



IN PARTNERSHIP WITH:
CNRS

**Institut Polytechnique de
Bordeaux**

Université de Bordeaux

Activity Report 2017

Project-Team HIEPACS

High-End Parallel Algorithms for Challenging Numerical Simulations

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

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
**Distributed and High Performance
Computing**

Table of contents

1. Personnel	1
2. Overall Objectives	2
3. Research Program	3
3.1. Introduction	3
3.2. High-performance computing on next generation architectures	4
3.3. High performance solvers for large linear algebra problems	6
3.3.1. Parallel sparse direct solver	6
3.3.2. Hybrid direct/iterative solvers based on algebraic domain decomposition techniques	7
3.3.3. Linear Krylov solvers	7
3.3.4. Eigensolvers	8
3.4. High performance Fast Multipole Method for N-body problems	8
3.4.1. Improvement of calculation efficiency	9
3.4.2. Non uniform distributions	9
3.4.3. Fast multipole method for dislocation operators	9
3.4.4. Fast multipole method for boundary element methods	9
3.5. Load balancing algorithms for complex simulations	9
3.5.1. Dynamic load-balancing with variable number of processors	10
3.5.2. Load balancing of coupled codes	10
3.5.3. Load balancing strategies for hybrid sparse linear solvers	11
4. Application Domains	11
4.1. Material physics	11
4.2. Co-design for scalable numerical algorithms in scientific applications	12
5. Highlights of the Year	12
6. New Software and Platforms	13
6.1. Chameleon	13
6.2. Fabulous	14
6.3. HIPS	15
6.4. MAPHYS	15
6.5. MetaPart	15
6.6. MPICPL	16
6.7. OptiDis	16
6.8. PaStiX	16
6.9. ScalFMM	17
6.10. VITE	17
6.11. PlaFRIM	18
7. New Results	18
7.1. High-performance computing on next generation architectures	18
7.1.1. Bridging the gap between OpenMP and task-based runtime systems	18
7.1.2. Modeling Irregular Kernels of Task-based codes: Illustration with the Fast Multipole Method	19
7.1.3. Task-based fast multipole method for clusters of multicore processors	19
7.1.4. Achieving high-performance with a sparse direct solver on Intel KNL	19
7.2. High performance solvers for large linear algebra problems	19
7.2.1. Blocking strategy optimizations for sparse direct linear solver on heterogeneous architectures	19
7.2.2. Sparse supernodal solver using block low-rank compression	20
7.2.3. Towards a hierarchical symbol factorization for data sparse direct solvers	20
7.3. High performance fast multipole method for N-body problems	21
7.3.1. Modeling Irregular Kernels of Task-based codes	21

7.3.2.	Task-based fast multipole method for clusters of multicore processors	21
7.4.	Efficient algorithmic for load balancing and code coupling in complex simulations	21
7.5.	Application Domains	22
7.5.1.	Material physics	22
7.5.1.1.	EigenSolver	22
7.5.1.2.	Dislocation	22
7.5.2.	Co-design for scalable numerical algorithms in scientific applications	22
7.5.2.1.	High performance simulation for ITER tokamak	22
7.5.2.2.	High performance simulation for 3D frequency-domain Maxwell's equations	23
7.5.2.3.	High performance simulation for atmospheric chemistry	23
8.	Bilateral Contracts and Grants with Industry	23
9.	Partnerships and Cooperations	23
9.1.	National Initiatives	23
9.1.1.1.	SOLHAR: SOLvers for Heterogeneous Architectures over Runtime systems	23
9.1.1.2.	DEDALES: Algebraic and geometric domain decomposition for subsurface/groundwater flows	24
9.1.1.3.	TECSER: Novel high performance numerical solution techniques for RCS computations	25
9.2.	European Initiatives	25
9.2.1.1.	EoCoE	25
9.2.1.2.	HPC4E	26
9.3.	International Initiatives	27
10.	Dissemination	27
10.1.	Promoting Scientific Activities	27
10.1.1.	Member of the Conference Program Committees	27
10.1.2.	Journal	28
10.1.2.1.	Member of the Editorial Boards	28
10.1.2.2.	Reviewer - Reviewing Activities	28
10.1.3.	Scientific Expertise	28
10.1.4.	Research Administration	28
10.2.	Teaching - Supervision - Juries	28
10.2.1.	Teaching	28
10.2.2.	Supervision	29
10.2.3.	Juries	29
11.	Bibliography	30

