

RESEARCH CENTRE

**Inria Center  
at the University of Bordeaux**

IN PARTNERSHIP WITH:

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

2022

ACTIVITY REPORT

Project-Team

STORM

## **STatic Optimizations, Runtime Methods**

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

### **DOMAIN**

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

### **THEME**

**Distributed and High Performance  
Computing**

*Inria*

# Contents

<b>Project-Team STORM</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
<b>3 Research program</b>	<b>5</b>
3.1 Parallel Computing and Architectures . . . . .	5
3.2 Scientific and Societal Stakes . . . . .	6
3.3 Towards More Abstraction . . . . .	6
<b>4 Application domains</b>	<b>7</b>
4.1 Application domains benefiting from HPC . . . . .	7
4.2 Application in High performance computing/Big Data . . . . .	7
<b>5 Highlights of the year</b>	<b>8</b>
5.1 Awards . . . . .	8
<b>6 New software and platforms</b>	<b>8</b>
6.1 New software . . . . .	8
6.1.1 Chameleon . . . . .	8
6.1.2 KStar . . . . .	9
6.1.3 AFF3CT . . . . .	10
6.1.4 VITE . . . . .	10
6.1.5 PARCOACH . . . . .	10
6.1.6 StarPU . . . . .	11
6.1.7 somp . . . . .	12
6.1.8 MIPP . . . . .	12
6.1.9 CERE . . . . .	13
6.1.10 DUF . . . . .	13
<b>7 New results</b>	<b>13</b>
7.1 MPI detach - Towards Automatic Asynchronous Local completion . . . . .	13
7.2 Code transformations for improving performance and productivity of PGAS applications . . . . .	14
7.3 Leveraging compiler analysis for NUMA and Prefetch optimization . . . . .	14
7.4 Generalizing NUMA and Prefetch optimization . . . . .	15
7.5 Optimizing energy consumption of Garbage Collectors in OpenJDK . . . . .	15
7.6 Survey on scheduling . . . . .	15
7.7 Scheduling Algorithms to Minimize the Energy Consumption of Federated Learning Devices . . . . .	15
7.8 Code optimization and generation for Cardiac simulation . . . . .	16
7.9 Predicting errors in parallel applications with ML . . . . .	16
7.10 Static Data Race Detection for MPI-RMA Programs . . . . .	17
7.11 Task scheduling with memory constraints . . . . .	17
7.12 Failure Tolerance for StarPU . . . . .	17
7.13 Energy-aware task scheduling in StarPU . . . . .	18
7.14 Scheduling iterative task graph for video games . . . . .	18
7.15 Task-based execution model for fine-grained tasks . . . . .	18
7.16 Hierarchical Tasks . . . . .	19
7.17 ADT Gordon . . . . .	19
7.18 High performance software defined radio with AFF3CT . . . . .	19
7.19 HPC Big Data Convergence . . . . .	20
7.20 Load-balancing of distributed sequential task flow programs . . . . .	20
7.21 Finite element simulation framework on a heterogeneous task-based runtime system . . . . .	21
7.22 StarPU interfacing with Tau performance evaluation toolkit . . . . .	21
7.23 Combining Uncore Frequency and Dynamic Power Capping to Improve Power Savings . . . . .	22

7.24 Study of the processor and memory power consumption of coupled sparse/dense solvers	22
<b>8 Bilateral contracts and grants with industry</b>	<b>22</b>
8.1 Bilateral contracts with industry	22
<b>9 Partnerships and cooperations</b>	<b>23</b>
9.1 International initiatives	23
9.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program	23
9.1.2 Inria associate team not involved in an IIL or an international program	24
9.1.3 Participation in other International Programs	24
9.2 International research visitors	25
9.2.1 Visits of international scientists	25
9.2.2 Visits to international teams	26
9.3 European initiatives	26
9.3.1 H2020 projects	26
9.4 National initiatives	29
9.4.1 ANR	29
9.4.2 IPL - Inria Project Lab	30
<b>10 Dissemination</b>	<b>30</b>
10.1 Promoting scientific activities	31
10.1.1 Scientific events: organisation	31
10.1.2 Scientific events: selection	31
10.1.3 Journal	31
10.1.4 Invited talks	32
10.1.5 Leadership within the scientific community	32
10.1.6 Scientific expertise	32
10.1.7 Research administration	32
10.2 Teaching - Supervision - Juries	33
10.2.1 Teaching	33
10.2.2 Supervision	34
10.2.3 Juries	35
10.3 Popularization	35
10.3.1 Internal or external Inria responsibilities	36
10.3.2 Articles and contents	36
10.3.3 Education	36
10.3.4 Interventions	36
<b>11 Scientific production</b>	<b>36</b>
11.1 Major publications	36
11.2 Publications of the year	37
11.3 Other	39
11.4 Cited publications	39

## 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.)
- A1.1.3. – Memory models
- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A2.1.7. – Distributed programming
- A2.2.1. – Static analysis
- A2.2.2. – Memory models
- A2.2.4. – Parallel architectures
- A2.2.5. – Run-time systems
- A2.2.6. – GPGPU, FPGA...

#### Other research topics and application domains

- B2.2.1. – Cardiovascular and respiratory diseases
- B3.2. – Climate and meteorology
- B3.3.1. – Earth and subsoil
- B3.4.1. – Natural risks
- B4.2. – Nuclear Energy Production
- B5.2.3. – Aviation
- B5.2.4. – Aerospace
- B6.2.2. – Radio technology
- B6.2.3. – Satellite technology
- B6.2.4. – Optic technology
- B9.2.3. – Video games

# 1 Team members, visitors, external collaborators

## Research Scientists

- Olivier Aumage [INRIA, Researcher, HDR]
- Scott Baden [UNIV CALIFORNIE, Advanced Research Position, from Apr 2022]
- Laercio Lima Pilla [CNRS, Researcher]
- Mihail Popov [Inria, Researcher]
- Emmanuelle Saillard [Inria, Researcher]

## Faculty Members

- Denis Barthou [Team leader, BORDEAUX INP, Professor, HDR]
- Marie-Christine Counilh [UNIV BORDEAUX, Associate Professor]
- Amina Guermouche [BORDEAUX INP, Associate Professor, from Sep 2022]
- Raymond Namyst [UNIV BORDEAUX, Professor, HDR]
- Samuel Thibault [UNIV BORDEAUX, Professor, HDR]
- Pierre-André Wacrenier [UNIV BORDEAUX, Associate Professor]

## PhD Students

- Célia Tassadit Ait Kaci [ATOS, CIFRE, until Feb 2022]
- Vincent Alba [UNIV BORDEAUX, from Sep 2022]
- Baptiste Coye [UBISOFT, CIFRE]
- Maxime Gonthier [Inria]
- Lise Jolicoeur [CEA, from Nov 2022]
- Alice Lasserre [Inria, from Sep 2022]
- Romain Lion [UNIV BORDEAUX]
- Gwenole Lucas [Inria]
- Van Man Nguyen [CEA]
- Diane Orhan [UNIV BORDEAUX, from Sep 2022]
- Lana Scravaglieri [IFPEN, CIFRE, from Nov 2022]
- Célia Tassadit Ait Kaci [Inria, from Mar 2022 until Sep 2022, PhD contract extension]
- Radjasouria Vinayagame [ATOS, CIFRE, from Dec 2022]

## Technical Staff

- Nathalie Furmento [CNRS, Engineer]
- Amina Guermouche [UNIV BORDEAUX, Engineer, until Aug 2022]
- Kun He [Inria, Engineer]
- Romain Lion [Inria, Engineer, until Jun 2022]
- Mariem Makni [UNIV BORDEAUX, Engineer, from Aug 2022]
- Chiheb Sakka [Inria, Engineer, until Jul 2022]
- Bastien Tagliaro [Inria, Engineer]
- Philippe Virouleau [Inria, Engineer, from Oct 2022]

## Interns and Apprentices

- Vincent Alba [Inria, Intern, from Feb 2022 until Jul 2022]
- Pélagie Alves [Inria, Intern, from Mar 2022 until Aug 2022]
- Edgar Baucher [Inria, until May 2022]
- Charles Goedefroit [Inria, from May 2022 until Jul 2022]
- Alice Lasserre [Inria, Intern, from Feb 2022 until Jul 2022]
- Charles Martin [Inria, from May 2022 until Jul 2022]
- Thomas Morin [Inria, Intern, from Aug 2022 until Sep 2022]
- Thomas Morin [Inria, until Apr 2022]
- Thomas Morin [Inria, Intern, from May 2022 until Jun 2022]
- Diane Orhan [ENS RENNES, Intern, from Feb 2022 until Jul 2022]
- Pierre-Antoine Rouby [Inria, Intern, from Feb 2022 until Jul 2022]

## Administrative Assistant

- Sabrina Duthil [Inria]

## External Collaborator

- Jean-Marie Couteyen [AIRBUS]

## 2 Overall objectives

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

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

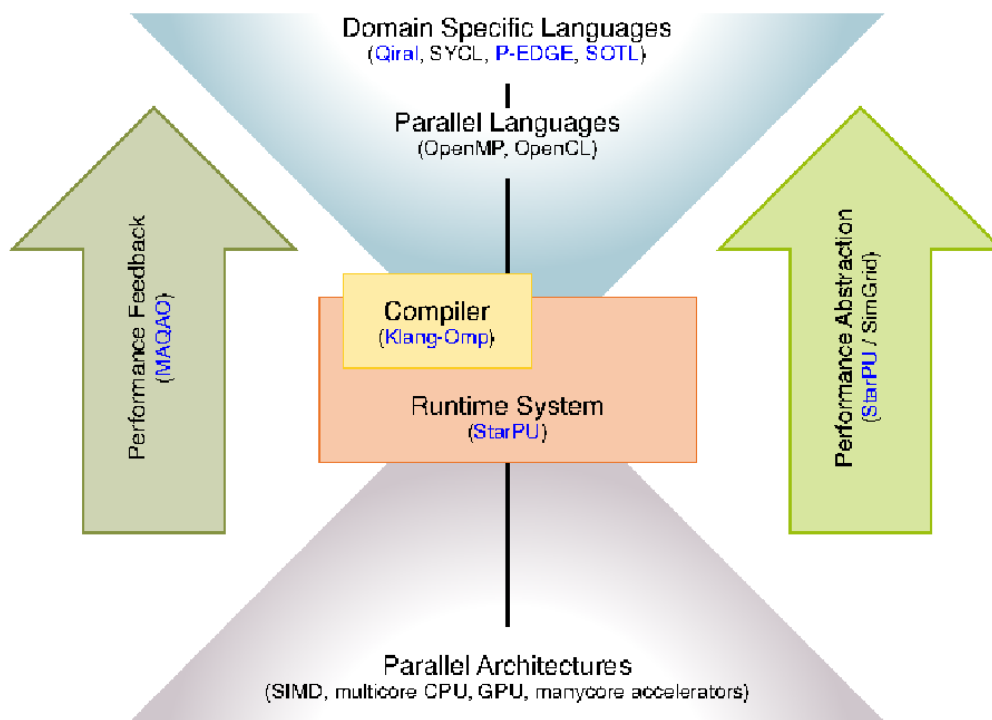


Figure 1: STORM Big Picture

1. Expressing parallelism: As shown in the following figure, we propose to work on parallelism expression through Domain Specific Languages, PGAS languages, C++ enhanced with libraries or even pragmas 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 concern 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.

