

2025 Activity Report

RESEARCH CENTRE: Inria Centre at the University of Bordeaux

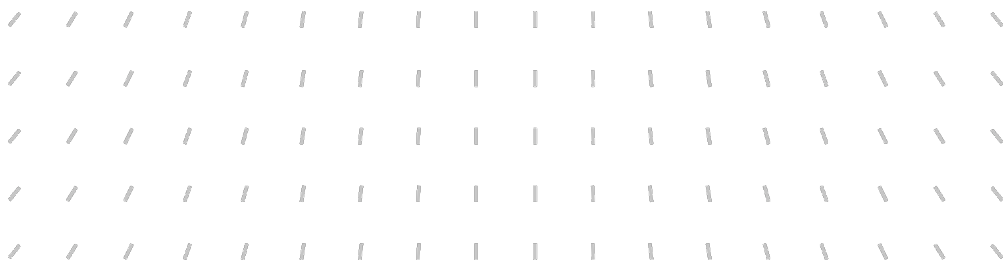
IN PARTNERSHIP WITH: Bordeaux INP, Université de Bordeaux, CNRS

Project-Team

STORM

Static Optimizations, Runtime Methods

In collaboration with Laboratoire Bordelais de Recherche en Informatique (LaBRI)



Project-Team STORM

Creation of the Project-Team: 2017 July 01

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

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
- A2.2.1. – Static analysis
- A6.2.7. – HPC for machine learning
- 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. – wireless networks
- B6.2.3. – Satellite networks
- B9.1. – Education

Contents

| | |
|---|-----------|
| Project-Team STORM | 1 |
| 1 Team members, visitors, external collaborators | 5 |
| 2 Overall objectives | 7 |
| 3 Research program | 8 |
| 3.1 Parallel Computing and Architectures | 8 |
| 3.2 Scientific and Societal Stakes | 8 |
| 3.3 Towards More Abstraction | 9 |
| 4 Application domains | 10 |
| 4.1 Application domains benefiting from HPC | 10 |
| 4.2 Application in High performance computing/Big Data | 10 |
| 5 Social and environmental responsibility | 10 |
| 5.1 Footprint of research activities | 10 |
| 5.2 Impact of research results | 10 |
| 6 Highlights of the year | 11 |
| 6.1 Awards | 11 |
| 6.2 Animation of the scientific community | 11 |
| 7 Latest software developments, platforms, open data | 11 |
| 7.1 Latest software developments | 11 |
| 7.1.1 AFF3CT | 11 |
| 7.1.2 PARCOACH | 11 |
| 7.1.3 MIPP | 12 |
| 7.1.4 CERE | 12 |
| 7.1.5 DUF | 13 |
| 7.1.6 MBI | 13 |
| 7.1.7 EasyPAP | 13 |
| 7.1.8 StarPU | 14 |
| 7.1.9 MPI-BugBench | 15 |
| 7.1.10 CORHPEX | 15 |
| 7.1.11 StreamPU | 16 |
| 7.2 New platforms | 16 |
| 7.3 Open data | 16 |
| 8 New results | 16 |
| 8.1 Scheduling for Pipelined and Replicated Task Chains and Graphs for Software-Defined Radio | 16 |
| 8.2 Optimization Space Exploration | 17 |
| 8.3 Price-performance analysis for task-based simulation in the Cloud | 17 |
| 8.4 Interoperable resource sharing between multiple runtime systems | 17 |
| 8.5 Portable EMI cardiac electrophysiology simulation | 18 |
| 8.6 Task scheduling with memory constraints | 18 |
| 8.7 Programming Heterogeneous Architectures Using Hierarchical Tasks | 18 |
| 8.8 C++ interfacing with StarPU | 19 |
| 8.9 Fault-Tolerance for task-based applications on large-scale systems | 19 |
| 8.10 Integration of asynchronous network communications scheduling and local task scheduling | 19 |
| 8.11 Task scheduling to improve throughput and reduce latency for deep neural network inference | 19 |
| 8.12 A Machine-Learning Approach to MPI Error Detection | 20 |
| 8.13 One-Sided Communications Automatic Rewriting | 20 |

| | | |
|-----------|---|-----------|
| 8.14 | Leveraging private container networks for increased user isolation and flexibility on HPC clusters | 20 |
| 8.15 | Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver - Best Paper Award | 21 |
| 8.16 | Highlighting EasyPAP Improvements | 21 |
| 8.17 | Automatic Dimensioning and Load Balancing on Heterogeneous Architectures | 21 |
| 8.18 | Improving energy efficiency of HPC applications using unbalanced GPU power capping | 22 |
| 8.19 | Fine-grain energy consumption modeling of HPC task-based programs | 22 |
| 8.20 | High-level Python programming interface for StreamPU | 23 |
| 8.21 | Code-based post-quantum cryptographical schemes in AFF3CT | 23 |
| 8.22 | 5G Physical Broadcast Channel in AFF3CT | 23 |
| 9 | Bilateral contracts and grants with industry | 23 |
| 9.1 | Bilateral contracts with industry | 23 |
| 9.1.1 | Airbus | 23 |
| 9.1.2 | ATOS / EVIDEN | 24 |
| 9.1.3 | IFPEN | 24 |
| 9.1.4 | CEA | 25 |
| 10 | Partnerships and cooperations | 25 |
| 10.1 | International initiatives | 25 |
| 10.1.1 | Horizon Europe | 25 |
| 10.2 | National initiatives | 27 |
| 10.2.1 | PEPR | 27 |
| 10.2.2 | AID | 28 |
| 10.2.3 | Inria exploratory actions | 28 |
| 10.3 | International research visitors | 28 |
| 10.3.1 | Visits of international scientists | 28 |
| 11 | Dissemination | 28 |
| 11.1 | Promoting scientific activities | 28 |
| 11.1.1 | Scientific events: organisation | 28 |
| 11.1.2 | Scientific events: selection | 29 |
| 11.1.3 | Journal | 29 |
| 11.1.4 | Invited talks | 29 |
| 11.1.5 | Research administration | 29 |
| 11.2 | Teaching - Supervision - Juries - Educational and pedagogical outreach | 30 |
| 11.2.1 | Teaching | 30 |
| 11.2.2 | Supervision | 30 |
| 11.2.3 | Juries | 31 |
| 11.3 | Popularization | 31 |
| 11.3.1 | Specific official responsibilities in science outreach structures | 31 |
| 11.3.2 | Productions (articles, videos, podcasts, serious games, ...) | 32 |
| 11.3.3 | Participation in Live events | 32 |
| 12 | Scientific production | 32 |
| 12.1 | Major publications | 32 |
| 12.2 | Publications of the year | 33 |

1 Team members, visitors, external collaborators

Research Scientists

- Olivier Aumage [Team leader, INRIA, Researcher, until Jul 2025]
- Olivier Aumage [INRIA, Researcher, from Aug 2025]
- Emmanuelle Saillard [INRIA]

Faculty Members

- Samuel Thibault [Team leader, UNIV BORDEAUX, Professor, from Aug 2025, HDR]
- Marie-Christine Counilh [UNIV BORDEAUX, Associate Professor]
- Amina Guermouche [BORDEAUX INP]
- Raymond Namyst [UNIV BORDEAUX, Professor, HDR]
- Benjamin Negrevergne [DAUPHINE PSL, Associate Professor Delegation, from Sep 2025]
- Samuel Thibault [UNIV BORDEAUX, Professor, until Jul 2025, HDR]
- Pierre-André Wacrenier [UNIV BORDEAUX, Associate Professor]