Project-Team HIEPACS

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

Keywords:

Computer Science and Digital Science:

- A1.1.4. - High performance computing
- A1.1.5. - Exascale
- A6.2.5. - Numerical Linear Algebra
- A6.2.7. - High performance computing
- A7.1. - Algorithms
- A8.1. - Discrete mathematics, combinatorics

Other Research Topics and Application Domains:

- B3.3.1. - Earth and subsoil
- B3.4.2. - Industrial risks and waste
- B4.1. - Fossile energy production (oil, gas)
- B4.2.2. - Fusion
- B5.5. - Materials
- B9.4.1. - Computer science
- B9.4.2. - Mathematics
- B9.4.4. - Chemistry

1. Personnel

Research Scientists

- Luc Giraud [Team leader, Inria, Senior Researcher, HDR]
- Emmanuel Agullo [Inria, Researcher]
- Olivier Coulaud [Inria, Senior Researcher, HDR]
- Jean Roman [Inria, Senior Researcher, HDR]

Faculty Members

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

Post-Doctoral Fellows

- Oguz Kaya [Inria, from Nov 2017]
- Ian Masliah [Inria]
- Cristobal Samaniego Alvarado [Inria, from Dec 2017]

PhD Students

- Arnaud Durocher [CEA]
- Aurélien Falco [granted by Conseil Régional Aquitaine - Airbus Group Innovations]
- Cyril Fournier [CERFACS, until Mar 2017]
- Grégoire Pichon [Inria, granted by DGA-Inria]
- Louis Poirel [Inria, granted by ANR **DEDALES**]

Technical staff

Yuval Harness [Inria, until Jun 2017]
Matias Hastaran [Inria, granted by PRACE 4IP until Feb 2017]
Quentin Khan [Inria, until Apr 2017]
Matthieu Kuhn [Inria, granted by H2020 **HPC4E**]
Gilles Marait [Inria, granted by H2020 **EoCoE**]
Cyrille Piacibello [Inria, granted by DGA HiBox until Apr 2017]

Interns

Said Amirouche [Inria, from Mar 2017 until Aug 2017]
Raphael Boucherie [Inria, from Mar 2017 until Sep 2017]
Benjamin Dufoyer [Inria, from Apr 2017 until Sep 2017]
Romain Echaniz [Inria, from Jun 2017 until Jul 2017]
Bilal Mellouki [Inria, from Jun 2017 until Jul 2017]
Thomas Mijieux [Inria, from Mar 2017 until Aug 2017]
Chaya Senichault [Inria, Jun 2017]

Administrative Assistant

Chrystel Plumejeau [Inria]

External Collaborators

Maria Predari [Univ de Bordeaux, until Jun 2017]
Guillaume Sylvand [Airbus Group]

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 confirmed 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 currently we are able to compute on the first leading architectures at a petaflop rate. Generalist petaflop supercomputers are available and exaflop computers are foreseen in early 2020.

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 a petaflop and more 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 likely in the framework of H2020 European 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 perspectives in scalable parallel linear algebra techniques are addressed, ranging from dense direct, sparse direct, iterative and hybrid approaches for large linear systems. 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/multiphysic 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 multidisciplinary context with colleagues in other academic or industrial groups where our contribution is related to our expertises. Not only these collaborations enable our knowledges 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 Group/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 ending 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 that 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 Coulaud, Mathieu Faverge, Luc Giraud, Abdou Guermouche, Matias Hastaran, Grégoire Pichon, Pierre Ramet, Jean Roman.