The results of the team research in 2022 reflect this focus. Results presented in Sections 7.18, 7.2, 9.4.1 correspond to efforts for higher abstractions through C++ or PGAS, and for decoupling algorithmics from parallel optimizations. Static and dynamic optimizations are presented in 7.2, 7.10, 7.1 are on the optimization of communications and their parallelism, while Sections 7.5, 7.3, 7.4 focus on efficient compilation schemes for parallelism. In particular, the works presented in Sections 7.3, 7.4 describes new methods resorting to AI to improve the compiled code. Results described in Sections 7.10, 7.9 provide feed-back information, through error detection for parallel executions. The work described in Sections 7.11, 7.20 7.21 7.22 7.13, 7.19, 7.12 and 7.16 focus in particular on StarPU and its development in order to better abstract architecture, resilience, energy saving, integration in other tools and optimizations. The works described in Sections 7.14 and 7.15 correspond to scheduling methods for lightweight tasks or repetitive task graphs. Sections 7.7 and 7.13 focus on optimizations for energy savings, for AI and for HPC applications, based on scheduling optimizations. A wider automatic survey on scheduling methods is proposed in 7.6.

Finally, Section 7.17 present an on-going effort on improving the Chameleon library and strengthening its relation with StarPU and the NewMadeleine communication library. They represent real-life applications for the runtime methods we develop. Section 7.19 presents application to bigdata application, and 7.8 to cardiac simulation.

## 4 Application domains

### 4.1 Application domains benefiting from HPC

The application domains of this research are the following:

- Bioinformatics
- Health and heart disease analysis (see EXACARD 9.4.1 and Microcard project projects 9.3.1)
- Software infrastructures for Telecommunications (see AFF3CT, 7.18)
- Aeronautics (collaboration with Airbus, J.-M. Couteyen)
- Video games (collaboration with Ubisoft, see 8.1)

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

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

## 5 Highlights of the year

### 5.1 Awards

- Amina Guermouche joined the permanent team members in September 2022.
- Three PhD students of the team defended their PhD at the end of the year: Célia Tassadit Ait Kaci, Van-Man Nguyen and Romain Lion.

## 6 New software and platforms

### 6.1 New software

#### 6.1.1 Chameleon

**Keywords:** Runtime system, Task-based algorithm, Dense linear algebra, HPC, Task scheduling

**Scientific Description:** Chameleon is part of the MORSE (Matrices Over Runtime Systems @ Exascale) project. The overall objective is to develop robust linear algebra libraries relying on innovative runtime systems that can fully benefit from the potential of those future large-scale complex machines.

We expect advances in three directions based first on strong and closed interactions between the runtime and numerical linear algebra communities. This initial activity will then naturally expand to more focused but still joint research in both fields.

1. Fine interaction between linear algebra and runtime systems. On parallel machines, HPC applications need to take care of data movement and consistency, which can be either explicitly managed at the level of the application itself or delegated to a runtime system. We adopt the latter approach in order to better keep up with hardware trends whose complexity is growing exponentially. One major task in this project is to define a proper interface between HPC applications and runtime systems in order to maximize productivity and expressivity. As mentioned in the next section, a widely used approach consists in abstracting the application as a DAG that the runtime system is in charge of scheduling. Scheduling such a DAG over a set of heterogeneous processing units introduces a lot of new challenges, such as predicting accurately the execution time of each type of task over each kind of unit, minimizing data transfers between memory banks, performing data prefetching, etc. Expected advances: In a nutshell, a new runtime system API will be designed to allow applications to provide scheduling hints to the runtime system and to get real-time feedback about the consequences of scheduling decisions.

2. Runtime systems. A runtime environment is an intermediate layer between the system and the application. It provides low-level functionality not provided by the system (such as scheduling or management of the heterogeneity) and high-level features (such as performance portability). In the framework of this proposal, we will work on the scalability of runtime environment. To achieve scalability it is required to avoid all centralization. Here, the main problem is the scheduling of the tasks. In many task-based runtime environments the scheduler is centralized and becomes a bottleneck as soon as too many cores are involved. It is therefore required to distribute the scheduling decision or to compute a data distribution that impose the mapping of task using, for instance the so-called “owner-compute” rule. Expected advances: We will design runtime systems that enable an efficient and scalable use of thousands of distributed multicore nodes enhanced with accelerators.

3. Linear algebra. Because of its central position in HPC and of the well understood structure of its algorithms, dense linear algebra has often pioneered new challenges that HPC had to face. Again, dense linear algebra has been in the vanguard of the new era of petascale computing with the design of new algorithms that can efficiently run on a multicore node with GPU accelerators. These algorithms are called “communication-avoiding” since they have been redesigned to limit the amount of communication between processing units (and between the different levels of memory hierarchy). They are expressed through Direct Acyclic Graphs (DAG) of fine-grained tasks that

are dynamically scheduled. Expected advances: First, we plan to investigate the impact of these principles in the case of sparse applications (whose algorithms are slightly more complicated but often rely on dense kernels). Furthermore, both in the dense and sparse cases, the scalability on thousands of nodes is still limited, new numerical approaches need to be found. We will specifically design sparse hybrid direct/iterative methods that represent a promising approach.

Overall end point. The overall goal of the MORSE associate team is to enable advanced numerical algorithms to be executed on a scalable unified runtime system for exploiting the full potential of future exascale machines.

**Functional Description:** Chameleon is a dense linear algebra software relying on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a Runtime system. A Runtime system such as StarPU is able to manage automatically data transfers between not shared memory area (CPUs-GPUs, distributed nodes). This kind of implementation paradigm allows to design high performing linear algebra algorithms on very different type of architecture: laptop, many-core nodes, CPUs-GPUs, multiple nodes. For example, Chameleon is able to perform a Cholesky factorization (double-precision) at 80 TFlop/s on a dense matrix of order 400 000 (i.e. 4 min 30 s).

**Release Contributions:** Chameleon includes the following features:

- BLAS 3, LAPACK one-sided and LAPACK norms tile algorithms
- Support QUARK and StarPU runtime systems and PaRSEC since 2018
- Exploitation of homogeneous and heterogeneous platforms through the use of BLAS/LAPACK CPU kernels and cuBLAS/MAGMA CUDA kernels
- Exploitation of clusters of interconnected nodes with distributed memory (using OpenMPI)

**URL:** <https://gitlab.inria.fr/solverstack/chameleon>

**Contact:** Emmanuel Agullo

**Participants:** Cédric Castagnede, Samuel Thibault, Emmanuel Agullo, Florent Pruvost, Mathieu Faverge

**Partners:** Innovative Computing Laboratory (ICL), King Abdullha University of Science and Technology, University of Colorado Denver

### 6.1.2 KStar

**Name:** The KStar OpenMP Compiler

**Keywords:** Source-to-source compiler, OpenMP, Task scheduling, Compilers, Data parallelism

**Functional Description:** The KStar software is a source-to-source OpenMP compiler for languages C and C++. The KStar compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system. The KStar compiler is virtually fully compliant with OpenMP 3.0 constructs. The KStar compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

**Release Contributions:** update support for StarPU data\_lookup to account for API change

**URL:** <https://gitlab.inria.fr/kstar/kstar>

**Publications:** [hal-01517153](#), [hal-01372022](#), [hal-01081974](#)

**Contact:** Olivier Aumage

**Participants:** Nathalie Furmento, Olivier Aumage, Philippe Virouleau, Samuel Thibault

### 6.1.3 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).

**News of the Year:** The AFF3CT toolbox was successfully used to develop a the software implementation of real- time DVB-S2 transceiver. For this purpose, USRP modules were combined with multicore and SIMD CPUs. Thus some components are directly from the AFF3CT library and others such as the synchronization functions have been added. The transceiver code is portable on x86 and ARM architectures.

**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

### 6.1.4 VITE

**Name:** Visual Trace Explorer

**Keywords:** Visualization, Execution trace

**Functional Description:** ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications.

**URL:** <https://solverstack.gitlabpages.inria.fr/vite/>

**Contact:** Mathieu Faverge

**Participant:** Mathieu Faverge

### 6.1.5 PARCOACH

**Name:** PARAllel Control flow Anomaly CHecker

**Keywords:** High-Performance Computing, Program verification, Debug, MPI, OpenMP, Compilation

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

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

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

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

**Contact:** Emmanuelle Saillard

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

**Partners:** CEA, Bull - Atos Technologies

### 6.1.6 StarPU

**Name:** The StarPU Runtime System

**Keywords:** Multicore, GPU, Scheduling, HPC, Performance

**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.

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

**Publications:** [hal-02943753](#), [hal-02970529](#), [hal-02985721](#), [hal-03290998](#), [hal-03552243](#), [hal-03273509](#), [hal-03773486](#), [inria-00378705](#), [inria-00384363](#), [inria-00411581](#), [inria-00421333](#), [inria-00467677](#), [inria-00523937](#), [inria-00547614](#), [inria-00547616](#), [inria-00550877](#), [inria-00590670](#), [inria-00606195](#), [inria-00606200](#), [inria-00619654](#), [hal-00643257](#), [hal-00648480](#), [hal-00654193](#), [hal-00661320](#), [hal-00697020](#), [hal-00725477](#), [hal-00772742](#), [hal-00773114](#), [hal-00773610](#), [hal-00776610](#), [tel-00777154](#), [hal-00803304](#), [hal-00807033](#), [hal-00824514](#), [hal-00853423](#), [hal-00858350](#), [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-01181135](#), [hal-01182746](#), [hal-01223573](#), [hal-01283949](#), [hal-01284004](#), [hal-01284136](#), [hal-01284235](#), [hal-01332774](#), [hal-01353962](#), [hal-01355385](#), [hal-01361992](#), [hal-01372022](#), [hal-01386174](#), [hal-01409965](#), [hal-01410103](#), [hal-01473475](#), [hal-01474556](#), [tel-01483666](#), [hal-01502749](#), [hal-01517153](#), [tel-01538516](#), [hal-01616632](#), [hal-01618526](#), [hal-01718280](#), [tel-01816341](#), [hal-01842038](#), [tel-01959127](#), [hal-02275363](#), [hal-02296118](#), [hal-02403109](#), [hal-02421327](#), [hal-02872765](#), [hal-02914793](#), [hal-02933803](#)

**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

### 6.1.7 somp

**Name:** SOMP

**Keywords:** Simulation, Task scheduling, OpenMP

**Functional Description:** sOMP is a simulator for task-based applications running on shared-memory architectures, utilizing the SimGrid framework. The aim is to predict the performance of applications on various machine designs while taking different memory models into account, using a trace from a sequential execution.

