

RESEARCH CENTRE

**Bordeaux - Sud-Ouest**

IN PARTNERSHIP WITH:

CNRS, Institut Polytechnique de  
Bordeaux, Université de Bordeaux

2020

ACTIVITY REPORT

Project-Team

HIEPACS

## **High-End Parallel Algorithms for Challenging Numerical Simulations**

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

**DOMAIN**

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

**THEME**

**Distributed and High Performance  
Computing**

# Contents

|  |           |
|--|-----------|
| <b>Project-Team HIEPACS</b>  | <b>1</b>  |
| <b>1 Team members, visitors, external collaborators</b>                                  | <b>2</b>  |
| <b>2 Overall objectives</b>  | <b>3</b>  |
| 2.1 Introduction   | 3         |
| <b>3 Research program</b>  | <b>4</b>  |
| 3.1 Introduction   | 4         |
| 3.2 High-performance computing on next generation architectures                          | 5         |
| 3.3 High performance solvers for large linear algebra problems                           | 6         |
| 3.3.1 Parallel sparse direct solvers   | 7         |
| 3.3.2 Hybrid direct/iterative solvers based on algebraic domain decomposition techniques | 8         |
| 3.3.3 Linear Krylov solvers  | 8         |
| 3.3.4 Eigensolvers   | 9         |
| 3.3.5 Fast Solvers for FEM/BEM Coupling  | 9         |
| 3.4 High performance Fast Multipole Method for N-body problems                           | 9         |
| 3.4.1 Improvement of calculation efficiency  | 10        |
| 3.4.2 Non uniform distributions  | 10        |
| 3.4.3 Fast multipole method for dislocation operators                                    | 10        |
| 3.4.4 Fast multipole method for boundary element methods                                 | 11        |
| 3.5 Load balancing algorithms for complex simulations                                    | 11        |
| 3.5.1 Dynamic load-balancing with variable number of processors                          | 11        |
| 3.5.2 Load balancing of coupled codes  | 12        |
| 3.5.3 Load balancing strategies for hybrid sparse linear solvers                         | 12        |
| <b>4 Application domains</b>   | <b>12</b> |
| 4.1 Material physics   | 12        |
| 4.2 Co-design of algorithms in scientific applications                                   | 13        |
| 4.2.1 High performance simulation for ITER tokamak                                       | 13        |
| 4.2.2 Numerical and parallel scalable hybrid solvers in large scale calculations         | 14        |
| 4.3 Aeroacoustics Simulation   | 14        |
| 4.4 Optimization for Deep Convolutional Neural Networks                                  | 15        |
| <b>5 Highlights of the year</b>  | <b>15</b> |
| <b>6 New software and platforms</b>  | <b>15</b> |
| 6.1 New software   | 15        |
| 6.1.1 AVCI   | 15        |
| 6.1.2 Chameleon  | 16        |
| 6.1.3 Diodon   | 17        |
| 6.1.4 DPLASMA  | 17        |
| 6.1.5 Fabulous   | 18        |
| 6.1.6 MAPHYS   | 18        |
| 6.1.7 MAPHYS++   | 19        |
| 6.1.8 StarPart   | 19        |
| 6.1.9 MPICPL   | 19        |
| 6.1.10 OptiDis   | 20        |
| 6.1.11 PaStiX  | 20        |
| 6.1.12 pmtool  | 21        |
| 6.1.13 rotor   | 21        |
| 6.1.14 ScalFMM   | 22        |
| 6.1.15 VITE  | 23        |
| 6.1.16 PlaFRIM   | 23        |

|          |   |           |
|----------|---|-----------|
| <b>7</b> | <b>New results</b>  | <b>23</b> |
| 7.1      | High performance solvers for large linear algebra problems  | 23        |
| 7.1.1    | Exploring variable accuracy storage through lossy compression techniques in numerical linear algebra: a first application to flexible GMRES | 23        |
| 7.1.2    | A block minimum residual norm subspace linear solver for sequences of multiple left and right-hand side linear systems                      | 24        |
| 7.1.3    | 2D Static Resource Allocation for Compressed Linear Algebra and Communication Constraints   | 24        |
| 7.1.4    | On soft errors in the Conjugate Gradient method: sensitivity and robust numerical detection   | 24        |
| 7.1.5    | Recent Developments Around the Block Low-Rank PaStiX Solver   | 25        |
| 7.1.6    | Deciding Non-Compressible Blocks in Sparse Direct Solvers using Incomplete Factorization  | 25        |
| 7.2      | Applications domains  | 25        |
| 7.2.1    | Material physics  | 25        |
| 7.2.2    | Metabarcoding   | 26        |
| 7.2.3    | Co-design for scalable algorithms in scientific applications  | 26        |
|          | New developments regarding the JOREK solver and physics based preconditioner  | 26        |
|          | Numerical and parallel scalable hybrid solvers in large scale calculations  | 27        |
|          | Fast Solvers for High Frequency Aeroacoustics   | 27        |
|          | Compression by Tensor Decompositions for a scattering problem   | 28        |
| 7.3      | Optimization for the training phase of deep convolutional networks  | 28        |
| 7.3.1    | Memory optimization   | 28        |
| 7.3.2    | Optimization of Pipelined Model Parallelism   | 28        |
| 7.4      | Scheduling  | 28        |
| 7.4.1    | Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing   | 28        |
| 7.4.2    | Efficient Ordering of Kernel Submission on GPUs   | 29        |
| 7.4.3    | Scheduling Tasks on Two Types of Resources  | 29        |
| 7.4.4    | Geometric Deep Reinforcement Learning for Dynamic DAG Scheduling  | 30        |
| 7.4.5    | Approximation Algorithm for Estimating Distances in Distributed Virtual Environments  | 30        |
| 7.4.6    | A Makespan Lower Bound for the Scheduling of the Tiled Cholesky Factorization based on ALAP Schedule  | 30        |
| 7.4.7    | Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs                                 | 31        |
| 7.5      | Efficient algorithmic for load balancing and code coupling in complex simulations   | 31        |
| <b>8</b> | <b>Partnerships and cooperations</b>  | <b>32</b> |
| 8.1      | International initiatives   | 32        |
| 8.1.1    | Inria International Labs  | 32        |
| 8.1.2    | Inria associate team not involved in an ILL   | 32        |
|          | MOLIERE   | 32        |
| 8.2      | European initiatives  | 32        |
| 8.2.1    | FP7 & H2020 Projects  | 32        |
|          | EoCoE-II  | 32        |
|          | PRACE-6IP   | 33        |
|          | EOSC-Pillar   | 35        |
| 8.3      | National initiatives  | 36        |
| 8.3.1    | ANR   | 36        |
|          | SASHIMI: Sparse Direct Solver using Hierarchical Matrices   | 36        |
|          | SOLHARIS: SOLvers for Heterogeneous Architectures over Runtime systems, Investigating Scalability   | 36        |
| 8.3.2    | FUI   | 37        |
|          | ICARUS: Intensive Calculation for AeRo and automotive engines Unsteady Simulations  | 37        |

|           |   |           |
|-----------|---|-----------|
| 8.3.3     | Inria Project Labs                          | 38        |
|           | Challenge HPC BigData                       | 38        |
| 8.4       | Regional initiatives                        | 38        |
| <b>9</b>  | <b>Dissemination</b>                        | <b>39</b> |
| 9.1       | Promoting scientific activities             | 39        |
| 9.1.1     | Scientific events: organisation             | 39        |
|           | Member of the organizing committees         | 39        |
|           | Chair of conference program committees      | 39        |
|           | Member of the conference program committees | 39        |
| 9.1.2     | Journal                                     | 39        |
|           | Member of the editorial boards              | 39        |
|           | Reviewer - reviewing activities             | 40        |
| 9.1.3     | Scientific expertise                        | 40        |
| 9.1.4     | Research administration                     | 40        |
| 9.2       | Teaching - Supervision - Juries             | 40        |
| 9.2.1     | Teaching                                    | 40        |
| 9.2.2     | Supervision                                 | 41        |
| 9.2.3     | Juries                                      | 41        |
| 9.2.4     | Interventions                               | 42        |
| <b>10</b> | <b>Scientific production</b>                | <b>42</b> |
| 10.1      | Major publications                          | 42        |
| 10.2      | Publications of the year                    | 43        |
| 10.3      | Other                                       | 45        |

## Project-Team HIEPACS

*Creation of the Team: 2009 January 01, updated into Project-Team: 2010 January 01*

### Keywords

#### Computer sciences and digital sciences

- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A1.1.9. – Fault tolerant systems
- A6.2.5. – Numerical Linear Algebra
- A6.2.7. – High performance computing
- A7.1. – Algorithms
- A8.1. – Discrete mathematics, combinatorics
- A8.2. – Optimization
- A9.2. – Machine learning
- A9.7. – AI algorithmics

#### Other research topics and application domains

- B3.3.1. – Earth and subsoil
- B3.6. – Ecology
- B3.6.1. – Biodiversity
- B4.2.2. – Fusion
- B5.5. – Materials
- B9.5.1. – Computer science
- B9.5.2. – Mathematics
- B9.5.4. – Chemistry

## 1 Team members, visitors, external collaborators

### Research Scientists

- Luc Giraud [Team leader, Inria, Senior Researcher, HDR]
- Emmanuel Agullo [Inria, Researcher]
- Olivier Beaumont [Inria, Senior Researcher, HDR]
- Olivier Coulaud [Inria, Senior Researcher, HDR]
- Lionel Eyraud-Dubois [Inria, Researcher]

### Faculty Members

- Aurélien Esnard [Univ de Bordeaux, Associate Professor]
- Mathieu Faverge [Institut National Polytechnique de Bordeaux, Associate Professor]
- Abdou Guermouche [Univ de Bordeaux, Associate Professor, HDR]
- Pierre Ramet [Univ de Bordeaux, Associate Professor, HDR]

### Post-Doctoral Fellows

- Pierre Esterie [Univ de Bordeaux, from Nov 2020]
- Mohammad Issa [Inria]
- Nick Schenkels [Inria, until Nov 2020]

### PhD Students

- Tobias Castanet [Univ de Bordeaux]
- Marek Felsoci [Inria]
- Martina Iannacito [Inria]
- Esragul Korkmaz [Inria]
- Aboul Karim Mohamed El Maarouf [IFPEN, from Mar 2020]
- Romain Peressoni [Univ de Bordeaux]
- Alena Shilova [Inria]
- Nicolas Venkovic [Cerfacs]
- Mathieu Verite [Inria]
- Yanfei Xiang [Inria, partially funded by CERFACS]

### Technical Staff

- Tony Delarue [Inria, Engineer]
- Pierre Esterie [Inria, Engineer, until Oct 2020]
- Gilles Marait [Inria, Engineer]
- Nick Schenkels [Inria, Engineer, from Dec 2020]
- Matthieu Simonin [Inria, Engineer, from Oct 2020]

## Interns and Apprentices

- Selmane Lebdaoui [ENSIERB-MATMECA Bordeaux, from Jun 2020 until Aug 2020]
- Victor Lederer [Inria, from Feb 2020 until Aug 2020]
- Matthieu Simonin [Inria, from Feb 2020 until Jul 2020]

## Administrative Assistant

- Chrystel Plumejeau [Inria]

## External Collaborators

- Rocio Carratala Saez [Universitat Jaume I-Espagne, until Sep 2020]
- Jean René Poirier [INP Toulouse, HDR]
- Guillaume Sylvand [Airbus R & T]

## 2 Overall objectives

### 2.1 Introduction

Over the last few decades, there have been innumerable science, engineering and societal breakthroughs enabled by the development of High Performance Computing (HPC) applications, algorithms and architectures. These powerful tools have provided researchers with the ability to computationally find efficient solutions for some of the most challenging scientific questions and problems in medicine and biology, climatology, nanotechnology, energy and environment. It is admitted today that *numerical simulation is the third pillar for the development of scientific discovery at the same level as theory and experimentation*. Numerous reports and papers also confirm that very high performance simulation will open new opportunities not only for research but also for a large spectrum of industrial sectors.

An important force which has continued to drive HPC has been to focus on frontier milestones which consist in technical goals that symbolize the next stage of progress in the field. In the 1990s, the HPC community sought to achieve computing at a teraflop rate and exascale machines are now expected in the next few months/years.

For application codes to sustain petaflops and more in the next few years, hundreds of thousands of processor cores or more are needed, regardless of processor technology. Currently, a few HPC simulation codes easily scale to this regime and major algorithms and codes development efforts are critical to achieve the potential of these new systems. Scaling to exaflop involves improving physical models, mathematical modeling, super scalable algorithms that will require paying particular attention to acquisition, management and visualization of huge amounts of scientific data.

In this context, the purpose of the **HIEPACS** project is to contribute performing efficiently frontier simulations arising from challenging academic and industrial research. The solution of these challenging problems require a multidisciplinary approach involving applied mathematics, computational and computer sciences. In applied mathematics, it essentially involves advanced numerical schemes. In computational science, it involves massively parallel computing and the design of highly scalable algorithms and codes to be executed on emerging hierarchical many-core, possibly heterogeneous, platforms. Through this approach, **HIEPACS** intends to contribute to all steps that go from the design of new high-performance more scalable, robust and more accurate numerical schemes to the optimized implementations of the associated algorithms and codes on very high performance supercomputers. This research will be conducted on close collaboration in particular with European and US initiatives and in the framework of EuroHPC collaborative projects.

The methodological part of **HIEPACS** covers several topics. First, we address generic studies concerning massively parallel computing, the design of high-end performance algorithms and software to be executed on future extreme scale platforms. Next, several research prospectives in scalable parallel linear algebra techniques are addressed, ranging from dense direct, sparse direct, iterative and hybrid

approaches for large linear systems. We are also interested in the general problem of minimizing memory consumption and data movements, by changing algorithms and possibly performing extra computations, in particular in the context of Deep Neural Networks. Then we consider research on N-body interaction computations based on efficient parallel fast multipole methods and finally, we address research tracks related to the algorithmic challenges for complex code couplings in multiscale/multiphysics simulations.

Currently, we have one major multiscale application that is in *material physics*. We contribute to all steps of the design of the parallel simulation tool. More precisely, our applied mathematics skill will contribute to the modeling and our advanced numerical schemes will help in the design and efficient software implementation for very large parallel multiscale simulations. Moreover, the robustness and efficiency of our algorithmic research in linear algebra are validated through industrial and academic collaborations with different partners involved in various application fields. Finally, we are also involved in a few collaborative initiatives in various application domains in a co-design like framework. These research activities are conducted in a wider multi-disciplinary context with colleagues in other academic or industrial groups where our contribution is related to our expertises. Not only these collaborations enable our expertise to have a stronger impact in various application domains through the promotion of advanced algorithms, methodologies or tools, but in return they open new avenues for research in the continuity of our core research activities.

Thanks to the two Inria collaborative agreements such as with Airbus/Conseil Régional Grande Aquitaine and with CEA, we have joint research efforts in a co-design framework enabling efficient and effective technological transfer towards industrial R&D. Furthermore, thanks to the past associate team **FASTLA** we contribute with world leading groups at Berkeley National Lab and Stanford University to the design of fast numerical solvers and their parallel implementations.

Our high performance software packages are integrated in several academic or industrial complex codes and are validated on very large scale simulations. For all our software developments, we use first the experimental platform **PLAFRIM**, the various large parallel platforms available through GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such as PRACE.

## 3 Research program

### 3.1 Introduction

The methodological component of **HIEPACS** concerns the expertise for the design as well as the efficient and scalable implementation of highly parallel numerical algorithms to perform frontier simulations. In order to address these computational challenges a hierarchical organization of the research is considered. In this bottom-up approach, we first consider in Section 3.2 generic topics concerning high performance computational science. The activities described in this section are transversal to the overall project and their outcome will support all the other research activities at various levels in order to ensure the parallel scalability of the algorithms. The aim of this activity is not to study general purpose solution but rather to address these problems in close relation with specialists of the field in order to adapt and tune advanced approaches in our algorithmic designs. The next activity, described in Section 3.3, is related to the study of parallel linear algebra techniques that currently appear as promising approaches to tackle huge problems on extreme scale platforms. We highlight the linear problems (linear systems or eigenproblems) because they are in many large scale applications the main computational intensive numerical kernels and often the main performance bottleneck. These parallel numerical techniques will be the basis of both academic and industrial collaborations, some are described in Section 4.1, but will also be closely related to some functionalities developed in the parallel fast multipole activity described in Section 3.4. Finally, as the accuracy of the physical models increases, there is a real need to go for parallel efficient algorithm implementation for multiphysics and multiscale modeling in particular in the context of code coupling. The challenges associated with this activity will be addressed in the framework of the activity described in Section 3.5.

Currently, we have one major application (see Section 4.1) that is in material physics. We will collaborate to all steps of the design of the parallel simulation tool. More precisely, our applied mathematics skill will contribute to the modelling, our advanced numerical schemes will help in the design and efficient



software implementation for very large parallel simulations. We also participate to a few co-design actions in close collaboration with some applicative groups. The objective of this activity is to instantiate our expertise in fields where they are critical for designing scalable simulation tools. We refer to Section 4.2 for a detailed description of these activities.

## 3.2 High-performance computing on next generation architectures

**Participants** Emmanuel Agullo, Olivier Beaumont, Olivier Coulaud, Pierre Esterie, Lionel Eyraud-Dubois, Mathieu Faverge, Luc Giraud, Abdou Guer-mouche, Gilles Marait, Pierre Ramet, Nick Schenkels, Alena Shilova, Mathieu Verite.