PhD Students

- Vincent Alba [UNIV BORDEAUX]
- Asia Auville [INRIA]
- Albert D Aviau De Piolant [INRIA]
- Nicolas Dias [AIRBUS, CIFRE, from Apr 2025]
- Nicolas Ducarton [INRIA, from Apr 2025]
- Lise Jolicoeur [CEA, CIFRE, until Nov 2025]
- Alice Lasserre [INRIA]
- Alan Lira Nunes [UNIV BORDEAUX]
- Thomas Morin [UNIV BORDEAUX]
- Vanderlei Munhoz Pereira Filho [INRIA, from Feb 2025]
- Diane Orhan [INRIA, from Sep 2025 until Nov 2025]
- Diane Orhan [UNIV BORDEAUX, until Aug 2025]
- Lana Scravaglieri [IFPEN, CIFRE, until Oct 2025]
- Radjasouria Vinayagame [ATOS, CIFRE, until Nov 2025]

Technical Staff

- Francois Cheminade [INRIA, Engineer]
- Guillaume Doyen [INRIA, Engineer]
- Nicolas Ducarton [UNIV BORDEAUX, Engineer, until Mar 2025]
- Nathalie Furmento [CNRS, Engineer]
- Romain Lion [INRIA, Engineer, until Mar 2025]
- Joachim Rosseel [INRIA, until Aug 2025]
- Victor-Benjamin Villain [INRIA, Engineer, until Mar 2025]

Interns and Apprentices

- Giorgio Bettonte [INRIA, Intern, from Mar 2025 until Jul 2025]
- Paul Bouchaud [ENS RENNES, Intern, from Jun 2025 until Aug 2025]
- Pierre-Antoine Creton [INRIA, Intern, from May 2025 until Jul 2025]
- Léo Fremery [ENS DE LYON, Intern, from Feb 2025 until Jul 2025]
- Nathan Houalet [INRIA, Intern, from May 2025 until Jul 2025]
- Raghid Osseiran [INRIA, Intern, from May 2025 until Jul 2025]
- Flavien Romanetti [INRIA, Intern, from Sep 2025 until Sep 2025]
- Flavien Romanetti [INRIA, Intern, from Mar 2025 until Sep 2025]
- Flavien Romanetti [INRIA, Intern, from Feb 2025 until Mar 2025]

Administrative Assistants

- Ellie Correa Da Costa De Castro Pinto [INRIA]
- Anne-Lise Pernel [INRIA]

Visiting Scientists

- David Alvarez [BSC-Espagne, from May 2025 until Aug 2025]
- Mariza Ferro [UFF]
- Ali Jannesari [UNIV IOWA]

External Collaborator

- Jean-Marie Couteyen [AIRBUS, until Feb 2025]

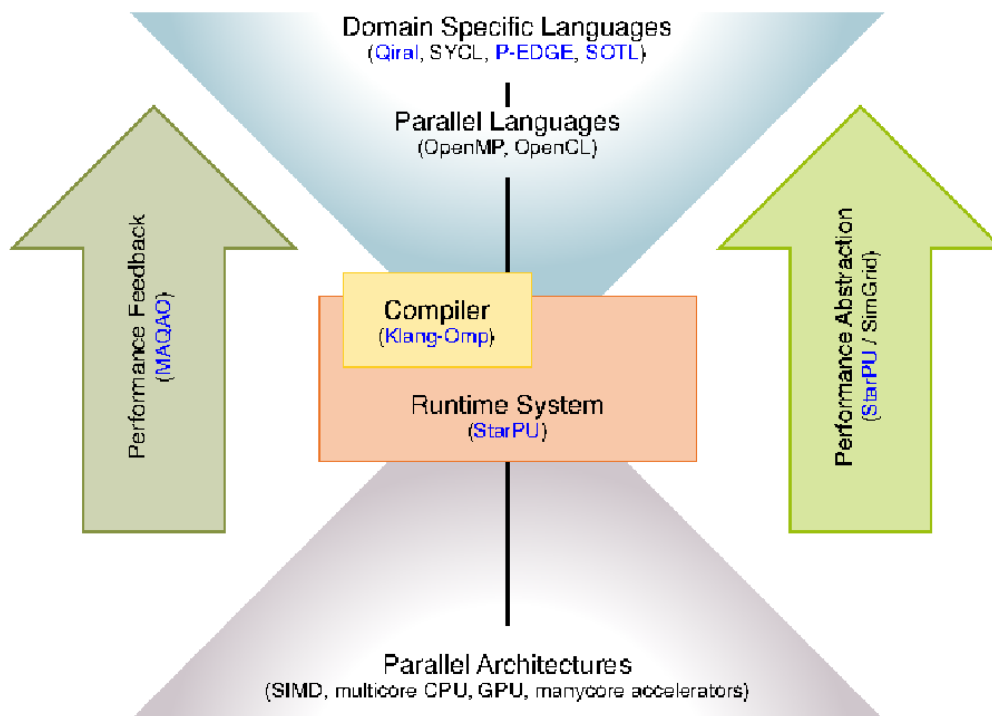


Figure 1: STORM Big Picture

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 feed back, 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 Kokkos, 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.

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

Exascale systems (i.e. Sustaining 10^{18} flops), such as the Jupiter Booster located in Jülich (DE), have appeared at the top of supercomputers, include millions of cores, and the future post-exascale platforms will aggravate this trend. To feed all computing units while hiding memory latencies on such systems, an overall concurrency level of $O(10^9)$ threads/tasks is required. It is obviously 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-2 projects [10.1.1](#))
- Software infrastructures for Telecommunications (see AFF3CT [10.2.2](#))
- Aeronautics (collaboration with Airbus, J.-M. Couteyen, MAMBO project [9.1.1](#))
- CO2 storage (collaboration with IFPEN, see [9.1.3](#))

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 Social and environmental responsibility

5.1 Footprint of research activities

- Nathalie Furmento and Amina Guermouche have been actively involved in the creation of the new GDR C4P (Calcul Paradigms, Parallelism, Performance, Precision), led by Alfredo Buttari and Théo Mary. This initiative explicitly integrates social and environmental responsibility into high-performance computing research. Within this framework, Nathalie Furmento will be responsible for the Software and Applications Working Group, with a focus on sustainable software practices and long-term impact. Amina Guermouche will lead the “Eco-responsible Computing” Working Group, which addresses energy efficiency, environmental impact, and responsible use of computational resources. Although the creation of the GDR has not yet been formally announced, it has already received official approval from the CNRS.
- Samuel Thibault is a member of the “Source Code and software group” of the national Committee for OpenScience (CoSO). He is involved in the working group for highlighting research software production, which produced a [report on a software catalog](#) and a [beta version of the catalog](#). He is involved in the working group for tools and recommended technical and social practices , which produced a [report on software forges](#).

5.2 Impact of research results

One of the main research axis of the team is energy efficiency. As a matter of fact, we designed tools and scheduling algorithms to reduce the energy consumption of HPC platforms. We also relied on AI to configure architectures in order to get the best energy efficiency. Finally, we are the co-leaders of the energy working group within the NumPEX PEPR. As such, we will organize a summer school for the all the project members where we will show how we can measure the energy consumed by a platform and what we can do about it.

An important socio-economic impact of our research activities corresponds to HPC and higher education. The EasyPAP project makes HPC more accessible to students, and contributes to its adoption with a larger audience. EasyPAP is actively used to train over 400 students each year across multiple French academic programs, including undergraduate, Master’s, and doctoral-level courses at multiple Universities (Bordeaux, Paris Sorbonne, Orléans) and at several engineering schools (ENSEIRB, Télécom SudParis, IMT Atlantique).