**URL:** <https://gitlab.inria.fr/idaoudi/omps/-/wikis/home>

**Publications:** [hal-02933803](#), [hal-03177026v2](#)

**Contact:** Samuel Thibault

**Participant:** Idriss Daoudi

### 6.1.8 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.

**News of the Year:** [2021] Prototyping of RISC-V RVV vector intrinsic support.

**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

### 6.1.9 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

### 6.1.10 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 Guerrouche

## 7 New results

### 7.1 MPI detach - Towards Automatic Asynchronous Local completion

**Participants:** V.M. Nguyen, E. Saillard, D. Barthou.



When aiming for large-scale parallel computing, waiting time due to network latency, synchronization, and load imbalance are the primary opponents of high parallel efficiency. A common approach to hide latency with computation is the use of non-blocking communication. In the presence of a consistent load imbalance, synchronization cost is just the visible symptom of the load imbalance. Tasking approaches as in OpenMP, TBB, OmpSs, or C++20 coroutines promise to expose a higher degree of concurrency, which can be distributed on available execution units and significantly increase load balance. Available MPI non-blocking functionality does not integrate seamlessly into such tasking parallelization. In this work, we present a slim extension of the MPI interface to allow seamless integration of non-blocking communication with available concepts of asynchronous execution in OpenMP and C++. Using our concept allows to span task dependency graphs for asynchronous execution over the full distributed memory application. We furthermore investigate compile-time analysis necessary to transform an application using blocking MPI communication into an application integrating OpenMP tasks with our proposed MPI interface extension [4].

## 7.2 Code transformations for improving performance and productivity of PGAS applications

**Participants:** S. Baden, E. Saillard, D. Barthou, O. Aumage.

The PGAS model is an attractive means of treating irregular fine-grained communication on distributed memory systems, providing a global memory abstraction that supports low-overhead Remote Memory Access (RMA), direct access to memory located in remote address spaces. RMA performance benefits from hardware support generally provided by modern high performance communication networks, delivering low over-head communication needed in irregular applications such as Metagenomics.

The proposed research program will apply source-to-source transformation to PGAS code. The project will target the UPC++ library [37], a US Department of Energy Exascale Computing Project that the applicant lead for three years at Lawrence Berkeley National Laboratory (LBNL) in Berkeley, California (USA). Source-to-source transformations will be investigated, and a translator will be built that realizes the transformations. UPC++ is open-source and the LBNL development team actively supports the software. The goal of the project is to investigate a source-to-source translator for re-structuring UPC++ code to improve performance. Four transformations will be investigated, as described next: Localization; Communication aggregation; Overlap communication with computation; Adaptive push-pull algorithms.

This work is in the context of the International Inria Chair for Scott Baden.

## 7.3 Leveraging compiler analysis for NUMA and Prefetch optimization

**Participants:** M. Popov, E. Saillard.

Performance counter based characterization is currently used to optimize the complex search space of threads and data mapping along with prefetchers on NUMA systems. While optimizing such spaces provides significant performance gains, it requires to dynamically profile applications resulting in huge a characterization overhead. We started a collaboration with the University of Iowa to reduce this overhead by using insights from the compiler. We develop a new static analysis method that characterizes the LLVM Intermediate Representation: it extracts information and exposes it to a deep learning network to optimize new unseen applications across the NUMA/prefetch space.

We demonstrated that the statically based optimizations achieve 80% of the gains compared to a dynamic approach but without any costly profiling. We further evaluated a hybrid model which predicts whether to use static or dynamic characterization. The hybrid model achieves similar gains as the dynamic model but only profiles 30% of the applications. These results were published at IPDPS 2022 [13]. We are currently investigating alternative methods for code embedding such as ir2vec.

## 7.4 Generalizing NUMA and Prefetch optimization

**Participants:** O. Aumage, A. Guermouche, L. Lima Pilla, M. Popov, E. Saillard, L. Scravaglieri.

In addition to the NUMA/prefetch characterization, we also studied how such optimizations generalize. Applications behavior change depending on their inputs. Therefore, an optimal configuration (thread and data mapping along with prefetch) for a fixed input might not be optimal when we change the inputs. We quantified how changing the inputs impacts the performance gains compared to a native per-input optimization. We also studied the energy impact of NUMA/prefetch and characterized how cross-inputs optimizations also affect energy.

We observed that applications can lose more than 25% of the gains due to input changes. We also showed that energy is more affected by NUMA/prefetch than performance and must to be optimized separately. Optimizing performance and energy provide on average 1.6x and 4x gains respectively. Performance-based optimizations can lose up to 50% of the energy gains compared to native energy-based optimizations. These results were published [29] and we plan to extend them in a journal.

These explorations were conducted thanks to the framework CERE 6.1.9 which is currently updated to a recent OS and LLVM version.

## 7.5 Optimizing energy consumption of Garbage Collectors in OpenJDK

**Participants:** M. Popov.

Recent studies demonstrate how important it is to select an efficient Java Virtual Machine to optimize the energy consumption. In our research, we further pushed the energy characterization and optimization by focusing on the Garbage Collector (GC) selection. In particular, we quantified the energy impact of existing garbage collectors, provided insights for developers to assist them in selecting the most efficient GC for their needs, and developed machine learning models that predict GCs to use for energy saving from simple performance counters [12].

## 7.6 Survey on scheduling

**Participants:** L. Lima Pilla, M. 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. This work is being developed in collaboration with researchers from the University of Basel.

## 7.7 Scheduling Algorithms to Minimize the Energy Consumption of Federated Learning Devices

**Participants:** L. 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 [28]. These ideas were presented on a Dagstuhl seminar on Power and Energy-aware Computing on Heterogeneous Systems.

We have also adapted the dynamic programming solution to the Multiple-Choice Minimum-Cost Maximal Knapsack Packing Problem in order to minimize the makespan of data-parallel applications on heterogeneous platforms, reducing the complexity to find an optimal solution by three orders of magnitude when compared to the state of the art [27].

## 7.8 Code optimization and generation for Cardiac simulation

**Participants:** C. Sakka, V. Alba, E. Saillard, D. Barthou, A. Guermouche, M.-C. Counilh, O. Aumage.

In the context of the ANR Exacard project, we optimized the code Propag for the cardiac electro-physiology simulation. We used for this MIPP [38] for writing the different SIMD kernels and optimized for the GPU. The results have been presented in the conference "Computing in Cardiology" [16].

Besides, during the M2 internship of V.Alba in the context of the European project Microcard [30], Vincent studied how to use StarPU in the code generated for the ionic model simulation. He also studied the impact of precision on the computation, and storage of the input data on the precision of the final results, for the ionic model simulation. The tool Verificarlo [39] was integrated into the compilation chain of OpenCarp for a fine-grained precision analysis. The results have been presented in the Microcard workshop in July 2022.

## 7.9 Predicting errors in parallel applications with ML

**Participants:** M. Popov, E. 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. Preliminary results over MBI show that we can train models that detect if a code is correct with 90% accuracy.

We plan to extend these codes by considering more benchmarks, either from other benchmarks suites for verification (e.g., DataRaceBench) or from student projects. We have the intuition that student

projects, in particular, contain valuable representative errors that beginner developers are susceptible to do.

Finally, we also consider crawling through Github to directly extract correct/incorrect codes. The idea is to associate issue descriptions to specific commits to get both a bug description as well as a code fix. We can further use natural language processing techniques to characterize the text description of the bug in the issue.

This work is also a collaboration with the Iowa State University.

## 7.10 Static Data Race Detection for MPI-RMA Programs

**Participants:** C.T. Ait Kaci, E. Saillard, D. Barthou.

Communications are a critical part of HPC simulations, and one of the main focuses of application developers when scaling on supercomputers. While classical message passing (also called two-sided communications) is the dominant communication paradigm, one-sided communications are often praised to be efficient to overlap communications with computations, but challenging to program. Their usage is then generally abstracted through languages and memory abstractions to ease programming (e.g. PGAS). Therefore, little work has been done to help programmers use intermediate runtime layers, such as MPI-RMA, that is often reserved to expert programmers. Indeed, programming with MPI-RMA presents several challenges that require handling the asynchronous nature of one-sided communications to ensure the proper semantics of the program while ensuring its memory consistency. To help programmers detect memory errors such as race conditions as early as possible, we propose a new static analysis of MPI-RMA codes that shows to the programmer the errors that can be detected at compile time [11]. The detection is based on a local concurrency errors detection algorithm that tracks accesses through BFS searches on the Control Flow Graphs of a program and is complementary to the runtime analysis developed before. We show on several tests and an MPI-RMA variant of the GUPS benchmark that the static analysis allows to detect such errors on user codes. The error codes are integrated in the MPI Bugs Initiative open-source test suite.

## 7.11 Task scheduling with memory constraints

**Participants:** M. Gonthier, S. 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. We extended previous results from mono-GPU to multi-GPU, and introduced a dynamic strategy with a locality-aware principle, and published results at IPDPS [8]. Previous mono-GPU results with a static ordering are currently submitted to the FGCS journal, pending minor revision, a preprint is available as RR [26].

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

## 7.12 Failure Tolerance for StarPU

**Participants:** A. Guermouche, R. Lion, S. Thibault.

Since supercomputers keep growing in terms of core numbers, the reliability decreases the same way. The project H2020 EXA2PRO and more precisely the PhD thesis of Romain Lion aimed to propose solutions for the failure tolerance problem, including StarPU. While exploring decades of research about the resilience techniques, we have identified properties in our task-based runtime's paradigm that can be exploited in order to propose a solution with lower overhead than the generic existing ones. Romain Lion finished writing his PhD thesis and defended it, which reveals itself a very good overview of how resilience can be integrated with task-based execution and leverage information from the task graph to optimize fault-tolerance support. Efforts on this support will continue in the context of the MicroCard project.

Results have been published in [40].

### 7.13 Energy-aware task scheduling in StarPU

**Participants:** A. Guermouche, M. Makni, S. Thibault.

In the context of the EXA2PRO project and the visit of A. Guermouche, we have investigated the time/energy behavior of several dense linear algebra kernels. We have found that they can exhibit largely different compromises, which raised the question of revising task scheduling to take into account energy efficiency. We have improved StarPU's ability to integrate energy performance models, and integrated helpers for performing energy measurement even with the coarse-grain support provided by the hardware. We have shown that the energy/time Pareto front can be presented to the application user, to decide which compromise should be chosen.

### 7.14 Scheduling iterative task graph for video games

**Participants:** B. Coye, D. Barthou, L. Lima Pilla, R. Namyst.

In the context of Baptiste Coye's PhD started in March 2020 in partnership with Ubisoft, Baptiste has studied the task graph and their scheduling for real games. The transition to a more modular architecture, with a central scheduler, is currently under study. There are opportunities for optimization stemming from the fact that the task graph is repeatedly executed each time frame, with few modifications from one frame to the next. The tasks have varying execution times, but differ only slightly from one frame to the other. Moreover, some tasks can be rescheduled from one time frame to the next. Taking into account these constraints the current research effort is on how to better define the tasks and their dependencies and constraints, and then propose an incremental modification of the task graph [10] [5].