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 log.html 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. Of course, this work requires the use/design of scheduling algorithms and models specifically to tackle our target problems. This has to be done in collaboration with our colleagues from the scheduling community like for example O. Beaumont (Inria **REALOPT** Project-Team). It is important to note that this topic is strongly linked to the underlying programming model. Indeed, considering multicore 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 behaviour 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 that framework, the **SOLHAR** project aims at studying and designing algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computers equipped with accelerators. Several attempts have been made to accomplish the porting of these methods on such architectures; the proposed approaches are mostly based on a simple offloading of some computational tasks (the coarsest grained ones) to the accelerators and rely on fine hand-tuning of the code and accurate performance modeling to achieve efficiency. **SOLHAR** proposes an innovative approach which relies on the efficiency and portability of runtime systems, such as the **StarPU** tool developed in the **STORM** team. Although the **SOLHAR** project will focus on heterogeneous computers equipped with GPUs due to their wide availability and affordable cost, the research accomplished on algorithms, methods and programming models will be readily applicable to other accelerator devices. Our final goal would be to have high performance solvers and tools which can efficiently run on all these types of complex architectures by exploiting all the resources of the platform (even if they are heterogeneous).

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, Mathieu Faverge, Aurélien Falco, Luc Giraud, Abdou Guermouche, Yuval Harness, Matias Hastaran, Matthieu Kuhn, Gilles Marait, Cyrille Piacibello, Grégoire Pichon, Louis Poirel, Pierre Ramet, Jean Roman, Cristobal Samaniego Alvarado, Guillaume Sylvand.

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 solver

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 grain 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 hierarchical matrices algebra.

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 **SOLHAR** 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 (**REALOPT** 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 Group Innovations.

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 libraries **HIPS** and **MaPHYs**. This activity will be developed in the context of the ANR **DEDALES** 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 breakdown 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.

- 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.

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

Participants: Emmanuel Agullo, Pierre Blanchard, Olivier Coulaud, Quentin Khan, 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 **HiEPCS**.

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, Maria Predari, Pierre Ramet, Jean Roman.

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 uses 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 **HIPS** or **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: Pierre Blanchard, Olivier Coulaud, Arnaud Durocher.

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 for scalable numerical algorithms in scientific applications

Participants: Nicolas Bouzat, Mathieu Faverge, Guillaume Latu, Michel Mehrenberger, Pierre Ramet, Jean Roman.

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 were involved in the Inria Project Lab **C2S@EXA** in collaboration with **CEA-IRFM** and are 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).

Currently, **GYSELA** is parallelized in an hybrid MPI+OpenMP way and can exploit the power of the current greatest supercomputers. To simulate faithfully the plasma physic, **GYSELA** handles a huge amount of data and today, the memory consumption is a bottleneck on very large simulations. In this context, mastering the memory consumption of the code becomes critical to consolidate its scalability and to enable the implementation of new numerical and physical features to fully benefit from the extreme scale architectures.

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.

5. Highlights of the Year

5.1. Highlights of the Year

We have presented two approaches using a Block Low-Rank (BLR) compression technique to reduce the memory footprint and/or the time-to-solution of the sparse supernodal solver **PaStiX**. Thanks to this

compression technique, we have been able to solve a 1 billion unknown system (a 3D Laplacian matrix $100 \times 100 \times 100.000$) on a single node with 3Tb of memory. The factorization time for this system was less than 6 hours using 96 cores, and the precision achieved at the first solve was 10^{-5} . With 10 additional iterative refinement steps, we reached easily 10^{-8} in double precision. The cost of one solve was limited to 280 seconds. We were able to save 9Tb over the 11Tb that would be requested by the direct solver. The last release of the software (**PaStiX** 6.0) includes these implementations and the description of the parameters are documented in [solverstack/pastix](#).

2017 has been the last year of the FASTLA associate team that has been for 6 years the framework of fruitful and intense research collaborations with Lawrence Berkeley National Laboratory and Stanford University on data sparse numerical algorithms; the joint research addressed especially fast multipole techniques and low rank calculation in sparse linear algebra. This successful collaboration has been concluded by the participation of E. Ng, head of Applied Mathematics Department at Berkeley, to the two HDR juries of A. Guermouche and P. Ramet that have been defended on the same day, November 27th.