The research directions proposed in HIEPACS are strongly influenced by both the applications we are studying and the architectures that we target (i.e., massively parallel heterogeneous many-core architectures, ...). Our main goal is to study the methodology needed to efficiently exploit the new generation of high-performance computers with all the constraints that it induces. To achieve this high-performance with complex applications we have to study both algorithmic problems and the impact of the architectures on the algorithm design.

From the application point of view, the project will be interested in multiresolution, multiscale and hierarchical approaches which lead to multi-level parallelism schemes. This hierarchical parallelism approach is necessary to achieve good performance and high-scalability on modern massively parallel platforms. In this context, more specific algorithmic problems are very important to obtain high performance. Indeed, the kind of applications we are interested in are often based on data redistribution for example (e.g., code coupling applications). This well-known issue becomes very challenging with the increase of both the number of computational nodes and the amount of data. Thus, we have both to study new algorithms and to adapt the existing ones. In addition, some issues like task scheduling have to be restudied in this new context. It is important to note that the work developed in this area will be applied for example in the context of code coupling (see Section 3.5).

Considering the complexity of modern architectures like massively parallel architectures or new generation heterogeneous multicore architectures, task scheduling becomes a challenging problem which is central to obtain a high efficiency. With the recent addition of colleagues from the scheduling community (O. Beaumont and L. Eyraud-Dubois), the team is better equipped than ever to design scheduling algorithms and models specifically tailored to our target problems. It is important to note that this topic is strongly linked to the underlying programming model. Indeed, considering multicore and heterogeneous architectures, it has appeared, in the last five years, that the best programming model is an approach mixing multi-threading within computational nodes and message passing between them. In the last five years, a lot of work has been developed in the high-performance computing community to understand what is critic to efficiently exploit massively multicore platforms that will appear in the near future. It appeared that the key for the performance is firstly the granularity of the computations. Indeed, in such platforms the granularity of the parallelism must be small so that we can feed all the computing units with a sufficient amount of work. It is thus very crucial for us to design new high performance tools for scientific computing in this new context. This will be developed in the context of our solvers, for example, to adapt to this new parallel scheme. Secondly, the larger the number of cores inside a node, the more complex the memory hierarchy. This remark impacts the behavior of the algorithms within the node. Indeed, on this kind of platforms, NUMA effects will be more and more problematic. Thus, it is very important to study and design data-aware algorithms which take into account the affinity between computational threads and the data they access. This is particularly important in the context of our high-performance tools. Note that this work has to be based on an intelligent cooperative underlying run-time (like the tools developed by the Inria STORM Project-Team) which allows a fine management of data distribution within a node.

Another very important issue concerns high-performance computing using “heterogeneous” resources within a computational node. Indeed, with the deployment of the GPU and the use of more specific co-processors, it is important for our algorithms to efficiently exploit these new type of architectures. To adapt our algorithms and tools to these accelerators, we need to identify what can be done

on the GPU for example and what cannot. Note that recent results in the field have shown the interest of using both regular cores and GPU to perform computations. Note also that in opposition to the case of the parallelism granularity needed by regular multicore architectures, GPU requires coarser grain parallelism. Thus, making both GPU and regular cores work all together will lead to two types of tasks in terms of granularity. This represents a challenging problem especially in terms of scheduling. From this perspective, we investigate new approaches for composing parallel applications within a runtime system for heterogeneous platforms.

In the context of scaling up, and particularly in the context of minimizing energy consumption, it is generally acknowledged that the solution lies in the use of heterogeneous architectures, where each resource is particularly suited to specific types of tasks, and in a fine control at the algorithmic level of data movements and the trade-offs to be made between computation and communication. In this context, we are particularly interested in the optimization of the training phase of deep convolutional neural networks which consumes a lot of memory and for which it is possible to exchange computations for data movements and memory occupation. We are also interested in the complexity introduced by resource heterogeneity itself, both from a theoretical point of view on the complexity of scheduling problems and from a more practical point of view on the implementation of specific kernels in dense or sparse linear algebra.

In order to achieve an advanced knowledge concerning the design of efficient computational kernels to be used on our high performance algorithms and codes, we will develop research activities first on regular frameworks before extending them to more irregular and complex situations. In particular, we will work first on optimized dense linear algebra kernels and we will use them in our more complicated direct and hybrid solvers for sparse linear algebra and in our fast multipole algorithms for interaction computations. In this context, we will participate to the development of those kernels in collaboration with groups specialized in dense linear algebra. In particular, we intend develop a strong collaboration with the group of Jack Dongarra at the University of Tennessee and collaborating research groups. The objectives will be to develop dense linear algebra algorithms and libraries for multicore architectures in the context the **PLASMA** project and for GPU and hybrid multicore/GPU architectures in the context of the **MAGMA** project. A new solver has emerged from the associate team, Chameleon. While **PLASMA** and **MAGMA** focus on multicore and GPU architectures, respectively, Chameleon makes the most out of heterogeneous architectures thanks to task-based dynamic runtime systems.

A more prospective objective is to study the resiliency in the context of large-scale scientific applications for massively parallel architectures. Indeed, with the increase of the number of computational cores per node, the probability of a hardware crash on a core or of a memory corruption is dramatically increased. This represents a crucial problem that needs to be addressed. However, we will only study it at the algorithmic/application level even if it needed lower-level mechanisms (at OS level or even hardware level). Of course, this work can be performed at lower levels (at operating system) level for example but we do believe that handling faults at the application level provides more knowledge about what has to be done (at application level we know what is critical and what is not). The approach that we will follow will be based on the use of a combination of fault-tolerant implementations of the run-time environments we use (like for example **ULFM**) and an adaptation of our algorithms to try to manage this kind of faults. This topic represents a very long range objective which needs to be addressed to guaranty the robustness of our solvers and applications.

Finally, it is important to note that the main goal of **HIEPACS** is to design tools and algorithms that will be used within complex simulation frameworks on next-generation parallel machines. Thus, we intend with our partners to use the proposed approach in complex scientific codes and to validate them within very large scale simulations as well as designing parallel solution in co-design collaborations.

### 3.3 High performance solvers for large linear algebra problems

**Participants** Emmanuel Agullo, Olivier Coulaud, Tony Delarue, Mathieu Faverge, Aurélien Falco, Marek Felsoci, Luc Giraud, Abdou Guermouche, Esragul Korkmaz, Gilles Marait, Van Gia Thinh Nguyen, Jean Rene Poirier, Pierre Ramet, Cristobal Samaniego Alvarado, Guillaume Sylvand, Nicolas Venkovic, Yanfei Xiang.

Starting with the developments of basic linear algebra kernels tuned for various classes of computers, a significant knowledge on the basic concepts for implementations on high-performance scientific computers has been accumulated. Further knowledge has been acquired through the design of more sophisticated linear algebra algorithms fully exploiting those basic intensive computational kernels. In that context, we still look at the development of new computing platforms and their associated programming tools. This enables us to identify the possible bottlenecks of new computer architectures (memory path, various level of caches, inter processor or node network) and to propose ways to overcome them in algorithmic design. With the goal of designing efficient scalable linear algebra solvers for large scale applications, various tracks will be followed in order to investigate different complementary approaches. Sparse direct solvers have been for years the methods of choice for solving linear systems of equations, it is nowadays admitted that classical approaches are not scalable neither from a computational complexity nor from a memory view point for large problems such as those arising from the discretization of large 3D PDE problems. We will continue to work on sparse direct solvers on the one hand to make sure they fully benefit from most advanced computing platforms and on the other hand to attempt to reduce their memory and computational costs for some classes of problems where data sparse ideas can be considered. Furthermore, sparse direct solvers are a key building boxes for the design of some of our parallel algorithms such as the hybrid solvers described in the sequel of this section. Our activities in that context will mainly address preconditioned Krylov subspace methods; both components, preconditioner and Krylov solvers, will be investigated. In this framework, and possibly in relation with the research activity on fast multipole, we intend to study how emerging  $\mathcal{H}$ -matrix arithmetic can benefit to our solver research efforts.

### 3.3.1 Parallel sparse direct solvers

For the solution of large sparse linear systems, we design numerical schemes and software packages for direct and hybrid parallel solvers. Sparse direct solvers are mandatory when the linear system is very ill-conditioned; such a situation is often encountered in structural mechanics codes, for example. Therefore, to obtain an industrial software tool that must be robust and versatile, high-performance sparse direct solvers are mandatory, and parallelism is then necessary for reasons of memory capability and acceptable solution time. Moreover, in order to solve efficiently 3D problems with more than 50 million unknowns, which is now a reachable challenge with new multicore supercomputers, we must achieve good scalability in time and control memory overhead. Solving a sparse linear system by a direct method is generally a highly irregular problem that induces some challenging algorithmic problems and requires a sophisticated implementation scheme in order to fully exploit the capabilities of modern supercomputers.

New supercomputers incorporate many microprocessors which are composed of one or many computational cores. These new architectures induce strongly hierarchical topologies. These are called NUMA architectures. In the context of distributed NUMA architectures, in collaboration with the Inria **STORM** team, we study optimization strategies to improve the scheduling of communications, threads and I/O. We have developed dynamic scheduling designed for NUMA architectures in the **PaStiX** solver. The data structures of the solver, as well as the patterns of communication have been modified to meet the needs of these architectures and dynamic scheduling. We are also interested in the dynamic adaptation of the computation gain to use efficiently multi-core architectures and shared memory. Experiments on several numerical test cases have been performed to prove the efficiency of the approach on different architectures. Sparse direct solvers such as **PaStiX** are currently limited by their memory requirements and computational cost. They are competitive for small matrices but are often less efficient than iterative methods for large matrices in terms of memory. We are currently accelerating the dense algebra components of direct solvers using block low-rank compression techniques.

In collaboration with the ICL team from the University of Tennessee, and the **STORM** team from Inria, we are evaluating the way to replace the embedded scheduling driver of the **PaStiX** solver by one of the generic frameworks, **PaRSEC** or **StarPU**, to execute the task graph corresponding to a sparse factorization. The aim is to design algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computer equipped with GPU accelerators. More generally, this work will be performed in the context of the ANR **SOLHARIS** project which aims at designing high performance sparse direct solvers for modern heterogeneous systems. This ANR project involves several groups working either on the sparse linear solver aspects (**HIEPACS** and **ROMA** from Inria and APO from IRIT), on runtime systems (**STORM** from Inria) or scheduling algorithms (**HIEPACS** and **ROMA** from Inria). The results of these efforts will be validated in the applications provided by the industrial project members, namely CEA-CESTA and Airbus Central R & T.

### 3.3.2 Hybrid direct/iterative solvers based on algebraic domain decomposition techniques

One route to the parallel scalable solution of large sparse linear systems in parallel scientific computing is the use of hybrid methods that hierarchically combine direct and iterative methods. These techniques inherit the advantages of each approach, namely the limited amount of memory and natural parallelization for the iterative component and the numerical robustness of the direct part. The general underlying ideas are not new since they have been intensively used to design domain decomposition techniques; those approaches cover a fairly large range of computing techniques for the numerical solution of partial differential equations (PDEs) in time and space. Generally speaking, it refers to the splitting of the computational domain into sub-domains with or without overlap. The splitting strategy is generally governed by various constraints/objectives but the main one is to express parallelism. The numerical properties of the PDEs to be solved are usually intensively exploited at the continuous or discrete levels to design the numerical algorithms so that the resulting specialized technique will only work for the class of linear systems associated with the targeted PDE.

In that context, we continue our effort on the design of algebraic non-overlapping domain decomposition techniques that rely on the solution of a Schur complement system defined on the interface introduced by the partitioning of the adjacency graph of the sparse matrix associated with the linear system. Although it is better conditioned than the original system the Schur complement needs to be preconditioned to be amenable to a solution using a Krylov subspace method. Different hierarchical preconditioners will be considered, possibly multilevel, to improve the numerical behaviour of the current approaches implemented in our software library **MaPHyS**. This activity will be developed further developed in the H2020 **EoCoE2** project. In addition to this numerical studies, advanced parallel implementation will be developed that will involve close collaborations between the hybrid and sparse direct activities.

### 3.3.3 Linear Krylov solvers

Preconditioning is the main focus of the two activities described above. They aim at speeding up the convergence of a Krylov subspace method that is the complementary component involved in the solvers of interest for us. In that framework, we believe that various aspects deserve to be investigated; we will consider the following ones:

- Preconditioned block Krylov solvers for multiple right-hand sides. In many large scientific and industrial applications, one has to solve a sequence of linear systems with several right-hand sides given simultaneously or in sequence (radar cross section calculation in electromagnetism, various source locations in seismic, parametric studies in general, ...). For “simultaneous” right-hand sides, the solvers of choice have been for years based on matrix factorizations as the factorization is performed once and simple and cheap block forward/backward substitutions are then performed. In order to effectively propose alternative to such solvers, we need to have efficient preconditioned Krylov subspace solvers. In that framework, block Krylov approaches, where the Krylov spaces associated with each right-hand side are shared to enlarge the search space will be considered. They are not only attractive because of this numerical feature (larger search space), but also from an implementation point of view. Their block-structures exhibit nice features with respect to data locality and re-usability that comply with the memory constraint of multicore architectures.

We will continue the numerical study and design of the block GMRES variant that combines inexact break-down detection, deflation at restart and subspace recycling. Beyond new numerical investigations, a software implementation to be included in our linear solver library **Fabulous** originally developed in the context of the DGA HiBOX project and further developed in the LynCs (Linear Algebra, Krylov-subspace methods, and multi-grid solvers for the discovery of New Physics) sub-project of **PRACE-6IP**.

- Extension or modification of Krylov subspace algorithms for multicore architectures: finally to match as much as possible to the computer architecture evolution and get as much as possible performance out of the computer, a particular attention will be paid to adapt, extend or develop numerical schemes that comply with the efficiency constraints associated with the available computers. Nowadays, multicore architectures seem to become widely used, where memory latency and bandwidth are the main bottlenecks; investigations on communication avoiding techniques will be undertaken in the framework of preconditioned Krylov subspace solvers as a general guideline for all the items mentioned above.

### 3.3.4 Eigensolvers

Many eigensolvers also rely on Krylov subspace techniques. Naturally some links exist between the Krylov subspace linear solvers and the Krylov subspace eigensolvers. We plan to study the computation of eigenvalue problems with respect to the following two different axes:

- Exploiting the link between Krylov subspace methods for linear system solution and eigensolvers, we intend to develop advanced iterative linear methods based on Krylov subspace methods that use some spectral information to build part of a subspace to be recycled, either through space augmentation or through preconditioner update. This spectral information may correspond to a certain part of the spectrum of the original large matrix or to some approximations of the eigenvalues obtained by solving a reduced eigenproblem. This technique will also be investigated in the framework of block Krylov subspace methods.
- In the context of the calculation of the ground state of an atomistic system, eigenvalue computation is a critical step; more accurate and more efficient parallel and scalable eigensolvers are required.

### 3.3.5 Fast Solvers for FEM/BEM Coupling

In this research project, we are interested in the design of new advanced techniques to solve large mixed dense/sparse linear systems, the extensive comparison of these new approaches to the existing ones, and the application of these innovative ideas on realistic industrial test cases in the domain of aeroacoustics (in collaboration with Airbus Central R & T).

- The use of  $\mathcal{H}$ -matrix solvers on these problems has been investigated in the context of the PhD of A. Falco. Airbus CR&T, in collaboration with Inria Bordeaux Sud-Ouest, has developed a task-based  $\mathcal{H}$ -matrix solver on top of the runtime engine StarPU. Ideas coming from the field of sparse direct solvers (such as nested dissection or symbolic factorization) have been tested within  $\mathcal{H}$ -matrices.
- The question of parallel scalability of task-based tools is an active subject of research, using new communication engine such as NewMadeleine, that will be investigated during this project, in conjunction with new algorithmic ideas on the task-based writing of  $\mathcal{H}$ -matrix algorithms.
- Naturally, comparison with existing tools will be performed on large realistic test cases. Coupling schemes between these tools and the hierarchical methods used in  $\mathcal{H}$ -matrix will be developed and benched as well.

## 3.4 High performance Fast Multipole Method for N-body problems

**Participants** Emmanuel Agullo, Olivier Coulaud, Pierre Esterie, Guillaume Sylvand.

In most scientific computing applications considered nowadays as computational challenges (like biological and material systems, astrophysics or electromagnetism), the introduction of hierarchical methods based on an octree structure has dramatically reduced the amount of computation needed to simulate those systems for a given accuracy. For instance, in the N-body problem arising from these application fields, we must compute all pairwise interactions among N objects (particles, lines, ...) at every timestep. Among these methods, the Fast Multipole Method (FMM) developed for gravitational potentials in astrophysics and for electrostatic (coulombic) potentials in molecular simulations solves this N-body problem for any given precision with  $O(N)$  runtime complexity against  $O(N^2)$  for the direct computation.

The potential field is decomposed in a near field part, directly computed, and a far field part approximated thanks to multipole and local expansions. We introduced a matrix formulation of the FMM that exploits the cache hierarchy on a processor through the Basic Linear Algebra Subprograms (BLAS). Moreover, we developed a parallel adaptive version of the FMM algorithm for heterogeneous particle distributions, which is very efficient on parallel clusters of SMP nodes. Finally on such computers, we developed the first hybrid MPI-thread algorithm, which enables to reach better parallel efficiency and better memory scalability. We plan to work on the following points in [HiEPACS](#).