### 7.15 Task-based execution model for fine-grained tasks

**Participants:** C. Castes, E. Saillard, O. Aumage.

Sequential task-based programming models paired with advanced runtime systems allow the programmer to write a sequential algorithm independently of the hardware architecture in a productive and portable manner, and let a third party software layer—the runtime system—, deal with the burden of scheduling a correct, parallel execution of that algorithm to ensure performance. Developing algorithms that specifically require fine-grained tasks along this model is still considered prohibitive, however, due to per-task management overhead, forcing the programmer to resort to a less abstract, and hence more

complex “task+X” model. We thus investigated the possibility to offer a tailored execution model, trading dynamic mapping for efficiency by using a decentralized, conservative in-order execution of the task flow, while preserving the benefits of relying on the sequential task-based programming model. We proposed a formal specification of the execution model as well as a prototype implementation, which we assess on a shared-memory multicore architecture with several synthetic workloads. The results showed that under the condition of a proper task mapping supplied by the programmer, the pressure on the runtime system is significantly reduced and the execution of fine-grained task flows is much more efficient. These results have been published in [6] and [24].

## 7.16 Hierarchical Tasks

**Participants:** N. Furmento, G. Lucas, R. Namyst, S. Thibault, P.A. Wacrenier.

Task-based systems have gained popularity as they promise to exploit the computational power of complex heterogeneous systems. StarPU is based on the Sequential Task Flow (STF) model, which, unfortunately, has the intrinsic limitation of supporting static task graphs only. This leads to potential submission overhead and to a static task graph not necessarily adapted for execution on heterogeneous systems. A standard approach is to find a trade-off between the granularity needed by accelerator devices and the one required by CPU cores to achieve performance.

To address these problems, we have extended the STF model of StarPU to enable tasks subgraphs at runtime. We refer to these tasks as hierarchical tasks. This approach allows for a more dynamic task graph. Combined with an automatic data manager, it allows to dynamically adapt the granularity to meet the optimal size of the targeted computing resource.

We have shown that the model is correct and provided an early evaluation on shared memory heterogeneous systems, using the Chameleon dense linear algebra library.

These results have been published in [7, 25, 14].

## 7.17 ADT Gordon

**Participants:** O. Aumage, N. Furmento, S. Thibault.

In collaboration with the HIEPACS and TADAAM Inria teams, we have strengthened the relations between the Chameleon linear algebra library from HIEPACS, our StarPU runtime scheduler, and the NewMadeleine high-performance communication library from TADAAM. More precisely, we have improved the interoperation between StarPU and NewMadeleine, to more carefully decide when NewMadeleine should proceed with communications. We have then introduced the notion of dynamic collective operations, which opportunistically introduce communication trees to balance the communication load. We have also evaluated the Chameleon + StarPU stack in the context of a biodiversity application of the PLEIADE team. Our stack proved to be able to process very large matrices (more than a million matrix side size), which was unachievable before, with reasonable processing time. We had to carefully integrate the I/O required for loading the matrices with the computation themselves. We have completed the writing of a common journal article, currently under submission, a preprint is available [20].

## 7.18 High performance software defined radio with AFF3CT

**Participants:** A. Cassagne, D. Barthou, D. Orhan, L. Lima Pilla, O. Aumage.

The AFF3CT library 6.1.3 developed jointly between IMS and the STORM team, which aims to model error correcting codes for numerical communications has been further improved in different ways. The

automatic parallelization of the tasks describing the simulation of a whole chain of signal transmission has been designed, using a Domain Specific Language. This allows the development of Software Defined Radio and has been put to work on use case with Airbus.

A developed 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 [23].

In the context of D. Orhan's internship and PhD, we have started to investigate the challenges related to the partition of chains of signal transmission in order to maximize their throughput using the minimal number of resources necessary. We have proposed a heuristic to automatically partition the chain into a pipeline and to choose how many resources are dedicated to each of its stages [34]. We are now working on an optimal algorithm for this problem and its performance measurement based on the previous DSEL.

## 7.19 HPC Big Data Convergence

**Participants:** O. Aumage, N. Furmento, K. He, S. Thibault.

This work is partly done within the framework of the project hpc-scalable-ecosystem from région Nouvelle Aquitaine. It is a collaboration with members of the Hiepac team and the LaBRI.

A Java interface for StarPU has been implemented and allows to execute Map Reduce applications on top of StarPU. We have made some preliminary experiments on Cornac, a big data application for visualising huge graphs.

We have also developed a new C++ library, called Yarn++, to allow the execution of HPC applications on Big Data clusters. We are doing initial evaluation with a FMM application written on top of StarPU on the PlaFRIM platform.

In the context of the HPC-BIGDATA IPL, a Python interface for StarPU has been completed, to allow executing Python tasks on top of StarPU. This allows to close the gap between the HPC and BigData communities by allowing the latter to directly execute their applications with the runtime system of the former. The challenge at stake was that the Python interpreter itself is not parallel, so data has to be transferred from one interpreter to another. We improved the TCP/IP-based master-slave distribution to support fully asynchronous transfers. We implemented memory mapping support, which allows to entirely avoid data transfers for specific types that implement memory views, such as the largely-used numpy arrays.

Also in the context of the HPC-BIGDATA IPL, we have continued integrating a machine-learning-based task scheduler (i.e. BigData for HPC), designed by Nathan Grinsztajn (from Lille) in the StarPU runtime system. The results which Nathan obtained in pure simulation were promising, the integration posed a lot of detailed issues, for instance StarPU pipelines task execution so as to prefetch data and overlap cosets, which the neural network was not prepared for.

In the context of the TEXTAROSSA project [3] and in collaboration with the TOPAL team, we have continued using the StarPU runtime system to execute machine-learning applications on heterogeneous platforms (i.e. HPC for BigData). We have started leveraging the onnx runtime with StarPU, so as to benefit from its CPU and CUDA kernel implementations as well as neural network support.

## 7.20 Load-balancing of distributed sequential task flow programs

**Participants:** Pélágie Alves, Laercio Lima Pilla, Olivier Aumage, Nathalie Furmento, Marie-Christine Counilh.

The StarPU runtime system developed by the STORM team implements the sequential task flow (STF) programming model, whereby applications are decomposed into tasks that are submitted in a sequential order and are then scheduled and executed in parallel by the runtime system. In the case where the computing platform is composed of multiple nodes (that is, without a shared memory), StarPU extends the STF model by submitting a single task graph to all nodes. However, it either offers a fully distributed execution model, but lets the application handle the distributed load balancing (while still taking in charge the scheduling of tasks within each node), or it takes responsibility for the distributed scheduling work, but it then manages the execution in a centralized, master-worker model. The centralized execution model is similarly employed by other runtime systems from the state of the art, and it may pose a challenge to their scalability in future computing platforms. In the fully distributed execution model, the application supplies a data distribution over the participating nodes, and StarPU then uses this data distribution to decide about the tasks mapping on these nodes. The application may therefore control the load balancing by altering this data distribution over the course of the execution lifespan. The fully distributed execution model is scalable by design, but the added value offered by StarPU to applications in this model is currently limited. We thus developed [31] a prototype extension of the fully distributed execution model of StarPU to design and experiment with the logic to take charge of the distributed load balancing work on behalf of applications, that is, to trigger data redistribution actions to ensure a balanced workload. Since a piece of data may be referred to from multiple tasks and a task may refer to multiple pieces of data, the main challenge is to capture the potentially complex relationship between the data distribution and the resulting work distribution in terms of tasks.

### 7.21 Finite element simulation framework on a heterogeneous task-based runtime system

**Participants:** Thomas Morin, Olivier Aumage, James Trotter, Xing Cai, Laercio Lima Pilla, Mihail Popov, Amina Guermouche, Denis Barthou.

The FEniCSx computing platform, of which SIMULA (Oslo, Norway) is the major contributor, is a popular open-source (LGPLv3) computing platform for solving partial differential equations. FEniCSx enables users to quickly translate scientific models into efficient finite element code. It provides scientific application programmers with a high-level Python and C++ front-end programming model that lets them express simulations in a formalism close to their mathematical expression. It allows them to focus on matters relevant to the problem being modeled, while abstracting away most computer-related technical details in its back-end. We investigated [33] a cooperation between FEniCSx's new DOLFINx computational back-end and the StarPU task-based runtime system developed by STORM. The DOLFINx C++ software layer is in charge of interfacing FEniCS applications with external HPC libraries such as PETSC for numerical aspects, ParMETIS and SCOTCH for mesh partitioning, as well as with MPI and OpenMP supports. The software architecture of FEniCS, involving a local, dense matrix level and the assembly of local matrices into a sparse global matrix, offers several alternative strategies for a port of DOLFINx on top of StarPU. The purpose of this work was to determine the relative benefits and drawbacks of each approach for a port of select parts of DOLFINx on StarPU, on heterogeneous CPU+GPU computing platforms.

### 7.22 StarPU interfacing with Tau performance evaluation toolkit

**Participants:** Olivier Aumage, Camille Coti.



We developed an interface between the StarPU task-based runtime system and the performance evaluation toolkit Tau (see: [TAU Performance System](#)). This interface involved the design of an extension to StarPU's performance metrics collection API to enable event-related data collection [22].

### 7.23 Combining Uncore Frequency and Dynamic Power Capping to Improve Power Savings

**Participants:** Amina Guermouche.

The US Department of Energy sets a limit of 20 to 30 MW for future exascale machines. In order to control their power consumption, modern processors provide many features. Power capping and uncore frequency scaling are examples of such features which allow to limit the power consumed by a processor. In this work [9], we propose to combine dynamic power capping to uncore frequency scaling. We propose DUF, an extension of DUE, an existing tool which dynamically adapts uncore frequency. DUF dynamically adapts the processor power cap to the application needs. Finally, just like DUE, DUF can tolerate performance loss up to a user-defined limit. With a controlled impact on performance, DUF is able to provide power savings with no energy loss. The evaluation of DUF shows that it manages to stay within the user-defined slowdown limits for most of the studied applications. Moreover, combining uncore frequency scaling to power capping: (i) improves power consumption by up to 13.98 % with additional energy savings for applications where uncore frequency scaling has a limited impact, (ii) improves power consumption by up to 7.90 % compared to using uncore frequency scaling by itself and (iii) leads to more than 5 % power savings at 5 % tolerated slowdown with no energy loss for most applications.

### 7.24 Study of the processor and memory power consumption of coupled sparse/dense solvers

**Participants:** Emmanuel Agullo, Marek Felšöci, Amina Guermouche, Hervé Mathieu, Guillaume Sylvand, Bastien Tagliaro.

In the aeronautical industry, aeroacoustics is used to model the propagation of acoustic waves in air flows enveloping an aircraft in flight. This for instance allows one to simulate the noise produced at ground level by an aircraft during the takeoff and landing phases, in order to validate that the regulatory environmental standards are met. Unlike most other complex physics simulations, the method resorts to solving coupled sparse/dense systems. In a previous study, we proposed two classes of algorithms for solving such large systems on a relatively small workstation (one or a few multicore nodes) based on compression techniques. The objective of this study [21] is to assess whether the positive impact of the proposed algorithms on time to solution and memory usage translates to the energy consumption as well. Because of the nature of the problem, coupling dense and sparse matrices, and the underlying solutions methods, including dense, sparse direct and compression steps, this yields an interesting processor and memory power profile which we aim to analyze in details.

