

RESEARCH CENTRE

**Inria Centre
at the University of Bordeaux**

IN PARTNERSHIP WITH:

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

2023

ACTIVITY REPORT

Project-Team
STORM

STatic Optimizations, Runtime Methods

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

DOMAIN

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

THEME

**Distributed and High Performance
Computing**

Inria

Contents

Project-Team STORM	1
1 Team members, visitors, external collaborators	2
2 Overall objectives	4
3 Research program	5
3.1 Parallel Computing and Architectures	5
3.2 Scientific and Societal Stakes	5
3.3 Towards More Abstraction	6
4 Application domains	7
4.1 Application domains benefiting from HPC	7
4.2 Application in High performance computing/Big Data	7
5 Highlights of the year	7
6 New software, platforms, open data	7
6.1 New software	7
6.1.1 AFF3CT	7
6.1.2 PARCOACH	8
6.1.3 StarPU	8
6.1.4 MIPP	9
6.1.5 CERE	10
6.1.6 DUF	10
6.1.7 MBI	10
6.1.8 EasyPAP	11
7 New results	11
7.1 DSEL for Stream-oriented Workloads	11
7.2 Optimizing Performance and Energy using AI for HPC	12
7.3 Optimal Scheduling for Pipelined and Replicated Task Chains for Software-Defined Radio	12
7.4 Generating Vectorized Kernels from a High Level Problem Description in a General-Purpose Finite Element Simulation Environment	12
7.5 Programming Heterogeneous Architectures Using Hierarchical Tasks	13
7.6 Task scheduling with memory constraints	13
7.7 I/O Classification	13
7.8 Optimizing Performance and Energy of MPI applications	14
7.9 Survey on scheduling	14
7.10 Scheduling Algorithms to Minimize the Energy Consumption of Federated Learning Devices	14
7.11 HPC/AI Convergence	15
7.12 Predicting errors in parallel applications with ML	15
7.13 Rethinking Data Race Detection in MPI-RMA Programs	16
7.14 Leveraging private container networks for increased user isolation and flexibility on HPC clusters	16
7.15 Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver	16
7.16 Highlighting PARCOACH Improvements on MBI	17
7.17 Task-based application tracing	17
7.18 Highlighting EasyPAP Improvements	17
7.19 Load balancing on heterogeneous architectures	17
7.20 Dynamic power capping on heterogeneous nodes	18
7.21 Impact of computation precision on performance and energy consumption	18

8	Bilateral contracts and grants with industry	18
8.1	Bilateral contracts with industry	18
8.1.1	Airbus	18
8.1.2	ATOS / EVIDEN	18
8.1.3	IFPEN	19
8.1.4	Qarnot	19
8.1.5	Ubisoft	20
9	Partnerships and cooperations	20
9.1	International initiatives	20
9.1.1	Inria associate team not involved in an IIL or an international program	20
9.2	International research visitors	20
9.2.1	Visits of international scientists	20
9.3	European initiatives	21
9.3.1	EuroHPC projects	21
9.4	National initiatives	22
9.4.1	ANR	22
9.4.2	PEPR	22
9.4.3	AID	23
9.4.4	Défis Inria	23
10	Dissemination	23
10.1	Promoting scientific activities	23
10.1.1	Scientific events: organisation	23
10.1.2	Scientific events: selection	24
10.1.3	Journal	24
10.1.4	Invited talks	25
10.1.5	Leadership within the scientific community	25
10.1.6	Scientific expertise	25
10.1.7	Research administration	26
10.2	Teaching - Supervision - Juries	26
10.2.1	Teaching	26
10.2.2	Supervision	28
10.2.3	Juries	29
10.3	Popularization	29
10.3.1	Internal or external Inria responsibilities	29
10.3.2	Articles and contents	30
10.3.3	Interventions	30
11	Scientific production	30
11.1	Major publications	30
11.2	Publications of the year	31
11.3	Cited publications	32

Project-Team STORM

Creation of the Project-Team: 2017 July 01

Keywords

Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)
- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A1.1.9. – Fault tolerant systems
- A1.1.13. – Virtualization
- A1.6. – Green Computing
- A2.1.6. – Concurrent programming
- A2.1.7. – Distributed programming
- A2.2.1. – Static analysis
- A2.2.4. – Parallel architectures
- A2.2.5. – Run-time systems
- A2.2.6. – GPGPU, FPGA...
- A2.2.8. – Code generation
- A2.4.1. – Analysis
- A2.4.2. – Model-checking
- A4.3. – Cryptography
- A6.2.7. – High performance computing
- A6.2.8. – Computational geometry and meshes
- A9.6. – Decision support

Other research topics and application domains

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

1 Team members, visitors, external collaborators

Research Scientists

- Olivier Aumage [Team leader, INRIA, Researcher, from Mar 2023, HDR]
- Olivier Aumage [INRIA, Researcher, until Feb 2023, HDR]
- Laercio Lima Pilla [CNRS, Researcher]
- Mihail Popov [INRIA, ISFP]
- Emmanuelle Saillard [INRIA, Researcher, Team permanent contact from 01/03/2023]

Faculty Members

- Denis Barthou [Team leader, BORDEAUX INP, Professor Delegation, until Feb 2023, HDR]
- Scott Baden [UCSD, Professor, Inria Internation Chair]
- Marie-Christine Counilh [UNIV BORDEAUX, Associate Professor]
- Amina Guermouche [BORDEAUX INP, Associate Professor]
- Raymond Namyst [UNIV BORDEAUX, Professor, HDR]
- Samuel Thibault [UNIV BORDEAUX, Professor, HDR]
- Pierre-André Wacrenier [UNIV BORDEAUX, Associate Professor]

Post-Doctoral Fellows

- Maxime Gonthier [INRIA, from Nov 2023]
- Gwenole Lucas [INRIA, from Nov 2023 until Nov 2023]

PhD Students

- Vincent Alba [UNIV BORDEAUX]
- Baptiste Coye [UBISOFT, until Mar 2023]
- Albert D'Aviau De Piolant [INRIA, from Oct 2023]
- Lise Jolicœur [CEA]
- Alice Lasserre [INRIA]
- Alan Lira Nunes [UNIV BORDEAUX, from Nov 2023]
- Gwenole Lucas [INRIA, until Sep 2023]
- Thomas Morin [UNIV BORDEAUX, from Oct 2023]
- Diane Orhan [UNIV BORDEAUX]
- Lana Scravaglieri [IFPEN]
- Radjasouria Vinayagame [ATOS, CIFRE]

Technical Staff

- Nicolas Ducarton [UNIV BORDEAUX, Engineer, from Sep 2023]
- Nathalie Furmento [CNRS, Engineer]
- Kun He [INRIA, Engineer, until May 2023]
- Andrea Lesavourey [INRIA, Engineer, from Oct 2023]
- Romain Lion [INRIA, Engineer]
- Mariem Makni [UNIV BORDEAUX, Engineer, until Jul 2023]
- Bastien Tagliaro [INRIA, Engineer, until Sep 2023]
- Philippe Virouleau [INRIA, Engineer, until Jun 2023]

Interns and Apprentices

- Albert D Aviau De Piolant [INRIA, Intern, from Feb 2023 until Jul 2023]
- Jad El Karchi [INRIA, Intern, until Apr 2023]
- Theo Grandsart [INRIA, Intern, from May 2023 until Jul 2023]
- Angel Hippolyte [INRIA, Intern, from May 2023 until Aug 2023]
- Thomas Morin [INRIA, from Sep 2023 until Sep 2023]
- Thomas Morin [INRIA, Intern, from Feb 2023 until Jul 2023]
- Thomas Morin [INRIA, Intern, from Mar 2023 until Aug 2023]
- Corentin Seutin [INRIA, Intern, from May 2023 until Jul 2023]

Administrative Assistants

- Fabienne Cuyollaa [INRIA, from Nov 2023]
- Sabrina Duthil [INRIA, until Oct 2023]