#### 3.4.1 Improvement of calculation efficiency

Nowadays, the high performance computing community is examining alternative architectures that address the limitations of modern cache-based designs. GPU (Graphics Processing Units) and the Cell processor have thus already been used in astrophysics and in molecular dynamics. The Fast Multipole Method has also been implemented on GPU. We intend to examine the potential of using these forthcoming processors as a building block for high-end parallel computing in N-body calculations. More precisely, we want to take advantage of our specific underlying BLAS routines to obtain an efficient and easily portable FMM for these new architectures. Algorithmic issues such as dynamic load balancing among heterogeneous cores will also have to be solved in order to gather all the available computation power. This research action will be conducted on close connection with the activity described in Section 3.2.

#### 3.4.2 Non uniform distributions

In many applications arising from material physics or astrophysics, the distribution of the data is highly non uniform and the data can grow between two time steps. As mentioned previously, we have proposed a hybrid MPI-thread algorithm to exploit the data locality within each node. We plan to further improve the load balancing for highly non uniform particle distributions with small computation grain thanks to dynamic load balancing at the thread level and thanks to a load balancing correction over several simulation time steps at the process level.

#### 3.4.3 Fast multipole method for dislocation operators

The engine that we develop will be extended to new potentials arising from material physics such as those used in dislocation simulations. The interaction between dislocations is long ranged ( $O(1/r)$ ) and anisotropic, leading to severe computational challenges for large-scale simulations. Several approaches based on the FMM or based on spatial decomposition in boxes are proposed to speed-up the computation. In dislocation codes, the calculation of the interaction forces between dislocations is still the most CPU time consuming. This computation has to be improved to obtain faster and more accurate simulations. Moreover, in such simulations, the number of dislocations grows while the phenomenon occurs and these dislocations are not uniformly distributed in the domain. This means that strategies to dynamically balance the computational load are crucial to achieve high performance.



### 3.4.4 Fast multipole method for boundary element methods

The boundary element method (BEM) is a well known solution of boundary value problems appearing in various fields of physics. With this approach, we only have to solve an integral equation on the boundary. This implies an interaction that decreases in space, but results in the solution of a dense linear system with  $O(N^3)$  complexity. The FMM calculation that performs the matrix-vector product enables the use of Krylov subspace methods. Based on the parallel data distribution of the underlying octree implemented to perform the FMM, parallel preconditioners can be designed that exploit the local interaction matrices computed at the finest level of the octree. This research action will be conducted on close connection with the activity described in Section 3.3. Following our earlier experience, we plan to first consider approximate inverse preconditioners that can efficiently exploit these data structures.

## 3.5 Load balancing algorithms for complex simulations

**Participants** Aurélien Esnard, Pierre Ramet.

Many important physical phenomena in material physics and climatology are inherently complex applications. They often use multi-physics or multi-scale approaches, which couple different models and codes. The key idea is to reuse available legacy codes through a coupling framework instead of merging them into a stand-alone application. There is typically one model per different scale or physics and each model is implemented by a parallel code.

For instance, to model a crack propagation, one uses a molecular dynamic code to represent the atomistic scale and an elasticity code using a finite element method to represent the continuum scale. Indeed, fully microscopic simulations of most domains of interest are not computationally feasible. Combining such different scales or physics is still a challenge to reach high performance and scalability.

Another prominent example is found in the field of aeronautic propulsion: the conjugate heat transfer simulation in complex geometries (as developed by the CFD team of CERFACS) requires to couple a fluid/convection solver (AVBP) with a solid/conduction solver (AVTP). As the AVBP code is much more CPU consuming than the AVTP code, there is an important computational imbalance between the two solvers.

In this context, one crucial issue is undoubtedly the load balancing of the whole coupled simulation that remains an open question. The goal here is to find the best data distribution for the whole coupled simulation and not only for each stand-alone code, as it is most usually done. Indeed, the naive balancing of each code on its own can lead to an important imbalance and to a communication bottleneck during the coupling phase, which can drastically decrease the overall performance. Therefore, we argue that it is required to model the coupling itself in order to ensure a good scalability, especially when running on massively parallel architectures (tens of thousands of processors/cores). In other words, one must develop new algorithms and software implementation to perform a *coupling-aware* partitioning of the whole application. Another related problem is the problem of resource allocation. This is particularly important for the global coupling efficiency and scalability, because each code involved in the coupling can be more or less computationally intensive, and there is a good trade-off to find between resources assigned to each code to avoid that one of them waits for the other(s). What does furthermore happen if the load of one code dynamically changes relatively to the other one? In such a case, it could be convenient to dynamically adapt the number of resources used during the execution.

There are several open algorithmic problems that we investigate in the HIEPACS project-team. All these problems use a similar methodology based upon the graph model and are expressed as variants of the classic graph partitioning problem, using additional constraints or different objectives.

### 3.5.1 Dynamic load-balancing with variable number of processors

As a preliminary step related to the dynamic load balancing of coupled codes, we focus on the problem of dynamic load balancing of a single parallel code, with variable number of processors. Indeed, if the workload varies drastically during the simulation, the load must be redistributed regularly among the processors. Dynamic load balancing is a well studied subject but most studies are limited to an initially

fixed number of processors. Adjusting the number of processors at runtime allows one to preserve the parallel code efficiency or keep running the simulation when the current memory resources are exceeded. We call this problem, *MxN graph repartitioning*.

We propose some methods based on graph repartitioning in order to re-balance the load while changing the number of processors. These methods are split in two main steps. Firstly, we study the migration phase and we build a “good” migration matrix minimizing several metrics like the migration volume or the number of exchanged messages. Secondly, we use graph partitioning heuristics to compute a new distribution optimizing the migration according to the previous step results.

### 3.5.2 Load balancing of coupled codes

As stated above, the load balancing of coupled code is a major issue, that determines the performance of the complex simulation, and reaching high performance can be a great challenge. In this context, we develop new graph partitioning techniques, called *co-partitioning*. They address the problem of load balancing for two coupled codes: the key idea is to perform a “coupling-aware” partitioning, instead of partitioning these codes independently, as it is classically done. More precisely, we propose to enrich the classic graph model with *inter-edges*, which represent the coupled code interactions. We describe two new algorithms, and compare them to the naive approach. In the preliminary experiments we perform on synthetically-generated graphs, we notice that our algorithms succeed to balance the computational load in the coupling phase and in some cases they succeed to reduce the coupling communications costs. Surprisingly, we notice that our algorithms do not degrade significantly the global graph edge-cut, despite the additional constraints that they impose.

Besides this, our co-partitioning technique requires to use graph partitioning with *fixed vertices*, that raises serious issues with state-of-the-art software, that are classically based on the well-known recursive bisection paradigm (RB). Indeed, the RB method often fails to produce partitions of good quality. To overcome this issue, we propose a *new* direct *k*-way greedy graph growing algorithm, called KGGGP, that overcomes this issue and succeeds to produce partition with better quality than RB while respecting the constraint of fixed vertices. Experimental results compare KGGGP against state-of-the-art methods, such as *Scotch*, for real-life graphs available from the popular *DIMACS'10* collection.

### 3.5.3 Load balancing strategies for hybrid sparse linear solvers

Graph handling and partitioning play a central role in the activity described here but also in other numerical techniques detailed in sparse linear algebra Section. The Nested Dissection is now a well-known heuristic for sparse matrix ordering to both reduce the fill-in during numerical factorization and to maximize the number of independent computation tasks. By using the block data structure induced by the partition of separators of the original graph, very efficient parallel block solvers have been designed and implemented according to super-nodal or multi-frontal approaches. Considering hybrid methods mixing both direct and iterative solvers such as *MaPHYs*, obtaining a domain decomposition leading to a good balancing of both the size of domain interiors and the size of interfaces is a key point for load balancing and efficiency in a parallel context.

We intend to revisit some well-known graph partitioning techniques in the light of the hybrid solvers and design new algorithms to be tested in the *Scotch* package.

## 4 Application domains

### 4.1 Material physics

**Participants** Olivier Coulaud, Pierre Esterie.

Due to the increase of available computer power, new applications in nano science and physics appear such as study of properties of new materials (photovoltaic materials, bio- and environmental sensors, ...), failure in materials, nano-indentation. Chemists, physicists now commonly perform simulations in these



fields. These computations simulate systems up to billion of atoms in materials, for large time scales up to several nanoseconds. The larger the simulation, the smaller the computational cost of the potential driving the phenomena, resulting in low precision results. So, if we need to increase the precision, there are two ways to decrease the computational cost. In the first approach, we improve algorithms and their parallelization and in the second way, we will consider a multiscale approach.

A domain of interest is the material aging for the nuclear industry. The materials are exposed to complex conditions due to the combination of thermo-mechanical loading, the effects of irradiation and the harsh operating environment. This operating regime makes experimentation extremely difficult and we must rely on multi-physics and multi-scale modeling for our understanding of how these materials behave in service. This fundamental understanding helps not only to ensure the longevity of existing nuclear reactors, but also to guide the development of new materials for 4th generation reactor programs and dedicated fusion reactors. For the study of crystalline materials, an important tool is dislocation dynamics (DD) modeling. This multiscale simulation method predicts the plastic response of a material from the underlying physics of dislocation motion. DD serves as a crucial link between the scale of molecular dynamics and macroscopic methods based on finite elements; it can be used to accurately describe the interactions of a small handful of dislocations, or equally well to investigate the global behavior of a massive collection of interacting defects.

To explore i.e. to simulate these new areas, we need to develop and/or to improve significantly models, schemes and solvers used in the classical codes. In the project, we want to accelerate algorithms arising in those fields.

We will focus on the following topics (in particular in the currently under definition **OPTIDIS** project in collaboration with CEA Saclay, CEA Ile-de-france and SIMaP Laboratory in Grenoble) in connection with research described at Sections 3.4 and 3.5.

- The interaction between dislocations is long ranged ( $O(1/r)$ ) and anisotropic, leading to severe computational challenges for large-scale simulations. In dislocation codes, the computation of interaction forces between dislocations is still the most CPU time consuming and has to be improved to obtain faster and more accurate simulations.
- In such simulations, the number of dislocations grows while the phenomenon occurs and these dislocations are not uniformly distributed in the domain. This means that strategies to dynamically construct a good load balancing are crucial to achieve high performance.
- From a physical and a simulation point of view, it will be interesting to couple a molecular dynamics model (atomistic model) with a dislocation one (mesoscale model). In such three-dimensional coupling, the main difficulties are firstly to find and characterize a dislocation in the atomistic region, secondly to understand how we can transmit with consistency the information between the two micro and meso scales.

## 4.2 Co-design of algorithms in scientific applications

**Participants** Emmanuel Agullo, Olivier Beaumont, Lionel Eyraud-Dubois, Mathieu Faverge, Marek Felsoci, Luc Giraud, Gilles Marait, Pierre Ramet, Matthieu Simonin, Guillaume Sylvand, Alena Shilova.

### 4.2.1 High performance simulation for ITER tokamak

Scientific simulation for ITER tokamak modeling provides a natural bridge between theory and experimentation and is also an essential tool for understanding and predicting plasma behavior. Recent progresses in numerical simulation of fine-scale turbulence and in large-scale dynamics of magnetically confined plasma have been enabled by access to petascale supercomputers. These progresses would have been unreachable without new computational methods and adapted reduced models. In particular, the plasma science community has developed codes for which computer runtime scales quite well with the number of processors up to thousands cores. The research activities of **HIEPACS** concerning the international ITER challenge have started in the Inria Project Lab **C2S@EXA** in collaboration with

**CEA-IRFM** and were related to two complementary studies: a first one concerning the turbulence of plasma particles inside a tokamak (in the context of **GYSELA** code) and a second one concerning the MHD instability edge localized modes (in the context of **JOREK** code). The activity concerning **GYSELA** was completed at the end of 2018.

Other numerical simulation tools designed for the ITER challenge aim at making a significant progress in understanding active control methods of plasma edge MHD instability Edge Localized Modes (ELMs) which represent a particular danger with respect to heat and particle loads for Plasma Facing Components (PFC) in the tokamak. The goal is to improve the understanding of the related physics and to propose possible new strategies to improve effectiveness of ELM control techniques. The simulation tool used (**JOREK** code) is related to non linear MHD modeling and is based on a fully implicit time evolution scheme that leads to 3D large very badly conditioned sparse linear systems to be solved at every time step. In this context, the use of **PaStiX** library to solve efficiently these large sparse problems by a direct method is a challenging issue.

This activity continues within the context of the **EoCoE2** project, in which the **PaStiX** solver is identified to allow the processing of very larger linear systems for the nuclear fusion code **TOKAM3X** from **CEA-IRFM**. Contrary to the **JOREK** code, the problem to be treated corresponds to the complete 3D volume of the plasma torus. The objective is to be competitive, for complex geometries, compared to an Algebraic MultiGrid approach designed by one partner of **EoCoE2**.

#### 4.2.2 Numerical and parallel scalable hybrid solvers in large scale calculations

Parallel and numerically scalable hybrid solvers based on a fully algebraic coarse space correction have been theoretically studied and various advanced parallel implementations have been designed. Their parallel scalability has been initially investigated on large scale problems within the **EoCoE** project thanks to a close collaboration with the BSC and the integration of **MaPhyS** within the Alya software. This activity will further develop in the **EoCoE2** project. The performance has also been assessed on PRACE Tier-0 machine within a PRACE Project Access through a collaboration with CERFACS and Laboratoire de Physique des Plasmas at Ecole Polytechnique for the calculation of plasma propulsion. A comparative parallel scalability study with the Algebraic MultiGrid from Petsc has been conducted in that framework.

#### 4.3 Aeroacoustics Simulation

This domain is in the context of a long term collaboration with Airbus Research Centers. Wave propagation phenomena intervene in many different aspects of systems design at Airbus. They drive the level of acoustic vibrations that mechanical components have to sustain, a level that one may want to diminish for comfort reason (in the case of aircraft passengers, for instance) or for safety reason (to avoid damage in the case of a payload in a rocket fairing at take-off). Numerical simulations of these phenomena play a central part in the upstream design phase of any such project. Airbus Central R & T has developed over the last decades an in-depth knowledge in the field of Boundary Element Method (BEM) for the simulation of wave propagation in homogeneous media and in frequency domain. To tackle heterogeneous media (such as the jet engine flows, in the case of acoustic simulation), these BEM approaches are coupled with volumic finite elements (FEM). We end up with the need to solve large (several millions unknowns) linear systems of equations composed of a dense part (coming from the BEM domain) and a sparse part (coming from the FEM domain). Various parallel solution techniques are available today, mixing tools created by the academic world (such as the Mumps and PaStiX sparse solvers) as well as parallel software tools developed in-house at Airbus (dense solver SPIDO, multipole solver,  $\mathcal{H}$ -matrix solver with an open sequential version available online). In the current state of knowledge and technologies, these methods do not permit to tackle the simulation of aeroacoustics problems at the highest acoustic frequencies (between 5 and 20 kHz, upper limits of human audition) while considering the whole complexity of geometries and phenomena involved (higher acoustic frequency implies smaller mesh sizes that lead to larger unknowns number, a number that grows like  $f^2$  for BEM and  $f^3$  for FEM, where  $f$  is the studied frequency). The purpose of the study in this domain is to develop advanced solvers able to tackle this kind of mixed dense/sparse linear systems efficiently on parallel architectures.

## 4.4 Optimization for Deep Convolutional Neural Networks

The training phase of Deep Convolutional Neural Networks represents nowadays a significant share of the computations performed on HPC supercomputers. It introduces several new problems concerning resource allocation and scheduling issues, because of the specific pattern of task graphs induced by the stochastic gradient descent and because memory consumption is particularly critical when performing training. As of today, the most classical parallelization methods consists in partitioning mini-batches, images, filters,... but all these methods induce high synchronization and communication costs, and only very partially resolve memory issues. Within the framework of the Inria IPL on HPC Big Data and Learning convergence, we are working on re-materialization techniques and on the use of model parallelism, in particular to be able to build on the research that has been carried out in a more traditional HPC framework on the exploitation of resource heterogeneity and dynamic runtime scheduling.

## 5 Highlights of the year

The year 2020 was marked by the covid crisis and its impact on society and its overall activity. The world of research was also greatly affected: Faculty members have seen their teaching load increase significantly; PhD students and post-docs have often had to deal with a worsening of their working conditions, as well as with reduced interactions with their supervisors and colleagues; most scientific collaborations have been greatly affected, with many international activities cancelled or postponed to dates still to be defined.

**HIEPACS** members have participated to preparation of the following successful 2020 H2020 proposals:

- Textarossa (Call H2020-JTI-EuroHPC-2019-01): Towards Extreme Scale Technologies for EuroHPC HW/SW Supercomputing Applications for Exascale.
- EUPEX (EuroHPC-2020-01-a): European Pilot for Exascale.
- HPCQS (H2020-JTI-EuroHPC-2020-2): High Performance Computer and Quantum Simulator hybrid.
- RISC-2 (H2020-FETHPC-2018-2020) : A network for supporting the coordination of High-Performance Computing research between Europe and Latin America.