## 8 Bilateral contracts and grants with industry

### 8.1 Bilateral contracts with industry

**Participants:** Denis Barthou, Emmanuelle Saillard, Raymond Namyst, Olivier Aumage, Mihail Popov.

- Contract with ATOS/Bull for the PhD CIFRE of Célia Tassadit Ait Kaci (2019-2022), Radjasouria VINAYAGAME (2022-2025)
- Contract with Ubisoft for the PhD CIFRE of Baptiste Coye (2020-2023),
- Contract with CEA for the PhD of Van Man Nguyen (2019-2022), P.Beziau (2018-2021) and other short contracts
- Contract with IFPEN for the PhD of Lana Scravaglieri (2022-2025).
- Contract with Qarnot (Défi PULSE 2022-2024)
- Contract "Plan de relance" with Atos to get an engineer "a disposition" working on detecting hardware errors in clusters.
- Contract "Plan de relance" with Atos to hire an engineer working on the integration of Célia Tassadit Ait Kaci's PhD work in PARCOACH.

## 9 Partnerships and cooperations

### 9.1 International initiatives

#### 9.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

##### COHPC

**Participants:** Emmanuelle Saillard, Denis Barthou, Scott Baden, Van Man Nguyen, Célia Tassadit Ait Kaci.

**Title:** Correctness and Performance of HPC Applications

**Duration:** 2019 -> 2022

**Coordinator:** Costin Iancu (cciancu@lbl.gov)

**Partners:**

- Lawrence Berkeley National Laboratory Berkeley (États-Unis)

**Inria contact:** Emmanuelle Saillard

**Summary:** High Performance Computing (HPC) plays an important role in many fields like health, materials science, security or environment. The current supercomputer hardware trends lead to more complex HPC applications (heterogeneity in hardware and combinations of parallel programming models) that pose programmability challenges. As indicated by a recent US DOE report, progress to Exascale stresses the requirement for convenient and scalable debugging and optimization methods to help developers fully exploit the future machines; despite all recent advances these still remain manual complex tasks.

This collaboration aims to develop tools to aid developers with problems of correctness and performance in HPC applications for Exascale systems. There are several requirements for such tools: precision, scalability, heterogeneity and soundness. In order to improve developer productivity, we aim to build tools for guided code transformations (semi-automatic) using a combination of static and dynamic analysis. Static analysis techniques will enable soundness and scalability in execution time. Dynamic analysis techniques will enable precision, scalability in LoCs and heterogeneity for hybrid parallelism. A key aspect of the collaboration is to give precise feedback to developers in order to help them understand what happens in their applications and facilitate the debugging and optimization processes.

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

#### Maelstrom

**Participants:** Olivier Aumage, Denis Barthou, Amina Guermouche, Laercio Lima Pilla, Thomas Morin, Mihail Popov.

**Title:** Maelstrom

**Duration:** 2022 ->

**Coordinator:** Olivier Aumage (olivier.aumage@inria.fr), Xing Cai (xingcai@simula.no)

**Partners:**

- Simula (Norvège)

**Inria contact:** Olivier Aumage

**Summary:** Scientific simulations are a prominent means for academic and industrial research and development efforts nowadays. Such simulations are extremely computing intensive due to the process involved in expressing modelled phenomena in a computer-enabled form. Exploiting supercomputer resources is essential to compute the high quality simulations in an affordable time. However, the complexity of supercomputer architectures makes it difficult to exploit them efficiently.

The FEniCS computing platform, of which Simula is the major contributor, 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.

Inria's Team Storm develops methodologies and tools to statically and dynamically optimize computations on HPC architectures, ranging from task-based parallel runtime systems to vector processing techniques, from performance-oriented scheduling to energy consumption reduction.

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

### 9.1.3 Participation in other International Programs

#### 3BEARS

**Participants:** L. Lima Pilla, M. Popov.

**Title:** Broad Bundle of Benchmarks for Scheduling in HPC, Big Data, and ML

**Partners:** • University of Basel, Switzerland

**Date/Duration:** 2021–2022

**Coordinator:** Laércio Lima Pilla, Florina Ciorba (UniBas)

**Program:** Partenariat Hubert Curien - Germaine de Staël

**Summary:** This project's main goal is to develop ways to co-design parallel applications and scheduling algorithms in order to achieve high performance and optimize resource utilization. Parallel applications nowadays are a mix of High-Performance Computing (HPC), Big Data, and Machine Learning (ML) software. They show varied computational profiles, being compute-, data-, I/O-intensive, or a combination thereof. Because of the varied nature of their parallelism, their performance can degrade due to factors such as synchronization, management of parallelism, communication, and load imbalance. In this situation, scheduling has to be done with care as to avoid causing new performance problems (e.g., fixing load imbalance may degrade communication performance). The main objective of this collaboration is to create the 3BEARS Benchmark Suite, a set of scheduling test applications to be freely available to the community to allow the study of state-of-the-art algorithms and enable the co-design of parallel applications and scheduling algorithms. The benchmarks will include applications from HPC, Big Data, and Machine Learning. Current efforts are dedicated to the identification and characterization of diverse scheduling problems in order to find the most representative benchmarks for the suite.

## 9.2 International research visitors

### 9.2.1 Visits of international scientists

#### Inria International Chair

- Scott Baden, U. California, visited the team two months, in April-May 2022

#### Other international visits to the team

- Xing Cai  
**Status** (Professor)  
**Institution of origin:** Simula  
**Country:** Norway  
**Dates:** 14-15 April  
**Context of the visit:** Maelstrom associate team
- Johannes Langguth  
**Status** (Researcher)  
**Institution of origin:** Simula  
**Country:** Norway  
**Dates:** 14-15 April  
**Context of the visit:** Maelstrom associate team
- James Trotter  
**Status** (Post-Doc)  
**Institution of origin:** Simula  
**Country:** Norway  
**Dates:** 14-15 April  
**Context of the visit:** Maelstrom associate team
- Kristian Hustad  
**Status** (PhD)  
**Institution of origin:** Simula  
**Country:** Norway

**Dates:** 14-15 April

**Context of the visit:** Maelstrom associate team

- David Black-Schaffer and Chang Hyun Park:

**Status** (Professor, Assistant Professor)

**Institution of origin:** Uppsala University

**Country:** Sweden

**Dates:** 4-5 May

## 9.2.2 Visits to international teams

### Research stays abroad

#### Maxime Gonthier

**Visited institution:** University of Uppsala

**Country:** Sweden

**Dates:** April-June

**Context of the visit:** Collaboration with Elisabeth Larsson and Carl Nettelblad

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

## 9.3 European initiatives

### 9.3.1 H2020 projects

**PRACE-6IP** [PRACE-6IP project on cordis.europa.eu](https://cordis.europa.eu/project/prace-6ip)

**Title:** PRACE 6th Implementation Phase Project

**Duration:** From May 1, 2019 to December 31, 2022

#### Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- CENTRUM SPOLOCNYCH CINNOSTI SLOVENSKEJ AKADEMIE VIED (CENTRE OF OPERATIONS OF THE SLOVAK ACADEMY OF SCIENCES), Slovakia
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (GENCI), France
- UNIVERSIDADE DO MINHO (UMINHO), Portugal
- LINKOPINGS UNIVERSITET (LIU), Sweden
- VSB - TECHNICAL UNIVERSITY OF OSTRAVA (VSB - TU Ostrava), Czechia
- MACHBA - INTERUNIVERSITY COMPUTATION CENTER (IUCC), Israel
- TECHNISCHE UNIVERSITAET WIEN (TU WIEN), Austria
- Gauss Centre for Supercomputing (GCS) e.V. (GCS), Germany
- FUNDACION PUBLICA GALLEGA CENTRO TECNOLOGICO DE SUPERCOMPUTACION DE GALICIA (CESGA), Spain
- UNIVERSITEIT ANTWERPEN (UANTWERPEN), Belgium
- NATIONAL UNIVERSITY OF IRELAND GALWAY (NUI GALWAY), Ireland
- AKADEMIA GORNICZO-HUTNICZA IM. STANISLAWA STASZICA W KRAKOWIE (AGH / AGH-UST), Poland

- KUNGLIGA TEKNISKA HOEGSKOLAN (KTH), Sweden
- FORSCHUNGSZENTRUM JULICH GMBH (FZJ), Germany
- EUDAT OY (EUDAT), Finland
- KORMANYZATI INFORMATIKAI FEJLESZTESI UGYNOKSEG (GOVERNMENTAL INFORMATION TECHNOLOGY DEVELOPMENT AGENCY), Hungary
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), France
- NATIONAL INFRASTRUCTURES FOR RESEARCH AND TECHNOLOGY (GRNET S.A.), Greece
- GEANT VERENIGING (GEANT VERENIGING), Netherlands
- UNIVERSIDADE DE EVORA (UNIVERSIDADE DE EVORA), Portugal
- KOBENHAVNS UNIVERSITET (UCPH), Denmark
- UPPSALA UNIVERSITET (UU), Sweden
- INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK, Poland
- ISTANBUL TEKNIK UNIVERSITESI (ITU), Türkiye
- BAYERISCHE AKADEMIE DER WISSENSCHAFTEN (BADW), Germany
- SURF BV, Netherlands
- ASSOCIACAO DO INSTITUTO SUPERIOR TECNICO PARA A INVESTIGACAO E DESENVOLVIMENTO (IST ID), Portugal
- PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE AISBL (PRACE), Belgium
- UMEA UNIVERSITET, Sweden
- UNIVERSIDADE DE COIMBRA (UNIVERSIDADE DE COIMBRA), Portugal
- UNITED KINGDOM RESEARCH AND INNOVATION (UKRI), United Kingdom
- UNIVERSITE DU LUXEMBOURG (uni.lu), Luxembourg
- EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH (ETH Zürich), Switzerland
- SYDDANSK UNIVERSITET (SDU), Denmark
- "ASSOCIATION " "NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS" (NCSA), Bulgaria
- BILKENT UNIVERSITESI VAKIF (BILKENTUNIVERSITY BILIM KENTI), Türkiye
- UNIVERSITETET I OSLO (UNIVERSITY OF OSLO), Norway
- DANMARKS TEKNISKE UNIVERSITET (DTU), Denmark
- UNIVERSIDADE DO PORTO (U.PORTO), Portugal
- SIGMA2 AS (SIGMA2), Norway
- UNIVERSITY OF STUTTGART (USTUTT), Germany
- MAX-PLANCK-GESELLSCHAFT ZUR FORDERUNG DER WISSENSCHAFTEN EV (MPG), Germany
- UNIVERSITAET INNSBRUCK (UIBK), Austria
- CINECA CONSORZIO INTERUNIVERSITARIO (CINECA), Italy
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (CNRS), France
- CENTRE INFORMATIQUE NATIONAL DE L'ENSEIGNEMENT SUPERIEUR (CINES), France
- POLITECHNIKA WROCLAWSKA (PWR), Poland
- POLITECHNIKA GDANSKA (GDANSK TECH), Poland
- NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET NTNU (NTNU), Norway
- THE CYPRUS INSTITUTE (THE CYPRUS INSTITUTE), Cyprus
- THE UNIVERSITY OF EDINBURGH (UEDIN), United Kingdom

- UNIVERZA V LJUBLJANI (UL), Slovenia
- BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION (BSC CNS), Spain
- CSC-TIETEEN TIETOTEKNIKAN KESKUS OY (CSC-IT CENTER FOR SCIENCE LTD), Finland

**Inria contact:** Jean ROMAM; Frédéric DESPREZ; Luc GIRAUD

**Coordinator:**

**Summary:** PRACE, the Partnership for Advanced Computing is the permanent pan-European High Performance Computing service providing world-class systems for world-class science. Systems at the highest performance level (Tier-0) are deployed by Germany, France, Italy, Spain and Switzerland, providing researchers with more than 17 billion core hours of compute time. HPC experts from 25 member states enabled users from academia and industry to ascertain leadership and remain competitive in the Global Race. Currently PRACE is finalizing the transition to PRACE 2, the successor of the initial five year period. The objectives of PRACE-6IP are to build on and seamlessly continue the successes of PRACE and start new innovative and collaborative activities proposed by the consortium. These include: assisting the development of PRACE 2; strengthening the internationally recognised PRACE brand; continuing and extend advanced training which so far provided more than 36 400 person-training days; preparing strategies and best practices towards Exascale computing, work on forward-looking SW solutions; coordinating and enhancing the operation of the multi-tier HPC systems and services; and supporting users to exploit massively parallel systems and novel architectures. A high level Service Catalogue is provided. The proven project structure will be used to achieve each of the objectives in 7 dedicated work packages. The activities are designed to increase Europe's research and innovation potential especially through: seamless and efficient Tier-0 services and a pan-European HPC ecosystem including national capabilities; promoting take-up by industry and new communities and special offers to SMEs; assistance to PRACE 2 development; proposing strategies for deployment of leadership systems; collaborating with the ETP4HPC, CoEs and other European and international organisations on future architectures, training, application support and policies. This will be monitored through a set of KPIs.