6. New Software and Platforms

6.1. Chameleon

KEYWORDS: Runtime system - Task-based algorithm - Dense linear algebra - HPC - Task scheduling

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

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

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

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

3. Linear algebra. Because of its central position in HPC and of the well understood structure of its algorithms, dense linear algebra has often pioneered new challenges that HPC had to face. Again, dense linear algebra has been in the vanguard of the new era of petascale computing with the design of new algorithms that can efficiently run on a multicore node with GPU accelerators. These algorithms are called “communication-avoiding” since they have been redesigned to limit the amount of communication between processing units (and between the different levels of memory hierarchy). They are expressed through Direct Acyclic Graphs (DAG) of fine-grained tasks that are dynamically scheduled. Expected advances: First, we plan to investigate the impact of these principles in the case of sparse applications (whose algorithms are slightly more complicated but often rely on dense kernels). Furthermore, both in the dense and sparse cases, the scalability on thousands of nodes is still limited, new numerical approaches need to be found. We will specifically design sparse hybrid direct/iterative methods that represent a promising approach.

Overall end point. The overall goal of the MORSE associate team is to enable advanced numerical algorithms to be executed on a scalable unified runtime system for exploiting the full potential of future exascale machines. FUNCTIONAL DESCRIPTION: Chameleon is a dense linear algebra software relying on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a Runtime system. A Runtime system such as StarPU is able to manage automatically data transfers between not shared memory area (CPUs-GPUs, distributed nodes). This kind of implementation paradigm allows to design high performing linear algebra algorithms on very different type of architecture: laptop, many-core nodes, CPUs-GPUs, multiple nodes. For example, Chameleon is able to perform a Cholesky factorization (double-precision) at 80 TFlop/s on a dense matrix of order 400 000 (e.i. 4 min).

RELEASE FUNCTIONAL DESCRIPTION: Chameleon includes the following features:

- BLAS 3, LAPACK one-sided and LAPACK norms tile algorithms - Support QUARK and StarPU runtime systems - 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)
- Participants: Cédric Castagnede, Samuel Thibault, Emmanuel Agullo, Florent Pruvost and Mathieu Faverge
- Partners: Innovative Computing Laboratory (ICL) - King Abdullha University of Science and Technology - University of Colorado Denver
- Contact: Emmanuel Agullo
- URL: <https://project.inria.fr/chameleon/>

6.2. Fabulous

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.

- Participants: Emmanuel Agullo, Luc Giraud and Cyrille Piacibello
- Contact: Luc Giraud
- Publication: [Block GMRES method with inexact breakdowns and deflated restarting](#)
- URL: <https://gitlab.inria.fr/solverstack/fabulous/>

6.3. HIPS

Hierarchical Iterative Parallel Solver

KEYWORDS: Simulation - HPC - Parallel calculation - Hybrid direct iterative method

SCIENTIFIC DESCRIPTION: The key point of the methods implemented in HIPS is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the subgraphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. HIPS exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file , it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, HIPS is a software library that provides several methods to build an efficient preconditioner in almost all situations.

FUNCTIONAL DESCRIPTION: HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

- Participants: Jérémie Gaidamour, Pascal Hénon and Yousef Saad
- Contact: Pierre Ramet
- URL: <http://hips.gforge.inria.fr/>

6.4. MAPHYS

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.

- Participants: Emmanuel Agullo, Luc Giraud, Matthieu Kuhn, Gilles Marait and Louis Poirel
- Contact: Emmanuel Agullo
- Publications: [Hierarchical hybrid sparse linear solver for multicore platforms](#) [Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems](#)
- URL: <https://gitlab.inria.fr/solverstack/maphys>

6.5. MetaPart

KEYWORDS: High performance computing - HPC - Parallel computing - Graph algorithmics - Graph - Hypergraph

FUNCTIONAL DESCRIPTION: MetaPart is a framework for graph or hypergraph manipulation that addresses different problems, like partitioning, repartitioning, or co-partitioning, ... MetaPart is made up of several projects, such as StarPart, LibGraph or CoPart. StarPart is the core of the MetaPart framework. It offers a wide variety of graph partitioning methods (Metis, Scotch, Zoltan, Patoh, ParMetis, Kahip, ...), which makes it easy to compare these different methods and to better adjust the parameters of these methods. It is built upon the LibGraph library, that provides basic graph & hypergraph routines. The Copart project is a library used on top of StarPart, that provides co-partitioning algorithms for the load-blancing of parallel coupled simulations.

- Participant: Aurélien Esnard
- Contact: Aurélien Esnard
- URL: <https://gitlab.inria.fr/metapart>

6.6. MPICPL

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.