## 6 New software and platforms

### 6.1 New software

#### 6.1.1 AVCI

**Name:** Adaptive vibrational configuration interaction

**Keywords:** Vibrational spectra, Eigen value

**Functional Description:** A-VCI is a theoretical vibrational spectroscopy algorithm developed to effectively reduce the number of vibrational states used in the configuration-interaction (CI) process. It constructs a nested basis for the discretization of the Hamiltonian operator inside a large CI approximation space and uses an a-posteriori error estimator (residue) to select the most relevant directions to expand the discretization space.

The Hamiltonian operator consists of 3 operators: a harmonic oscillator sum, the potential energy surface operator and the Coriolis operators. In addition, the code can compute the intensity of eigenvectors.

The code can handle molecules up to 10 atoms, which corresponds to solving an eigenvalue problem in a 24-dimensional space.

**Publications:** [hal-01310708](#), [hal-02518243](#), [hal-01534134](#), [hal-02524533](#)

**Author:** Olivier Coulaud

**Contact:** Olivier Coulaud

**Partner:** IPREM

### 6.1.2 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>

**Authors:** Emmanuel Agullo, Mathieu Faverge, Samuel Thibault, Florent Pruvost

**Contacts:** Emmanuel Agullo, Florent Pruvost, Mathieu Faverge, Samuel Thibault

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

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

### 6.1.3 Diodon

**Name:** Diodon

**Keywords:** Dimensionality reduction, Data analysis

**Functional Description:** Most of dimension reduction methods inherited from Multivariate Data Analysis, and currently implemented as element in statistical learning for handling very large datasets (the dimension of spaces is the number of features) rely on a chain of pretreatments, a core with a SVD for low rank approximation of a given matrix, and a post-treatment for interpreting results. The costly part in computations is the SVD, which is in cubic complexity. Diodon is a list of functions and drivers which implement (i) pre-treatments, SVD and post-treatments on a large diversity of methods, (ii) random projection methods for running the SVD which permits to bypass the time limit in computing the SVD, and (iii) an implementation in C++ of the SVD with random projection at prescribed rank or precision, connected to MDS.

**URL:** <https://gitlab.inria.fr/afranc/diodon>

**Authors:** Alain Franc, Florent Pruvost, Romain Peressoni, Romain Peressoni

**Contacts:** Alain Franc, Olivier Coulaud

### 6.1.4 DPLASMA

**Name:** Distributed Parallel Linear Algebra Software for Multicore Architectures

**Keywords:** Dense linear algebra, Linear algebra

**Functional Description:** DPLASMA is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. DPLASMA achieves this objective through the state of the art PaRSEC runtime, porting the PLASMA algorithms to the distributed memory realm.

**URL:** <http://icl.cs.utk.edu/parsec/index.html>

**Author:** Mathieu Faverge

**Contact:** Mathieu Faverge

### 6.1.5 Fabulous

**Name:** Fast Accurate Block Linear krylOv Solver

**Keywords:** Numerical algorithm, Block Krylov solver

**Scientific Description:** Versatile and flexible numerical library that implements Block Krylov iterative schemes for the solution of linear systems of equations with multiple right-hand sides

**Functional Description:** Versatile and flexible numerical library that implements Block Krylov iterative schemes for the solution of linear systems of equations with multiple right-hand sides. The library implements block variants of minimal norm residual variants with partial convergence management and spectral information recycling. The package already implements regular block-GMRES (BGMRES), Inexact Breakdown BGMRES (IB-BMGRES), Inexact Breakdown BGMRES with Deflated Restarting (IB-BGMRES-DR), Block Generalized Conjugate Residual with partial convergence management. The C++ library relies on callback mechanisms to implement the calculations (matrix-vector, dot-product, ...) that depend on the parallel data distribution selected by the user.

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

**Publication:** hal-01067159

**Authors:** Luc Giraud, Emmanuel Agullo, Gilles Marait, Cyrille Piacibello, Thomas Mijieux

**Contacts:** Luc Giraud, Emmanuel Agullo

**Participants:** Emmanuel Agullo, Luc Giraud, Gilles Marait, Cyrille Piacibello, Matthieu Simonin

### 6.1.6 MAPHYS

**Name:** Massively Parallel Hybrid Solver

**Keyword:** Parallel hybrid direct/iterative solution of large linear systems

**Functional Description:** MaPHyS is a software package that implements a parallel linear solver coupling direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz. A fully algebraic coarse space is available for symmetric positive definite problems, that insures the numerical scalability of the preconditioner.

The parallel implementation is based on MPI+thread. Maphys relies on state-of-the art sparse and dense direct solvers.

MaPHyS is essentially a preconditioner that can be used to speed-up the convergence of any Krylov subspace method and is coupled with the ones implemented in the Fabulous package.

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

**Publications:** [hal-01379227](#), [hal-01399203](#)

**Contacts:** Luc Giraud, Emmanuel Agullo

**Participants:** Emmanuel Agullo, Luc Giraud, Matthieu Kuhn, Gilles Marait, Louis Poirel

### 6.1.7 MAPHYS++

**Name:** Massively Parallel Hybrid Solver in modern C++

**Keywords:** Linear Systems Solver, Parallel computing, C++, Parallel numerical solvers

**Scientific Description:** Maphys++ is a parallel linear solver for the resolution of large hollow linear systems that attempts to best combine the advantages of direct methods in terms of robustness and iterative methods in terms of memory footprint and adapted precision.

**Functional Description:** MAPHYS++ is a parallel linear solver for large sparse linear systems. It implements a modern C++ interface (C++17/20), giving the user a wide range of solving methods : efficient direct solver (wrapping MUMPS, Pastix...), iterative solver (CG, GMRES...) and also allowing for a combination of those (hybrid solve using the Schur complement or directly on the original matrix, with various preconditioners). Parallelism is based on domain decomposition methods and is implemented in MPI.

**News of the Year:** The redesign of the package has been completed

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

**Contacts:** Emmanuel Agullo, Gilles Marait, Luc Giraud

**Participants:** Emmanuel Agullo, Luc Giraud, Gilles Marait, Matthieu Simonin

### 6.1.8 StarPart

**Keywords:** High performance computing, HPC, Parallel computing, Graph algorithmics, Graph, Hypergraph

**Functional Description:** StarPart is a flexible and extensible framework that integrates state-of-the-art methods for graph partitioning and sparse matrix ordering. More precisely, StarPart is a framework that offers a uniform API to manipulate graph, hypergraph and mesh structures. It is designed to be easily extensible by adding new methods and to plug all these methods into a comprehensive framework. It is initially designed to provide graph partitioning and sparse matrix ordering methods, that come from state-of-the-art software such as Metis, Scotch, Patoh, Zoltan, etc. Besides, it provides some facilities for IO, diagnostic, benchmark, visualization (VTK, SVG, ...). StarPart is the core of the MetaPart project. It is built upon the LibGraph library.

**URL:** <https://gitlab.inria.fr/metapart/starpart>

**Contact:** Aurélien Esnard

**Participant:** Aurélien Esnard

### 6.1.9 MPICPL

**Name:** MPI CouPLing

**Keywords:** MPI, Coupling software



**Functional Description:** MPICPL is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the mpicplrun tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the mpicplrun tool.

**URL:** <https://gitlab.inria.fr/esnard/mpicpl>

**Contact:** Aurélien Esnard

**Participant:** Aurélien Esnard

### 6.1.10 OptiDis

**Keywords:** Dislocation dynamics simulation, Fast multipole method, Large scale, Collision

**Functional Description:** OptiDis is a new code for large scale dislocation dynamics simulations. Its purpose is to simulate real life dislocation densities (up to 5.1022 dislocations/m<sup>2</sup>) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation of irradiated zirconium. Zirconium alloys are the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking about channeling mechanism, which means to stick with the reality, more than tens of thousands of induced loops, i. e. 100 million degrees of freedom in the simulation. The code is based on Numodis code developed at CEA Saclay and the ScalFMM library developed in HiePACS project. The code is written in C++ language and using the last features of C++11/14. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rises. In order to achieve that, we use different levels of parallelism. First of all, the simulation box is distributed over MPI processes, then we use a thinner level for threads, dividing the domain by an Octree representation. All these parts are controlled by the ScalFMM library. On the last level, our data are stored in an adaptive structure that absorbs the dynamics of this type of simulation and manages the parallelism of tasks..

**URL:** <http://optidis.gforge.inria.fr/>

**Publication:** hal-01086371

**Authors:** Olivier Coulaud, Arnaud Etcheverry, Pierre Blanchard, Arnaud Durocher

**Contact:** Olivier Coulaud

**Participant:** Olivier Coulaud

**Partner:** CEA

### 6.1.11 PaStiX

**Name:** Parallel Sparse matrix package

**Keywords:** Linear algebra, High-performance calculation, Sparse Matrices, Linear Systems Solver, Low-Rank compression

**Scientific Description:** PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handle a combination of 1D and 2D block distributions. A dynamic scheduling can also be applied to take care of NUMA architectures while taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations.



**Functional Description:** PaStiX is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) methods. It can handle low-rank compression techniques to reduce the computation and the memory complexity. Numerical algorithms are implemented in single or double precision (real or complex) for LLt, LDLt and LU factorization with static pivoting (for non symmetric matrices having a symmetric pattern). The PaStiX library uses the graph partitioning and sparse matrix block ordering packages Scotch or Metis.

The PaStiX solver is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes with GPU accelerators or KNL processors. In particular, we provide a high-performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

The solver also provides some low-rank compression methods to reduce the memory footprint and/or the time-to-solution.

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

**Authors:** Xavier Lacoste, Pierre Ramet, Mathieu Faverge, Pascal Hénon, Tony Delarue, Eragul Korkmaz, Grégoire Pichon

**Contacts:** Pierre Ramet, Mathieu Faverge

**Participants:** Tony Delarue, Grégoire Pichon, Mathieu Faverge, Eragul Korkmaz, Pierre Ramet

**Partners:** INP Bordeaux, Université de Bordeaux

#### 6.1.12 pmtool

**Keywords:** Scheduling, Task scheduling, StarPU, Heterogeneity, GPGPU, Performance analysis

**Functional Description:** Analyse post-mortem the behavior of StarPU applications. Provide lower bounds on makespan. Study the performance of different schedulers in a simple context. Provide implementations of many scheduling algorithms from the literature

**News of the Year:** Included many new algorithms, in particular online algorithms Better integration with StarPU by accepting .rec files as input

**URL:** <https://gitlab.inria.fr/eyrauddu/pmtool>

**Publications:** [hal-01386174](#), [hal-01878606](#)

**Contact:** Lionel Eyraud-Dubois

**Participant:** Lionel Eyraud-Dubois

#### 6.1.13 rotor

**Name:** Re-materializing Optimally with pyTORch

**Keywords:** Deep learning, Optimization, Python, GPU, Automatic differentiation

**Scientific Description:** This software implements in PyTorch a new activation checkpointing method which allows to significantly decrease memory usage when training Deep Neural Networks with the back-propagation algorithm. Similarly to checkpointing techniques coming from the literature on Automatic Differentiation, it consists in dynamically selecting the forward activations that are saved during the training phase, and then automatically recomputing missing activations from those previously recorded. We propose an original computation model that combines two types of activation savings: either only storing the layer inputs, or recording the complete history of operations that produced the outputs (this uses more memory, but requires fewer recomputations in the

backward phase), and we provide in <https://hal.inria.fr/hal-02352969> an algorithm to compute the optimal computation sequence for this model.

Our PyTorch implementation processes the entire chain, dealing with any sequential DNN whose internal layers may be arbitrarily complex and automatically executing it according to the optimal checkpointing strategy computed given a memory limit. In <https://hal.inria.fr/hal-02352969>, through extensive experiments, we show that our implementation consistently outperforms existing checkpointing approaches for a large class of networks, image sizes and batch sizes.

**Functional Description:** Allows to train very large convolutional networks on limited memory by optimally selecting which activations should be kept and which should be recomputed. This code is meant to replace the `checkpoint.py` utility available in `pytorch`, by providing more efficient rematerialization strategies. The algorithm is easier to tune: the only required parameter is the available memory, instead of the number of segments.

**Publication:** [hal-02352969](https://hal.inria.fr/hal-02352969)

**Authors:** Lionel Eyraud-Dubois, Olivier Beaumont, Alena Shilova, Julien Herrmann

**Contacts:** Lionel Eyraud-Dubois, Olivier Beaumont, Alena Shilova

**Participants:** Olivier Beaumont, Alena Shilova, Alexis Joly, Lionel Eyraud-Dubois, Julien Herrmann

#### 6.1.14 ScalFMM

**Name:** Scalable Fast Multipole Method

**Keywords:** N-body, Fast multipole method, Parallelism, MPI, OpenMP

**Scientific Description:** ScalFMM is a software library to simulate N-body interactions using the Fast Multipole Method. The library offers two methods to compute interactions between bodies when the potential decays like  $1/r$ . The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve @ Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There is two main parts:

the management of the octree and the parallelization of the method the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization.

**Functional Description:** Compute N-body interactions using the Fast Multipole Method for large number of objects

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

**Authors:** Bramas Bérenger, Olivier Coulaud

**Contact:** Olivier Coulaud

**Participants:** Bramas Bérenger, Olivier Coulaud, Pierre Estérie

### 6.1.15 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:** <http://vite.gforge.inria.fr/>

**Authors:** Mathieu Faverge, Nicolas Richart, Cédric Augonnet

**Contact:** Mathieu Faverge

**Participant:** Mathieu Faverge

### 6.1.16 PlaFRIM

**Name:** Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

**Keywords:** High-Performance Computing, Hardware platform

**Functional Description:** PlaFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. It is now more than 1,000 cores (excluding GPU and Xeon Phi ) that are available for all research teams of Inria Bordeaux, Labri and IMB. This computer is in particular used by all the engineers who work in HiePACS and are advised by F. Rue from the SED.

**URL:** <https://www.plafrim.fr/en/home/>

**Contact:** Olivier Coulaud

## 7 New results

### 7.1 High performance solvers for large linear algebra problems

#### 7.1.1 Exploring variable accuracy storage through lossy compression techniques in numerical linear algebra: a first application to flexible GMRES

In the context of the Inria International Lab JLESC we have an ongoing collaboration with Argonne National Laboratory on the use of agnostic compression techniques to reduce the memory footprint of iterative linear solvers. Large scale applications running on HPC systems often require a substantial amount of memory and can have a large data movement overhead. Lossy data compression techniques can reduce the size of the data and associated communication cost, but the effect of the loss of accuracy on the numerical algorithm can be hard to predict. In this work we examine the FGMRES algorithm, which requires the storage of a basis for the search space and another one for the space spanned by the residuals. We show that the vectors spanning this search space can be compressed by looking at the combination of FGMRES and compression in the context of inexact Krylov subspace methods. This allows us to derive a bound on the normwise relative compression error in each iteration. We use this bound to formulate a number of different practical compression strategies, and validate and compare them through numerical experiments. We refer the reader to [34] for a detailed presentation of this work.

### 7.1.2 A block minimum residual norm subspace linear solver for sequences of multiple left and right-hand side linear systems

We are concerned with the iterative solution of linear systems with multiple right-hand sides available one group after another, including the case where there are massive number (like tens of thousands) of right-hand sides associated with a single matrix so that all of them cannot be solved at once but rather need to be split into chunks of possible variable sizes. For such sequences of linear systems with multiple left and right-hand sides, we develop a new recycling block generalized conjugate residual method with inner orthogonalization and inexact breakdown (IB-BGCRO-DR), which glues subspace recycling technique in GCRO-DR [SIAM J. Sci. Comput., 28(5) (2006), pp 1651,1674] and inexact breakdown mechanism in IB-BGMRES [Linear Algebra Appl., 419 (2006), pp. 265,285] to guarantee this new algorithm could reuse spectral information for subsequent cycles as well as for the remaining linear systems to be solved. Related variant IB-BFGCRO-DR that suits for flexible preconditioning is devised to cope with constraints on some applications while also enabling mixed-precision calculation, which provides advantages in speed and memory usage over double precision as well as in perspective of emerging computing units such as the GPUs. Besides, we also discuss why/ how many possible choices are available when building recycling subspace as well as how to exploit the inexact breakdown mechanism for realizing the flexibility of search space expansion policies and monitoring individual convergence thresholds for each single right-hand side. As a side effect one also illustrates that this method can be applied to constant/ slowly-varying left-hand side. Finally we demonstrate the numerical and computational benefits of combining these ideas in such algorithms on a set of simple academic examples with a sequence of right-hand sides.

A scientific report should appear shortly that will detailed this work. Furthermore, a C++ implementation in the Fabulous package is also an ongoing work.

### 7.1.3 2D Static Resource Allocation for Compressed Linear Algebra and Communication Constraints

In [24], we consider the design of static resource allocation problems for irregular distributed parallel applications. More precisely, we focus on two classical tiled linear algebra kernels: the Matrix Multiplication (MM) and the LU decomposition (LU) algorithms on large linear systems. In the context of parallel distributed platforms, data exchanges can dramatically degrade the performance of linear algebra kernels and in this context, compression techniques such as Block Low Rank (BLR) compression techniques are good candidates both for limiting data storage on each processor and data exchanges between processors. On the other hand, the use of BLR representation makes the static allocation problem of tiles to processors more complex. Indeed, the load associated to each tile depends on its compression factor, which induces an heterogeneous load balancing problem. In turn, solving this load balancing problem optimally might lead to complex allocation schemes, where the tiles allocated to a given processor are scattered all over the matrix. This in turn induces communication costs, since matrix multiplication and LU decompositions heavily rely on broadcasting operations along rows and columns of processors, so that the communication volume is minimized when the maximal number of different processors on any row and column is minimized. In the fully homogeneous case, 2D Block Cyclic (BC) allocation solves both load balancing and communication minimization issues simultaneously, but it might lead to bad load balancing in the heterogeneous case. Our goal in this paper is to propose data allocation schemes dedicated to BLR format and to prove that it is possible to obtain good performance on makespan when simultaneously balancing the load and minimizing the maximal number of different processor in any row or column.