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

**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
- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- MEGWARE COMPUTER VERTRIEB UND SERVICE GMBH (Megware), Germany
- SIMULA RESEARCH LABORATORY AS, Norway
- UNIVERSITE DE STRASBOURG (UNISTRA), France
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 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:**

**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. We will develop, in concert, numerical schemes suitable for exascale parallelism, problem-tailored linear-system solvers and preconditioners, and a compiler to translate high-level model descriptions into optimized, energy-efficient system code for heterogeneous computing systems. The code will be parallelized with a recently developed runtime system that is resilient to hardware failures and will use an energy-aware task placement strategy.

The platform will be applied in real-life use cases with high impact in the biomedical domain and will showcase HPC in this area where it is painfully underused. It will be made accessible for a wide range of users both as code and through a web interface.

We will further employ our HPC and biomedical expertise to accelerate the development of parallel segmentation and (re)meshing software, necessary to create the extremely large and complex meshes needed from available large volumes of microscopy data.

The platform will be adaptable to similar biological systems such as nerves, and components of the platform will be reusable in a wide range of applications.

## 9.4 National initiatives

**ELCI** The ELCI PIA project (Software Environment for HPC) aims to develop a new generation of software stack for supercomputers, numerical solvers, runtime and programming development environments for HPC simulation. The ELCI project also aims to validate this software stack by showing its capacity to offer improved scalability, resilience, security, modularity and abstraction on real applications. The coordinator is Bull, and the different partners are CEA, INRIA, SAFRAN, CERFACS, CNRS CORIA, CENAERO, ONERA, UVSQ, Kitware and AlgoTech.

### 9.4.1 ANR

#### **ANR SOLHARIS** SOLHARIS

- 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. The challenges that SOLHARIS will be confronted with are related to the complexity, irregularity and, to some extent, unpredictability of sparse, direct solvers, to the large scale, complexity and heterogeneity of supercomputing platforms and to the ever increasing performance gap between processing units, memories and interconnects. SOLHARIS will tackle these challenges with three, tightly combined research efforts. First, it will develop dense and sparse linear algebra algorithms that can achieve better scalability by means of higher concurrency and efficiency and lower memory consumption. Second, it will improve runtime systems with novel features that enhance their performance and scalability and extend their programming interface to allow for an efficient and portable implementation of scalable algorithms. Third, it will develop scheduling methods for achieving high performance and scalability of both runtime systems and sparse direct solvers on large heterogeneous supercomputers.

**ANR EXACARD** • AAPG ANR 2018 (42 months)

- Coordinator: Yves Coudière (Carmen) INRIA Bordeaux
- Abstract: Cardiac arrhythmia affect millions of patients and cause 300,000 deaths each year in Europe. Most of these arrhythmia are due to interaction between structural and electrophysiological changes in the heart muscle. A true understanding of these phenomena requires numerical simulations at a much finer resolution, and larger scale, than currently possible. Next-generation, heterogeneous, high-performance computing (HPC) systems provide the power for this. But the large scale of the computations pushes the limits of current runtime optimization systems, and together with task-based parallelism, prompts for the development of dedicated numerical methods and HPC runtime optimizations. With a consortium including specialists of these domains and cardiac modeling, we will investigate new task-based optimization techniques and numerical methods to utilize these systems for cardiac simulations at an unprecedented scale, and pave the way for future use cases.

#### 9.4.2 IPL - Inria Project Lab

**HPC-BigData** (High Performance Computing and Big Data)

- Inria IPL 2018 - 2022 (48 months)
- Coordinator: Bruno Raffin (team DataMove, Inria Rhône Alpes)

Since June 2018, the team is participating to the **HPC-BigData** Inria Project Lab (IPL). The goal of this HPC-BigData IPL is to gather teams from the HPC, Big Data and Machine Learning (ML) areas to work at the intersection between these domains. Research is organized along three main axes: high performance analytics for scientific computing applications, high performance analytics for big data applications, infrastructure and resource management.

**EQIP** (Engineering for Quantum Information Processors)

- Inria IPL 2020 - 2024 (48 months)
- Coordinator: Anthony Leverrier (team Cosmiq)

Since Nov 2020, the team is participating to the **EQIP** Inria Project Lab (IPL). The goal of this EQIP IPL is to build a quantum processor, develop the software stack needed to operate a large-scale quantum computer, and develop quantum solutions to overcome classical computers. The EQIP challenge requires expertise in (1) superconducting qubits, (2) simulation of quantum systems, (3) numerical methods, (4) control theory, (5) programming languages and formal methods, (6) compilation, (7) quantum error correction, (8) quantum emulation, (9) cryptography and cryptanalysis, (10) quantum algorithms, (11) high-performance computing and gathers teams in this domains.

## 10 Dissemination

Olivier Aumage Denis Barthou Marie-Christine Counilh Amina Guermouche Laercio Lima Pilla Raymond Namyst Mihail Popov Emmanuelle Saillard Samuel Thibault Pierre-André Wacrenier

## 10.1 Promoting scientific activities

### 10.1.1 Scientific events: organisation

#### Member of the organizing committees

- Olivier Aumage, Laércio Lima Pilla, Emmanuelle Saillard: Participation to the organization of the SBAC-PAD'22 international conference.
- Mihail Popov, Emmanuelle Saillard: Participation to the organization of the ICPP'22 international conference as web chair, communication chair, and local arrangement chair.
- Emmanuelle Saillard: participation to the organization of the C3PO workshop as associate chair.
- Amina Guermouche: Chair of the parallelism track for Compas 2022

### 10.1.2 Scientific events: selection

#### Member of the conference program committees

- Laércio Lima Pilla: COMPAS 2022, HCW 2022, HPCEuroLatam 2022, ICCS 2022, PAW-ATM 2022
- Mihail Popov: COMPAS 2022, ICCS 2022, PDP 2022
- Olivier Aumage: ExHET'22
- Emmanuelle Saillard: COMPAS 2022, Correctness 2022, ICCS 2022
- Samuel Thibault: HCW'22, SPAC-PAD 2022, Cluster 2022
- Amina Guermouche: HiPC 2022, SBACPAD 2022, FTXS 2022

**Reviewer** The members of the team contribute to the reviews of the main conferences, workshops and journals of their domain.

### 10.1.3 Journal

#### Member of the editorial boards

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

#### Reviewer - reviewing activities

- Laércio Lima Pilla: Journal of Parallel and Distributed Computing
- Mihail Popov: IPDPS 2022
- Olivier Aumage: Journal of Parallel and Distributed Computing, CLUSTER 2022, PPOPP 2023
- Emmanuelle Saillard: CLUSTER 2022
- Samuel Thibault: Journal of Systems & Software
- Amina Guermouche: Journal of Parallel and Distributed Computing

#### 10.1.4 Invited talks

- Laércio Lima Pilla: Dagstuhl Seminar on "Power and Energy-aware Computing on Heterogeneous Systems (PEACHES)".
- Mihail Popov: Talk at University of Versailles on "Exploring software and hardware optimizations with machine learning".
- Olivier Aumage:
  - SIAM PP'22, minisymposium on "Targeting Future Exascale and Extreme Heterogeneity Era".
  - PPOPP ExHET 2022 Workshop on "Extreme Heterogeneity Solutions".
  - Talk at IDRIS supercomputing center in Orsay.
  - SC'22 PAW-ATM Workshop on "Extreme Heterogeneity Solutions".
- Emmanuelle Saillard:
  - CLAP seminar (online) "LLVM pour les nuls"
  - MTV seminar at University of Bordeaux, "PARCOACH, a tool that combines static and dynamic analyses to detect misuse of MPI communications"
  - Midi de la bidouille, "Introduction à l'infrastructure de compilation LLVM"
  - MICROCARD local seminar
- Samuel Thibault
  - Stanford University seminar
  - Uppsala University seminar

#### 10.1.5 Leadership within the scientific community

- Olivier Aumage: SC'22 Panel "Runtimes Systems for Extreme Heterogeneity: Challenges and Opportunities".

#### 10.1.6 Scientific expertise

- Olivier Aumage: Member of ATOS "Joseph Fourier Prize 2022" selection committee for HPC and IA.

#### 10.1.7 Research administration

- Olivier Aumage: Head of Team "SEHP – *Supports d'exécutions hautes performances*" (High performance executive supports) at LaBRI computer science Laboratory.
- Olivier Aumage: Elected member of LaBRI's Scientific Council.
- Samuel Thibault: Nominated member of LaBRI's Laboratory Council
- Samuel Thibault: Vice-president of recruitment committee for an associate professor position
- Samuel Thibault is member of the "Source Code and software group" of the national Comity for OpenScience (CoSO).
- Laércio Lima Pilla:
  - co-creation and participation in the Gender Equality and Equal Opportunities work group of the Inria centre at the university of Bordeaux.
  - member of the societal challenges commission at LaBRI.
- Emmanuelle Saillard: Member of the Commission de délégation at Inria Research Centre of the University of Bordeaux.

- Nathalie Furmento
  - member of the CDT (commission développement technologique) for Inria Research Center of the University of Bordeaux.
  - elected member of the council of the LaBRI.
  - member of the parity and equality group of the Inria Bordeaux center and member of the societal challenges commission at the LaBRI.
- Denis Barthou: WP leader for the European project MICROCARD.

