop on Asynchronous Many-Task Systems and Applications 2024. Knoxville, United States: <https://wamta24.icl.utk.edu/>, 14th Feb. 2024. URL: <https://inria.hal.science/hal-04548787>.
- [4] T. Jammer, E. Saillard, S. Schwitanski, J. Jenke, R. Vinayagame, A. Hück and C. Bischof. 'MPI-BugBench: A Framework for Assessing MPI Correctness Tools'. In: *Lecture Notes in Computer Science*. EuroMPI/Australia 2024. Vol. LNCS-15267. Recent Advances in the Message Passing Interface 31st European MPI Users' Group Meeting, EuroMPI 2024, Perth, WA, Australia, September 25–27, 2024, Proceedings. Perth, Australia: Springer Nature Switzerland, 25th Sept. 2025, pp. 121–137. DOI: [10.1007/978-3-031-73370-3\\_8](https://doi.org/10.1007/978-3-031-73370-3_8). URL: <https://hal.science/hal-04878321>.
- [5] L. Jolicoeur, F. Diakhaté and R. Namyst. 'Leveraging private container networks for increased user isolation and flexibility on HPC clusters'. In: *High Performance Computing. ISC High Performance 2024 International Workshops*. WOCC 2024 - 2nd International Workshop on Converged Computing on Edge, Cloud, and HPC. Hamburg, Germany, Dec. 2024. URL: <https://inria.hal.science/hal-04740275>.
- [6] J. E. Karchi, H. Chen, A. Tehranijamsaz, A. Jannesari, M. Popov and E. Saillard. 'MPI Errors Detection using GNN Embedding and Vector Embedding over LLVM IR'. In: IPDPS 2024 - 38th International Symposium on Parallel and Distributed Processing. San Francisco, United States, 27th May 2024. URL: <https://inria.hal.science/hal-04724011> (cit. on p. 14).
- [7] A. Lasserre, J. M. Couteyen Carpaye, A. Guermouche and R. Namyst. 'Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver'. In: The 25th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2024). San Francisco, United States, 31st May 2024, p. 10. URL: <https://inria.hal.science/hal-04403209>.



- [8] A. Nunes, D. Sodré, C. Boeres, J. Viterbo, L. Drummond, V. Rebello, L. Teylo, F. Portella, P. Estrela and R. Malini. ‘A Framework for Executing Long Simulation Jobs Cheaply in the Cloud’. In: IC2E 2024 - IEEE International Conference on Cloud Engineering. Paphos, Cyprus: IEEE, 12th Sept. 2024, pp. 233–244. DOI: [10.1109/IC2E61754.2024.00033](https://doi.org/10.1109/IC2E61754.2024.00033). URL: <https://hal.science/hal-04839966>.
- [9] R. Vinayagame, V. M. Nguyen, M. Sergent, S. Thibault and E. Saillard. ‘Static-Dynamic analysis for Performance and Accuracy of Data Race Detection in MPI One-Sided Programs’. In: C3PO 2024 - Compiler-assisted Correctness Checking and Performance Optimization for HPC. Hambourg, Germany, 12th May 2024. URL: <https://hal.science/hal-04581890>.