### 7.1.4 On soft errors in the Conjugate Gradient method: sensitivity and robust numerical detection

The conjugate gradient (CG) method is the most widely used iterative scheme for the solution of large sparse systems of linear equations when the matrix is symmetric positive definite. Although more than sixty year old, it is still a serious candidate for extreme-scale computations on large computing platforms. On the technological side, the continuous shrinking of transistor geometry and the increasing complexity of these devices affect dramatically their sensitivity to natural radiation, and thus diminish their reliability. One of the most common effects produced by natural radiation is the single event upset which consists in a bit-flip in a memory cell producing unexpected results at application level. Consequently, the future extreme scale computing facilities will be more prone to errors of any kind, including bit-flips, during

their calculations. These numerical and technological observations are the main motivations for this work, where we first investigate through extensive numerical experiments the sensitivity of CG to bit-flips in its main computationally intensive kernels, namely the matrix-vector product and the preconditioner application. We further propose numerical criteria to detect the occurrence of such soft errors and assess their robustness through extensive numerical experiments.

We have been working on the revision of this work that has been eventually published in [15]; a complementary study on persistent faults is detailed in [35].

### 7.1.5 Recent Developments Around the Block Low-Rank PaStiX Solver

Recent works in sparse direct solvers have multiplied the solutions to exploit data-sparsity additionally to sparse structure in these solvers. Among these solution, the PaStiX solver exploits block low-rank structures to reduce both its memory consumption and time to solution. In this talk, we will discuss recent changes made to the solver in this context such as the study of suitable low-rank compression kernels, adapted ordering heuristics, or scheduling strategies through the use of runtime systems. Performance results of the latest release including the presented features will be shown and discussed.

This work has been presented at the international conference SIAM PP 2020 [33].

### 7.1.6 Deciding Non-Compressible Blocks in Sparse Direct Solvers using Incomplete Factorization

Low-rank compression techniques are very promising for reducing memory footprint and execution time on a large spectrum of linear solvers. Sparse direct supernodal approaches are one these techniques. However, despite providing a very good scalability and reducing the memory footprint, they suffer from an important flops overhead in their unstructured low-rank updates. As a consequence, the execution time is not improved as expected.

In this paper, we study a solution to improve low-rank compression techniques in sparse supernodal solvers. The proposed method tackles the overprice of the low-rank updates by identifying the blocks that have poor compression rates. We show that block incomplete LU factorization, thanks to the block fill-in levels, allows us to identify most of these non-compressible blocks at low cost. This identification enables us to postpone the low-rank compression step to trade small extra memory consumption for a better time to solution. The solution is validated within the PaStiX library with a large set of application matrices. It demonstrates sequential and multi-threaded speedup up to 84%, for small memory overhead of less than 3% with respect to the original version.

This work will be submitted at the international conference EuroPar 2021.

## 7.2 Applications domains

### 7.2.1 Material physics

The Adaptive Vibrational Configuration Interaction (A-VCI) algorithm is an iterative process that computes the anharmonic spectrum of a molecule using nested bases to discretize the Hamiltonian operator in high dimension. For large molecular systems, the size of the discretization space, the computation time and the memory consumption become prohibitive especially when are looking eigenvalues in the middle of the spectra. It is therefore necessary to develop new methods to further limit the number of basis functions.

To overcome this issues, we have worked this year in two direction.

- Most of the time, the interpretation of an experimental infrared spectrum does not require the calculation of all eigenvalues but only those corresponding to vibrational states with significant intensity. In [20] a technique that uses infrared intensities is introduced to select a subset of eigenvalues to be precisely calculated. Thus, we build smaller nested bases and reduce both the memory footprint and the computational time. We validate the advantages of this new approach on a well-studied 7-atom molecular system (C<sub>2</sub>H<sub>4</sub>O), and we apply it on a larger 10-atom molecule (C<sub>4</sub>H<sub>4</sub>N<sub>2</sub>).
- We have adapted the FEAST algorithm to search for eigenvalue in an interval to the tensor formalism and more particularly to the tensor train (TT) format when both operators and eigenvectors are

written in TT format. The method uses the contour integral theory to construct the projector of the space generated by the target eigenvectors. The main problem is to construct the projector and to control its rank, in order to quickly compute a solution. We have proposed an orthogonalization scheme based on the gram matrix and the preliminary results obtained are promising on small molecules (dimensions 3 and 6). An article is in progress.

### 7.2.2 Metabarcoding

This year, we focused mainly on the work of comparing point clouds obtained by multidimensional scaling calculations. Our goal is to find a way to automatically compare these clouds without having to look at them manually in order to judge whether or not they have the same shape. To do this, we took into account Hausdorff's distance. However this metric had several problems like the presence of outliers i.e. few points that are very far away but affect the distance by lot, different orientations of the cloud shape. We have proposed an algorithm that overcomes these difficulties and gives very unsuccessful results. To improve these performances, we now compute distances in larger spaces. That led to issues with the cost of alignment which rise exponentially with the number of dimensions studied. We have proposed a heuristic that allows us to compare the shapes after passing them through a process that have the same output regardless of the symmetries present. A report is being written.

We are involved in the ADT Gordon ((partners: TADAAM (coordinator), STORM, HIEPACS, PLEIADE). The objectives of this ADT is to scale our solver stack on a PLEIADE dimensioning metabarcoding application (multidimensional scaling method). Our goal is to be able to handle a problem leading to a distance matrix around 100 million individuals. Our contribution concerns the the scalability of the multidimensional scaling method and more particularly the random projection methods to speed up the SVD solver. Experiments on occigen platform has allowed us to show that the solver stack was able to solve efficiently a large problem up to one million individuals in less than 22 minutes on 100 nodes. This has highlighted that for these problem sizes the management of I/O, inputs and outputs with the disks, becomes critical and dominates calculation times.

Concerning the use of tensor formalism for data analysis, we worked on two aspects in collaboration with EPC PLEIADE (A. Franc). Firstly, a methodology was set up to analyze the structured data of the Malabar project. These data are genetic data collected in the Arcachon bay. The frequencies of these genetic sequences were classified according to their species, zone, time and position in the water column of collection. On the frequency tensor, the Tucker decomposition without or with the positivity constraint of the tensor was applied and made it possible to review the techniques for extracting the relevant information. In a second step, we were interested in the Correspondance Analysis and its geometric interpretation (of point clouds) from frequency matrices and their studies. The analysis of point clouds is possible thanks to a linear relationship existing between the point clouds. We have studied how to extend this relationship to three point clouds, by applying some properties developed in the context of Tucker's hierarchy. An article is in progress.

### 7.2.3 Co-design for scalable algorithms in scientific applications

**New developments regarding the JOREK solver and physics based preconditioner** The JOREK extended magneto-hydrodynamic (MHD) code is a widely used simulation code for studying the non-linear dynamics of large-scale instabilities in divertor tokamak plasmas. Due to the large scale-separation intrinsic to these phenomena both in space and time, the computational costs for simulations in realistic geometry and with realistic parameters can be very high, motivating the investment of considerable effort for optimization. The code is usually run with a fully implicit time integration allowing to use large time steps independent of a CFL criterion. This is particularly important due to the fast time scales associated with MHD waves and fast parallel heat transport. For solving the resulting large sparse-matrix system iteratively in each time step, a physics-based preconditioner building on the assumption of a weak coupling between the toroidal harmonics is applied. The solution for each harmonic matrix is determined independently in this preconditioner using a direct sparse matrix library. In this article, a set of developments regarding the JOREK solver and preconditioner is described, which lead to overall significant benefits for large production simulations. This comprises in particular enhanced convergence in highly non-linear scenarios and a general reduction of memory consumption and computational



costs. The developments include faster construction of preconditioner matrices, a domain decomposition of preconditioning matrices for solver libraries that can handle distributed matrices, interfaces for additional solver libraries, an option to use matrix compression methods, and the implementation of a complex solver interface for the preconditioner. The most significant development presented consists in a generalization of the physics based preconditioner to “mode groups”, which allows to account for the dominant interactions between toroidal Fourier modes in highly non-linear simulations. At the cost of a moderate increase of memory consumption, the technique can strongly enhance convergence in suitable cases allowing to use significantly larger time steps. For all developments, benchmarks based on typical simulation cases demonstrate the resulting improvements.

**Numerical and parallel scalable hybrid solvers in large scale calculations** In the context of a parallel plasma physics simulation code, we perform a qualitative performance study between two natural candidates for the parallel solution of 3D Poisson problems that are multigrid and domain decomposition. We selected one representative of each of these numerical techniques implemented in state of the art parallel packages and show that depending on the regime used in terms of number of unknowns per computing cores the best alternative in terms of time to solution varies. Those results show the interest of having both types of numerical solvers integrated in a simulation code that can be used in very different configurations in terms of problem sizes and parallel computing platforms. More information on these results will be shortly available in an Inria scientific report.

**Fast Solvers for High Frequency Aeroacoustics** Wave propagation phenomena intervene in many different aspects of systems design at Airbus. They drive the level of acoustic vibrations that mechanical components have to sustain, a level that one may want to diminish for comfort reason (in the case of aircraft passengers, for instance) or for safety reason (to avoid damage in the case of a payload in a rocket fairing at take-off). Numerical simulations of these phenomena plays a central part in the upstream design phase of any such project. Airbus Central R&T has developed over the last decades an in-depth knowledge in the field of Boundary Element Method (BEM) for the simulation of wave propagation in homogeneous media and in frequency domain. To tackle heterogeneous media (such as the jet engine flows, in the case of acoustic simulation), these BEM approaches are coupled with volumic finite elements (FEM). We end up with the need to solve large (several millions unknowns) linear systems of equations composed of a dense part (coming for the BEM domain) and a sparse part (coming from the FEM domain). Various parallel solution techniques are available today, mixing tools created by the academic world (such as the Mumps and Pastix sparse solvers) as well as parallel software tools developed in-house at Airbus (dense solver SPIDO, multipole solver, H-matrix solver with an open sequential version available online `hmatoss`). In the current state of knowledge and technologies, these methods do not permit to tackle the simulation of aeroacoustics problems at the highest acoustic frequencies (between 5 and 20 kHz, upper limits of human audition) while considering the whole complexity of geometries and phenomena involved (higher acoustic frequency implies smaller mesh sizes that lead to larger unknowns number, a number that grows like  $f^2$  for BEM and  $f^3$  for FEM, where  $f$  is the studied frequency). In this research project, we are interested in many aspects that include the design of new advanced techniques to solve these large mixed dense/sparse linear systems, the extensive comparison of these new approaches to the existing ones, and the application of these innovative ideas on realistic industrial test cases.

- The use of H-matrix solvers on these problems will be investigated (in the continuation of a PhD currently in progress). Airbus CR&T, in collaboration with Inria Bordeaux Sud-Ouest, has developed a task-based H-matrix solver on top of the runtime engine StarPU. The question of parallel scalability of task-based tools is an active subject of research, using new communication engine such as NewMadeleine, that will be investigated during this project, in conjunction with new algorithmic ideas on the task-based writing of H-matrix algorithms.
- Ideas coming from the field of sparse direct solvers (such as nested dissection or symbolic factorization) have been tested within H-matrices. This activity will be continued and extended.
- Comparison with existing tools will be performed on large realistic test cases. Coupling schemes between these tools and the hierarchical methods used in H-matrix will be developed and benched as well.

**Compression by Tensor Decompositions for a scattering problem** It is common to accelerate the boundary element method by compression techniques (FMM,  $\mathcal{H}$ -matrix/ACA) that enable a more accurate solution or a solution in higher frequency. In this article, we present a compression method based on a transformation of the linear system into the tensor-train format by the quantization technique. The method is applied to a scattering problem on a canonical object with a regular mesh and improves the performance obtained from existing methods. This method has been published in [21].

## 7.3 Optimization for the training phase of deep convolutional networks

### 7.3.1 Memory optimization

Training Deep Neural Networks is known to be an expensive operation, both in terms of computational cost and memory load. Indeed, during training, all intermediate layer outputs (called activations) computed during the forward phase must be stored until the corresponding gradient has been computed in the backward phase. These memory requirements sometimes prevent to consider larger batch sizes and deeper networks, so that they can limit both convergence speed and accuracy.

Recent works have proposed to offload some of the computed forward activations from the memory of the GPU to the memory of the CPU. This requires to determine which activations should be offloaded and when these transfers from and to the memory of the GPU should take place. In [23], We prove that this problem is NP-hard in the strong sense, and we propose two heuristics based on relaxations of the problem. We perform extensive experimental evaluation on standard Deep Neural Networks. We compare the performance of our heuristics against previous approaches from the literature, showing that they achieve much better performance in a wide variety of situations.

### 7.3.2 Optimization of Pipelined Model Parallelism

In [38] we consider the complexity of the optimization problems induced by the use of pipelined model parallelism for the training phase in Deep Neural Networks has become an important source of computing resource usage and because of the resulting volume of computation, it is crucial to perform it efficiently on parallel architectures. Even today, data parallelism is the most widely used method, but the associated requirement to replicate all the weights on the totality of computation resources poses problems of memory at the level of each node and of collective communications at the level of the platform. In this context, the model parallelism, which consists in distributing the different layers of the network over the computing nodes, is an attractive alternative. Indeed, it is expected to better distribute weights (to cope with memory problems) and it does not imply large collective communications since only forward activations are communicated. However, to be efficient, it must be combined with a pipelined / streaming approach, which leads in turn to new memory costs. The goal of this paper is to model these memory costs in detail, to analyze the complexity of the associated throughput optimization problem under memory constraints and to show that it is possible to formalize this optimization problem as an Integer Linear Program (ILP). In [37], we formalize in detail the optimization problem associated with the placement of DNN layers onto computation resources when using pipelined model parallelism, and we derive a dynamic programming based heuristic, MadPipe, that allows to significantly improve the performance of the parallel model approach compared to the literature.

## 7.4 Scheduling

### 7.4.1 Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing

In order to express parallelism, parallel sparse direct solvers take advantage of the elimination tree to exhibit tree-shaped task graphs, where nodes represent computational tasks and edges represent data dependencies. One of the pre-processing stages of sparse direct solvers consists of mapping computational resources (processors) to these tasks. The objective is to minimize the factorization time by exhibiting good data locality and load balancing. The proportional mapping technique is a widely used approach to solve this resource-allocation problem. It achieves good data locality by assigning the same processors to large parts of the elimination tree. However, it may limit load balancing in some cases. In this paper, we



propose a dynamic mapping algorithm based on proportional mapping. This new approach, named Steal, relaxes the data locality criterion to improve load balancing. In order to validate the newly introduced method, we perform extensive experiments on the PaStiX sparse direct solver. It demonstrates that our algorithm enables better static scheduling of the numerical factorization while keeping good data locality.

This work has been presented at the international conference EuroPar 2020 [29].

#### 7.4.2 Efficient Ordering of Kernel Submission on GPUs

In distributed memory systems, it is paramount to develop strategies to overlap the data transfers between memory nodes with the computations in order to exploit their full potential. In [28], we consider the problem of determining the order of data transfers between two memory nodes for a set of independent tasks with the objective of minimizing the makespan. We prove that, with limited memory capacity, the problem of obtaining the optimal data transfer order is NP-complete. We propose several heuristics to determine this order and discuss the conditions that might be favorable to different heuristics. We analyze our heuristics on traces obtained by running two molecular chemistry kernels, namely, Hartree-Fock (HF) and Coupled Cluster Singles Doubles (CCSD), on 10 nodes of an HPC system. Our results show that some of our heuristics achieve significant overlap for moderate memory capacities and resulting in makespans that are very close to the lower bound.

Concurrent kernel execution is a relatively new feature in modern GPUs, which was designed to improve hardware utilization and the overall system throughput. However, the decision on the simultaneous execution of tasks is performed by the hardware with a leftover policy, that assigns as many resources as possible for one task and then assigns the remaining resources to the next task. This can lead to unreasonable use of resources. In [40, 27], we tackle the problem of co-scheduling for GPUs with and without preemption, with the focus on determining the kernels submission order to reduce the number of preemptions and the kernels makespan, respectively. We propose a graph-based theoretical model to build preemptive and non-preemptive schedules. We show that the optimal preemptive makespan can be computed by solving a Linear Program in polynomial time, and we propose an algorithm based on this solution which minimizes the number of preemptions. We also propose an algorithm that transforms a preemptive solution of optimal makespan into a non-preemptive solution with the smallest possible preemption overhead. We show, however, that finding the minimal amount of preemptions among all preemptive solutions of optimal makespan is a NP-hard problem, and computing the optimal non-preemptive schedule is also NP-hard. In addition, we study the non-preemptive problem, without searching first for a good preemptive solution, and present a Mixed Integer Linear Program solution to this problem. We performed experiments on real-world GPU applications and our approach can achieve optimal makespan by preempting 6 to 9% of the tasks. Our non-preemptive approach, on the other side, obtains makespan within 2.5% of the optimal preemptive schedules, while previous approaches exceed the preemptive makespan by 5 to 12%.