## 10.2 Teaching - Supervision - Juries

### 10.2.1 Teaching

- Licence: Emmanuelle Saillard, Introduction to research, 1HeTD, L3, University of Bordeaux.
- Engineering School: Emmanuelle Saillard, Languages of parallelism, 12HeC, M2, ENSEIRB-MATMECA.
- Engineering School: Mihail Popov, Project C, 25HeC, L3, ENSEIRB-MATMECA.
- Engineering School: Mihail Popov, Cryptography, 33HeC, M1, ENSEIRB-MATMECA.
- Engineering School: Denis Barthou, Architectures (L3), Parallel Architectures (M2), Large development project (M1).
- 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: Romain Lion, System Programming, 18HeTD, M1, ENSEIRB-MATMECA.
- Licence: Marie-Christine Counilh, Introduction to Computer Science (58HeTD), Introduction to C programming (36HeTD), Scientific methodology (21HeTD) L1, University of Bordeaux.
- Master MIAGE: Marie-Christine Counilh, Object oriented programming in Java (30HeTD), M1, University of Bordeaux.
- 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: Samuel Thibault, Software Security, 60HeTD, M1, University of Bordeaux.
- Master: Samuel Thibault, System Security, 20HeTD, M2, University of Bordeaux.
- Licence, Pierre-André Wacrenier, is responsible for the 3rd year of the computer science Licence.
- Licence, Pierre-André Wacrenier, Introduction to Computer Science (64HeTD, L1), Systems Programming (64HeTD, L3), University of Bordeaux.
- Master, Pierre-André Wacrenier, Parallel Programming (64HeTD), University of Bordeaux.

- Raymond Namyst was involved in the introduction of Computer Science courses in the French High School (Lycée) scholar program. In particular, he was in charge of organizing a one-week condensed training session to 96 High School teachers on the following topics: Computer Architecture, Operating Systems, Networks and Robotics. The goal was to prepare then to teach computer science basics to students starting from September 2019, and to help them to prepare material for practice sessions.
- 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, Programming Project, 25HeTD, ENSEIRB-MATMECA
- 2nd year : Amina Guermouche, System Programming, 18HeTD, ENSEIRB-MATMECA
- 3rd year : Amina Guermouche, GPU Programming, 39HeTD, ENSEIRB-MATMECA

### 10.2.2 Supervision

- PhD in progress: Célia Ait Kaci, "Analysis and optimizations for partitioned global address space based HPC applications", 2019 - 2022, supervised by E. Saillard and D. Barthou.
- PhD in progress: Van-Man Nguyen, "Automatic optimization of parallel applications using non-blocking communications", 2020 - 2022, supervised by E. Saillard and D. Barthou.
- PhD in progress: Baptiste Coye, "Dynamic scheduling of task graph by composition", supervised by R. Namyst and D. Barthou.
- PhD in progress: Maxime Gonthier, "Memory-constrained Scheduling in a task-based model", october 2020 - , supervised by L. Marchal and S. Thibault.
- PhD in progress: 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 A. Guermouche, D.Barthou
- PhD in progress: Alice Lasserre, "Optimization of a task-based simulation code on a distributed supercomputer", supervised by R.Namyst
- PhD in progress: Lise Jolicoeur, "Towards secure cluster architectures for HPC workflows", supervised by R.Namyst
- PhD in progress: Diane Orhan, "Modeling and dynamic optimization of software radio chains on heterogeneous architectures.", supervised by L. Lima Pilla, D.Barthou, C.Jego
- PhD in progress: Lana Scravaglieri, "Portable vectorization with numerical accuracy control for multi-precision simulation codes", supervised by M. Popov, O.Aumage
- PhD in progress: Radjasouria Vinayagame, "Optimization of porting and performance of HPC applications with distributed and globally addressed memory", december 2022 - , supervised by E. Saillard and S. Thibault.
- Internship: Radjasouria Vinayagame, Jan. 2022 - April 2022, S. Baden, E. Saillard, D. Barthou.
- Internship: Pierre-Antoine Roubey, Feb. 2022 - Jul. 2022, E. Saillard
- Internship: Charles Goedefroit, May 2022 - Jul. 2022, D. Barthou
- Internship: Charles Martin, May 2022 - Jul. 2022, A. Guermouche
- Internship: Vincent Alba, Feb.2022 - Jul. 2022, D. Barthou and M.-C. Counilh

- Internship: Diane Orhan, Feb.2022 - Jul. 2022, D. Barthou and L.Lima Pilla
- Internship: Alice Lasserre, Feb.2022 - Jul. 2022, R. Namyst
- Internship: Pélégie Alves, Feb. 2022 - Jul. 2022, L. Lima Pilla, O. Aumage
- Internship: Thomas Morin, Jun. 2022 - Aug. 2022, P.-A. Wacrenier

### 10.2.3 Juries

- Denis Barthou:
  - Reviewer for the PhD of H. Wang, "Implicit parallelism for neural network acceleration", U.Orléans,
  - Reviewer for the PhD of A. Hennequin, "Performance optimization for the LHCb experiment", Sorbonne U.
  - President of the PhD committee for P.Swartzvagher, "On the Interactions between HPC Task-based Runtime Systems and Communication Libraries", U.Bordeaux
- Emmanuelle Saillard:
  - Associate Professor position at Bordeaux INP, May 2022
  - Inria CRCN/ISFP selection committee for Inria BSO Research Center
  - Examiner for the PhD of J. Gliksberg, "New routing algorithms for heterogeneous exaflop supercomputers", U. Paris-Saclay
- Amina Guermouche:
  - Examiner for the PhD of Mathieu Vérité, "Algorithmes d'allocation statique pour la planification d'applications haute performance", U. Bordeaux
  - Examiner for the PhD of Romain Lion, "Réplication des données pour la tolérance aux pannes dans un support d'exécution à base de tâches", U. Bordeaux
- Samuel Thibault:
  - Reviewer for the PhD of E. Lenormand, "Unification des mémoires réparties dans les systèmes hétérogènes"
  - Reviewer for the PhD of A. Colin, "De la collecte de trace à la prédiction du comportement d'applications parallèles"
- Nathalie Furmento:
  - Technical expert for several Inria engineer recruitments

### 10.3 Popularization

- Olivier Aumage: 1-hour webinar on Task-Based Performance Portability in HPC, ETP4HPC, [link](#).
- Olivier Aumage, Nathalie Furmento, Kun He and Samuel Thibault: 1-day tutorial on heterogeneous parallel computing with the StarPU Runtime System, TREX European HPC Centre of Excellence for Quantum Chemistry, CALMIP supercomputing center, Toulouse, [link](#).
- Denis Barthou: combined tutorial on MIPP wrapper for vectorization, combined with MAQAO, TREX European HPC Centre of Excellence for Quantum Chemistry, CALMIP supercomputing center, Toulouse, [link](#).

### 10.3.1 Internal or external Inria responsibilities

- Emmanuelle Saillard:
  - Organization of a meet-a-researcher day for ENS Lyon, December 2022.
  - Emmanuelle Saillard is responsible of popularization activities for Inria Research Centre of the University of Bordeaux.

### 10.3.2 Articles and contents

- Emmanuelle Saillard: "1 minute avec.." [35]
- "À quoi servent les ordinateurs les plus puissants du monde?", published at The Conversation, written by Emmanuelle Saillard and Mark Potse.

### 10.3.3 Education

Alice Lasserre, Raymond Namyst and Pierre-André Wacrenier contributed to a publication on education, relative to teaching HPC and in particular to high-quality assignments that are easy for other instructors to adopt and use in their own classes [15].

### 10.3.4 Interventions

- Emmanuelle Saillard:
  - Fête de la Science, Inria, Oct. 2022: atelier "La grande muraille d’Egypte", debate "être un citoyen numérique", Tchat onisep
  - Chiche!: March 2022 (Vaclav Havel high school), May 2022 (Gustave Eiffel high school), Oct. 2022 (Pape Clément high school)
  - Cordées de la réussite, May 2022 ("La grande muraille d’Egypte")
  - Welcoming ENS-Lyon undergraduate students, December 2022.
  - Atelier "La grande muraille d’Egypte", May 2022, E. Vaillant high school
  - Welcoming students fom F Mauriac high school, May 2022
  - Moi Informaticienne - Moi Mathématicienne, University of Bordeaux, April 2022.
  - Table ronde "Le casse-tête de l’enquête sur les algorithmes" at the "Institut de Journalisme Bordeaux Aquitaine" (IJBA), Nov. 2022, [link](#)
- Mihail Popov, Alice Lasserre, Laércio Lima Pilla, Amina Guermouche, Diane Orhan: Welcoming ENS-Lyon undergraduate students, December 2022.
- Nathalie Furmento: Fête de la Science, Inria, October 2022, debat "être un citoyen numérique"

## 11 Scientific production

### 11.1 Major publications

- [1] M. Gonthier, L. Marchal and S. Thibault. ‘Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs with Dynamic Runtime Systems’. In: IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium. Lyon, France: IEEE, 30th May 2022, pp. 1–11. DOI: [10.1109/IPDPS53621.2022.00073](https://doi.org/10.1109/IPDPS53621.2022.00073). URL: <https://hal.inria.fr/hal-03552243>.
- [2] A. Tehranijamsaz, M. Popov, A. Dutta, E. Saillard and A. Jannesari. ‘Learning Intermediate Representations using Graph Neural Networks for NUMA and Prefetchers Optimization’. In: IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium. Lyon / Virtual, France, 30th May 2022. URL: <https://hal.inria.fr/hal-03603118>.

## 11.2 Publications of the year

### International journals

- [3] G. Agosta, M. Aldinucci, C. Alvarez, R. Ammendola, Y. Arfat, O. Beaumont, M. Bernaschi, A. Biagioni, T. Boccali, B. Bramas et al. 'Towards EXtreme scale technologies and accelerators for euROhpc hw/Sw supercomputing applications for exascale: The TEXTAROSSA approach'. In: *Microprocessors and Microsystems: Embedded Hardware Design* 95 (Nov. 2022), p. 104679. DOI: [10.1016/j.micpro.2022.104679](https://doi.org/10.1016/j.micpro.2022.104679). URL: <https://hal.inria.fr/hal-03936864>.
- [4] J. Protze, M.-A. Hermanns, M. S. Müller, V. M. Nguyen, J. Jaeger, E. Saillard, P. Carribault and D. Barthou. 'MPI detach — Towards automatic asynchronous local completion'. In: *Parallel Computing* 109 (Mar. 2022), p. 102859. DOI: [10.1016/j.parco.2021.102859](https://doi.org/10.1016/j.parco.2021.102859). URL: <https://hal-cea.archives-ouvertes.fr/cea-03537990>.
- [5] M. Regragui, B. Coye, L. Lima Pilla, R. Namyst and D. Barthou. 'SimSGamE : Scheduling simulator for modern game engines'. In: *Journal of Open Source Software* 7.76 (Aug. 2022), p. 4592. DOI: [10.21105/joss.04592](https://doi.org/10.21105/joss.04592). URL: <https://hal.archives-ouvertes.fr/hal-03753664>.