It is also used for science outreach, reaching around 40 middle and high school students annually through workshops.

Emmanuelle Saillard is responsible of the popularization activities for the Inria Center at the University of Bordeaux since 2021. She co-created new outreach format (e.g., 1 minute avec..., Désassemblons le numérique) and has been co-organizing the doctoral training program on outreach activities for Inria Center at the University of Bordeaux since 2024. She is also a member of the SIF (Société Informatique de France) executive board, and the Blaise Pascal Fondation scientific board.

6 Highlights of the year

6.1 Awards

- Diane Orhan got the Best Paper Award at the HCW'25 workshop [17].
- Lana Scravaglieri got the Best Open-Source Contribution Award at the IPDPS'25 conference [20].

6.2 Animation of the scientific community

- Nathalie Furmento and Amina Guermouche have been involved in the creation of the new GDR C4P (Calcul, Paradigmes, Parallélisme, Performance, Précision), lead by Alfredo Buttari and Théo Mary. Nathalie Furmento will be responsible for the Software and Application WG. Amina Guermouche will be responsible for the WG “calcul éco-responsable”.

The creation of the GDR is not yet formally announced but the CNRS has already given its green light.

7 Latest software developments, platforms, open data

7.1 Latest software developments

7.1.1 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

7.1.2 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 hardwares 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>

Publications: [hal-03882459](#), [hal-03374614](#), [hal-04320261](#), [hal-00920901](#), [hal-01078762](#), [hal-01078759](#), [hal-01252321](#), [hal-01253204](#), [hal-01199718](#), [hal-01420655](#), [hal-01937316](#), [hal-02390025](#)

Contact: Emmanuelle Saillard

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

7.1.3 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, Olivier Aumage, an anonymous participant

Partner: LIP6

7.1.4 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.

URL: <https://benchmark-subsetting.github.io/cere/>

Contact: Mihail Popov

Participant: Mihail Popov

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

7.1.5 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

Participant: Amina Guermouche

7.1.6 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

7.1.7 EasyPAP

Name: easyPAP

Keyword: Log visualisation

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/>

Publications: [hal-03126887](#), [hal-03938420](#)

Contact: Raymond Namyst

Participants: Raymond Namyst, Pierre Wacrenier, Alice Lasserre

7.1.8 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, hal-05147860, hal-05199066, hal-05226796

Contact: Nathalie Furmento

Participants: Olivier Aumage, Nathalie Furmento, Samuel Thibault, 38 anonymous participants

7.1.9 MPI-BugBench

Name: A Framework for Assessing MPI Correctness Tools

Keyword: MPI

Functional Description: MPI-BugBench is an benchmark suite to evaluate the classification quality of MPI correctness tools. It covers various error classes in MPI while incorporating a broad range of real-world MPI usage scenarios.

Publication: [hal-04878321](#)

Contact: Emmanuelle Saillard

Participant: 5 anonymous participants

Partners: RWTH Aachen University, TU Darmstadt

7.1.10 CORHPEX

Name: COmpiler, Runtime and Hardware Parameter EXplorer

Keywords: Energy, Performance, Optimization, Design space exploration

Scientific Description: COmpiler, Runtime and Hardware Parameter EXplorer (CORHPEX), is a framework to explore performance optimization spaces for HPC applications.

Functional Description: CORHPEX enables application developers to discover the influence of configurations of hardware, compilers and run-time options on optimization targets such as performance and energy consumption by proposing the efficient collection and exploration of metrics in application design spaces.

Release Contributions: First public diffusion

URL: <https://gitlab.inria.fr/corhpex/corhpex>

Publications: [hal-05311328](#), [hal-05224515](#), [hal-04969854](#)

Contact: Mihail Popov

Participants: Mihail Popov, 4 anonymous participants

Partner: IFPEN

7.1.11 StreamPU

Name: Domain Specific Embedded Language (DSEL) and runtime system for streaming applications

Keywords: Runtime system, Streaming, Workflow, Parallelism, Task scheduling

Functional Description: StreamPU is a C++ library defining building blocks to construct chains of tasks for streaming applications, and schedule their parallel execution on multicore CPUs.

URL: <https://github.com/aff3ct/StreamPU>

Contact: Olivier Aumage

Partners: LIP6, IMS

7.2 New platforms

Participants: Nathalie Furmento.

- Nathalie Furmento is a member of the technical team of PlaFRIM, the parallel computing platform shared by Inria Center at the University of Bordeaux, the LaBRI and the IMB (mathematics laboratory of the University of Bordeaux). The platform consists of about 100 servers with different architectures (x86 and ARM) and many different generations of CPUs, often equipped with high-speed interconnects (InfiniBand or OmniPath), GPUs (NVIDIA, AMD or Intel), or large pool of memories. It is used by 500 users from the local laboratories as well as academic and industrial partners worldwide, for early HPC, IA and graphics research before moving to larger/production computing centers ([PlaFRIM website](#)).

7.3 Open data

MPI-BugBench (<https://git-ce.rwth-aachen.de/hpc-public/mpi-bugbench>, [15]) is a collection of MPI programs that can be used to evaluate the classification quality of MPI correctness tools. It covers various error classes in MPI while incorporating a broad range of real-world MPI usage scenarios.

8 New results

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

Participants: Olivier Aumage, Denis Barthou, Laércio Lima Pilla, Diane Orhan, Pierre-Antoine Creton.

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 [13], an algorithm we 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. With the internship of Pierre-Antoine Creton, we also started investigating dynamic setups where the duration of tasks in the pipeline may vary over time [27], and the mapping of work onto CPU core resources needs to adapt.

8.2 Optimization Space Exploration

Participants: Olivier Aumage, Mihail Popov, Lana Scravaglieri, Raghid Osseiran.

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 the context of Lana Scravaglieri Ph.D. thesis and in collaboration with IFP Energies nouvelles (IFPEN), we developed this research topic by focusing, in particular, on the exploration of IFPEN's carbon storage applications. To do so, we designed a general purpose application design space exploration infrastructure, CORHPEX [20], that can easily incorporate diverse optimization knobs for platform, compiler, runtime and application related settings, and investigate their relative impact on metrics such as execution time and energy consumption. We also explored multiple visualization strategies for CORPHEX's results with the internship of Raghid Osseiran. [29]

8.3 Price-performance analysis for task-based simulation in the Cloud

Participants: Olivier Aumage, Vanderlei Munhoz Pereira Filho.

In the PhD Thesis of Vanderlei Munhoz Pereira Filho in cotutella with the University of Santa Catarina in Brazil, we evaluated two N-body simulations execution on the AWS cloud service, following multiple resource allocation strategies [21]. These simulations were implemented using a task-based parallel programming model, leveraging the StarPU runtime system, to dynamically schedule computational tasks across various processing units. Our experimental results demonstrated three key findings: (1) smaller GPU-equipped instances (g6.2xlarge) achieve performance comparable to larger instances while costing approximately one-sixth the price, challenging conventional scaling assumptions for cloud-based HPC; (2) strategic GPU utilization yields up to 8.2× performance improvements over CPU-only configurations while reducing total execution costs by 24.4×; and (3) while task-based programming models effectively address network limitations through dynamic scheduling, complex tree-based algorithms like TBFMM face significant optimization challenges in cloud environments due to load balancing issues and expensive parameter tuning requirements. These findings provide practical guidance for researchers and practitioners seeking cost-effective cloud HPC deployments, demonstrating that commodity cloud infrastructures can be viable for regular computational workloads but require careful algorithmic-resource matching for optimal efficiency.