Visiting Scientists

- Mariza Ferro [UFE, from Dec 2023]
- James Trotter [SIMULA, from May 2023 until Jul 2023]

External Collaborator

- Jean-Marie Couteyen [AIRBUS]

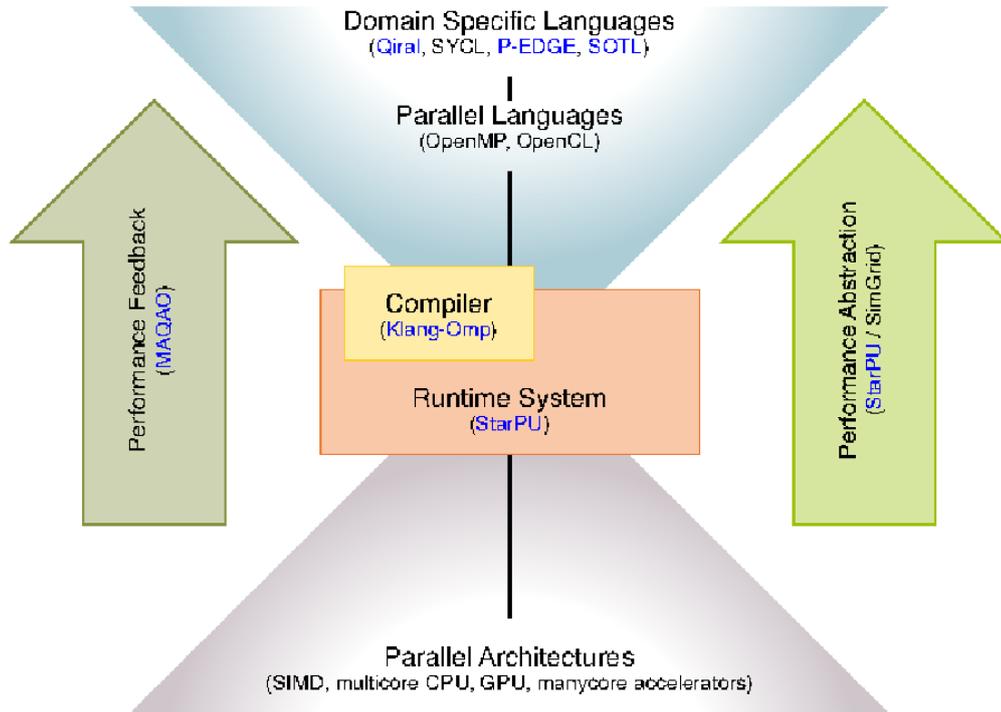


Figure 1: STORM Big Picture

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 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

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

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

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

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

3.2 Scientific and Societal Stakes

In the HPC context, simulation is already considered as a third pillar of science with experiments and theory. Additional computing power means more scientific results, and the possibility to open new fields of simulation requiring more performance, such as multi-scale, multi-physics simulations. Many scientific domains able to take advantage of Exascale computers, these “Grand Challenges” cover large panels of

science, from seismic, climate, molecular dynamics, theoretical and astrophysics physics... Besides, more widespread compute intensive applications are also able to take advantage of the performance increase at the node level. For embedded systems, there is still an on-going trend where dedicated hardware is progressively replaced by off-the-shelf components, adding more adaptability and lowering the cost of devices. For instance, Error Correcting Codes in cell phones are still hardware chips, but new software and adaptative 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 project projects [9.3.1](#))
- Software infrastructures for Telecommunications (see AFF3CT [9.4.3](#))
- Aeronautics (collaboration with Airbus, J.-M. Couteyen, MAMBO project [8.1.1](#))
- Video games (collaboration with Ubisoft, see [8.1.5](#))
- CO2 storage (collaboration with IFPEN, see [8.1.3](#))

4.2 Application in High performance computing/Big Data

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

5 Highlights of the year

Three PhD students of the team defended their PhD at the end of the year: Baptiste COYE, Maxime GONTHIER, Gwenolé LUCAS.

6 New software, platforms, open data

6.1 New software

6.1.1 AFF3CT

Name: A Fast Forward Error Correction Toolbox

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

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

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

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

Authors: Adrien Cassagne, Bertrand Le Gal, Camille Leroux, Denis Barthou, Olivier Aumage

Contact: Denis Barthou

Partner: IMS

6.1.2 PARCOACH

Name: PARallel Control flow Anomaly CHecker

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/index.html>

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

Contact: Emmanuelle Saillard

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

6.1.3 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-04236246

Contact: Olivier Aumage

Participants: Corentin Salingue, Andra Hugo, Benoît Lize, Cédric Augonnet, Cyril Roelandt, François Tessier, Jérôme Clet-Ortega, Ludovic Courtes, Ludovic Stordeur, Marc Sergent, Mehdi Juhour, Nathalie Furmento, Nicolas Collin, Olivier Aumage, Pierre Wacrenier, Raymond Namyst, Samuel Thibault, Simon Archipoff, Xavier Lacoste, Terry Cojean, Yanis Khorsi, Philippe Virouleau, Loïc Jouans, Leo Villeveygoux, Maxime Gonthier, Philippe Swartvagher, Gwenole Lucas, Romain Lion

6.1.4 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: Implement int8 / int16 AVX2 shuffle. Add GEMM example.

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

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

Contact: Denis Barthou

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

Partners: INP Bordeaux, Université de Bordeaux

6.1.5 CERE

Name: Codelet Extractor and REplayer

Keywords: Checkpointing, Profiling

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

Contact: Mihail Popov

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

6.1.6 DUF

Name: Dynamic Uncore Frequency Scaling

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

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

Contact: Amina Guermouche

6.1.7 MBI

Name: MPI Bugs Initiative

Keywords: MPI, Verification, Benchmarking, Tools

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

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

Contact: Emmanuelle Saillard

Participants: Emmanuelle Saillard, Martin Quinson

6.1.8 EasyPAP

Name: easyPAP

Keywords: Parallel programming, Education, OpenMP, MPI, Visualization, Execution trace, OpenCL, Curve plotting

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/>

Authors: Raymond Namyst, Pierre Wacrenier, Alice Lasserre

Contact: Raymond Namyst

Partner: Université de Bordeaux

7 New results

7.1 DSEL for Stream-oriented Workloads

Participants: Olivier Aumage, Denis Barthou.

[6] introduces a new Domain Specific Embedded Language (DSEL) dedicated to Software-Defined Radio (SDR). From a set of carefully designed components, it enables to build efficient software digital communication systems, able to take advantage of the parallelism of modern processor architectures, in a straightforward and safe manner for the programmer. In particular, proposed DSEL enables the combination of pipelining and sequence duplication techniques to extract both temporal and spatial parallelism from digital communication systems. We leverage the DSEL capabilities on a real use case: a fully digital transceiver for the widely used DVB-S2 standard designed entirely in software. Through evaluation, we show how proposed software DVB-S2 transceiver is able to get the most from modern, high-end multicore CPU targets.

7.2 Optimizing Performance and Energy using AI for HPC

Participants: Olivier Aumage, Amina Guermouche, Laercio Lima Pilla, Mihail Popov, Emmanuelle Saillard, Lana Scravaglieri.

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

In these works [11][26], we investigate the impact of configuration options and different problem sizes over both performance and energy. Through a search space exploration, we have found that well-adapted NUMA-related options and cache prefetchers provide significantly more gains for energy (5.9x) than performance (1.85x) over a standard baseline configuration. Moreover, reusing optimization strategies from performance to energy only provides 40% of the gains found when natively optimizing for energy, while transferring strategies across problem sizes is limited to about 70% of the original gains. In order to fill this gap and to avoid exploring the whole search space in multiple scenarios for each new application, we have proposed a new Machine Learning framework. Taking information from one problem size enables us to predict the best configurations for other sizes. Overall, our Machine Learning models achieve 88% of the native gains when cross-predicting across performance and energy, and 85% when predicting across problem sizes.