### International peer-reviewed conferences

- [6] C. Castes, E. Agullo, O. Aumage and E. Saillard. 'Decentralized in-order execution of a sequential task-based code for shared-memory architectures'. In: *IPDPSW 2022 - IEEE International Parallel and Distributed Processing Symposium Workshops*. Lyon, France: IEEE, May 2022, pp. 552–561. DOI: [10.1109/IPDPSW55747.2022.00095](https://doi.org/10.1109/IPDPSW55747.2022.00095). URL: <https://hal.inria.fr/hal-03896030>.
- [7] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, R. Namyst, S. Thibault and P.-A. Wacrenier. 'Programming Heterogeneous Architectures Using Hierarchical Tasks'. In: *HeteroPar 2022*. Glasgow, United Kingdom, 23rd Aug. 2022, p. 12. URL: <https://hal.science/hal-03789625>.
- [8] M. Gonthier, L. Marchal and S. Thibault. 'Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs with Dynamic Runtime Systems'. In: *IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium*. Lyon, France: IEEE, 30th May 2022, pp. 1–11. DOI: [10.1109/IPDPS53621.2022.00073](https://doi.org/10.1109/IPDPS53621.2022.00073). URL: <https://hal.inria.fr/hal-03552243>.
- [9] A. Guermouche. 'Combining uncore frequency and dynamic power capping to improve power savings'. In: *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. Lyon, France, 30th May 2022, pp. 1028–1037. URL: <https://hal.archives-ouvertes.fr/hal-03563120>.
- [10] M. Regragui, B. Coye, L. Lima Pilla, R. Namyst and D. Barthou. 'Exploring scheduling algorithms for parallel task graphs: a modern game engine case study'. In: *International European Conference on Parallel and Distributed Computing (Euro-Par)*. Euro-Par 2022: Parallel Processing. Glasgow, United Kingdom, 1st Aug. 2022, pp. 103–118. DOI: [10.1007/978-3-031-12597-3\\_7](https://doi.org/10.1007/978-3-031-12597-3_7). URL: <https://hal.archives-ouvertes.fr/hal-03580775>.
- [11] E. Saillard, M. Sergent, T. C. Aitkaci and D. Barthou. 'Static Local Concurrency Errors Detection in MPI-RMA Programs'. In: *Correctness 2022 - Sixth International Workshop on Software Correctness for HPC Applications*. Dallas, United States, 18th Nov. 2022. URL: <https://hal.inria.fr/hal-03882459>.
- [12] M. Shimchenko, M. Popov and T. Wrigstad. 'Analysing and Predicting Energy Consumption of Garbage Collectors in OpenJDK'. In: *MPLR 2022 - 19th International Conference on Managed Programming Languages and Runtimes*. Brussels, Belgium: ACM, 14th Sept. 2022, pp. 3–15. DOI: [10.1145/3546918.3546925](https://doi.org/10.1145/3546918.3546925). URL: <https://hal.inria.fr/hal-03884425>.
- [13] A. Tehranijamsaz, M. Popov, A. Dutta, E. Saillard and A. Jannesari. 'Learning Intermediate Representations using Graph Neural Networks for NUMA and Prefetchers Optimization'. In: *IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium*. Lyon / Virtual, France, 30th May 2022. URL: <https://hal.inria.fr/hal-03603118>.



### National peer-reviewed Conferences

- [14] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, S. Thibault and P.-A. Wacrenier. ‘Programmation des architectures hétérogènes à l’aide de tâches hiérarchiques’. In: COMPAS 2022 - Conférence francophone d’informatique en Parallélisme, Architecture et Système. Amiens, France, 5th July 2022. URL: <https://hal.inria.fr/hal-03773486>.

### Conferences without proceedings

- [15] H. M. Bucker, H. Casanova, R. F. da Silva, A. Lasserre, D. Luyen, R. Namyst, J. Schoder, P.-A. Wacrenier and D. Bunde. ‘Peachy Parallel Assignments (EduPar 2022)’. In: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Lyon, France: IEEE, 30th May 2022, pp. 361–368. DOI: [10.1109/IPDPSW55747.2022.00068](https://doi.org/10.1109/IPDPSW55747.2022.00068). URL: <https://hal.science/hal-03938420>.
- [16] C. Sakka, A. Guermouche, O. Aumage, E. Saillard, M. Potse, Y. Coudière and D. Barthou. ‘A comparison of multithreading, vectorization, and GPU computing for the acceleration of cardiac electrophysiology models’. In: Computing in Cardiology. Tampere, Finland, 5th Sept. 2022. URL: <https://hal.inria.fr/hal-03936903>.

### Doctoral dissertations and habilitation theses

- [17] C. T. A. Kaci. ‘Analysis and optimizations for partitioned global address space based HPC applications’. Bordeaux: Université de Bordeaux, Dec. 2022. URL: <https://www.theses.fr/s345102>.
- [18] R. Lion. ‘Data replication for failure-tolerance in a distributed task-based runtime system’. Bordeaux: Université de Bordeaux, Dec. 2022. URL: <https://www.theses.fr/s346124>.
- [19] V. M. Nguyen. ‘Compile-time Validation and Optimization of MPI Nonblocking Communications’. Bordeaux: Université de Bordeaux, Dec. 2022. URL: <https://www.theses.fr/s346345>.

### Reports & preprints

- [20] E. Agullo, O. Coulaud, A. Denis, M. Faverge, A. Franc, J.-M. Frigerio, N. Furmento, A. Guilbaud, E. Jeannot, R. Peresson, F. Pruvost and S. Thibault. *Task-based randomized singular value decomposition and multidimensional scaling*. RR-9482. Inria Bordeaux - Sud Ouest; Inrae - BioGeCo, 9th Sept. 2022, p. 37. URL: <https://hal.inria.fr/hal-03773985>.
- [21] E. Agullo, M. Felšöci, A. Guermouche, H. Mathieu, G. Sylvand and B. Tagliaro. *Study of the processor and memory power consumption of coupled sparse/dense solvers*. RR-9463. Inria Bordeaux Sud-Ouest, Feb. 2022, p. 17. URL: <https://hal.inria.fr/hal-03589695>.
- [22] O. Aumage and C. Coti. *StarPU profiling interface*. Inria & Labri, Université de Bordeaux; Université du Québec à Montréal, 23rd Nov. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03868526>.
- [23] 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*. 26th July 2022. DOI: [10.48550/arXiv.2206.06147](https://doi.org/10.48550/arXiv.2206.06147). URL: <https://hal.archives-ouvertes.fr/hal-03690685>.
- [24] C. Castes, E. Agullo, O. Aumage and E. Saillard. *Decentralized in-order execution of a sequential task-based code for shared-memory architectures*. RR-9450. Inria Bordeaux - Sud Ouest, 28th Jan. 2022, p. 30. URL: <https://hal.inria.fr/hal-03547334>.
- [25] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, R. Namyst, S. Thibault and P.-A. Wacrenier. *Programming Heterogeneous Architectures Using Hierarchical Tasks*. RR-9466. Inria Bordeaux Sud-Ouest, 15th Mar. 2022, p. 21. URL: <https://hal.inria.fr/hal-03609275>.
- [26] M. Gonthier, L. Marchal and S. Thibault. *Taming data locality for GPU task scheduling in runtime systems*. 29th Mar. 2022. URL: <https://hal.inria.fr/hal-03623220>.

- [27] L. Lima Pilla. *Optimal Workload Scheduling Algorithm for Data-Parallel Applications on Heterogeneous Platforms based on Dynamic Programming*. RR-9487. CNRS; LaBRI; Inria; Université de Bordeaux; Bordeaux INP, 13th Sept. 2022, pp. 1–6. URL: <https://hal.inria.fr/hal-03776372>.
- [28] L. Lima Pilla. *Scheduling Algorithms for Federated Learning with Minimal Energy Consumption*. 12th Sept. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03775491>.
- [29] L. Scravaglieri, L. Lima Pilla, A. Guermouche, O. Aumage, E. Saillard and M. Popov. *Optimizing Performance and Energy Across Problem Sizes Through a Search Space Exploration and Machine Learning*. 11th Oct. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03810305>.

### Other scientific publications

- [30] V. Alba. ‘Load balancing and precision analysis for cardiac simulation M2 Internship report’. Université de Bordeaux (UB), France, 23rd June 2022. URL: <https://hal.inria.fr/hal-03784546>.
- [31] P. Alves. ‘Equilibrage de charge distribué sur StarPU’. Université de bordeaux, 23rd June 2022. URL: <https://hal.inria.fr/hal-03756492>.
- [32] F. de Barros, T. Gergouil, R. Grelard and S. Thibault. ‘Analyse de systèmes de vote électronique’. Université de Bordeaux, 18th Feb. 2022. URL: <https://hal.inria.fr/hal-03580506>.
- [33] T. Morin. ‘Exploring the collaboration between FEniCSx and StarPU’. Université de Bordeaux (UB), FRA., 14th Dec. 2022. URL: <https://hal.inria.fr/hal-03897912>.
- [34] D. Orhan. ‘Ordonnancement automatique et parallèle du flux de données appliqué à la radiologie et notamment au logiciel AFF3CT’. Université de Bordeaux (UB), France, 23rd June 2022. URL: <https://hal.inria.fr/hal-03768106>.
- [35] E. Saillard. *1 minute avec...Emmanuelle Saillard*. 11th Apr. 2022. URL: <https://hal.inria.fr/hal-03656775>.

## 11.3 Other

### Scientific popularization

- [36] E. Saillard and M. Potse. ‘À quoi servent les ordinateurs les plus puissants au monde ? Un exemple en cardiologie’. In: *The Conversation* (28th Dec. 2022). URL: <https://hal.inria.fr/hal-03939093>.

## 11.4 Cited publications

- [37] J. Bachan, S. B. Baden, S. Hofmeyr, M. Jacquelin, A. Kamil, D. Bonachea, P. H. Hargrove and H. Ahmed. ‘UPC++: A High-Performance Communication Framework for Asynchronous Computation’. In: *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 2019, pp. 963–973. DOI: [10.1109/IPDPS.2019.00104](https://doi.org/10.1109/IPDPS.2019.00104).
- [38] A. Cassagne, O. Aumage, D. Barthou, C. Leroux and C. Jego. ‘MIPP: a Portable C++ SIMD Wrapper and its use for Error Correction Coding in 5G Standard’. In: *The 4th Workshop on Programming Models for SIMD/Vector Processing (WPMVP 2018)*. Vienna, Austria: ACM Press, Feb. 2018. DOI: [10.1145/3178433.3178435](https://doi.org/10.1145/3178433.3178435). URL: <https://hal.inria.fr/hal-01888010>.
- [39] C. Denis, P. de Oliveira Castro and E. Petit. ‘Verificarlo: checking floating point accuracy through Monte Carlo Arithmetic’. In: *2016 IEEE 23rd Symposium on Computer Arithmetic (ARITH)*. Ed. by IEEE. Santa Clara, United States, July 2016. DOI: [10.1109/ARITH.2016.31](https://doi.org/10.1109/ARITH.2016.31). URL: <https://hal.science/hal-01192668>.
- [40] R. Lion and S. Thibault. ‘From tasks graphs to asynchronous distributed checkpointing with local restart’. In: *FTXS 2020 - IEEE/ACM 10th Workshop on Fault Tolerance for HPC at eXtreme Scale*. Atlanta / Virtual, United States, Nov. 2020. DOI: [10.1109/FTXS51974.2020.00009](https://doi.org/10.1109/FTXS51974.2020.00009). URL: <https://hal.archives-ouvertes.fr/hal-02970529>.