- Participant: Aurélien Esnard
- Contact: Aurélien Esnard
- URL: <https://gitlab.inria.fr/esnard/mpicpl>

6.7. 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-2) 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 H/14iePACS project. The code is written in C++ language and using the last features of C++11. 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..

- Participant: Olivier Coulaud
- Contact: Olivier Coulaud
- URL: <http://optidis.gforge.inria.fr/>

6.8. PaStiX

Parallel Sparse matrix package

KEYWORDS: Sparse Matrices - Factorisation - High-performance calculation - Linear algebra - Linear Systems Solver

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.

- Participants: Grégoire Pichon, Mathieu Faverge and Pierre Ramet
- Partner: Université Bordeaux 1
- Contact: Pierre Ramet
- URL: <http://pastix.gforge.inria.fr/>

6.9. ScalFMM

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

- Participants: Bramas Bérenger and Olivier Coulaud
- Contact: Olivier Coulaud
- URL: <https://gitlab.inria.fr/solverstack/ScalFMM>

6.10. VITE

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.

- Participant: Mathieu Faverge
- Contact: Mathieu Faverge
- URL: <http://vite.gforge.inria.fr/>

6.11. PlaFRIM

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.

- Contact: Olivier Coulaud
- URL: <https://www.plafrim.fr/en/home/>

7. New Results

7.1. High-performance computing on next generation architectures

7.1.1. Bridging the gap between OpenMP and task-based runtime systems

With the advent of complex modern architectures, the low-level paradigms long considered sufficient to build High Performance Computing (HPC) numerical codes have met their limits. Achieving efficiency, ensuring portability, while preserving programming tractability on such hardware prompted the HPC community to design new, higher level paradigms while relying on runtime systems to maintain performance. However, the common weakness of these projects is to deeply tie applications to specific expert-only runtime system APIs. The OpenMP specification, which aims at providing common parallel programming means for shared-memory platforms, appears as a good candidate to address this issue thanks to the latest task-based constructs introduced in its revision 4.0. The goal of this paper is to assess the effectiveness and limits of this support for designing a high-performance numerical library, ScalFMM, implementing the fast multipole method (FMM) that we have deeply redesigned with respect to the most advanced features provided by OpenMP 4. We show that OpenMP 4 allows for significant performance improvements over previous OpenMP revisions on recent multicore processors and that extensions to the 4.0 standard allow for strongly improving the performance, bridging the gap with the very high performance that was so far reserved to expert-only runtime system APIs. More details on this work can be found in [17].

7.1.2. Modeling Irregular Kernels of Task-based codes: Illustration with the Fast Multipole Method

The significant increase of the hardware complexity that occurred in the last few years led the high performance community to design many scientific libraries according to a task-based parallelization. The modeling of the performance of the individual tasks (or kernels) they are composed of is crucial for facing multiple challenges as diverse as performing accurate performance predictions, designing robust scheduling algorithms, tuning the applications, etc. Fine-grain modeling such as emulation and cycle-accurate simulation may lead to very accurate results. However, not only their high cost may be prohibitive but they furthermore require a high fidelity modeling of the processor, which makes them hard to deploy in practice. In this paper, we propose an alternative coarse-grain, empirical methodology oblivious to both the target code and the hardware architecture, which leads to robust and accurate timing predictions. We illustrate our approach with a task-based Fast Multipole Method (FMM) algorithm, whose kernels are highly irregular, implemented in the **ScalFMM** library on top of the **StarPU** task-based runtime system and the simgrid simulator. More details on this work can be found in [41].

7.1.3. Task-based fast multipole method for clusters of multicore processors

Most high-performance, scientific libraries have adopted hybrid parallelization schemes - such as the popular MPI+OpenMP hybridization - to benefit from the capacities of modern distributed-memory machines. While these approaches have shown to achieve high performance, they require a lot of effort to design and maintain sophisticated synchronization/communication strategies. On the other hand, task-based programming paradigms aim at delegating this burden to a runtime system for maximizing productivity. In this article, we assess the potential of task-based fast multipole methods (FMM) on clusters of multicore processors. We propose both a hybrid MPI+task FMM parallelization and a pure task-based parallelization where the MPI communications are implicitly handled by the runtime system. The latter approach yields a very compact code following a sequential task-based programming model. We show that task-based approaches can compete with a hybrid MPI+OpenMP highly optimized code and that furthermore the compact task-based scheme fully matches the performance of the sophisticated, hybrid MPI+task version, ensuring performance while maximizing productivity. We illustrate our discussion with the **ScalFMM** FMM library and the **StarPU** runtime system. More details on this work can be found in [40].