We further carry this research in collaboration with IFP Energies nouvelles (IFPEN) by focusing on the exploration of SIMD transformations over carbon storage applications. To do so, we are designing a more general exploration infrastructure that can easily incorporate more diverse optimization knobs and applications.

7.3 Optimal Scheduling for Pipelined and Replicated Task Chains for Software-Defined Radio

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

Software-Defined Radio (SDR) represents a move from dedicated hardware to software implementations of digital communication standards. This approach offers flexibility, shorter time to market, maintainability, and lower costs, but it requires an optimized distribution of SDR tasks in order to meet performance requirements. In this context, we study [20] the problem of scheduling SDR linear task chains of stateless and stateful tasks. We model this problem as a pipelined workflow scheduling problem based on pipelined and replicated parallelism on homogeneous resources. Based on this model, we propose a scheduling algorithm named OTAC for maximizing throughput while also minimizing the number of allocated hardware resources, and we prove its optimality. We evaluate our approach and compare it to other algorithms in a simulation campaign, and with an actual implementation of the DVB-S2 communication standard on the AFF3CT SDR Domain Specific Language. Our results demonstrate how OTAC finds optimal schedules, leading consistently to better results than other algorithms, or equivalent results with much fewer hardware resources.

7.4 Generating Vectorized Kernels from a High Level Problem Description in a General-Purpose Finite Element Simulation Environment

Participants: Olivier Aumage, Amina Guermouche, Angel Hippolyte, Laércio Lima Pilla, Mihail Popov, James Trotter.

We studied [23] the process of vectorizing the code generated by the Finite Form Compiler (FFCx), a key component of the FEniCS programming environment (of which Simula is a major contributor) for building scientific simulations using the Finite Element Method. The goal of this work was to explore how the FFCx compiler could target the MIPP library developed by STORM to access the Single-Instruction / Multiple-Data (SIMD) instruction sets available on all modern processors.

7.5 Programming Heterogeneous Architectures Using Hierarchical Tasks

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

Task-based systems have become popular due to their ability to utilize the computational power of complex heterogeneous systems. A typical programming model used is the Sequential Task Flow (STF) model, which unfortunately only supports static task graphs. This can result in submission overhead and a static task graph that is not well-suited for execution on heterogeneous systems. A common approach is to find a balance between the granularity needed for accelerator devices and the granularity required by CPU cores to achieve optimal performance. To address these issues, we have extended the STF model in the StarPU6.1.3 runtime system by introducing the concept of hierarchical tasks. This allows for a more dynamic task graph and, when combined with an automatic data manager, it is possible to adjust granularity at runtime to best match the targeted computing resource. That data manager makes it possible to switch between various data layout without programmer input and allows us to enforce the correctness of the DAG as hierarchical tasks alter it during runtime. Additionally, submission overhead is reduced by using large-grain hierarchical tasks, as the submission process can now be done in parallel. We have shown in [8, 18, 12] that the hierarchical task model is correct and have conducted an early evaluation on shared memory heterogeneous systems using the CHAMELEON dense linear algebra library.

7.6 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 have devised a new scheduling strategies which reorder tasks to strive for locality and thus reduce the amount of communications to be performed. It was shown to be more effective than the current state of the art, particularly in the most constrained cases. The initial mono-GPU results have been published in the FGCS journal [9]. We have introduced a dynamic strategy with a locality-aware principle. We have extended the previous prototype so as to support tasks dependencies, which allowed to complete the initial results on matrix multiplication with the Cholesky and LU cases. We have observed that the obtained behavior is actually very close to the proven-optimal behavior. We have presented the results as poster at ISC [21] and as invited talk at JLESC [22]. We have also extended the support from the GPU-CPU situation to the CPU-disk situation, and shown that the strategy remained effective. Maxime Gonthier defended his PhD thesis [17] The eventual results are pending submission to the JPDC journal, an preliminary RR is available [19]

7.7 I/O Classification

Participants: Francieli Zanon Boito, Luan Teylo, Mihail Popov.

Characterizing the I/O behavior of HPC applications is a challenging task, but informing the system about it is valuable for techniques such as I/O scheduling or burst buffer management, thus yielding potential to improve the overall system performance and energy consumption. In this preliminary work, we investigate I/O traces from several clusters and look at features to group similar I/O job activities: our goal is to define a topology (a la Berkeley's Dwarf) of I/O patterns over HPC systems to help practitioners and users. This study is currently lead by the TADaaM team.

7.8 Optimizing Performance and Energy of MPI applications

Participants: Frédéric Becerril, Emmanuel Jeannot, Laércio Lima Pilla, Mihail Popov.

The balance between performance and energy consumption is a critical challenge in HPC systems. This study focuses on this challenge by exploring and modeling different MPI parameters (e.g., number of processes, process placement across NUMA nodes) across different code patterns (e.g., stencil pattern, memory footprint, communication protocol, strong/weak scalability). A key take away is that optimizing MPI codes for time performance can lead to poor energy consumption: energy consumption of the MiniGhost proto-application could be optimized by more than five times by considering different execution options.

7.9 Survey on scheduling

Participants: Francieli Zanon Boito, Laércio Lima Pilla, Mihail Popov.

Scheduling is a research and engineering topic that encompasses many domains. Given the increasing pace and diversity of publications in conferences and journals, keeping track of the state of the art in their own subdomain and learning about new results in other related topics has become an even more challenging task. This has negative effects on research overall, as algorithms are rediscovered or proposed multiple times, outdated baselines are employed in comparisons, and the quality of research results is decreased.

In order to address these issues and to answer research questions related to the evolution of research in scheduling, we are proposing a data-driven, systematic review of the state of the art in scheduling. Using scheduling-oriented keywords, we extract relevant papers from the major Computer Science publishers and libraries (e.g., IEEE and ACM). We employ natural language processing techniques and clustering algorithms to help organize tens of thousands or more publications found in these libraries, and we adopt a sampling-based approach to identify and characterize groups of publications. In particular, we applied Term Frequency - Inverse Document Frequency (TF-IDF) over the paper corpus and analysed the resulting vectors with semantic Non-Negative Matrix Factorization (NMF) to find the number of latent topics which are used to define clusters of papers. We are currently studying the small representative set of centroids from the aforementioned clusters to extract the research insights. This work is being developed in collaboration with researchers from the University of Basel.

7.10 Scheduling Algorithms to Minimize the Energy Consumption of Federated Learning Devices

Participants: Laércio Lima Pilla.

Federated Learning is a distributed machine learning technique used for training a shared model (such as a neural network) collaboratively while not sharing local data. With an increase in the adoption of

these machine learning techniques, a growing concern is related to their economic and environmental costs. Unfortunately, little work had been done to optimize the energy consumption or emissions of carbon dioxide or equivalents of Federated Learning, with energy minimization usually left as a secondary objective.

In our research, we investigated the problem of minimizing the energy consumption of Federated Learning training on heterogeneous devices by controlling the workload distribution. We modeled this as a total cost minimization problem with identical, independent, and atomic tasks that have to be assigned to heterogeneous resources with arbitrary cost functions. We proposed a pseudo-polynomial optimal solution to the problem based on the previously unexplored Multiple-Choice Minimum-Cost Maximal Knapsack Packing Problem. We also provided four algorithms for scenarios where cost functions are monotonically increasing and follow the same behavior. These solutions are likewise applicable on the minimization of other kinds of costs, and in other one-dimensional data partition problems [4]. Current work is focused on the combined optimization of energy and execution time.

7.11 HPC/AI Convergence

Participants: Bastien Tagliaro, Samuel Thibault.

In the context of the HPC-BIGDATA IPL, we have completed integrating a machine-learning-based task scheduler (i.e. AI for HPC), designed by Nathan Grinsztajn (from Lille) in the StarPU6.1.3 runtime system. The results which Nathan obtained in pure simulation were promising, the integration posed a lot of detailed issues. We completed the integration of not only the inference process but also the training process, written in Python as the AI community is used to. This allows, with very little modification of StarPU applications (essentially add a `while` loop around the main code), to train the neural network at scheduling this application, and then let the network schedule the application. The results are not yet on par with state-of-the-art task scheduling, but significantly better than base heuristics.