#### 7.4.3 Scheduling Tasks on Two Types of Resources

We study in [17] the problem of executing an application represented by a precedence task graph on a parallel machine composed of standard computing cores and accelerators. Contrary to most existing approaches, we distinguish the allocation and the scheduling phases and we mainly focus on the allocation part of the problem: choose the most appropriate type of computing unit for each task. We address both off-line and on-line settings and design generic scheduling approaches. In the first case, we establish strong lower bounds on the worst-case performance of a known approach based on Linear Programming for solving the allocation problem. Then, we refine the scheduling phase and we replace the greedy List Scheduling policy used in this approach by a better ordering of the tasks. Although this modification leads to the same approximability guarantees, it performs much better in practice. We also extend this algorithm to more types of computing units, achieving an approximation ratio which depends on the number of different types. In the on-line case, we assume that the tasks arrive in any, not known in advance, order which respects the precedence relations and the scheduler has to take irrevocable decisions about their allocation and execution. In this setting, we propose the first on-line scheduling algorithm which takes into account precedences. Our algorithm is based on adequate rules for selecting the type of processor where to allocate the tasks and it achieves a constant factor approximation

guarantee if the ratio of the number of CPUs over the number of GPUs is bounded. Finally, all the previous algorithms for hybrid architectures have been experimented , on a large set of common benchmarks that we made available for the community. These simulations assess the good practical behavior of the algorithms with respect to the state-of-the-art solutions, whenever these exist, or baseline algorithms.

#### 7.4.4 Geometric Deep Reinforcement Learning for Dynamic DAG Scheduling

Along a slightly different direction, we consider in [30] the use of reinforcement learning to help to design efficient dynamic schedulers to be used in runtime systems such as StarPU. In practice, it is quite common to face combinatorial optimization problems which contain uncertainty along with non-determinism and dynamicity. These three properties call for appropriate algorithms; reinforcement learning (RL) is dealing with them in a very natural way. Today, despite some efforts, most real-life combinatorial optimization problems remain out of the reach of reinforcement learning algorithms. In this paper, we propose a reinforcement learning approach to solve a realistic scheduling problem, and apply it to an algorithm commonly executed in the high performance computing community, the Cholesky factorization. On the contrary to static scheduling, where tasks are assigned to processors in a predetermined ordering before the beginning of the parallel execution, our method is dynamic: task allocations and their execution ordering are decided at runtime, based on the system state and unexpected events, which allows much more flexibility. To do so, our algorithm uses graph neural networks in combination with an actor-critic algorithm (A2C) to build an adaptive representation of the problem on the fly. We show that this approach is competitive with state-of-the-art heuristics used in high-performance computing runtime systems. Moreover, our algorithm does not require an explicit model of the environment, but we demonstrate that extra knowledge can easily be incorporated and improves performance. We also exhibit key properties provided by this RL approach, and study its transfer abilities to other instances.

#### 7.4.5 Approximation Algorithm for Estimating Distances in Distributed Virtual Environments

We are also interested in the possibility of performing extra (redundant) computations in order to save messages. This possibility might be of interest, for instance, in the context of energy savings, and it is crucial in contexts such as on-line games. In [22], we consider the issue of guaranteeing properties in Distributed Virtual Environments (DVEs) without a server and without global knowledge of the system state and therefore only by exchanging messages. This issue is particularly relevant in the case of online games, that operate in a fully distributed framework and for which network resources such as bandwidth are the critical resources. In the context of games, players typically need to know the distance between their character and other characters, at least approximately. Players all share the same position estimation algorithm but, in general, do not know the current positions of others. We provide a synchronized distributed algorithm Alc to guarantee, at any time, that the estimated distance  $d_{est}$  between any pair of characters A and B is always a  $(1 + \epsilon)$  approximation of the current distance. Our result is twofold: (1) we prove that if characters move randomly on a  $d$ -dimensional grid, or follow a random continuous movement on up to three dimensions, the number of messages of Alc is optimal up to a constant factor; (2) in a more practical setting, we also observe that the number of messages of Alc for actual game traces is much less than the standard algorithm sending actual positions at a given frequency.

#### 7.4.6 A Makespan Lower Bound for the Scheduling of the Tiled Cholesky Factorization based on ALAP Schedule

In [25], we consider the theoretical problem of deriving strong bounds on the running time of Cholesky factorization, performed on homogeneous (GPUs or CPUs) resources. Due to the advent of multicore architectures and massive parallelism, the tiled Cholesky factorization algorithm has recently received plenty of attention and is often referenced by practitioners as a case study. It is also implemented in mainstream dense linear algebra libraries and is used as a testbed for runtime systems. However, we note that theoretical study of the parallelism of this algorithm is currently lacking. In this paper, we present new theoretical results about the tiled Cholesky factorization in the context of a parallel homogeneous model without communication costs. Based on the relative costs of involved kernels, we prove that only two different situations must be considered, typically corresponding to CPUs and GPUs. By a careful analysis on the number of tasks of each type that run simultaneously in the ALAP (As Late As Possible)

schedule without resource limitation, we are able to determine precisely the number of busy processors at any time (as degree 2 polynomials). We then use this information to find a closed form formula for the minimum time to schedule a tiled Cholesky factorization of size  $n$  on  $P$  processors. We show that this bound outperforms classical bounds from the literature. We also prove that ALAP( $P$ ), an ALAP-based schedule where the number of resources is limited to  $P$ , has a makespan extremely close to the lower bound, thus proving both the effectiveness of ALAP( $P$ ) schedule and of the lower bound on the makespan.

#### 7.4.7 Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs

In [18], we consider the influence on data-locality of the replication of data files, as automatically performed by Distributed File Systems such as HDFS. Replication of data files, as automatically performed by Distributed File Systems such as HDFS, is known to have a crucial impact on data locality in addition to system fault tolerance. Indeed, intuitively, having more replicas of the same input file gives more opportunities for this task to be processed locally, i.e. without any input file transfer. Given the practical importance of this problem, a vast literature has been proposed to schedule tasks, based on a random placement of replicated input files. Our goal in this paper is to study the performance of these algorithms, both in terms of makespan minimization (minimize the completion time of the last task when non-local processing is forbidden) and communication minimization (minimize the number of non-local tasks when no idle time on resources is allowed). In the case of homogenous tasks, we are able to prove, using models based on "balls into bins" and "power of two choices" problems, that the well known good behavior of classical strategies can be theoretically grounded. Going further, we even establish that it is possible, using semi-matchings theory, to find the optimal solution in very small time. We also use known graph-orientation results to prove that this optimal solution is indeed near-perfect with strong probability. In the more general case of heterogeneous tasks, we propose heuristics solutions both in the clairvoyant and non-clairvoyant cases (i.e. task length is known in advance or not), and we evaluate them through simulations, using actual traces of a Hadoop cluster.

### 7.5 Efficient algorithmic for load balancing and code coupling in complex simulations

In the context of the french ICARUS project (FUI), which focuses the development of high-fidelity calculation tools for the design of hot engine parts (aeronautics & automotive), we are looking to develop new load-balancing algorithms to optimize the complex numerical simulations of our industrial and academic partners (Turbomeca, Siemens, Cerfacs, Onera, ...). Indeed, the efficient execution of large-scale coupled simulations on powerful computers is a real challenge, which requires revisiting traditional load-balancing algorithms based on graph partitioning. A thesis on this subject has already been conducted in the Inria HiePACS team in 2016, which has successfully developed a co-partitioning algorithm that balances the load of two coupled codes by taking into account the coupling interactions between these codes.

This work was initially integrated into the StarPart platform. The necessary extension of our algorithms to parallel & distributed (increasingly dynamic) versions has led to a complete redesign of StarPart, which has been the focus of our efforts this year (as in the previous year).

The StarPart framework provides the necessary building blocks to develop new graph algorithms in the context of HPC, such as those we are targeting. The strength of StarPart lies in the fact that it is a light runtime system applied to the issue of "graph computing". It provides a unified data model and a uniform programming interface that allows easy access to a dozen partitioning libraries, including Metis, Scotch, Zoltan, etc. Thus, it is possible, for example, to load a mesh from an industrial test case provided by our partners (or an academic graph collection as DIMACS'10) and to easily compare the results for the different partitioners integrated in StarPart.

Alongside this work, we are beginning to work on the application of learning techniques to the problem of graph partitioning. Recent work on GCNs (Graph Convolutional Networks) is an interesting approach that we will explore.

## 8 Partnerships and cooperations

### 8.1 International initiatives

#### 8.1.1 Inria International Labs

There is an ongoing research activity with Argonne National Laboratory in the framework of the JLESC International Lab, through a postdoc funded by the DPI, namely Nick Schenkels, who work on data compression techniques in Krylov methods for the solution of large linear systems. The objective is to use agnostic compressor developed at Argonne to compress the basis involved in Krylov methods that have a large memory footprint. The challenge is to design algorithm that reduce the memory consumption, hence the energy, while preserving the numerical convergence of the numerical technique. We refer to [34] for a detailed description of the work.

#### 8.1.2 Inria associate team not involved in an ILL

##### MOLIERE

**Title:** Moliere: Memory Optimization for parallel training of deep neural networks

**Duration:** 2020 - 2022

**Coordinator:** Olivier Beaumont

**Partner:** Skoltech – Skolkovo Institute of Science and Technology (Russian Federation)

**Inria contact:** Olivier Beaumont

**Summary:** Training for Deep Learning Networks (DNNs) has become a major compute intensive application. Nevertheless, even if training is still in general performed on small clusters of GPU machines, the use of large HPC infrastructures is becoming popular, in particular because they offer high bandwidth and low latency networks. An important limitation of hyper-parameter tuning and data parallelism approaches is that they do not help in considering larger models by solving memory issues during the training phase. Indeed, in both settings, the whole set of weights has to be stored on all participating resources. In general, the memory consumed by the training phase consists in two main parts. The first part is related to the storage of the parameters of the network and is directly related to the size of the model. For both hyper-parameters and data parallelism approaches, these weights have to be replicated on every node, and, in the case of data parallelism, they even have to be aggregated and broadcast on the network after each parallel mini-batch training phase. The second source of memory consumption is due to the storage on each node of all the forward activations (i.e. all the outputs of the different stages of the network) until the backward phase. This part is directly proportional to the size of the mini-batch itself. In this context, our goal in the framework of this associated team is to provide new techniques to both increase parallelism while limiting memory requirements, by exploring techniques based on compression and tensor train decomposition and checkpointing and offloading techniques to control the memory consumed during the training phase.

### 8.2 European initiatives

#### 8.2.1 FP7 & H2020 Projects

##### EoCoE-II

**Title:** Energy oriented Centre of Excellence for computer applications

**Duration:** January 2019 - December 2021

**Coordinator:** CEA

**Partners:**

- Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)
- Commissariat A L' Energie Atomique et Aux Energies Alternatives (France)
- Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (France)
- Consiglio Nazionale Delle Ricerche (Italy)
- The Cyprus Institute (Cyprus)
- Agenzia Nazionale Per le Nuove Tecnologie, l'energia E Lo Sviluppo Economico Sostenibile (Italy)
- Fraunhofer Gesellschaft Zur Forderung Der Angewandten Forschung Ev (Germany) Instytut Chemii Bioorganicznej Polskiej Akademii Nauk (Poland) Forschungszentrum Julich (Germany)
- Max Planck Gesellschaft Zur Foerderung Der Wissenschaften E.V. (Germany) University of Bath (United Kingdom)
- Université Libre de Bruxelles (Belgium)
- Universita Degli Studi di Trento (Italy)

**Inria contact:** Bruno Raffin

**Summary:** The Energy-oriented Centre of Excellence (EoCoE) applies cutting-edge computational methods in its mission to accelerate the transition to the production, storage and management of clean, decarbonized energy. EoCoE is anchored in the High Performance Computing (HPC) community and targets research institutes, key commercial players and SMEs who develop and enable energy-relevant numerical models to be run on exascale supercomputers, demonstrating their benefits for low carbon energy technology. The present project will draw on a successful proof-of-principle phase of EoCoE-I, where a large set of diverse computer applications from four such energy domains achieved significant efficiency gains thanks to its multidisciplinary expertise in applied mathematics and supercomputing. During this 2nd round, EoCoE-II will channel its efforts into 5 scientific Exascale challenges in the low-carbon sectors of Energy Meteorology, Materials, Water, Wind and Fusion. This multidisciplinary effort will harness innovations in computer science and mathematical algorithms within a tightly integrated co-design approach to overcome performance bottlenecks and to anticipate future HPC hardware developments. A world-class consortium of 18 complementary partners from 7 countries will form a unique network of expertise in energy science, scientific computing and HPC, including 3 leading European supercomputing centres. New modeling capabilities in selected energy sectors will be created at unprecedented scale, demonstrating the potential benefits to the energy industry, such as accelerated design of storage devices, high-resolution probabilistic wind and solar forecasting for the power grid and quantitative understanding of plasma core-edge interactions in ITER-scale tokamaks. These flagship applications will provide a high-visibility platform for high-performance computational energy science, cross-fertilized through close working connections to the EERA and EUROfusion consortia.

## **PRACE-6IP**

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

**Duration:** May 2019 - December 2021

**Coordinator:** FORSCHUNGSZENTRUM JULICH GMBH

### **Partners:**

- AKADEMIA GORNICZO-HUTNICZA IM. STANISLAWA STASZICA W KRAKOWIE (Poland)
- ASSOCIACAO DO INSTITUTO SUPERIOR TECNICO PARA A INVESTIGACAO E DESENVOLVIMENTO (Portugal)

- ASSOCIATION "NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS (Bulgaria)
- BARCELONA SUPERCOMPUTING CENTER - CENTRO NACIONAL DE SUPERCOMPUTACION (Spain)
- BAYERISCHE AKADEMIE DER WISSENSCHAFTEN (Germany)
- BILKENT UNIVERSITESI VAKIF (Turkey)
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (France)
- CENTRUM SPOLOCNYCH CINNOSTI SLOVENSKEJ AKADEMIE VIED (Slovakia)
- CINECA CONSORZIO INTERUNIVERSITARIO (Italy)
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (France)
- DANMARKS TEKNISKE UNIVERSITET (Denmark)
- EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH (Switzerland)
- FORSCHUNGSZENTRUM JULICH GMBH (Germany)
- FUNDACION PUBLICA GALLEGA CENTRO TECNOLOGICO DE SUPERCOMPUTACION DE GALICIA (Spain)
- GEANT VERENIGING (Netherlands)
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (France)
- Gauss Centre for Supercomputing (GCS) e.V. (Germany)
- ISTANBUL TEKNIK UNIVERSITESI (Turkey)
- KOBENHAVNS UNIVERSITET (Denmark)
- KORMANYZATI INFORMATIKAI FEJLESZTESI UGYNOKSEG (Hungary)
- KUNGLIGA TEKNISKA HOEGSKOLAN (Sweden)
- LINKOPINGS UNIVERSITET (Sweden)
- MACHBA - INTERUNIVERSITY COMPUTATION CENTER (Israel)
- MAX-PLANCK-GESELLSCHAFT ZUR FORDERUNG DER WISSENSCHAFTEN EV (Germany)
- NATIONAL UNIVERSITY OF IRELAND GALWAY (Ireland)
- NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET NTNU (Norway)
- PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE AISBL (Belgium)
- POLITECHNIKA GDANSKA (Poland)
- POLITECHNIKA WROCLAWSKA (Poland)
- SYDDANSK UNIVERSITET (Denmark)
- TECHNISCHE UNIVERSITAET WIEN (Austria)
- THE CYPRUS INSTITUTE (Cyprus)
- THE UNIVERSITY OF EDINBURGH (U.K.)
- UNINETT SIGMA2 AS (Norway)
- UNITED KINGDOM RESEARCH AND INNOVATION (U.K.)
- UNIVERSIDADE DE COIMBRA (Portugal)
- UNIVERSIDADE DE EVORA (Portugal)
- UNIVERSIDADE DO MINHO (Portugal)
- UNIVERSIDADE DO PORTO (Portugal)
- UNIVERSITAET INNSBRUCK (Austria)
- UNIVERSITAET STUTTGART (Germany)
- UNIVERSITE DU LUXEMBOURG (Luxembourg)

- UNIVERSITEIT ANTWERPEN (Belgium)
- UNIVERSITETET I OSLO (Norway)
- UNIVERZA V LJUBLJANI (Slovenia)
- UPPSALA UNIVERSITET (Sweden)
- VSB - Technical University of Ostrava (Czech Republic)

**Inria contact:** Luc Giraud

**Summary:** PRACE-6IP, 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.

HIEPACS is involved in WP6 where it acts as an expert for a Preparatory Access entitled "Fast Multidimensional Methods" led by INRAe. HIEPACS is also involved in a project part of WP8 entitled "Linear Algebra, Krylov-subspace methods, and multi-grid solvers for the discovery of New Physics" lead by CaSToRC (The Cyprus Institute) in collaboration with Leibniz Supercomputing Center.