8.4 Interoperable resource sharing between multiple runtime systems

Participants: Olivier Aumage, David Alvarez.

As part of JLESC [project on task-based runtime system interoperability](#), we welcomed David Alvarez (PhD at Barcelona Supercomputing Center (BSC)) for a 3-Month internship in team STORM. We developed a port of the StarPU task-based runtime system on BSC's nOS-V runtime hypervisor. This port enable StarPU to seamlessly share CPU core resources with other nOS-V enabled runtime systems, without resulting in cores over-subscription or under-subscription. We explored two strategies for the port. A lightweight port where StarPU's scheduling engine cooperates with nOS-V hypervizing scheduler, and a more deeply integrated port where StarPU's tasks are directly scheduled by nOS-v's scheduler.

8.5 Portable EMI cardiac electrophysiology simulation

Participants: Olivier Aumage, Giorgio Bettonte.

We built on the work initiated last year to develop a port of Simula's DEMI cardiac electrophysiology simulation code on top of StarPU's parallel and distributed programming model [30]. While the initial DEMI application is divided into multiple code variants for supporting single-node parallel execution and multi-node distributed execution, the port of StarPU enables a single shared code base to target both execution models simultaneously, in a portable manner.

8.6 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 published the results in JPDC [12].

8.7 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. We propose a new policy for the Splitter, which is designed for heterogeneous platforms, that relies on linear programming aimed at minimizing 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 was published in the JPDC journal [11]. We have then introduced another strategy, GASPP, that overcomes the limitations of the linear programming approach. It is a greedy approach that leverages runtime predictions and integrates scheduling constraint to ensure effective

CPU/GPU utilization. Its performance results surpass the linear programming approach while exhibiting much lower overhead. This work is currently submitted for IPDPS' HCW'26 workshop.

8.8 C++ interfacing with StarPU

Participants: Olivier Aumage, Paul Bouchaud.

Paul Bouchaud's internship did a preliminary investigation of the interfacing of StarPU's programming model with modern C++ (e.g C++ 20 and more recent) [25]. We used FEniCSx finite element programming environment as target applicative code, with a focus on the finite element assembly step, and the usage of C++'s `span` and `mdspan` objects.

8.9 Fault-Tolerance for task-based applications on large-scale systems

Participants: Nicolas Ducarton, Amina Guermouche, Thomas Hérault, Samuel Thibault.

Since supercomputers keep growing in terms of core numbers, the reliability decreases the same way. In the context of the NUMPEX Exa-Soft projects, the PhD Thesis of Nicolas Ducarton aims to propose solutions for the failure tolerance problem in the particular context of task-based runtime systems such as StarPU. We had previously identified properties in our task-based runtime's paradigm that can be exploited in order to propose a completely asynchronous checkpoint solution with local restart thanks to message logging which exhibits lower overhead than the generic existing ones. During 2025, we have finished putting the theory into practice, which now allows to seamlessly restart a StarPU application on MPI node failure. We will present the general approach and the requirement of MPI communicator replacement at the WAMTA'26 workshop. We will pursue with redefining the effectiveness of such approach, since without a global synchronization point for checkpoints, the Young/Daly principle does not hold any more.

8.10 Integration of asynchronous network communications scheduling and local task scheduling

Participants: Tristan Riehs, Alexandre Denis, Philippe Swartvagher, Samuel Thibault.

Runtime systems for heterogeneous distributed systems include two essential parts: scheduling computation tasks and scheduling network communications. The latter is essentially always hidden behind the MPI programming interface, so that optimization is performed on both sides, with low transfer of information between the two. The StarPU task-based runtime system includes a NewMadeleine communication driver, which allows to reach integration of the two beyond what the MPI interface provides. We have notably shown [23] that StarPU can then provide NewMadeleine with a crucial information: the future. When StarPU starts a task whose output will have to be sent to the network, it notifies NewMadeleine about the upcoming communication, so that the latter can prepare both emission and reception buffers. This entails that the receiving system can allocate reception buffers on-the-fly a few milliseconds before data is received from the network. This allows to much better re-use data buffers, and control overall memory allocation. The NewMadeleine packet scheduler will also be able to anticipate future communications and take better advantage of communication priorities.

8.11 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. Jean-François defended his PhD thesis [24]

8.12 A Machine-Learning Approach to MPI Error Detection

Participants: Asia Auville, Mihail Popov, Emmanuelle Saillard.

MPI errors are challenging to identify despite the significant number of expert verification tools. Dynamic tools (i.e., requiring profiling) are computationally expensive and accurate in error detection, whereas static analysis (i.e., operating at source code or compilation) is computationally cheap but less accurate. Interestingly, the recent success of AI and LLMs offers an alternative to increase static analysis accuracy while preserving its low overhead. Yet current models remain too general and poorly adapted to the specific challenges of high-performance computing. In the context of Asia Auville Ph.D. thesis [22], we are currently investigating how AI-powered tools can efficiently and accurately detect errors in real-world MPI applications. We propose a novel MPI Mutated Dataset (MMD), constructed from MPI programs extracted from thousands of open-source GitHub projects. After sorting and filtering these files, we inject errors that realistically emulate developers' mistakes using synthetic code mutations. We leverage the dataset to train different AI models and assess their generalization capabilities against standard verification tools. This work is done in collaboration with the University of Versailles and Intel.

8.13 One-Sided Communications Automatic Rewriting

Participants: Emmanuelle Saillard, Samuel Thibault, Radjasouria Vinayagame.

The Message Passing Interface (MPI) provides communication routines named MPI One-Sided Communications (OSC) with which a process can access the memory space of another process without requiring any action from the latter. Thanks to these operations, applications can improve the overlap of communications with computations and thus achieve better performance. However, OSC are complex to write because they bring constraints on memory consistency and synchronizations that are trickier than with two-sided communications. In the context of Radjasouria's PhD, we have developed OSCAR, a solution to help developers migrate toward the use of one-sided communications. OSCAR automatically and statically transforms two-sided communications into their one-sided counterparts, using the LLVM compilation infrastructure.

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

Participants: Lise Jolicoeur, Raymond Namyst.

The diversification of high performance computing (HPC) workloads and the development of heterogeneous workflows involving HPC, artificial intelligence and machine learning (AI/ML), as well as in-situ analysis, have challenged the typical architecture of HPC clusters. Modern HPC workflows increasingly rely on services and cloud-native applications that are not typically supported on HPC environments. Cloud computing has historically been the platform of choice for running services, leading to the development of a rich ecosystem of cloud-native software. At the same time, it is increasingly supporting HPC applications by offering HPC-grade hardware and batch scheduling through either standard HPC or cloud-native tools. Driven by the converging needs of both communities, there is a growing interest in a "best of both worlds" architecture that supports both HPC and cloud-native applications within the same environment, without compromising on performance, security, and usability. During the last year of her PhD, Lise has studied a new approach Towards secure cluster architectures for HPC workflows based on deploying on-demand Kubernetes clusters on HPC resources offering different tradeoffs in terms of resource management flexibility and supported Kubernetes capabilities. This work is a collaboration with Daniel Milroy and Vanessa Sochat from the Lawrence Livermore National Laboratory (LLNL) [16].

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

Participants: Léo Fremery, 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.

In collaboration with Airbus and NVIDIA, we have extended their C++ CFD application demonstrator to enable automatic GPU offload using the NVIDIA NVHPC compiler. While being an elegant solution to the problem of porting complex codes on GPUs, the approach still need further optimization to remove unnecessary synchronizations between kernels.