In the context of the TEXTAROSSA project and in collaboration with the TOPAL team, we have continued using the StarPU6.1.3 runtime system to execute machine-learning applications on heterogeneous platforms (i.e. HPC for AI). We have completed leveraging the onnx runtime with StarPU, so as to benefit from its CPU and CUDA kernel implementations as well as neural network support. We have experimented with an inference batching situation, where inference jobs are submitted periodically, to be processed by a system equipped with both CPUs and GPUs. We have compared three approaches: the Triton software from NVIDIA, which was recently tailor-cut for such situation, a StarPU-based approach that lets an eager scheduler distribute tasks among CPUs and GPUs, and a StarPU-based approach where we have split the neural network in two pieces, one for GPUs and a smaller one for CPUs. We have observed that the long-standing StarPU runtime takes much better benefit of task/transfer overlapping and asynchronism in general than Triton, and that the splitting approach can provide better bandwidth/latency compromises.

7.12 Predicting errors in parallel applications with ML

Participants: Jad El-Karchi, Mihail Popov, Emmanuelle Saillard.

Investigating if parallel applications are correct is a very challenging task. Yet, recent progress in ML and text embedding show promising results in characterizing source code or the compiler intermediate representation to identify optimizations. We propose to transpose such characterization methods to the context of verification. In particular, we train ML models that take as labels the code correctness along with intermediate representations embeddings as features. Results over MBI and DataRaceBench demonstrate that we can train models that detect if a code is correct with 90% accuracy and up to 75% over new unseen errors. These results are to appear at IPDPS'24. This work is a collaboration with the Iowa State University.

7.13 Rethinking Data Race Detection in MPI-RMA Programs

Participants: Radjasouria Vinayagame, Samuel Thibault, Emmanuelle Saillard.

Supercomputers are capable of more and more computations, and nodes forming them need to communicate even more efficiently with each other. Thus, other types of communication models gain traction in the community. For instance, the Message Passing Interface (MPI) proposes a communication model based on one-sided communications called the MPI Remote Memory Access (MPI-RMA). Thanks to these operations, applications can improve the overlap of communications with computations. However, one-sided communications are complex to write since they are subject to data races. Tools trying to help developers by providing a data race detection for one-sided programs are thus emerging. This work rethinks an existing data race detection algorithm for MPI-RMA programs by improving the way it stores memory accesses, thus improving its accuracy and reducing the overhead at runtime [16, 14].

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

Participants: Lise Jolicoeur, Raymond Namyst.

To address the increasing complexity of modern scientific computing workflows, HPC clusters must be able to accommodate a wider range of workloads without compromising their efficiency in processing batches of highly parallel jobs. Cloud computing providers have a long history of leveraging all forms of virtualization to let their clients easily and securely deploy complex distributed applications and similar capabilities are now expected from HPC facilities. In recent years, containers have been progressively adopted by HPC practitioners to facilitate the installation of applications along with their software dependencies. However little attention has been given to the use of containers with virtualized networks to securely orchestrate distributed applications on HPC resources.

In her PhD thesis, Lise Jolicoeur is designing containerization mechanisms to leverage network virtualization and benefit from the flexibility and isolation typically found in a cloud environment while being as transparent and as easy to use as possible for people familiar with HPC clusters. Users are automatically isolated in their own private network which prevents unwanted network accesses and allows them to easily define network addresses so that components of a distributed workflow can reliably reach each other. This has been implemented the pcocc (private cloud on a compute cluster) container runtime, developed at CEA. Scalability tests on several hundreds of virtual nodes confirm the low impact on user experience, and the execution of benchmarks such as HPL in a production environment on up to 100 nodes confirmed the very small overhead incurred by our implementation.

7.15 Multi-Criteria Mesh Partitioning for an Explicit Temporal Adaptive Task-Distributed Finite-Volume Solver

Participants: Alice Lasserre, Raymond Namyst.

The aerospace industry is one of the largest users of numerical simulation, which is an essential tool in the field of aerodynamic engineering, where many fluid dynamics simulations are involved. In order to obtain the most accurate solutions, some of these simulations use unstructured finite volume solvers that cope with irregular meshes by using explicit time-adaptive integration methods. Modern parallel implementations of these solvers rely on task-based runtime systems to perform fine-grained load balancing and to avoid unnecessary synchronizations. Although such implementations greatly improve performance compared to a classical fork-join MPI+OpenMP variants, it remains a challenge to keep all cores busy throughout the simulation loop.

This work takes place in the context of the FLUSEPA CFD solver, developed by ArianeGroup, and used by Airbus evaluate jet noise in installed nozzle configurations. In her PhD thesis, Alice Lasserre has investigated the origins of this lack of parallelism. She has showed that the irregular structure of the task graph plays a major role in the inefficiency of the computation distribution. Her main contribution is to improve the shape of the task graph by using a new mesh partitioning strategy. The originality of the approach is to take the temporal level of mesh cells into account during the mesh partitioning phase. She has implemented this approach in the FLUSEPA ArianeGroup production code. Our partitioning method leads to a more balanced task graph. The resulting task scheduling is up to two times faster for meshes ranging from 200,000 to 12,000,000 components.

7.16 Highlighting PARCOACH Improvements on MBI

Participants: Philippe Virouleau, Emmanuelle Saillard.

PARCOACH is one of the few verification tools that mainly relies on a static analysis to detect errors in MPI programs. First focused on the detection of call ordering errors with collectives, it has been extended to detect local concurrency errors in MPI-RMA programs. Furthermore, the new version of the tool fixes multiple errors and is easier to use [15].

7.17 Task-based application tracing

Participants: Samuel Thibault.

In collaboration with the TADAAM team [7], we have investigated the common issues encountered when tracing a task-based application, and devised various solutions, including: (1) managing trace flushing on the disk, (2) filtering the different types of events, (3) decentralizing the tracing locking. The execution in a distributed context also requires careful clock synchronization between the different nodes.

7.18 Highlighting EasyPAP Improvements

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

EasyPAP 6.1.8 is a simple and attractive programming environment to facilitate the 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.

We have enhanced EasyPAP to take into account hardware counters and added support for CUDA and MIPP interfaces.

7.19 Load balancing on heterogeneous architectures

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

In collaboration with the CAMUS team and in the context of the MICROCARD project, we are investigating the scheduling of generated simulation kernels on heterogeneous architectures. We designed a heuristic to improve the load balancing on these architecture and also an algorithm to dynamically choose the optimal number of GPUs to use. Additionally, we are integrating these mechanisms into OpenCARP, leveraging previous efforts in the Microcard project. These efforts capitalized on LLVM's MLIR framework to generate various device-specialized versions of kernels derived from the ionic model expressed in OpenCARP's high-level DSL.

7.20 Dynamic power capping on heterogeneous nodes

Participants: Albert D'Aviau De Piolant, Abdou Guermouche, Amina Guermouche.

We are working on a dynamic power capping algorithm for heterogeneous nodes. Based on DUF 6.1.6, we are extending the algorithm in order to account for GPUs. The idea is to benefit from a low CPU-intensive phase to transfer the power budget the GPU in order to improve performance. A first version of the algorithm was implemented and evaluated [25].

7.21 Impact of computation precision on performance and energy consumption

Participants: Hicham Nekt, Abdou Guermouche, Amina Guermouche.

We studied [24] the impact of the computations precision on the performance and energy consumption of an application. We conducted a set of experiments on CPU-intensive applications. For such applications, the faster the application, the better the energy consumption.

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

8.1.1 Airbus

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

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

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

8.1.2 ATOS / EVIDEN

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

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

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

Contract "Plan de relance" to integrate Célia Tassadit Ait Kaci's PhD work in PARCOACH

PARCOACH is a framework that detects errors in MPI applications. This "Plan de relance" aims at integrating the work of Célia Tassadit Ait Kaci's PhD (RMA-Analyzer) and improving user experience.