## 11.2 Publications of the year

### International peer-reviewed conferences

- [10] O. Beaumont, R. Bouzel, L. Eyraud-Dubois, E. Korkmaz, L. Lima Pilla and A. van Kempen. ‘A  $1.25(1+\epsilon)$ -Approximation Algorithm for Scheduling with Rejection Costs Proportional to Processing Times’. In: International European Conference on Parallel and Distributed Computing (EuroPar). Vol. 14801. Lecture Notes in Computer Science. Madrid, Spain: Springer Nature Switzerland, 26th Aug. 2024, pp. 225–238. DOI: [10.1007/978-3-031-69577-3\\_16](https://doi.org/10.1007/978-3-031-69577-3_16). URL: <https://hal.science/hal-04670834> (cit. on p. 17).
- [11] O. Beaumont, J.-F. David, L. Eyraud-Dubois and S. Thibault. ‘Exploiting Processor Heterogeneity to Improve Throughput and Reduce Latency for Deep Neural Network Inference’. In: SBAC-PAD 2024 - IEEE 36th International Symposium on Computer Architecture and High Performance Computing. Hilo, Hawaii, United States, 13th Nov. 2024. URL: <https://hal.science/hal-04690154> (cit. on p. 13).
- [12] O. Beaumont, J.-F. David, L. Eyraud-Dubois and S. Thibault. ‘StarONNX: a Dynamic Scheduler for Low Latency and High Throughput Inference on Heterogeneous Resources’. In: HeteroPar 2024 - 22ND INTERNATIONAL WORKSHOP Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms. HeteroPar’24 Proceedings. Madrid, Spain, 2024. URL: <https://inria.hal.science/hal-04646530> (cit. on p. 13).
- [13] N. Furmento, A. Guermouche, G. Lucas, T. Morin, S. Thibault and P.-A. Wacrenier. ‘Optimizing Parallel System Efficiency: Dynamic Task Graph Adaptation with Recursive Tasks’. In: WAMTA 2024 - Workshop on Asynchronous Many-Task Systems and Applications 2024. Knoxville, United States: <https://wamta24.icl.utk.edu/>, 14th Feb. 2024. URL: <https://inria.hal.science/hal-04548787> (cit. on p. 12).
- [14] M. Gonthier, E. Larsson, L. Marchal, C. Nettelblad and S. Thibault. ‘Data-Driven Locality-Aware Batch Scheduling’. In: APDCM 2024 - 26th Workshop on Advances in Parallel and Distributed Computational Models. San Francisco, United States, 27th May 2024. URL: <https://inria.hal.science/hal-04500281> (cit. on p. 12).
- [15] T. Jammer, E. Saillard, S. Schwitanski, J. Jenke, R. Vinayagame, A. Hück and C. Bischof. ‘MPI-BugBench: A Framework for Assessing MPI Correctness Tools’. In: *Lecture Notes in Computer Science*. EuroMPI/Australia 2024. Vol. LNCS-15267. Recent Advances in the Message Passing Interface 31st European MPI Users’ Group Meeting, EuroMPI 2024, Perth, WA, Australia, September 25–27, 2024, Proceedings. Perth, Australia: Springer Nature Switzerland, 25th Sept. 2025, pp. 121–137. DOI: [10.1007/978-3-031-73370-3\\_8](https://doi.org/10.1007/978-3-031-73370-3_8). URL: <https://hal.science/hal-04878321>.
- [16] T. Jammer, S. Schwitanski, E. Saillard, A. Hück, J. Jenke, R. Vinayagame and C. Bischof. ‘Designing Quality MPI Correctness Benchmarks: Insights and Metrics’. In: *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 8th International Workshop on Software Correctness for HPC Applications (Correctness’24). Atlanta, United States, 18th Nov. 2024. URL: <https://hal.science/hal-04878332>.

- [17] L. Jolicoeur, F. Diakhaté and R. Namyst. ‘Leveraging private container networks for increased user isolation and flexibility on HPC clusters’. In: *High Performance Computing. ISC High Performance 2024 International Workshops*. WOCC 2024 - 2nd International Workshop on Converged Computing on Edge, Cloud, and HPC. Hamburg, Germany, Dec. 2024. URL: <https://inria.hal.science/hal-04740275>.
- [18] J. E. Karchi, H. Chen, A. Tehranijamsaz, A. Jannesari, M. Popov and E. Saillard. ‘MPI Errors Detection using GNN Embedding and Vector Embedding over LLVM IR’. In: IPDPS 2024 - 38th International Symposium on Parallel and Distributed Processing. San Francisco, United States, 27th May 2024. URL: <https://inria.hal.science/hal-04724011>.
- [19] A. Lasserre, J. M. Couteyen Carpaye, A. Guermouche and R. Namyst. ‘Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver’. In: The 25th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2024). San Francisco, United States, 31st May 2024, p. 10. URL: <https://inria.hal.science/hal-04403209>.
- [20] A. Nunes, D. Sodré, C. Boeres, J. Viterbo, L. Drummond, V. Rebello, L. Teylo, F. Portella, P. Estrela and R. Malini. ‘A Framework for Executing Long Simulation Jobs Cheaply in the Cloud’. In: IC2E 2024 - IEEE International Conference on Cloud Engineering. Paphos, Cyprus: IEEE, 12th Sept. 2024, pp. 233–244. DOI: [10.1109/IC2E61754.2024.00033](https://doi.org/10.1109/IC2E61754.2024.00033). URL: <https://hal.science/hal-04839966> (cit. on p. 13).
- [21] R. Vinayagame, V. M. Nguyen, M. Sergent, S. Thibault and E. Saillard. ‘Static-Dynamic analysis for Performance and Accuracy of Data Race Detection in MPI One-Sided Programs’. In: C3PO 2024 - Compiler-assisted Correctness Checking and Performance Optimization for HPC. Hamburg, Germany, 12th May 2024. URL: <https://hal.science/hal-04581890> (cit. on p. 14).

#### Conferences without proceedings

- [22] O. Beaumont, J.-F. David, L. Eyraud-Dubois and S. Thibault. ‘StarONNX : Un ordonnanceur dynamique pour une inférence rapide et à haut débit sur des ressources hétérogènes’. In: Compas 2024 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Nantes, France, 2nd July 2024. URL: <https://inria.hal.science/hal-04668550> (cit. on p. 13).
- [23] A. Lira Nunes, C. Boeres, L. M. de A. Drummond and L. Lima Pilla. ‘Optimal Time and Energy-Aware Client Selection Algorithms for Federated Learning on Heterogeneous Resources’. In: 2024 IEEE 36th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). Hilo, France: IEEE, 13th Nov. 2024, pp. 148–158. DOI: [10.1109/SBAC-PAD63648.2024.00021](https://doi.org/10.1109/SBAC-PAD63648.2024.00021). URL: <https://hal.science/hal-04690494>.
- [24] T. Morin. ‘Optimiser l’Efficacité des Systèmes Parallèles : Adaptation Dynamique des Graphes de Tâches Récursives’. In: COMPAS 2024 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Nantes, France, 2nd July 2024. URL: <https://inria.hal.science/hal-04672417> (cit. on p. 12).