### **EOSC-Pillar**

**Title:** Coordination and Harmonisation of National Initiatives, Infrastructures and Data services in Central and Western Europe

**Duration:** 2019 - 2023

**Coordinator:** Federico Ruggeri

### **Partners:**

- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (France)
- CINECA CONSORZIO INTERUNIVERSITARIO (Italy)
- COMPLA SRL (Italy)
- CONSIGLIO NAZIONALE DELLE RICERCHE (Italy)
- CONSORTIUM GARR (Italy)
- FONDAZIONE CENTRO EURO-MEDITERRANEOSUI CAMBIAMENTI CLIMATICI (Italy)



- FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (Germany)
- HELMHOLTZ ZENTRUM POTSDAM DEUTSCHESGEOFORSCHUNGSZENTRUM GFZ (Germany)
- INSTITUT NATIONAL DE RECHERCHE POUR L'AGRICULTURE, L'ALIMENTATION ET L'ENVIRONNEMENT (France)
- ISTITUTO NAZIONALE DI FISICA NUCLEARE (Italy)
- KARLSRUHER INSTITUT FUER TECHNOLOGIE (Germany)
- UNIVERSITAT WIEN (Austria)
- UNIVERSITEIT GENT (Belgium)

**Inria contact:** Stefano Zacchiroli

**Summary:** he project aims to support the coordination and harmonization of national initiatives relevant to EOSC in Europe and investigate the option for them to interfederate at a later stage, help integrating initiatives and data/cloud providers through the development of common policies and tools, and facilitate user communities in adopting and using these services and propose new ones born from their scientific domain. To this end, the project will integrate a bottom-up approach (by voicing the requirements and needs expressed by the different scientific communities operating at the national level) and a top-down one (by harmonising the national strategies and translating them in a viable work plan). In the longer term, this is expected to facilitate the design and adoption of common policies and streamline the process of joining EOSC for service providers and user communities while helping populating the EOSC with useful services of wider European interest, based on the real needs and interests of the European scientific communities. In order to maximise this simplification process, the project will collaborate with related regional and thematic initiatives.

### 8.3 National initiatives

#### 8.3.1 ANR

**SASHIMI: Sparse Direct Solver using Hierarchical Matrices**

**Duration:** 2018 – 2022

**Coordinator:** Mathieu Faverge

**Summary:** Nowadays, the number of computational cores in supercomputers has grown largely to a few millions. However, the amount of memory available has not followed this trend, and the memory per core ratio is decreasing quickly with the advent of accelerators. To face this problem, the SaSHiMi project wants to tackle the memory consumption of linear solver libraries used by many major simulation applications by using low-rank compression techniques. In particular, the direct solvers which offer the most robust solution to strategy but suffer from their memory cost. The project will especially investigate the super-nodal approaches for which low-rank compression techniques have been less studied despite the attraction of their large parallelism and their lower memory cost than for the multi-frontal approaches. The results will be integrated in the PaStiX solver that supports distributed and heterogeneous architectures.

**SOLHARIS: SOLvers for Heterogeneous Architectures over Runtime systems, Investigating Scalability**

**Duration:** 2018 – 2022

**Coordinator:** Alfredo Buttari (IRIT)

**Inria contact:** Abdou Guermouche



**Partners:**

- IRIT Institut de Recherche en Informatique de Toulouse
- Inria Bordeaux - Sud-Ouest and Lyon
- Airbus Central R&T
- CEA Commissariat à l'énergie atomique et aux énergies alternatives

**Summary:** The **SOLHARIS** project aims at addressing the issues related to the development of fast and scalable linear solvers for large-scale, heterogeneous supercomputers. Because of the complexity and heterogeneity of the targeted algorithms and platforms, this project intends to rely on modern runtime systems to achieve high performance, programmability and portability. By gathering experts in computational linear algebra, scheduling algorithms and runtimes, **SOLHARIS** intends to tackle these issues through a considerable research effort for the development of numerical algorithms and scheduling methods that are better suited to the characteristics of large scale, heterogeneous systems and for the improvement and extension of runtime systems with novel features that more accurately fulfill the requirements of these methods. This is expected to lead to fundamental research results and software of great interest for researchers of the scientific computing community.

**8.3.2 FUI****ICARUS: Intensive Calculation for AeRo and automotive engines Unsteady Simulations**

**Duration:** 2018 – 2022

**Coordinator:** SAFRAN

**Inria contact:** Aurélien Esnard

**Partners:**

- CENAERO
- CERFACS
- CORIA
- DISTENE
- GDTECH
- IFPEN
- ONERA
- SAFRAN
- SIEMENS

**Summary:** Large Eddy Simulation (LES) is an increasingly attractive unsteady modelling approach for modelling reactive turbulent flows due to the constant development of massively parallel supercomputers. It can provide open and robust design tools that allow access to new concepts (technological breakthroughs) or a global consideration of a structure (currently processed locally). The mastery of this method is therefore a major competitive lever for industry. However, it is currently constrained by its access and implementation costs in an industrial context. The ICARUS project aims to significantly reduce them (costs and deadlines) by bringing together major industrial and research players to work on the entire high-fidelity LES computing process by:

- increasing the performance of existing reference tools (for 3D codes: AVBP, Yales2, ARGO) both in the field of code coupling and code/machine matching;
- developing methodologies and networking tools for the LES;

- adapting the ergonomics of these tools to the industrial world: interfaces, data management, code interoperability and integrated chains;
- validating this work on existing demonstrators, representative of the aeronautics and automotive industries.

### 8.3.3 Inria Project Labs

#### Challenge HPC BigData

**Duration:** 2018 – 2022

**Coordinator:** Bruno Raffin

**HIEPACS contact:** Olivier Beaumont & Olivier Coulaud

#### Inria teams:

- KerData
- SequeL
- Sierra
- Tau
- Zenith
- Parietal
- TADaaM
- HiePACSD
- Storm

**Summary:** The goal of the Challenge on HPC-BigData is to gather teams from the HPC, Big Data and Machine Learning (ML) areas to work at the intersection between these domains. HPC and Big Data evolved with their own infrastructures (supercomputers versus clouds), applications (scientific simulations versus data analytics) and software tools (MPI and OpenMP versus Map/Reduce or Deep Learning frameworks). But Big Data analytics is becoming more compute-intensive (thanks to deep learning), while data handling is becoming a major concern for scientific computing. Within the IPL, we are in particular involved in a tight collaboration with Zenith Team (Montpellier) on how to parallelize and how to deal with memory issues in the context of the training phase of Pl@ntnet (<https://www.plantnet.org>). Alexis Joly (Zenith) co supervises with Olivier Beaumont the PhD Thesis of Alena Shilova. We are also collaborating with Sequel (Nathan Grinsztajn and Philippe Preux) and Tadaam (Emmanuel Jeannot) teams on the design of dynamic runtime schedulers based on reinforcement learning.

## 8.4 Regional initiatives

**Title:** HPC-Ecosystem

**Duration:** 2018-2020

**Coordinator:** E. Agullo

#### Partners:

- **STORM, TADAAM** from Inria Bordeaux Sud-Ouest
- Airbus
- CEA-CESTA

**Description:** Numerical simulation is today integrated in all cycles of scientific design and studies, whether academic or industrial, to predict or understand the behavior of complex phenomena often coupled or multi-physical. The quality of the prediction requires having precise and adapted models, but also to have computation algorithms efficiently implemented on computers with architectures in permanent evolution. Given the ever increasing size and sophistication of simulations implemented, the use of parallel computing on computers with up to several hundred thousand computing cores and consuming / generating massive volumes of data becomes unavoidable; this domain corresponds to what is now called High Performance Computing (HPC). On the other hand, the digitization of many processes and the proliferation of connected objects of all kinds generate ever-increasing volumes of data that contain multiple valuable information; these can only be highlighted through sophisticated treatments; we are talking about Big Data. The intrinsic complexity of these digital treatments requires a holistic approach with collaborations of multidisciplinary teams capable of mastering all the scientific skills required for each component of this chain of expertise. To have a real impact on scientific progress and advances, these skills must include the efficient management of the massive number of compute nodes using programming paradigms with a high level of expressiveness, exploiting high-performance communications layers, effective management for intensive I / O, efficient scheduling mechanisms on platforms with a large number of computing units and massive I / O volumes, innovative and powerful numerical methods for analyzing volumes of data produced and efficient algorithms that can be integrated into applications representing recognized scientific challenges with high societal and economic impacts. The project we propose aims to consider each of these links in a consistent, coherent and consolidated way. For this purpose, we propose to develop a unified Execution Support (SE) for large-scale numerical simulation and the processing of large volumes of data. We identified four Application Challenges (DA) identified by the Nouvelle-Aquitaine region that we propose to carry over this unified support. We will finally develop four Methodological Challenges (CM) to evaluate the impact of the project. This project will make a significant contribution to the emerging synergy on the convergence between two yet relatively distinct domains, namely High Performance Computing (HPC) and the processing, management of large masses of data (Big Data); this project is therefore clearly part of the emerging field of High Performance Data Analytics (HPDA).

## 9 Dissemination

### 9.1 Promoting scientific activities

#### 9.1.1 Scientific events: organisation

**Member of the organizing committees** SIAM Conference on Parallel Processing for Scientific Computing, SIAM PP 2020 <https://www.siam.org/conferences/cm/conference/pp20> (O. Beaumont)

**Chair of conference program committees** SBAC-PAD'21 (O. Beaumont, Program Co-Chair)

**Member of the conference program committees** EuroPar'20 (O. Beaumont), AlgoCloud'20 (O. Beaumont), HPML'20 (O. Beaumont), MSPW'20 (O. Beaumont), IPDPS Workshops'20 (O. Beaumont), ISC'20 Workshops (E. Agullo), PDSEC'20 (O. Coulaud, L. Giraud), SC'20 (A. Guermouche), ICPP'20 (A. Guermouche), SC'21 (O. Beaumont)

#### 9.1.2 Journal

##### Member of the editorial boards

- O. Beaumont is Associate Editor in Chief for [the Journal of Parallel and Distributed Computing \(Elsevier\)](#).
- L. Giraud is member of the editorial board of the SIAM Journal on Scientific Computing ([SISC](#)) and SIAM Journal on Matrix Analysis and Applications ([SIMAX](#)).

**Reviewer - reviewing activities** The members of the **HIEPACS** project have performed reviewing for the following list of journals: Computer Methods in Applied Mechanics and Engineering, SIAM J. Matrix Analysis and Applications, SIAM J. Scientific Computing, Elsevier Applied Numerical Mathematics, Elsevier Journal of Computational Physics, Elsevier Journal of Parallel and Distributed Computing, Elsevier Journal of Systems and Software, Elsevier Journal of Scheduling, IEEE Transactions on Parallel and Distributed Systems.

### 9.1.3 Scientific expertise

- Olivier Beaumont acted as expert for **2020 FET Open Evaluation Committee**.
- Luc Giraud is member of the board on Modelization, Simulation and data analysis of the **Competitiveness Cluster for Aeronautics, Space and Embedded Systems**. He is also member of the scientific council of the **ONERA Lab LMA2S** (Laboratoire de Mathématiques Appliquées à l'Aéronautique et au Spatial).
- Pierre Ramet is Scientific Advisor at the CEA-DAM CESTA.
- Guillaume Sylvand is Expert in Numerical Simulation and HPC at Airbus.

### 9.1.4 Research administration

- P. Ramet is the head of the SATANAS (Supports and Algorithms for High Performance Numerical Applications) team from LaBRI UMR CNRS 5800.

## 9.2 Teaching - Supervision - Juries

### 9.2.1 Teaching

- Undergraduate level/Licence:
  - A. Esnard: System programming 36h, Computer architecture 40h, Network 23h, C programming 35h at Bordeaux University. He is also responsible for the second year of the computer science degree (L2 Informatique), which involves managing about 200 students each year.
  - M. Faverge: Programming environment 26h, Numerical algorithmic 40h, C projects 25h at Bordeaux INP (ENSEIRB-MatMeca).
  - A. Guermouche: System programming 36h at Bordeaux University.
  - P. Ramet: System programming 24h, Databases 32h, Object programming 48h, Distributed programming 16h, Cryptography 16h, Introduction to AI Deep Learning and Data Analytics 16h at Bordeaux University.
- Post graduate level/Master:
  - E. Agullo: Operating systems 24h at Bordeaux University ; Dense linear algebra kernels 8h, Numerical algorithms 30h at Bordeaux INP (ENSEIRB-MatMeca).
  - O. Coulaud: Paradigms for parallel computing 8h, Introduction to Tensor methods 4h at Bordeaux INP (ENSEIRB-MatMeca).
  - A. Esnard: Network management 27h, Network security 27h at Bordeaux University; Programming distributed applications 35h at Bordeaux INP (ENSEIRB-MatMeca).
  - L. Eyraud-Dubois and Olivier Beaumont: Approximation and BigData 24h at Bordeaux University.
  - M. Faverge: System programming 72h, Linear Algebra for high Performance Computing 13h at Bordeaux INP (ENSEIRB-MatMeca). He is also in charge of the master 2 internship for the Computer Science department at Bordeaux INP (ENSEIRB-MatMeca). Starting in September, he is in charge with Raymond Namyst of the High Performance Computing -

High Performance Data Analytics option at ENSEIRB-MatMeca. This is a common training curriculum between the Computer Science and MatMeca departments at Bordeaux INP and with Bordeaux University in the context of the Computer Science Research Master.

- Alena Shilova and Olivier Beaumont: Deep Learning Frameworks, at Bordeaux INP (ENSEIRB-MatMeca), 20h.
  - Olivier Beaumont, Sketching and Streaming Algorithms, ENS Lyon, 8h
  - L. Giraud: Introduction to intensive computing and related programming tools 30h, INSA Toulouse; On mathematical tools for numerical simulations 10h, ENSEEIHT Toulouse.
  - A. Guermouche: Network management 92h, Network security 64h, Operating system 24h at Bordeaux University.
  - P. Ramet: Cryptography 20h and Numerical algorithmic 40h at Bordeaux INP (ENSEIRB-Matmeca).
- High School teachers:
    - A. Esnard, M. Faverge, and A. Guermouche participated to the training of the High School teachers (DIU Enseigner l’Informatique au Lycée) in computer science for the new computer science program starting in September 2019.

### 9.2.2 Supervision

- PhD in progress: Tobias Castanet; Replication algorithms for multi-player virtual worlds; started Sep. 2019; O. Beaumont, N. Hanusse (LaBRI), C. Travers (Bordeaux INP - LaBRI).
- PhD in progress: Marek Felsoci; Fast solvers for high-frequency aeroacoustics; started Oct. 2019; G. Sylvand, E. Agullo.
- PhD in progress: Martina Iannacito; Linear solvers in tensorial format for high dimensional problems; started Oct 2019; O. Coulaud, L. Giraud.
- PhD in progress: Eragul Korkmaz; Sparse linear solver and hierachical matrices; started Nov. 2018; M. Faverge, P. Ramet.
- PhD in progress: Romain Peressoni; Fast multidimensional scaling method for the study of biodiversity; started Oct 2019; E. Agullo, O. Coulaud, A. Franc ([PLEIADE](#))
- PhD in progress: Alena Shilova; Scheduling for deep learning applications; started Oct. 2018; L. Eyraud-Dubois, O. Beaumont.
- PhD in progress: Nicolas Venkovic; Domain decomposition techniques for the solution of stochastic elliptic PDEs; started Nov. 2018; L. Giraud, P. Mycek (Cerfacs).
- PhD in progress: Mathieu Vérité; Static allocation algorithms for scheduling High-Performance applications; started Sept. 2019; L. Eyraud-Dubois, O. Beaumont.
- PhD in progress: Yanfei Xiang; Solution of large linear systems with massive numbers of right-hand sides. Started Nov. 2019; L. Giraud, P. Mycek (Cerfacs).

### 9.2.3 Juries

- Clément Mommessin, “Efficient Management of Resources in Heterogeneous Platforms”; referees: Anne Benoit and Krzysztof Rządca; members: Olivier Beaumont, Anne Benoit, Noel de Palma, Giorgio Lucarelli, Yanick Ngoko, Krzysztof Rządca and Denis Trystram; Université de Grenoble, December 11, 2020.
- Valentin Le Fevre, “Resilient scheduling algorithms for large-scale platforms”; referees: Henri Casanova and Rami Mehlem; members: Olivier Beaumont, Anne Benoit, Henri Casanova, Amina Guermouche, Rami Mehlem and Yves Robert; ENS Lyon, June 18, 2020.

### 9.2.4 Interventions

In the framework of the "Digital Career Week", Olivier Beaumont met high school students (with the NSI option) from the Lycée de la Mer (Gujan Mestras) for about two hours.