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

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

8.1.3 IFPEN

Participants: Olivier Aumage, Mihail Popov, Lana Scravaglieri.

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

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

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

8.1.4 Qarnot

Participants: Laércio Lima Pilla.

Among the different HPC centers, data centers, and Cloud providers, Qarnot distinguishes itself by proposing a decentralized and geo-distributed solution with an aim of promoting a more virtuous approach to the emissions generated by the execution of compute-intensive tasks. With their compute clusters, Qarnot focuses on capturing the heat released by the processors that carry out computing tasks. This heat is then used to power third-party systems (boilers, heaters, etc.). By reusing the energy from computing as heating, Qarnot provides a low-carbon infrastructure to its compute and heating users.

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

8.1.5 Ubisoft

Participants: Denis Barthou, Baptiste Coye, Laercio Lima-Pilla, Raymond Namyst.

Inria and Ubisoft have been working for three years (2020-2023) on improving task scheduling in modern video game engines, where it is essential to keep a high frame rate even if thousands of task are generated during each frame. These tasks are organized in a soft real-time parallel task graph where some tasks could be postponed to a later frame while others can not be delayed. Baptiste Coye defended his PhD thesis on 2023, July the 10th. He has proposed a new formalization of the scheduling problem for game engines, a dynamic scheduling solution based on a Monte Carlo Graph Search approach combined with task skipping to adapt execution for a target framerate. He has evaluate'd this approach using two recent commercial games from Ubisoft with various loads on different architectures. He was able to increase by 14% the obtained framerate on a real sequence of gameplay on PS4 Pro and by 12.8% on Xbox One, at the cost of moderate AI degradation.

9 Partnerships and cooperations

9.1 International initiatives

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

MAELSTROM

Participants: Olivier Aumage, Amina Guermouche, Angel Hippolyte, Laercio Lima Pilla, Mihail Popov, James Trotter.

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

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

9.2 International research visitors

9.2.1 Visits of international scientists

Inria International Chair

Participants: Scott Baden.

Scott Baden, U. California, visited the team one month in May and another month in September 2023.

Other international visits to the team

Participants: James Trotter.

Status Post-Doc

Institution of origin: SIMULA

Country: Norway

Dates: From May 30 to July 21

Context of the visit: MAELSTROM Associate Team

Mobility program/type of mobility: Research stay

9.3 European initiatives**9.3.1 EuroHPC projects****MICROCARD**

Participants: Vincent Alba, Olivier Aumage, Denis Barthou, Marie-Christine Counilh, Nicolas Ducarton, Amina Guermouche, Mariem Makni, Emmanuelle Saillard.

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

Title: Numerical modeling of cardiac electrophysiology at the cellular scale

Duration: From April 1, 2021 to September 30, 2024

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- MEGWARE COMPUTER VERTRIEB UND SERVICE GMBH, Germany
- SIMULA RESEARCH LABORATORY AS, Norway
- UNIVERSITE DE STRASBOURG (UNISTRA), France
- ZUSE-INSTITUT BERLIN (ZUSE INSTITUTE BERLIN), Germany
- UNIVERSITA DELLA SVIZZERA ITALIANA (USI), Switzerland
- KARLSRUHER INSTITUT FUER TECHNOLOGIE (KIT), Germany
- UNIVERSITE DE BORDEAUX (UBx), France
- UNIVERSITA DEGLI STUDI DI PAVIA (UNIPV), Italy
- INSTITUT POLYTECHNIQUE DE BORDEAUX (Bordeaux INP), France
- NUMERICOR GMBH, Austria

- OROBIX SRL (OROBIX), Italy

Inria contact: Mark Potse

Coordinator: Mark Potse

Summary: Cardiovascular diseases are the most frequent cause of death worldwide and half of these deaths are due to cardiac arrhythmia, a disorder of the heart's electrical synchronization system. Numerical models of this complex system are highly sophisticated 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 implies a different, harder numerical problem and a 10,000-fold increase in problem size. Exascale computers will be needed to run such models. We propose to develop an exascale application platform for cardiac electrophysiology simulations that is usable for cell-by-cell simulations. The platform will be co-designed by HPC experts, numerical scientists, biomedical engineers, and biomedical scientists, from academia and industry.

9.4 National initiatives

9.4.1 ANR

ANR SOLHARIS **SOLHARIS**

Participants: Olivier Aumage, Nathalie Furmento, Maxime Gonthier, Gwénoél Lucas, Samuel Thibault, Pierre-André Wacrenier.

- ANR PRCE 2019 Program, 2019 - 2023 (48 months)
- Identification: ANR-19-CE46-0009
- Coordinator: CNRS-IRIT-INPT
- Other partners: INRIA-LaBRI Bordeaux, INRIA-LIP Lyon, CEA/CESTA, Airbus CRT
- Abstract: SOLHARIS aims at achieving strong and weak scalability (i.e., the ability to solve problems of increasingly large size while making an effective use of the available computational resources) of sparse, direct solvers on large scale, distributed memory, heterogeneous computers. These solvers will rely on asynchronous task-based parallelism, rather than traditional and widely adopted message-passing and multithreading techniques; this paradigm will be implemented by means of modern runtime systems which have proven to be good tools for the development of scientific computing applications.

9.4.2 PEPR

PEPR NumPEX / Exa-Soft focused project

Participants: Raymond Namyst, Samuel Thibault, Albert D'Aviau De Piolant, Amina Guermouche, Nathalie Furmento.

- 2023 - 2028 (60 months)
- Coordinator: Raymond Namyst
- Other partners: CEA, CNRS, Univ. Paris-Saclay, Telecom SudParis, Univ. of Bordeaux, Bordeaux INP, Univ. Rennes, Univ. Strasbourg, Univ. Toulouse 3, Univ. Grenoble Alpes.
- Abstract: The NumPEX project (High Performance numerics for Exascale) aims to design and develop the software components and tools that will equip future exascale machines and to prepare the major application domains to fully exploit the capabilities of these machines. It is composed of 5 scientific focused project. The Exa-Soft project aims at consolidating the

exascale software ecosystem by providing a coherent, exascale-ready software stack featuring breakthrough research advances enabled by multidisciplinary collaborations between researchers. Meeting the needs of complex parallel applications and the requirements of exascale architectures raises numerous challenges which are still left unaddressed. As a result, several parts of the software stack must evolve to better support these architectures. More importantly, the links between these parts must be strengthened to form a coherent, tightly integrated software suite. The main scientific challenges we intend to address are: productivity, performance portability, heterogeneity, scalability and resilience, performance and energy efficiency.

9.4.3 AID

AID AFF3CT

Participants: Olivier Aumage, Andrea Lesavourey, Laercio Lima Pilla, Diane Orhan.

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

9.4.4 Défis Inria

Défi PULSE

Participants: Laercio Lima Pilla.

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

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: organisation

Participants: Olivier Aumage, Nathalie Furmento, Laércio Lima Pilla, Mihail Popov, Emmanuelle Saillard.

General chair, scientific chair

- Emmanuelle Saillard and Mihail Popov participated in the organisation of the first HPC Bugs Fest during the "Correctness" Workshop of the SuperComputing SC23 Conference in Denver.

- Olivier Aumage and Laércio Lima Pilla organized the [2nd AFF3CT User Day Workshop](#) at LIP6 Laboratory, Sorbonne University, Paris.

Member of the organizing committees

- Nathalie Furmento was part of the organizing committee of the Inria national seminar for engineers working on HPC related subjects.
- Emmanuelle Saillard was both Publicity and Web Chair and Publisher Chair of the EuroMPI 2023 conference, Bristol, UK.

10.1.2 Scientific events: selection

Participants: Olivier Aumage, Amina Guermouche, Laercio Lima Pilla, Mihail Popov, Emmanuelle Saillard, Samuel Thibault, Raymond Namyst.