8.16 Highlighting EasyPAP Improvements

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

In 2025, the EasyPAP educationnal environment has been extended along two major directions. First, the code structure was completely revised to extract two standalone libraries: EasyVisualization (EZV) and EasyMonitoring (EZM). The APIs of these libraries are simple enough to easily instrument existing parallel codes based on Pthreads, OpenMP, TBB, Kokkos, MPI, OpenCL, CUDA, or a combination of any. It was notably used to monitor a acoustic wave simulation code developed by Airbus. Second, we have leveraged the functionalities of the Hardware Locality software to display the mapping of threads on the underlying computing units, both in the real time monitor and in the trace visualizer. Among others, this enables users to verify that threads are correctly bound to resources in accordance with the definitions of execution places and processor bindind directives.

8.17 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 executing these kernels while taking into account variations in kernel execution time and heterogeneous processing unit speeds, is crucial for overall simulation performance. Ensuring that this execution strategy adapts automatically to the underlying hardware architecture is important to ensure performance portability.

To address these challenges, this work [9] introduces a method for efficiently distributing ionic model kernels across heterogeneous resources, guided by a resource dimensioning heuristic that adapts to each model's computational profile. These mechanisms are integrated into `OPENCARP` and evaluated on 30 representative ionic models, with a focus on both performance and energy efficiency. We demonstrate that on a node with 8 GPUs, our method achieves a geometric mean speedup of 1.45 compared to using all GPUs, while also improving energy efficiency.

8.18 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 [18].

8.19 Fine-grain energy consumption modeling of HPC task-based programs

Participants: Jules Risse, Amina Guermouche, François Trahay.

Monitoring the energy consumption of HPC programs is a good first step to reduce the power consumption of HPC applications: using external or software power meters, one can measure the energy consumption of an entire compute node or some of its hardware components. Unfortunately, the differences in scope and time scale between power meters and code level functions prevent the identification of power hungry code blocks. For this work, we propose leveraging the tracing mechanism of the StarPU runtime system in order to estimate task level power consumption. We trace the execution of the application while regularly measuring coarse-grain energy consumption of central processing units (CPUs) and graphics processing units (GPUs) using vendor software interfaces. After execution, we identify the executed tasks on each processing unit for every coarse-grain energy measurement interval. We then use this information to generate an overdetermined linear system linking tasks and energy measurements. Subsequently, solving the system allows us to estimate

the fine-grain power consumption of each task independently of its actual duration. We achieve mean average percentage errors (MAPE) ranging from 0.5 on GPUs. We show that a solution generated from a run can be used to predict the energy consumption of other runs with different scheduling policies.[19]

8.20 High-level Python programming interface for StreamPU

Participants: Olivier Aumage, Flavien Romanetti.

Flavien Romanetti's development internship in the team [31], in collaboration with Romain Tajan from IMS laboratory, was dedicated to bring the new Python programming interface of StreamPU ready for its upcoming diffusion on the Python Package Index (pypi.org), in particular by enabling the support from the Python API for StreamPU's `stateful` and `stateless` class inheritance paths. A demonstrator code for the ADS-B (Automatic Dependent Surveillance — Broadcast) civil aircraft identification protocol reception was also developed by Flavien Romanetti as a testcase for StreamPU's Python programming interface.

8.21 Code-based post-quantum cryptographical schemes in AFF3CT

Participants: Olivier Aumage, Victor-Benjamin Villain.

Quantum computers bring the prospect of breaking classical asymmetrical cryptography schemes such as those based on the factorization of integers in large prime number pairs since the development of the Shor algorithm in 1994. Post-quantum cryptographical schemes aim at proactively addressing this evolution by being robust to quantum computers specific properties. Multiple approaches have been explored by the cryptographical community to design such schemes, and a family of them relies on code theory. Building on the relationship of this family with error correction codes, with Andrea Lesavourey from XLIM laboratory in Limoges, we have implemented a selection of three code-based schemes, Classic McEliece, BIKE and Hamming Quasi-Cyclic (HQC), in AFF3CT (`libpqc`) among those proposed by the community. Their availability in AFF3CT make them interoperable with the other numerical communication building blocks and modules offered by the library.

8.22 5G Physical Broadcast Channel in AFF3CT

Participants: Olivier Aumage, Joachim Rosseel.

We have developed with the cooperation of Romain Tajan from IMS laboratory a prototype implementation of the 5G wireless communication standard PBCH channel (Physical Broadcast Channel) in AFF3CT. The PBCH channel enables user devices to synchronize with the base station and is therefore critical in the 5G standard architecture. The AFF3CT implementation focusses in particular on the processing of the Synchronization Signal Block (SSB), using the primary and secondary signals (PSS and SSS) to synchronize and equalize the received frames to correct the effects of the transmission channel such as noise, frequency shifts and delays. It serves as an example for the OFDM (orthogonal frequency-division multiplexing) modulation / demodulation scheme, on which the PBCH channel is based.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

9.1.1 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.

9.1.2 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.

9.1.3 IFPEN

Participants: Olivier Aumage, Mihail Popov, Lana Scraglieri.

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 parameters (e.g., degree of parallelism, simd instructions, compiler optimizations) during an execution.

9.1.4 CEA

Participants: Raymond Namyst.

CEA STORM has a long-standing collaboration record with CEA/DAM (Military Applications). We have conducted research on designing secured and flexible Cloud deployment facilities for HPC clusters (PhD of L. Jolicoeur [16], 2022-25).

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Horizon Europe

EoCoE-III [EoCoE-III project on cordis.europa.eu](https://cordis.europa.eu)

Title: FOSTERING THE EUROPEAN ENERGY TRANSITION WITH EXASCALE

Duration: From January 1, 2024 to December 31, 2026

Partners:

- DATADIRECT NETWORKS FRANCE, France
- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- UNIVERSITA DEGLI STUDI DI ROMA TOR VERGATA (UNITOV), Italy
- FRIEDRICH-ALEXANDER-UNIVERSITAET ERLANGEN-NUERNBERG (FAU), Germany
- FORSCHUNGSZENTRUM JULICH GMBH (FZJ), Germany
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), France
- CENTRO DE INVESTIGACIONES ENERGETICAS MEDIOAMBIENTALES Y TECNOLOGICAS (CIEMAT), Spain
- INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK, Poland
- UNIVERSITE LIBRE DE BRUXELLES (ULB), Belgium
- AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE (ENEA), Italy
- CENTRE EUROPEEN DE RECHERCHE ET DEFORMATION AVANCEE EN CALCUL SCIENTIFIQUE (CERFACS), France
- E 4 COMPUTER ENGINEERING SPA (E4), Italy
- CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), Italy
- UNIVERSITA DEGLI STUDI DI TRENTO (UNITN), Italy
- IFP Energies nouvelles (IFPEN), France

- MAX-PLANCK-GESELLSCHAFT ZUR FORDERUNG DER WISSENSCHAFTEN EV (MPG), Germany
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (CNRS), France
- BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION (BSC CNS), Spain

Inria contact: Bruno Raffin

Coordinator:

Summary: The Energy-oriented Centre of Excellence for exascale HPC applications (EoCoE-III) applies cutting-edge computational methods in its mission to foster the transition to decarbonized energy in Europe. EoCoE-III is anchored both in the High Performance Computing (HPC) community and in the energy field. It will demonstrate the benefit of HPC for the net-zero energy transition for research institutes and also for key industry in the energy sector. The present project will draw the experience of two successful previous projects EoCoE-I and EoCoE-II, where a set of diverse computer applications from four energy domains achieved significant efficiency gains thanks to its multidisciplinary expertise in applied mathematics and supercomputing. During this 3rd round, EoCoE-III will channel its efforts into 5 exascale lighthouse applications covering the key domains of Energy Materials, Water, Wind and Fusion. A world-class consortium of 18 complementary partners from 6 countries will form a unique network of expertise in energy science, scientific computing and HPC, including 3 leading European supercomputing centres. This multidisciplinary effort will harness innovations in computer science and mathematical algorithms within a tightly integrated co-design approach to overcome performance bottlenecks, to deploy the lighthouse applications on the coming European exascale infrastructure and to anticipate future HPC hardware developments. New modelling capabilities will be created at unprecedented scale, demonstrating the potential benefits to the energy industry, such as accelerated design of photovoltaic devices, high-resolution wind farm modelling over complex terrains and quantitative understanding of plasma core-edge interactions in ITER-scale tokamaks. These lighthouse applications will provide a high-visibility platform for high-performance computational energy science, cross-fertilized through close working connections to the EERA consortium.