7.1.4. Achieving high-performance with a sparse direct solver on Intel KNL

The need for energy-efficient high-end systems has led hardware vendors to design new types of chips for general purpose computing. However, designing or porting a code tailored for these new types of processing units is often considered as a major hurdle for their broad adoption. In this paper, we consider a modern Intel Xeon Phi processor, namely the Intel Knights Landing (KNL) and a numerical code initially designed for a classical multi-core system. More precisely, we consider the **qr_mumps** scientific library implementing a sparse direct method on top of the StarPU runtime system. We show that with a portable programming model (task-based programming), a good software support (a robust runtime system coupled with an efficient scheduler) and some well defined hardware and software settings, we are able to transparently run the exact same numerical code. This code not only achieves very high performance (up to 1 TFlop/s) on the KNL but also significantly outperforms a modern Intel Xeon multi-core processor both in terms of time to solution and energy efficiency up to a factor of 2.0. More details on this work can be found in [42].

7.2. High performance solvers for large linear algebra problems

7.2.1. Blocking strategy optimizations for sparse direct linear solver on heterogeneous architectures

The preprocessing steps of sparse direct solvers, ordering and block-symbolic factorization, are two major steps that lead to a reduced amount of computation and memory and to a better task granularity to reach a good level of performance when using BLAS kernels. With the advent of GPUs, the granularity of the block computation became more important than ever. In this paper, we present a reordering strategy that increases

this block granularity. This strategy relies on the block-symbolic factorization to refine the ordering produced by tools such as METIS or **Scotch**, but it does not impact the number of operations required to solve the problem. We integrate this algorithm in the **PaStiX** solver and show an important reduction of the number of off-diagonal blocks on a large spectrum of matrices. This improvement leads to an increase in efficiency of up to 20% on GPUs.

These contributions have been published in SIAM Journal on Matrix Analysis and Applications [22].

7.2.2. *Sparse supernodal solver using block low-rank compression*

In the context of **FASTLA** associate team, during the last 4 years, we are collaborating with Eric Darve, professor in the Institute for Computational and Mathematical Engineering and the Mechanical Engineering Department at Stanford, on the design of a new efficient sparse direct solvers. We have been working on applying fast direct solvers for dense matrices to the solution of sparse direct systems. We observed that the extend-add operation (during the sparse factorization) is the most time-consuming step. We have therefore developed a series of algorithms to reduce this computational cost.

We presented two approaches using a Block Low-Rank (BLR) compression technique to reduce the memory footprint and/or the time-to-solution of the sparse supernodal solver **PaStiX**. This flat, non-hierarchical, compression method allows to take advantage of the low-rank property of the blocks appearing during the factorization of sparse linear systems, which come from the discretization of partial differential equations. The first approach, called *Minimal Memory*, illustrates the maximum memory gain that can be obtained with the BLR compression method, while the second approach, called *Just-In-Time*, mainly focuses on reducing the computational complexity and thus the time-to-solution. Singular Value Decomposition (SVD) and Rank-Revealing QR (RRQR), as compression kernels, are both compared in terms of factorization time, memory consumption, as well as numerical properties. Experiments on a single node with 24 threads and 128 GB of memory are performed to evaluate the potential of both strategies. On a set of matrices from real-life problems, we demonstrate a memory footprint reduction of up to 4 times using the *Minimal Memory* strategy and a computational time speedup of up to 3.5 times with the *Just-In-Time* strategy. Then, we study the impact of configuration parameters of the BLR solver that allowed us to solve a 3D laplacian of 36 million unknowns a single node, while the full-rank solver stopped at 8 million due to memory limitation.

These contributions have been presented at the PDSEC workshop of IPDPS'17 conference [30] and an extended version has been submitted in Journal of Computational Science [48].