Member of the conference program committees

- Olivier Aumage: "ExHET / Extreme Heterogeneity Solutions" Workshop of the PPOPP 2023 Conference (Montreal), "D-HPC / Democratizing HPC" Workshop of the SuperComputing SC23 Conference (Denver), "iWAPT / International Workshop on Automatic Performance Tuning" of the IPDPS 2023 Conference (St Petersburg, FL, USA)
- Amina Guermouche: HiPC 2023, SC23 tutorials, SC23 PhD Posters
- Laércio Lima Pilla: HCW 2023, SBAC-PAD 2023, WCC 2023.
- Mihail Popov: PDP 2023, Correctness 2023
- Emmanuelle Saillard: COMPAS 2023, Correctness 2023, CC 2024
- Samuel Thibault: Euro-Par 2023, HCW'23, Cluster 2023
- Raymond Namyst: Heteropar 2023

Reviewer

- Laércio Lima Pilla: IPDPS 2024
- Mihail Popov: IPDPS 2024, SC 2023, Cluster 2023

10.1.3 Journal

Participants: Laércio Lima Pilla, Mihail Popov, Samuel Thibault.

Member of the editorial boards

- Samuel Thibault: Subject Area Editor for Journal of Parallel and Distributed Computing

Reviewer - reviewing activities

- Laércio Lima Pilla: Future Generation Computing Systems, Journal of Parallel and Distributed Computing, ACM Transactions on Cyber-Physical Systems, ACM Surveys.
- Mihail Popov: PeerJ Computer Science.

10.1.4 Invited talks

Participants: Olivier Aumage, Maxime Gonthier, Mihail Popov, Emmanuelle Saillard, Samuel Thibault, Andrea Lesavourey.

- Olivier Aumage gave a talk about "Using the StarPU task-based runtime system for heterogeneous platforms as the core engine for a linear algebra software stack" at the ICIAM 2023 Conference in Tokyo.
- Maxime Gonthier gave a talk about "Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs" at the JLESC'23 meeting in Bordeaux.
- Mihail Popov gave a talk about "Characterizing Design Spaces Optimizations" at the Intel's VSSAD seminar (virtual).
- Emmanuelle Saillard gave a talk about "Latest Advancements in PARCOACH for MPI Code Verification" at the International parallel Tools workshop (virtual).
- Samuel Thibault:
 - talk about "Scaling StarPU's Task-Based Sequential Task Flow (STF) Inside Nodes and Between Nodes" at SIAM CSE'23 conference in Asmterdam.
 - keynote about "Vector operations, tiled operations, distributed operations, task graphes, ... What next?" as keynote speaker for the JLESC'23 meeting in Bordeaux.
 - talk about "Les graphes de tâches, une opportunité pour l'hétérogénéité" at the GDR SOC2 workshop.
- Andrea Lesavourey gave a talk on "Affect"

10.1.5 Leadership within the scientific community

Participants: Olivier Aumage, Raymond Namyst.

- Olivier Aumage participated to the Panel "Runtimes & Workflow Systems for Extreme Heterogeneity: Challenges and Opportunitie" at the SuperComputing SC23 Conference in Denver.
- Raymond Namyst is the coordinator of the national Exa-SofT (PEPR NumPEX) ANR project (2023-2028).

10.1.6 Scientific expertise

Participants: Olivier Aumage, Marie-Christine Counilh, Nathalie Furmento, Emmanuelle Saillard, Samuel Thibault, Raymond Namyst.

- Olivier Aumage participated to 2023 CRCN / ISFP Inria Researcher selection jury for the Inria Research Center at the University of Bordeaux.
- Nathalie Furmento was a member of several recruiting committees for engineer positions.
- Emmanuelle Saillard participated to the Associate Professor selection jury at Bordeaux INP.
- Samuel Thibault was a RIPEC reviewer.
- Marie-Christine Counilh was a RIPEC reviewer and member of the committee for ATER (Attaché Temporaire d'Enseignement et de Recherche) positions at the University of Bordeaux.
- Raymond Namyst has served as an computer science expert for the Belgium AEQES evaluation committee.

10.1.7 Research administration

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

- Olivier Aumage
 - Head of Team "SEHP – Supports d'exécutions hautes performances" (High performance executive supports) at LaBRI computer science Laboratory.
 - Elected member of LaBRI's Scientific Council.
- 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.
- Laércio Lima Pilla
 - co-coordinator of the parity and equality group of the Inria centre of the University of Bordeaux.
 - 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.
 - external member of the Equality, Diversity & Inclusion Committee of the Department of Computer Science of the University of Sheffield.
- Emmanuelle Saillard
 - Member of the Commission de délégation at Inria Research Centre of the University of Bordeaux.
- Samuel Thibault
 - nominated member of the council of the LaBRI.
 - member of the "Source Code and software group" of the national Comity for OpenScience (CoSO).

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

Participants: Olivier Aumage, Albert D'Aviau De Piolant, Denis Barthou, Marie-Christine Counilh, Nathalie Furmento, Amina Guermouche, Kun He, Laércio Lima Pilla, Mihail Popov, Emmanuelle Saillard, Samuel Thibault, Diane Orhan, Lise Jolicoeur, Alice Lasserre, Radjasouria Vinayagame, Vincent Alba, Thomas Morin, Raymond Namyst.

- Training Management
 - Raymond Namyst is vice chair of the Computer Science Training Department of University of Bordeaux.

- Academic Teaching

- Engineering School + Master: Olivier Aumage, Multicore Architecture Programming, 24HeTD, M2, ENSEIRB-MATMECA + University of Bordeaux.
- Engineering School: Emmanuelle Saillard, Languages of parallelism, 12HeC, M2, ENSEIRB-MATMECA.
- Licence: Emmanuelle Saillard, Introduction to research, 1HeTD, L3, University of Bordeaux.
- Master: Laércio Lima Pilla, Algorithms for High-Performance Computing Platforms, 17HeTD, M2, ENSEIRB-MATMECA and University of Bordeaux.
- Master: Laércio Lima Pilla, Scheduling and Runtime Systems, 27.75 HeTD, M2, University of Paris-Saclay.
- Engineering School: Mihail Popov, Project C, 25HeC, L3, ENSEIRB-MATMECA.
- Engineering School: Mihail Popov, Cryptography, 33HeC, M1, ENSEIRB-MATMECA.
- Amina Guermouche is responsible for the computer science first year at ENSEIRB-MATMECA
- 1st year : Amina Guermouche, Linux Environment, 24HeTD, ENSEIRB-MATMECA
- 1st year : Amina Guermouche, Computer architecture, 36 HeTD, ENSEIRB-MATMECA
- 1st year : Amina Guermouche, Programming project, 25HeTD, ENSEIRB-MATMECA
- 1st year : Amina Guermouche, Programming Project, 25HeTD, ENSEIRB-MATMECA
- 2nd year : Amina Guermouche, System Programming, 18HeTD, ENSEIRB-MATMECA
- 3rd year : Amina Guermouche, GPU Programming, 39HeTD, ENSEIRB-MATMECA + University of Bordeaux
- 1st year : Albert d'Aviau de Piolant, Computer architecture, 16HeTD, ENSEIRB-MATMECA
- Licence: Samuel Thibault is responsible for the computer science course of the science university first year.
- 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, Introduction to Computer Science, 56HeTD, L1, University of Bordeaux.
- Licence: Samuel Thibault, Networking, 51HeTD, Licence Pro, University of Bordeaux.
- Master: Samuel Thibault, Operating Systems, 24HeTD, M1, University of Bordeaux.
- Master: Alice Lasserre, Operating Systems, 24HeTD, M1, University of Bordeaux.
- Master: Samuel Thibault, System Security, 20HeTD, M2, University of Bordeaux.
- Licence : Marie-Christine Counilh, Introduction to Computer Science, 45HeTD, L1, University of Bordeaux.
- Licence : Marie-Christine Counilh, Introduction to C programming, 36HeTD, L1, University of Bordeaux.
- Licence : Marie-Christine Counilh, Object oriented programming in Java, 32HeTD, L2, University of Bordeaux.
- Master MIAGE : Marie-Christine Counilh, Object oriented programming in Java, 28HeTD, M1, University of Bordeaux.
- 1st year : Diane Orhan, Computer architecture, 16HeTD, ENSEIRB-MATMECA
- 2nd year : Diane Orhan, Systems programming, 20HeTD, ENSEIRB-MATMECA
- 2nd year : Diane Orhan, Operating systems project, 14HeTD, ENSEIRB-MATMECA
- 3rd year : Diane Orhan, , HeTD, ENSEIRB-MATMECA
- 1st year: Radjasouria Vinayagame, Linux Environment, 24HeTD, ENSEIRB-MATMECA