#### Reports & preprints

- [25] V. Alba, O. Aumage, D. Barthou, R. Colin, M.-C. Counilh, S. Genaud, A. Guermouche, V. Loechner and A. Thangamani. *Performance portability of generated cardiac simulation kernels through automatic dimensioning and load balancing on heterogeneous nodes*. 10th June 2024. URL: <https://hal.science/hal-04606388> (cit. on p. 16).
- [26] O. Beaumont, R. Bouzel, L. Eyraud-Dubois, E. Korkmaz, L. Lima Pilla and A. van Kempen. *Approximation Algorithms for Scheduling with/without Deadline Constraints where Rejection Costs are Proportional to Processing Times*. 15th Oct. 2024. URL: <https://hal.science/hal-04745701> (cit. on p. 17).
- [27] M. Gonthier, S. Thibault and L. Marchal. *A generic scheduler to foster data locality for GPU and out-of-core task-based applications*. 13th Sept. 2024. URL: <https://inria.hal.science/hal-04146714> (cit. on p. 12).

- [28] A. Lesavourey, K. Fukushima, T. Plantard and A. Sipasseuth. *Diagonally dominant matrices for cryptography*. 2nd Feb. 2024. URL: <https://hal.science/hal-03728051>.
- [29] A. d'Aviau de Piolant, H. Tayeb, B. Bramas, M. Faverge, A. Guermouche and A. Guermouche. *Improving energy efficiency of HPC applications using unbalanced GPU power capping*. 11th Oct. 2024. URL: <https://inria.hal.science/hal-04883872> (cit. on p. 17).

### Other scientific publications

- [30] A. Auville. 'Utilisation des grands modèles de langages pour la détection et la correction des erreurs'. Inria & Labri, Université de Bordeaux, 30th Sept. 2024. URL: <https://inria.hal.science/hal-04877570>.
- [31] P. Carpenter, G. Antoniu, M. Arenaz, O. Aumage, J. Beránek, A. Buttari, A. Costan, S. Happ, V. Kannan, C. Perez, A. Peña, A. Scionti, X. Vigouroux and P. Viviani. *ETP4HPC SRA White Paper - Programming Environment*. Dec. 2024. DOI: [10.5281/zenodo.14446622](https://doi.org/10.5281/zenodo.14446622). URL: <https://hal.science/hal-04905035> (cit. on p. 23).
- [32] A. Cassagne, R. Tajan and O. Aumage. 'Combined Computer Vision and SDR Transmission on Multicore CPUs'. In: Journée des Développeurs. Bruz, France, 10th Oct. 2024. URL: <https://hal.science/hal-04771981>.
- [33] A. El Metni. 'Support d'exécution à base de tâches et programmation de haut niveau pour la simulation par éléments finis'. Bordeaux INP, 30th Aug. 2024. URL: <https://inria.hal.science/hal-04755787>.
- [34] A. Lasserre, J. M. Couteyen Carpaye, A. Guermouche and R. Namyst. 'Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver'. In: Doctoral students' day of the Mathematics and Computer Science doctoral school of the University of Bordeaux (EDMI). Bordeaux, France, 11th Apr. 2024. URL: <https://inria.hal.science/hal-04895715>.
- [35] E. Potin. 'Mémoire de stage'. Université de Bordeaux, 19th June 2024. URL: <https://inria.hal.science/hal-04755831>.

### Software

- [36] [SW] O. Beaumont, L. Eyraud-Dubois, E. Korkmaz and L. Lima Pilla, *Experimental codes and results for the paper "A 5/4(1+eps)-Approximation Algorithm for Scheduling with Rejection Costs Proportional to Processing Times"* 22nd Mar. 2024. Inria & Labri, Univ. Bordeaux. LIC: CeCILL Free Software License Agreement v2.0. HAL: ([hal-04517532](https://hal.science/hal-04517532)), URL: <https://inria.hal.science/hal-04517532>, SWHID: ([swh:1:dir:53aa25178b70f7d119690440f64912c226521893;origin=https://hal.archives-ouvertes.fr/hal-04517532;visit=swh:1:snp:690ab9c97d792e39eb94c530093dc36623bb9dac;anchor=swh:1:rel:f7390f430900bcc0a290b2fc33d2ab5285922588;path=/](https://sw.hal.archives-ouvertes.fr/hal-04517532)) (cit. on p. 17).

### 11.3 Cited publications

- [37] L. Scravaglieri, M. Popov, L. Lima Pilla, A. Guermouche, O. Aumage and E. Saillard. 'Optimizing performance and energy across problem sizes through a search space exploration and machine learning'. In: *Journal of Parallel and Distributed Computing* 180 (2023), p. 104720. DOI: <https://doi.org/10.1016/j.jpdc.2023.104720>. URL: <https://www.sciencedirect.com/science/article/pii/S0743731523000904> (cit. on p. 12).