MICROCARD-2 [MICROCARD-2 project on cordis.europa.eu](https://cordis.europa.eu)

Title: Numerical modeling of cardiac electrophysiology at the cellular scale

Duration: From November 1, 2024 to April 30, 2027

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- MEGWARE COMPUTER VERTRIEB UND SERVICE GMBH, Germany
- TECHNISCHE UNIVERSITAET MUENCHEN (TUM), Germany
- SIMULA RESEARCH LABORATORY AS, Norway
- UNIVERSITE DE STRASBOURG (UNISTRA), France
- ZUSE-INSTITUT BERLIN (ZUSE INSTITUTE BERLIN), Germany
- KARLSRUHER INSTITUT FUER TECHNOLOGIE (KIT), Germany
- UNIVERSITE DE BORDEAUX (UBx), France
- UNIVERSITA DEGLI STUDI DI PAVIA (UNIPV), Italy
- UNIVERSITA DEGLI STUDI DI TRENTO (UNITN), Italy

Inria contact: Olivier Aumage

Coordinator:

Summary: Cardiac function is coordinated by an electric system whose disorders are among the most frequent causes of death and disease. Numerical models of this complex system are mature and widely used, but to match observations in aging and diseased hearts they need to move from a continuum approach to a representation of individual cells and their interconnections. This makes the problem more complex, harder to solve, and four orders of magnitude larger, necessitating exascale computers.

The EuroHPC-2019 MICROCARD project is developing a simulation platform that can meet this challenge, by a joint effort of HPC experts, numerical scientists, biomedical engineers, and biomedical scientists, from academia and industry. Our proposal is to establish a Centre of Excellence that will consolidate and scale up the MICROCARD results enabling digital twins of cardiac tissue.

With a consortium gathering the core partners of MICROCARD, we will further develop MICROCARD's numerical schemes, moving to second-order spatial discretization. Based on MICROCARD results, we will develop mixed-precision preconditioners and data compression to reduce communication bandwidth. The highly successful efforts towards automated compilation of high-level model descriptions into optimized, energy-efficient system code for different CPUs and GPUs will be extended to upcoming architectures. We will continue efforts to robustify parallel remeshing software and add necessary functionality for parallel mesh partitioning and production of realistic synthetic tissue meshes needed for simulations.

The platform will be benchmarked with realistic test cases and be made accessible for a wide range of users with tailored workflows.

The platform will be adaptable to similar biological systems such as nerves, and several of our products such as improved solvers, preconditioners, remeshers, and partitioners will be reusable in a wide range of applications.

10.2 National initiatives

10.2.1 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.

10.2.2 AID AID AFF3CT

Participants: Olivier Aumage, François Cheminade, Diane Orhan, Joachim Rosseel, Victor-Benjamin Villain.

- 2023 - 2025 (24 months)
- Coordinator: Olivier Aumage
- Other partners: 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.

10.2.3 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.

10.3 International research visitors

10.3.1 Visits of international scientists

Other international visits to the team David Alvarez, PhD student from BSC, visited STORM for 3 months in 2025 to work on the interfacing of StarPU with the nOS-V runtime hypervisor.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

Member of the organizing committees

- Emmanuelle Saillard: C3PO @ISC'25
- Olivier Aumage: AFF3CT User Day 2025.

11.1.2 Scientific events: selection

Member of the conference program committees

- Samuel Thibault: Euro-Par'25, IPDPS'25, HiPC'25, HCW'25
- Emmanuelle Saillard: BoF @SC'25
- Amina Guermouche: CCGrid'25
- Olivier Aumage: ExHET'25, LLM4HPC, ICPP25

Reviewer The team participates to reviewing in various conferences, among which SuperComputing, Euro-Par, IPDPS, HiPC, HCW, ...

11.1.3 Journal

Member of the editorial boards

- Samuel Thibault: JPDC Associate Editor

Reviewer - reviewing activities

- Samuel Thibault: FGCS, IJHPCA, PARCO
- Olivier Aumage: JPDC, SoftwareX

11.1.4 Invited talks

- Samuel Thibault: J3 Fortran: Tasking Discussions, online, may 2025.
- Olivier Aumage: FOSDEM'25 DevRoom Radio, Brussels, Feb. 2025. Talk on TaskStubs task tracing framework at Parallel Tools Workshop at HLRS, Stuttgart, Nov. 2025.

11.1.5 Research administration

- 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
- Emmanuelle Saillard is a member of the Commission de délégation at Inria Research Centre of the University of Bordeaux.
- Samuel Thibault is head of LaBRI's STORM team, and thusly member of the scientific council of the LaBRI

11.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

11.2.1 Teaching

- Academic Teaching
 - Engineering School: Emmanuelle Saillard, Languages of parallelism, 12HeC, M2, 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: 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.
 - 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.

11.2.2 Supervision

- PhD defended:
 - Jean-François David, Dynamic scheduling for inference in deep neural networks. Advisors: Olivier Beaumont, Samuel Thibault, and Lionel Eyraud-Dubois.
 - Radjasouria Vinayagame, Optimization of porting and performance of HPC applications with distributed and globally addressed memory. Advisors: Emmanuelle Saillard and Samuel Thibault.
 - 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).
 - Diane Orhan, Modeling and dynamic optimization of software radio chains on heterogeneous architectures. Advisors: Denis Barthou and Christophe Jégo.
 - Lise Jolicoeur, Towards architectures of secured clusters for HPC workflow execution. Advisors: Raymond Namyst and François Diakhate (CEA).
 - Vincent Alba, "Task scheduling for exascale". Advisors: Denis Barthou and Amina Guermouche.
- PhD in progress:
 - Asia Auville, Large Language Models for Detection and Correction of Errors in HPC Applications. Advisors: Emmanuelle Saillard and Mihail Popov.

- Albert D’Aviau de Piolant, Energy aware scheduling for exascale architectures. Advisors: Abdou Guermouche and Amina Guermouche.
- Nicolas Dias Hybrid CPU/GPU programming via a runtime system for complex simulations. Advisors: Raymond Namyst
- Nicolas Ducarton, Fault tolerance for task based runtime systems. Advisors : Thomas Hernaut and Samuel Thibault.
- Alice Lasserre Optimising a task-based computation code on distributed memory systems. Advisors: Raymond Namyst and Abdou Guermouche.
- Thomas Morin, Scheduling recursive task graphs. Advisors: Abdou Guermouche, Samuel Thibault, Pierre-André Wacrenier.
- Vanderlei Munhoz Pereira Filho Task-based HPC applications from supercomputers to cloud service platforms. In cotutella with the University of Santa Catarina, Brazil. Advisors : Olivier Aumage, Márcio Castro (UFSC), Laércio Lima Pilla (Team TOPAL)
- Jules Risse, Fine-grain energy consumption measurement of HPC task-based programs. Advisors: Amina Guermouche and François Trahay.