- 1st year: Radjasouria Vinayagame, Logic and proof of program, 14HeTD, ENSEIRB-MATMECA
- 1st year: Vincent Alba, Logic and proof of program, 14HeTD, ENSEIRB-MATMECA
- Master: Thomas Morin, Computability and complexity, 24HeTD, M1, University of Bordeaux.
- Engineering School: Lise Jolicoeur, Software for HPC Clusters (Logiciels Cluster), 13.5HeTD, M1, ENSIE.
- Tutorials
 - Emmanuelle Saillard: tutorial on PARCOACH during EuroMPI'23 in Bristol
 - Olivier Aumage, Nathalie Furmento, Kun He, Samuel Thibault: tutorial on STARPU at CALMIP computing center in Toulouse
 - Denis Barthou: tutorial on MIPP at CALMIP computing center in Toulouse

10.2.2 Supervision

Participants: Olivier Aumage, Amina Guermouche, Laercio Lima Pilla, Raymond Namyst, Mihail Popov, Emmanuelle Saillard, Samuel Thibault, Pierre-André Wacrenier.

- PhD completed: Baptiste Coye, "Dynamic scheduling of task graph by composition", supervised by Raymond Namyst and Denis Barthou.
- PhD completed: Maxime Gonthier, "Memory-constrained Scheduling in a task-based model", supervised by L. Marchal and S. Thibault
- PhD completed: Gwenolé Lucas, "Programming Heterogeneous Architectures using Divisible Tasks", supervised by M. Faverge, A. Guermouche, R. Namyst and P.-A. Wacrenier.
- PhD in progress: Vincent Alba, "Task scheduling for Exascale", supervised by Denis Barthou and Amina Guermouche.
- PhD in progress: Albert d'Aviau de Piolant, "Energy-aware scheduling for exascale architectures", supervised by Abdou Guermouche and Amina Guermouche.
- PhD in progress: Lise Jolicoeur, "Towards secure cluster architectures for HPC workflows", supervised by R.Namyst.
- PhD in progress: Alice Lasserre, "Optimization of a task-based simulation code on a distributed supercomputer", supervised by R.Namyst.
- PhD in progress: Alan Lira Nunes, "Scheduling Algorithms for the Optimization of Distributed Machine Learning Models on Heterogeneous Resources", supervised by Laércio Lima Pilla and Lúcia Drummond.
- PhD in progress: Thomas Morin, "Scheduling recursive task graphs", supervised by Abdou Guermouche and Samuel Thibault.
- PhD in progress: Diane Orhan, "Modeling and dynamic optimization of software radio chains on heterogeneous architectures", supervised by Denis Barthou, Christophe Jégo, and Laércio Lima Pilla.
- PhD in progress: Lana Scravaglieri, "Portable vectorization with numerical accuracy control for multi-precision simulation codes", supervised by Olivier Aumage, Mihail Popov, Thomas Guignon (IFPEN) and Ani Anciaux-Sedrakian (IFPEN).
- PhD in progress: Radjasouria Vinayagame, "Optimization of porting and performance of HPC applications with distributed and globally addressed memory", supervised by Emmanuelle Saillard and Samuel Thibault.

- Internship: Albert d'Aviau de Piolant, Feb. - Jul. 2023, Amina Guermouche
- Internship: Jad El Karchi, Feb. - Jul. 2023, Emmanuelle Saillard, Mihail Popov
- Internship: Angel Hippolyte, May. - Aug. 2023, Olivier Aumage, Amina Guermouche, Laércio Lima Pilla, Mihail Popov, James Trotter
- Internship: Hicham Nekt, Jun - Sept. 2023, Abdou Guermouche, Amina Guermouche
- Internship: Frederic Becerril, Jun - July. 2023, Emmanuel Jeannot, Mihail Popov, Laércio Lima Pilla

10.2.3 Juries

Participants: Laércio Lima Pilla, Emmanuelle Saillard, Samuel Thibault.

- Laércio Lima Pilla:
 - Member for the PhD of Z. Yao, "Intelligent Orchestration of High Performance, Scalable, and Correct Data Centres As a Network Service", École Polytechnique, Institut Polytechnique de Paris
- Emmanuelle Saillard:
 - Guest for the PhD of A. Maillé, "Simple, Safe and Efficient Abstractions for Communication and Streaming in Parallel Computing", ENS Lyon
 - Reviewer for the PhD of A. Thune, "High performance computing for reservoir simulation", University of Oslo
- Samuel Thibault:
 - President for the HDR of G. Pallez, "Model Design and Accuracy for Resource Management in HPC", University of Bordeaux
 - Reviewer for the PhD of A. Maillé, "Simple, Safe and Efficient Abstractions for Communication and Streaming in Parallel Computing", ENS de Lyon
 - Member for the PhD of A. Jegou, "Modèles de programmation avancés à base de tâches pour les algorithmes d'algèbre linéaire qui passent à l'échelle", ENSEEIHT

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

Participants: Emmanuelle Saillard, , Raymond Namyst.

- Emmanuelle Saillard, Raymond Namyst: Organization of a hackathon with IJBA, November 2023.
- Emmanuelle Saillard:
 - responsible of popularization activities for Inria Research Centre of the University of Bordeaux.
 - member of the Blaise Pascal Fondation
 - organisation of Moi Informaticienne - Moi Mathématicienne, February 2023.

10.3.2 Articles and contents

Participants: Emmanuelle Saillard.

- "Ces super-ordinateurs peuvent éviter les arythmies cardiaques !", video published by Curieux for the "Nuit Européenne des chercheur.e.s"

10.3.3 Interventions

Participants: Nathalie Furmento, Alice Lasserre, Laércio Lima Pilla, Thomas Morin, Diane Orhan, Mihail Popov, Emmanuelle Saillard, Lana Scravaglieri.

- Nathalie Furmento, Alice Lasserre, Thomas Morin, Diane Orhan: Participation to the Open Days of the University of Bordeaux
- Mihail Popov, Diane Orhan, Lana Scravaglieri, Emmanuelle Saillard: Welcoming ENS-Lyon undergraduate students, November 2023.
- Laércio Lima Pilla: coach and jury on a hackathon with IJBA, November 2023.
- Diane Orhan: coach on a hackathon with IJBA, November 2023.
- Emmanuelle Saillard:
 - Fête de la Science, Inria, Oct. 2023: atelier "La grande muraille d’Egypte", "Jeu à débattre"
 - Moi Informaticienne - Moi Mathématicienne, University of Bordeaux, February 2023.
 - Chiche!: Gustave Eiffel high school (May and December 2023), Daguin high school (December 2023)
 - Participation of a panel organised by the association "Femmes et science", June 2023 (University of Bordeaux)
 - Nuit européenne des chercheur.e.s: speedsearching, "Parle-moi du futur"
 - Participation of the JSI to talk about Chiche!, August 2023, Bordeaux
 - Atelier "La grande muraille d’Egypte", April 2023, Bordeaux-lac middle school

11 Scientific production

11.1 Major publications

- [1] A. Cassagne, R. Tajan, O. Aumage, C. Leroux, D. Barthou and C. Jégo. ‘A DSEL for high throughput and low latency software-defined radio on multicore CPUs’. In: *Concurrency and Computation: Practice and Experience* (3rd July 2023), e7820. DOI: [10.1002/cpe.7820](https://doi.org/10.1002/cpe.7820). URL: <https://hal.science/hal-04156404>.
- [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] M. Gonthier, L. Marchal and S. Thibault. ‘Taming data locality for task scheduling under memory constraint in runtime systems’. In: *Future Generation Computer Systems* (2023). DOI: [10.1016/j.future.2023.01.024](https://doi.org/10.1016/j.future.2023.01.024). URL: <https://inria.hal.science/hal-03623220>.
- [4] L. Lima Pilla. ‘Scheduling Algorithms for Federated Learning with Minimal Energy Consumption’. In: *IEEE Transactions on Parallel and Distributed Systems* (31st Jan. 2023). DOI: [10.1109/TPDS.2023.3240833](https://doi.org/10.1109/TPDS.2023.3240833). URL: <https://hal.science/hal-03775491>.