## 10 Scientific production

### 10.1 Major publications

- [1] E. Agullo, B. Bramas, O. Coulaud, E. Darve, M. Messner and T. Takahashi. 'Task-Based FMM for Multicore Architectures'. In: *SIAM Journal on Scientific Computing* 36.1 (2014), pp. 66–93. DOI: [10.1137/130915662](https://doi.org/10.1137/130915662). URL: <https://hal.inria.fr/hal-00911856>.
- [2] E. Agullo, A. Buttari, A. Guermouche and F. Lopez. 'Implementing multifrontal sparse solvers for multicore architectures with Sequential Task Flow runtime systems'. In: *ACM Transactions on Mathematical Software* (July 2016). DOI: [10.1145/2898348](https://doi.org/10.1145/2898348). URL: <https://hal.inria.fr/hal-01333645>.
- [3] E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, N. Schenkels and W. Vanroose. 'On soft errors in the conjugate gradient method: sensitivity and robust numerical detection'. In: *SIAM Journal on Scientific Computing* 42.6 (Nov. 2020). DOI: [10.1137/18M122858X](https://doi.org/10.1137/18M122858X). URL: <https://hal.inria.fr/hal-03022845>.
- [4] E. Agullo, L. Giraud and Y.-F. Jing. 'Block GMRES method with inexact breakdowns and deflated restarting'. In: *SIAM Journal on Matrix Analysis and Applications* 35.4 (Nov. 2014), pp. 1625–1651. DOI: [10.1137/140961912](https://doi.org/10.1137/140961912). URL: <https://hal.inria.fr/hal-01067159>.
- [5] E. Agullo, L. Giraud and L. Poirel. 'Robust preconditioners via generalized eigenproblems for hybrid sparse linear solvers'. In: *SIAM Journal on Matrix Analysis and Applications* 40.2 (2019), pp. 417–439. DOI: [10.1137/17M1153765](https://doi.org/10.1137/17M1153765). URL: <https://hal.inria.fr/hal-02074474>.
- [6] O. Beaumont, L.-C. Canon, L. Eyraud-Dubois, G. Lucarelli, L. Marchal, C. Mommessin, B. Simon and D. Trystram. 'Scheduling on Two Types of Resources: a Survey'. In: *ACM Computing Surveys* 53.3 (May 2020). DOI: [10.1145/3387110](https://doi.org/10.1145/3387110). URL: <https://hal.inria.fr/hal-02432381>.
- [7] O. Beaumont, L. Eyraud-Dubois and S. Kumar. 'Fast Approximation Algorithms for Task-Based Runtime Systems'. In: *Concurrency and Computation: Practice and Experience* 30.17 (Sept. 2018). DOI: [10.1002/cpe.4502](https://doi.org/10.1002/cpe.4502). URL: <https://hal.inria.fr/hal-01878606>.
- [8] O. Beaumont, L. Eyraud-Dubois and A. Shilova. 'Optimal GPU-CPU Offloading Strategies for Deep Neural Network Training'. In: *Euro-Par 2020 - 26th International Conference on Parallel and Distributed Computing*. Ed. by M. Malawski and K. Rzadca. Euro-Par 2020: Parallel Processing 12247. Warsaw / Virtual, Poland: Springer, Aug. 2020, pp. 151–166. DOI: [10.1007/978-3-030-57675-2\\_10](https://doi.org/10.1007/978-3-030-57675-2_10). URL: <https://hal.inria.fr/hal-02316266>.
- [9] O. Beaumont, T. Lambert, L. Marchal and B. Thomas. 'Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs'. In: *Future Generation Computer Systems* 111 (Oct. 2020), pp. 582–598. DOI: [10.1016/j.future.2019.08.024](https://doi.org/10.1016/j.future.2019.08.024). URL: <https://hal.inria.fr/hal-02275473>.
- [10] A. Hugo, A. Guermouche, P.-A. Wacrenier and R. Namyst. 'Composing multiple StarPU applications over heterogeneous machines: A supervised approach'. In: *International Journal of High Performance Computing Applications* 28 (Feb. 2014), pp. 285–300. DOI: [10.1177/1094342014527575](https://doi.org/10.1177/1094342014527575). URL: <https://hal.inria.fr/hal-01101045>.
- [11] M. Odunlami, V. Le Bris, D. Bégué, I. Baraille and O. Coulaud. 'A-VCI: A flexible method to efficiently compute vibrational spectra'. In: *Journal of Chemical Physics* 146.21 (June 2017). DOI: [10.1063/1.4984266](https://doi.org/10.1063/1.4984266). URL: <https://hal.inria.fr/hal-01534134>.
- [12] G. Pichon, E. Darve, M. Faverge, P. Ramet and J. Roman. 'Sparse supernodal solver using block low-rank compression: Design, performance and analysis'. In: *International Journal of Computational Science and Engineering* 27 (July 2018), pp. 255–270. DOI: [10.1016/J.JOCS.2018.06.007](https://doi.org/10.1016/J.JOCS.2018.06.007). URL: <https://hal.inria.fr/hal-01824275>.

- [13] G. Pichon, M. Faverge, P. Ramet and J. Roman. ‘Reordering Strategy for Blocking Optimization in Sparse Linear Solvers’. In: *SIAM Journal on Matrix Analysis and Applications*. SIAM Journal on Matrix Analysis and Applications 38.1 (2017), pp. 226–248. DOI: [10.1137/16M1062454](https://doi.org/10.1137/16M1062454). URL: <https://hal.inria.fr/hal-01485507>.
- [14] M. Predari, A. Esnard and J. Roman. ‘Comparison of initial partitioning methods for multilevel direct k-way graph partitioning with fixed vertices’. In: *Parallel Computing* (2017). DOI: [10.1016/j.parco.2017.05.002](https://doi.org/10.1016/j.parco.2017.05.002). URL: <https://hal.inria.fr/hal-01538600>.

## 10.2 Publications of the year

### International journals

- [15] E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, N. Schenkels and W. Vanroose. ‘On soft errors in the conjugate gradient method: sensitivity and robust numerical detection’. In: *SIAM Journal on Scientific Computing* 42.6 (2nd Nov. 2020). DOI: [10.1137/18M122858X](https://doi.org/10.1137/18M122858X). URL: <https://hal.inria.fr/hal-03022845>.
- [16] B. Alzaix, L. Giraud, B. Michielsen and J.-R. Poirier. ‘A Plane Wave Scattering Dedicated Integral Equation’. In: *IEEE Transactions on Antennas and Propagation* (Feb. 2020), pp. 1–10. DOI: [10.1109/TAP.2019.2948390](https://doi.org/10.1109/TAP.2019.2948390). URL: <https://hal.archives-ouvertes.fr/hal-02448907>.
- [17] O. Beaumont, L.-C. Canon, L. Eyraud-Dubois, G. Lucarelli, L. Marchal, C. Mommessin, B. Simon and D. Trystram. ‘Scheduling on Two Types of Resources: a Survey’. In: *ACM Computing Surveys* 53.3 (1st May 2020). DOI: [10.1145/3387110](https://doi.org/10.1145/3387110). URL: <https://hal.inria.fr/hal-02432381>.
- [18] O. Beaumont, T. Lambert, L. Marchal and B. Thomas. ‘Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs’. In: *Future Generation Computer Systems* 111 (Oct. 2020), pp. 582–598. DOI: [10.1016/j.future.2019.08.024](https://doi.org/10.1016/j.future.2019.08.024). URL: <https://hal.inria.fr/hal-02275473>.
- [19] T. Benacchio, L. Bonaventura, M. Altenbernd, C. D. Cantwell, P. D. Düben, M. Gillard, L. Giraud, D. Göddeke, E. Raffin, K. Teranishi and N. Wedi. ‘Resilience and fault tolerance in high-performance computing for numerical weather and climate prediction’. In: *International Journal of High Performance Computing Applications* (8th Feb. 2021). DOI: [10.1177/1094342021990433](https://doi.org/10.1177/1094342021990433). URL: <https://hal.inria.fr/hal-03138061>.
- [20] V. Le Bris, M. Odunlami, D. Bégué, I. Baraille and O. Coulaud. ‘Using computed infrared intensities for the reduction of vibrational configuration interaction bases’. In: *Physical Chemistry Chemical Physics* (2020). DOI: [10.1039/D0CP00593B](https://doi.org/10.1039/D0CP00593B). URL: <https://hal.archives-ouvertes.fr/hal-02524533>.
- [21] J.-R. Poirier, O. Coulaud and O. Kaya. ‘Fast BEM Solution for 2-D Scattering Problems Using Quantized Tensor-Train Format’. In: *IEEE Transactions on Magnetism* 56.3 (Mar. 2020), pp. 1–4. DOI: [10.1109/TMAG.2019.2954584](https://doi.org/10.1109/TMAG.2019.2954584). URL: <https://hal.inria.fr/hal-03150956>.

### International peer-reviewed conferences

- [22] O. Beaumont, T. Castanet, N. Hanusse and C. Travers. ‘Approximation Algorithm for Estimating Distances in Distributed Virtual Environments’. In: *Euro-Par 2020: Parallel Processing*. Euro-Par 2020 - 26th International European Conference on Parallel and Distributed Computing. Vol. Lecture Notes in Computer Science. 12247. Warsaw / Virtual, Poland: <https://2020.euro-par.org/>, 20th Feb. 2020. DOI: [10.1007/978-3-030-57675-2\\_23](https://doi.org/10.1007/978-3-030-57675-2_23). URL: <https://hal.archives-ouvertes.fr/hal-02486218>.
- [23] O. Beaumont, L. Eyraud-Dubois and A. Shilova. ‘Optimal GPU-CPU Offloading Strategies for Deep Neural Network Training’. In: Euro-Par 2020 - 26th International Conference on Parallel and Distributed Computing. Euro-Par 2020: Parallel Processing 12247. Warsaw / Virtual, Poland: <https://2020.euro-par.org/>, Aug. 2020, pp. 151–166. DOI: [10.1007/978-3-030-57675-2\\_10](https://doi.org/10.1007/978-3-030-57675-2_10). URL: <https://hal.inria.fr/hal-02316266>.



- [24] O. Beaumont, L. Eyraud-Dubois and M. Verite. ‘2D Static Resource Allocation for Compressed Linear Algebra and Communication Constraints’. In: HIPC 2020: 27th IEEE International Conference on High Performance Computing, Data, and Analytics. Proceedings of the 27th IEEE International Conference on High Performance Computing, Data, and Analytics. (virtual), India, Dec. 2020. URL: <https://hal.inria.fr/hal-02900244>.
- [25] O. Beaumont, J. Langou, W. Quach and A. Shilova. ‘A Makespan Lower Bound for the Scheduling of the Tiled Cholesky Factorization based on ALAP Schedule’. In: *EuroPar 2020 Proceedings*. EuroPar 2020 - 26th International European Conference on Parallel and Distributed Computing. Proceedings of EuroPar 2020. Warsaw / Virtual, Poland: 2020.euro-par.org, Aug. 2020. URL: <https://hal.inria.fr/hal-02487920>.
- [26] R. Carratalá-Sáez, M. Faverge, G. Pichon, G. Sylvand and E. S. Quintana-Ortí. ‘Tiled Algorithms for Efficient Task-Parallel H-Matrix Solvers’. In: PDSEC 2020 - 21st IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing. News Orleans, United States, 22nd May 2020, pp. 1–10. URL: <https://hal.inria.fr/hal-02513433>.
- [27] L. Eyraud-Dubois and C. Bentes. ‘Algorithms for Preemptive Co-scheduling of Kernels on GPUs’. In: HiPC 2020 : 27th IEEE International Conference on High Performance Computing, Data, and Analytics. Pune / Virtual, India, 16th Dec. 2020. URL: <https://hal.inria.fr/hal-03148711>.
- [28] L. Eyraud-Dubois and S. Kumar. ‘Analysis of a List Scheduling Algorithm for Task Graphs on Two Types of Resources’. In: IPDPS 2020 - 34th IEEE International Parallel and Distributed Processing Symposium. New Orleans / Virtual, United States, 18th May 2020. URL: <https://hal.inria.fr/hal-02431810>.
- [29] C. Gou, A. Al Zoobi, A. Benoit, M. Faverge, L. Marchal, G. Pichon and P. Ramet. ‘Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing’. In: EuroPar 2020 - 26th International European Conference on Parallel and Distributed Computing. Warsaw / Virtual, Poland, 24th Aug. 2020, pp. 1–16. URL: <https://hal.inria.fr/hal-02973315>.
- [30] N. Grinsztajn, O. Beaumont, E. Jeannot and P. Preux. ‘Geometric Deep Reinforcement Learning for Dynamic DAG Scheduling’. In: IEEE SSCI 2020 - Symposium Series on Computational Intelligence. SSCI 2020 proceedings. Canberra / Virtual, Australia, Dec. 2020. URL: <https://hal.inria.fr/hal-03028981>.

### Conferences without proceedings

- [31] M. A. Abouabdallah, O. Coulaud, A. A. Franc and N. Peyrard. ‘Statistical learning for OTUs identification’. In: ISEC 2020 - International Statistical Ecology Conference. Sydney / Virtual, Australia, 22nd June 2020. URL: <https://hal.inrae.fr/hal-02941708>.
- [32] R. Carratalá-Sáez, M. Faverge, G. Pichon, E. S. Quintana-Ortí and G. Sylvand. ‘Exploiting Generic Tiled Algorithms Toward Scalable H-Matrices Factorizations on Top of Runtime Systems’. In: SIAM PP20 - SIAM Conference on Parallel Processing for Scientific Computing. Seattle, United States, 12th Feb. 2020. URL: <https://hal.inria.fr/hal-03149953>.
- [33] G. Pichon, M. Faverge and P. Ramet. ‘Recent Developments Around the Block Low-Rank PaStiX Solver’. In: PP 2020 - SIAM Conference on Parallel Processing for Scientific Computing. Seattle, United States: <https://www.siam.org/conferences/cm/conference/pp20>, Feb. 2020. URL: <https://hal.inria.fr/hal-03140189>.

### Reports & preprints

- [34] E. Agullo, F. Cappelletto, S. Di, L. Giraud, X. Liang and N. Schenkels. *Exploring variable accuracy storage through lossy compression techniques in numerical linear algebra: a first application to flexible GMRES*. Inria Bordeaux Sud-Ouest, 13th May 2020. URL: <https://hal.inria.fr/hal-02572910>.
- [35] E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, N. Schenkels and W. Vanroose. *A complementary note on soft errors in the Conjugate Gradient method: the persistent error case*. Inria Bordeaux Sud-Ouest, 25th Aug. 2020. URL: <https://hal.inria.fr/hal-02921669>.

- [36] E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, N. Schenkels and W. Vanroose. *On soft errors in the Conjugate Gradient method: sensitivity and robust numerical detection -revised*. Inria Bordeaux Sud-Ouest, 1st Mar. 2020, p. 43. URL: <https://hal.inria.fr/hal-02495301>.
- [37] O. Beaumont, L. Eyraud-Dubois and A. Shilova. *MadPipe: Memory Aware Dynamic Programming Algorithm for Pipelined Model Parallelism*. 26th Nov. 2020. URL: <https://hal.archives-ouvertes.fr/hal-03025305>.
- [38] O. Beaumont, L. Eyraud-Dubois and A. Shilova. *Pipelined Model Parallelism: Complexity Results and Memory Considerations*. 18th Feb. 2021. URL: <https://hal.inria.fr/hal-02968802>.
- [39] R. Carratalá-Sáez, M. Faverge, G. Pichon, G. Sylvand and E. S. Quintana-Ortí. *Tiled Algorithms for Efficient Task-Parallel H-Matrix Solvers*. Inria, Feb. 2020. URL: <https://hal.inria.fr/hal-02489269>.
- [40] L. Eyraud-Dubois and C. Bentes. *Independent tasks on 2 resources with co-scheduling effects*. 8th Jan. 2020. URL: <https://hal.inria.fr/hal-02431897>.
- [41] L. Giraud, Y.-F. Jing and Y. Xiang. *A block minimum residual norm subspace solver for sequences of multiple left and right-hand side linear systems*. Inria Bordeaux Sud-Ouest, Feb. 2021, p. 52. URL: <https://hal.inria.fr/hal-03146213>.
- [42] C. Gou, A. Al Zoobi, A. Benoit, M. Faverge, L. Marchal, G. Pichon and P. Ramet. *Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing*. Inria Rhône-Alpes, Feb. 2020, p. 21. URL: <https://hal.inria.fr/hal-02491495>.
- [43] N. Venkovic, P. Mycek, L. Giraud and O. Le Maitre. *Comparative study of harmonic and Rayleigh-Ritz procedures with applications to deflated conjugate gradients*. Cerfacs, 13th Jan. 2020. URL: <https://hal.archives-ouvertes.fr/hal-02434043>.

### 10.3 Other

#### Softwares

- [44] [SW] S. Archipoff, C. Augonnet, O. Aumage, G. Beauchamp, B. Bramas, A. Buttari, A. Cassagne, J. Clet-Ortega, T. Cojean, N. Collin, V. Danjean, A. Denis, L. Eyraud-Dubois, N. Furmento, S. Henry, A. Hugo, M. Juhour, A. Juven, M. Keryell-Even, Y. Khorsi, T. Lambert, E. Leria, B. Lizé, M. Makni, S. Nakov, R. Namyst, L. Nesi Lucas, P. Joris, D. Pasqualinotto, S. Pitoiset, Q.-D. Nguyen, C. Roelandt, C. Sakka, C. Salingue, L. Mello Schnorr, M. Sergent, A. Simonet, L. Stanisic, B. Subervie, F. Tessier, S. Thibault, B. Videau, L. Villeveygoux and P.-A. Wacrenier, *StarPU* version 1.3.3, 20th Jan. 2020. HAL: [hal-02443512](https://hal.inria.fr/hal-02443512), URL: <https://hal.inria.fr/hal-02443512>, SWHID: [swh:1:dir:b6e19d99449a78805e7a55a341fbaba2bc431973;origin=https://hal.archives-ouvertes.fr/hal-02443512;visit=swh:1:snp:c21d3dfbd96e4fb502c534e59644dba14c542100;anchor=swh:1:rev:31be198773f103324593d26369f135fbde5b97f8;path=/](https://hal.archives-ouvertes.fr/hal-02443512).