11.2.3 Juries

- Amina Guermouche
 - Member of the PhD jury of Simon Lambert, Ecole Normal Supérieure de Lyon
 - Member of the PhD jury of Sylvain Joube, University of Paris Saclay
- Raymond Namyst
 - President for the PhD of Marie Reinbigler, Institut polytechnique de Paris
 - President for the PhD of Hayfa Tayeb, University of Bordeaux
- Emmanuelle Saillard: member of the PhD jury of Nahuel Palumbo, University of Lille
- Samuel Thibault
 - Reviewer for PhD of Joseph John, Canberra (Australia)
 - Reviewer for PhD of Aymeric Pablo Millan, University of Paris Saclay
 - President for the PhD of Atte Johannes Torri, University of Paris Saclay
 - Member of the PhD jury of Louis boulanger, University of Grenoble Alpes

11.3 Popularization

11.3.1 Specific official responsibilities in science outreach structures

- Emmanuelle Saillard, Raymond Namyst: Organization of Moi Informaticienne - Moi Mathématicienne, April 2025.
- 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)

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

- Emmanuelle Saillard:
 - Elles font le numerique #8: [link](#)
 - La grande muraille d'Égypte. 1024 : Bulletin de la Société Informatique de France, 2025 [32]
- Pierre-André Wacrenier, Raymond Namyst: Les supports d'exécution pour le calcul haute performance, in *Le calcul à découvert*, 2025, [33]

11.3.3 Participation in Live events

- Emmanuelle Saillard
 - Participation at the "Circuit scientifique Bordelais", Inria, Oct. 2025: ("La grande muraille d'Égypte")
 - Participation at the "Nuit européenne de la recherche", Cap Science, Sept. 2025 (speedsearching)
- Olivier Aumage
 - Session Chiche! at Val de Garonne high school in Marmande for two classes, Fev. 2025.
- Marie-Christine Counilh, Nathalie Furmento, Mihail Popov, Pierre-André Wacrenier: Half-day supervision of high-school students during a practical session on HPC, June 2025

12 Scientific production

12.1 Major publications

- [1] V. Alba, O. Aumage, D. Barthou, M.-C. Counilh and A. Guermouche. 'Performance portability of generated cardiac simulation kernels through automatic dimensioning and load balancing on heterogeneous nodes'. In: *Journal of Supercomputing* 81.9 (18th June 2025), p. 1047. DOI: [10.1007/s11227-025-07510-5](https://doi.org/10.1007/s11227-025-07510-5). URL: <https://inria.hal.science/hal-05235567>.
- [2] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, R. Namyst, S. Thibault and P.-a. Wacrenier. 'Programming Heterogeneous Architectures Using Hierarchical Tasks'. In: *Concurrency and Computation: Practice and Experience* 35.25 (2023). DOI: [10.1002/cpe.7811](https://doi.org/10.1002/cpe.7811). URL: <https://hal.science/hal-04088833>.
- [3] N. Furmento, A. Guermouche, G. Lucas, T. Morin, S. Thibault and P.-A. Wacrenier. 'Optimizing Parallel Heterogeneous System Efficiency: Dynamic Task Graph Adaptation with Recursive Tasks'. In: *Journal of Parallel and Distributed Computing* 205 (16th June 2025), p. 105157. DOI: [10.1016/j.jpdc.2025.105157](https://doi.org/10.1016/j.jpdc.2025.105157). URL: <https://hal.science/hal-05199066>.
- [4] M. Gonthier, L. Marchal and S. Thibault. 'A scheduler to foster data locality for GPU and out-of-core task-based linear algebra applications'. In: *Journal of Parallel and Distributed Computing* 206 (11th Aug. 2025), p. 105170. DOI: [10.1016/j.jpdc.2025.105170](https://doi.org/10.1016/j.jpdc.2025.105170). URL: <https://inria.hal.science/hal-05226796>.
- [5] L. Jolicoeur, V. Sochat, F. Diakhaté and D. Milroy. 'Enabling RDMA and GPUs in Rootless Kubernetes for Accelerated HPC and AI Applications'. In: *VHPC 2025 - 20th Workshop on Virtualization in High-Performance Cloud Computing (held in conjunction with Europ-Par 2025 - the European Conference on Parallel and Distributed Computing)*. Dresden, Germany, 2025. URL: <https://inria.hal.science/hal-05236403>.
- [6] D. Orhan, L. Lima Pilla, D. Barthou, A. Cassagne, O. Aumage, R. Tajan, C. Jégo and C. Leroux. 'Optimal Scheduling Algorithms for Software-Defined Radio Pipelined and Replicated Task Chains on Multicore Architectures'. In: *Journal of Parallel and Distributed Computing* (2025), p. 105106. DOI: [10.1016/j.jpdc.2025.105106](https://doi.org/10.1016/j.jpdc.2025.105106). URL: <https://hal.science/hal-04228117>. In press.

- [7] 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'. In: HCW (Ipdps workshop). Milan (Italie), Italy, 2nd June 2025. URL: <https://inria.hal.science/hal-04883872>.
- [8] L. Scravaglieri, A. Anciaux-Sedrakian, O. Aumage, T. Guignon and M. Popov. 'Compiler, Runtime, and Hardware Parameters Design Space Exploration'. In: IPDPS 2025 - 39th IEEE International Parallel and Distributed Processing Symposium. Milan, Italy, Feb. 2025. URL: <https://inria.hal.science/hal-04969854>.

12.2 Publications of the year

International journals

- [9] V. Alba, O. Aumage, D. Barthou, M.-C. Counilh and A. Guermouche. 'Performance portability of generated cardiac simulation kernels through automatic dimensioning and load balancing on heterogeneous nodes'. In: *Journal of Supercomputing* 81.9 (18th June 2025), p. 1047. DOI: [10.1007/s11227-025-07510-5](https://doi.org/10.1007/s11227-025-07510-5). URL: <https://inria.hal.science/hal-05235567> (cit. on p. 22).
- [10] O. Beaumont, R. Bouzel, L. Eyraud-Dubois, E. Korkmaz, L. L. Pilla and A. van Kempen. 'Approximation Algorithms for Scheduling with/without Deadline Constraints where Rejection Costs are Proportional to Processing Times'. In: *IEEE Transactions on Parallel and Distributed Systems* 36.12 (Dec. 2025), pp. 2596–2608. DOI: [10.1109/TPDS.2025.3605674](https://doi.org/10.1109/TPDS.2025.3605674). URL: <https://hal.science/hal-04745701>.
- [11] N. Furmento, A. Guermouche, G. Lucas, T. Morin, S. Thibault and P.-A. Wacrenier. 'Optimizing Parallel Heterogeneous System Efficiency: Dynamic Task Graph Adaptation with Recursive Tasks'. In: *Journal of Parallel and Distributed Computing* 205 (16th June 2025), p. 105157. DOI: [10.1016/j.jpdc.2025.105157](https://doi.org/10.1016/j.jpdc.2025.105157). URL: <https://hal.science/hal-05199066> (cit. on p. 18).
- [12] M. Gonthier, L. Marchal and S. Thibault. 'A scheduler to foster data locality for GPU and out-of-core task-based linear algebra applications'. In: *Journal of Parallel and Distributed Computing* 206 (11th Aug. 2025), p. 105170. DOI: [10.1016/j.jpdc.2025.105170](https://doi.org/10.1016/j.jpdc.2025.105170). URL: <https://inria.hal.science/hal-05226796> (cit. on p. 18).
- [13] D. Orhan, L. Lima Pilla, D. Barthou, A. Cassagne, O. Aumage, R. Tajan, C. Jégo and C. Leroux. 'Optimal Scheduling Algorithms for Software-Defined Radio Pipelined and Replicated Task Chains on Multicore Architectures'. In: *Journal of Parallel and Distributed Computing* (2025), p. 105106. DOI: [10.1016/j.jpdc.2025.105106](https://doi.org/10.1016/j.jpdc.2025.105106). URL: <https://hal.science/hal-04228117>. In press (cit. on p. 17).