- [5] L. Scravaglieri, M. Popov, L. Lima Pilla, A. Guermouche, O. Aumage and E. Saillard. ‘Optimizing Performance and Energy Across Problem Sizes Through a Search Space Exploration and Machine Learning’. In: *Journal of Parallel and Distributed Computing* 180 (28th June 2023), p. 104720. DOI: [10.1016/j.jpdc.2023.104720](https://hal.science/hal-03810305). URL: <https://hal.science/hal-03810305>.

11.2 Publications of the year

International journals

- [6] A. Cassagne, R. Tajan, O. Aumage, C. Leroux, D. Barthou and C. Jégo. ‘A DSEL for high throughput and low latency software-defined radio on multicore CPUs’. In: *Concurrency and Computation: Practice and Experience* (3rd July 2023), e7820. DOI: [10.1002/cpe.7820](https://hal.science/hal-04156404). URL: <https://hal.science/hal-04156404>.
- [7] A. Denis, E. Jeannot, P. Swartvagher and S. Thibault. ‘Tracing task-based runtime systems: Feedbacks from the StarPU case’. In: *Concurrency and Computation: Practice and Experience* (10th Oct. 2023), p. 24. DOI: [10.1002/cpe.7920](https://inria.hal.science/hal-04236246). URL: <https://inria.hal.science/hal-04236246>.
- [8] 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://hal.science/hal-04088833). URL: <https://hal.science/hal-04088833>.
- [9] M. Gonthier, L. Marchal and S. Thibault. ‘Taming data locality for task scheduling under memory constraint in runtime systems’. In: *Future Generation Computer Systems* (2023). DOI: [10.1016/j.future.2023.01.024](https://inria.hal.science/hal-03623220). URL: <https://inria.hal.science/hal-03623220>.
- [10] L. Lima Pilla. ‘Scheduling Algorithms for Federated Learning with Minimal Energy Consumption’. In: *IEEE Transactions on Parallel and Distributed Systems* (31st Jan. 2023). DOI: [10.1109/TPDS.2023.3240833](https://hal.science/hal-03775491). URL: <https://hal.science/hal-03775491>.
- [11] L. Scravaglieri, M. Popov, L. Lima Pilla, A. Guermouche, O. Aumage and E. Saillard. ‘Optimizing Performance and Energy Across Problem Sizes Through a Search Space Exploration and Machine Learning’. In: *Journal of Parallel and Distributed Computing* 180 (28th June 2023), p. 104720. DOI: [10.1016/j.jpdc.2023.104720](https://hal.science/hal-03810305). URL: <https://hal.science/hal-03810305>.

Invited conferences

- [12] S. Thibault. ‘Vector operations, tiled operations, distributed execution, task graphs, ... What next?’ In: JLESC 15 - 15th Joint Laboratory for Extreme Scale Computing Workshop. Talence, France, 21st Mar. 2023. URL: <https://inria.hal.science/hal-04115280>.

International peer-reviewed conferences

- [13] I. Daoudi, T. Gautier, S. Thibault and S. Perarnau. ‘Improving Simulations of Task-Based Applications on Complex NUMA Architectures’. In: IWOMP 2023 - 19th International Workshop on OpenMP. Vol. 14114. Lecture Notes in Computer Science. Bristol, United Kingdom: Springer Nature Switzerland, 1st Sept. 2023, pp. 195–209. DOI: [10.1007/978-3-031-40744-4_13](https://inria.hal.science/hal-04201317). URL: <https://inria.hal.science/hal-04201317>.
- [14] R. Vinayagame, V. M. Nguyen, M. Sergent, S. Thibault and E. Saillard. ‘Rethinking Data Race Detection in MPI-RMA Programs’. In: *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*. 7th International Workshop on Software Correctness for HPC Applications (Correctness '23). Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis. Denver (Colorado, USA), United States, 2023, pp. 196–204. DOI: [10.1145/3624062.3624086](https://inria.hal.science/hal-04272399). URL: <https://inria.hal.science/hal-04272399>.

- [15] P. Virouleau, E. Saillard, M. Sergent and P. Lemarinier. ‘Highlighting PARCOACH Improvements on MBI’. In: SC-W 2023 - Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis. Denver CO, United States: ACM, 12th Nov. 2023, pp. 238–241. DOI: [10.1145/3624062.3624093](https://doi.org/10.1145/3624062.3624093). URL: <https://hal.science/hal-04320261>.

Conferences without proceedings

- [16] R. Vinayagame. ‘Rethinking Data Race Detection in RMA-Analyzer’. In: COMPAS 2023 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Annecy, France, 4th July 2023. URL: <https://inria.hal.science/hal-04272083>.

Doctoral dissertations and habilitation theses

- [17] M. Gonthier. ‘Scheduling Under Memory Constraint in Task-based Runtime Systems’. Ecole normale supérieure de lyon - ENS LYON, 25th Sept. 2023. URL: <https://theses.hal.science/tel-04260094>.
- [18] G. Lucas. ‘On the Use of Hierarchical Task for Heterogeneous Architectures’. Université de Bordeaux, 10th Oct. 2023. URL: <https://theses.hal.science/tel-04316145>.

Reports & preprints

- [19] M. Gonthier, S. Thibault and L. Marchal. *A generic scheduler to foster data locality for GPU and out-of-core task-based applications*. 30th June 2023. DOI: [10.1145/nnnnnnnn.nnnnnnnn](https://doi.org/10.1145/nnnnnnnn.nnnnnnnn). URL: <https://inria.hal.science/hal-04146714>.
- [20] D. Orhan, L. Lima Pilla, D. Barthou, A. Cassagne, O. Aumage, R. Tajan, C. Jégo and C. Leroux. *OTAC: Optimal Scheduling for Pipelined and Replicated Task Chains for Software-Defined Radio*. 4th Oct. 2023. URL: <https://hal.science/hal-04228117>.

Other scientific publications

- [21] M. Gonthier, L. Marchal and S. Thibault. ‘Memory-Aware Scheduling Of Tasks Sharing Data On Multiple GPUs’. In: ISC 2023 - ISC High Performance 2023. Hamburg, Germany, 21st May 2023. URL: <https://inria.hal.science/hal-04090595>.
- [22] M. Gonthier, L. Marchal and S. Thibault. *Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs*. Bordeaux, France, 21st Mar. 2023. URL: <https://inria.hal.science/hal-04090612>.
- [23] A. Hippolyte. ‘Vector code generation and high level programming for finite element simulation’. Talence (33): Université de Bordeaux, 1st Sept. 2023, p. 39. URL: <https://inria.hal.science/hal-04315144>.
- [24] H. Nekt. ‘Énergie ou performance : impact de l’implémentation sur la consommation’. Bordeaux-INP, 26th Oct. 2023. URL: <https://inria.hal.science/hal-04394261>.
- [25] A. d’Aviau de Piolant. ‘Etude de l’impact du power capping sur les performances des GPU’. Université de Bordeaux, 22nd June 2023. URL: <https://inria.hal.science/hal-04395148>.

11.3 Cited publications

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