International peer-reviewed conferences

- [14] S. Humenda, S. Thibault and H. Schirmeier. 'Screen Readers – Out of Sight, but in the TCB'. In: *Tagungsband des FG-BS Herbsttreffens. 2025 - Herbsttreffen der Fachgruppe Betriebssysteme*. Aachen, Germany: Gesellschaft für Informatik e.V., 2025. DOI: [10.18420/fgbs2025h-03](https://doi.org/10.18420/fgbs2025h-03). URL: <https://hal.science/hal-05273382>.
- [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> (cit. on p. 16).
- [16] L. Jolicoeur, V. Sochat, F. Diakhaté and D. Milroy. 'Enabling RDMA and GPUs in Rootless Kubernetes for Accelerated HPC and AI Applications'. In: *VHPC 2025 - 20th Workshop on Virtualization in High-Performance Cloud Computing (held in conjunction with Europ-Par 2025 - the European Conference on Parallel and Distributed Computing)*. Dresden, Germany, 2025. URL: <https://inria.hal.science/hal-05236403> (cit. on pp. 21, 25).

- [17] D. Orhan, Y. Idouar, L. Lima Pilla, A. Cassagne, D. Barthou and C. Jégo. ‘Scheduling Strategies for Partially-Replicable Task Chains on Two Types of Resources’. In: 2025 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Milano, Italy: IEEE, 3rd June 2025, pp. 896–905. DOI: [10.1109/IPDPSW66978.2025.00140](https://doi.org/10.1109/IPDPSW66978.2025.00140). URL: <https://hal.science/hal-04941123> (cit. on p. 11).
- [18] 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’. In: HCW (Ipdps workshop). Milan (Italie), Italy, 2nd June 2025. URL: <https://inria.hal.science/hal-04883872> (cit. on p. 22).
- [19] J. Risse, A. Guermouche and F. Trahay. ‘Fine-grain energy consumption modeling of HPC task-based programs’. In: *CLUSTER 2025: IEEE International Conference on Cluster Computing*. IEEE International Conference on Cluster Computing (CLUSTER). Edimbourg, United Kingdom: IEEE, 7th Oct. 2025. DOI: [10.1109/CLUSTER59342.2025.11186478](https://doi.org/10.1109/CLUSTER59342.2025.11186478). URL: <https://hal.science/hal-05200287> (cit. on p. 23).
- [20] L. Scravaglieri, A. Anciaux-Sedrakian, O. Aumage, T. Guignon and M. Popov. ‘Compiler, Runtime, and Hardware Parameters Design Space Exploration’. In: IPDPS 2025 - 39th IEEE International Parallel and Distributed Processing Symposium. Milan, Italy, Feb. 2025. URL: <https://inria.hal.science/hal-04969854> (cit. on pp. 11, 17).
- [21] N. Vanz, V. Munhoz, M. Castro, L. Lima Pilla and O. Aumage. ‘Task-Based HPC in the Cloud: Price-Performance Analysis of N-Body Simulations with StarPU’. In: IC2E 2025 - 13th IEEE International Conference on Cloud Engineering. Rennes, France, 23rd Sept. 2025. URL: <https://inria.hal.science/hal-05235814> (cit. on p. 17).

Conferences without proceedings

- [22] A. Auville. ‘Un modèle d’intelligence artificielle pour détecter et corriger des erreurs dans du code MPI’. In: COMPAS 2025 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Bordeaux, France, 24th June 2025. URL: <https://hal.science/hal-05139769> (cit. on p. 20).
- [23] T. Chatelain. ‘Anticipation des communications réseau grâce à la connaissance du futur dans le parallélisme à tâches’. In: COMPAS 2025 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Bordeaux, France, 24th June 2025. URL: <https://hal.science/hal-05147860> (cit. on p. 19).

Doctoral dissertations and habilitation theses

- [24] J.-F. David. ‘Dynamic scheduling for deep neural network inference’. Université de Bordeaux, 28th Mar. 2025. URL: <https://theses.hal.science/tel-05047108> (cit. on p. 20).

Reports & preprints

- [25] P. Bouchaud. *Task-Based Runtime Support and Programming for Finite Element Simulations*. ENS de Lyon; Inria, 18th Aug. 2025. URL: <https://inria.hal.science/hal-05224253> (cit. on p. 19).
- [26] G. da Costa and A. Guermouche. *Measurement methods sheet, WG6 Exa-Soft*. Université de Toulouse; Université de bordeaux, 22nd Sept. 2025. URL: <https://hal.science/hal-05272179>.
- [27] P.-A. Creton. *Ordonnancement dynamique de chaînes de communication numérique pour la radio logicielle*. Université de Bordeaux; Inria, 15th July 2025. URL: <https://inria.hal.science/hal-05224528> (cit. on p. 17).
- [28] Y. Idouar, A. Cassagne, L. Lima Pilla, J. Sopena, M. Bouyer, D. Orhan, L. Lacassagne, D. Galayko, D. Barthou and C. Jégo. *Energy-Aware Scheduling Strategies for Partially-Replicable Task Chains on Heterogeneous Processors*. 5th Sept. 2025. URL: <https://hal.science/hal-05242657>.

- [29] R. Osseiran. *Internship report - CRPVIS, Visualization for application optimization, CORHPEX (COmpiler Runtime and Hardware Parameter EXplorer) visualization tool*. Université de Bordeaux; Inria, 11th July 2025. URL: <https://inria.hal.science/hal-05224515> (cit. on p. 17).

Other scientific publications

- [30] G. Bettonte. ‘High-performance cell-based simulations of cardiac electrophysiology using task parallelism’. University of Luxembourg; Inria, 25th Aug. 2025. URL: <https://inria.hal.science/hal-05224224> (cit. on p. 18).
- [31] F. Romanetti. ‘Passage en production et diffusion du logiciel StreamPU sur le python package index’. Bordeaux INP Enseirb - Matmeca, 17th Oct. 2025. URL: <https://inria.hal.science/hal-05311915> (cit. on p. 23).

Scientific popularization

- [32] E. Saillard. ‘La grande muraille d’Égypte’. In: *1024 : Bulletin de la Société Informatique de France* 26 (Dec. 2025), pp. 35–44. DOI: [10.48556/SIF.1024.26.35](https://doi.org/10.48556/SIF.1024.26.35). URL: <https://hal.science/hal-05417312> (cit. on p. 32).
- [33] P.-A. Wacrenier and R. Namyst. ‘Les supports d’exécution pour le calcul haute performance’. In: *Le calcul à découvert*. CNRS, 23rd Jan. 2025, pp. 113–115. URL: <https://hal.science/hal-05223927> (cit. on p. 32).