7.2.3. *Towards a hierarchical symbol factorization for data sparse direct solvers*

Hierarchical algorithms based on low-rank compression techniques have led to fully re-design the methods of solving dense linear systems at the dawn of the twenty-first century, significantly reducing the computational costs. However, their application to the treatment of sparse linear systems remains today a major challenge to which both the community of hierarchical matrices and that of the sparse matrices are tackling. For this purpose, a first class of approach has been developed by the community of hierarchical matrices to exploit the sparse matrix structure. If the strong point of these methods is that the resulting algorithm remains hierarchical, these do not manage to exploit some zeros as naturally do sparse solvers. In contrast, the fact that a sparse factorization can be seen as a sequence of smaller, dense operations, the community of hollow matrices has explored this property to introduce hierarchical techniques within these elementary operations. However, the resulting algorithm loses the fundamental property of hierarchical algorithms, since the compression hierarchy is only local. As part of this doctorate, we introduce a new algorithm, performing a sparse hierarchical symbolic factorization that allows to exploit precisely the sparse structure of the matrix and its factors while preserving a global hierarchical structure to ensure effective compression. We have shown experimentally that this new approach allows us to obtain at the same time a reduced number of operations (because of its hierarchical character) and a number of non-zero elements as small as a hollow method (through the use of a symbolic factorization).

This work is developed in the A. Falco PhD thesis, it led to a publication in a national conference [31] and will give rise to a submission in an international journal in 2018

7.3. High performance fast multipole method for N-body problems

7.3.1. Modeling Irregular Kernels of Task-based codes

The significant increase of the hardware complexity that occurred in the last few years led the high performance community to design many scientific libraries according to a task-based parallelization. The modeling of the performance of the individual tasks (or kernels) they are composed of is crucial for facing multiple challenges as diverse as performing accurate performance predictions, designing robust scheduling algorithms, tuning the applications, etc. Fine-grain modeling such as emulation and cycle-accurate simulation may lead to very accurate results. However, not only their high cost may be prohibitive but they furthermore require a high fidelity modeling of the processor, which makes them hard to deploy in practice. In this paper, we propose an alternative coarse-grain, empirical methodology oblivious to both the target code and the hardware architecture, which leads to robust and accurate timing predictions. We illustrate our approach with a task-based Fast Multipole Method (FMM) algorithm, whose kernels are highly irregular, implemented in the **ScalFMM** library on top of the starpu task-based runtime system and the simgrid simulator. More details on this work can be found in [41].

7.3.2. Task-based fast multipole method for clusters of multicore processors

Most high-performance, scientific libraries have adopted hybrid parallelization schemes - such as the popular MPI+OpenMP hybridization - to benefit from the capacities of modern distributed-memory machines. While these approaches have shown to achieve high performance, they require a lot of effort to design and maintain sophisticated synchronization/communication strategies. On the other hand, task-based programming paradigms aim at delegating this burden to a runtime system for maximizing productivity. In this article, we assess the potential of task-based fast multipole methods (FMM) on clusters of multicore processors. We propose both a hybrid MPI+task FMM parallelization and a pure task-based parallelization where the MPI communications are implicitly handled by the runtime system. The latter approach yields a very compact code following a sequential task-based programming model. We show that task-based approaches can compete with a hybrid MPI+OpenMP highly optimized code and that furthermore the compact task-based scheme fully matches the performance of the sophisticated, hybrid MPI+task version, ensuring performance while maximizing productivity. We illustrate our discussion with the ScalFMM FMM library and the StarPU runtime system. More details on this work can be found in [40].

7.4. Efficient algorithmic for load balancing and code coupling in complex simulations

7.4.1. Comparison of initial partitioning methods for multilevel direct k-way graph partitioning with fixed vertices

In scientific computing, load balancing is a crucial step conditioning the performance of large-scale applications. In this case, an efficient decomposition of the workload to a number of processors is highly necessary. A common approach to solve this problem is to use graph representation and perform a graph partitioning in k parts using the multilevel framework and the recursive bisection (RB) paradigm. However, in graph instances where fixed vertices are used to model additional constraints, RB often produces partitions of poor quality. In this paper, we investigate the difficulties of RB to handle fixed vertices and we compare its results with two different alternatives. The first one, called KGGGP is a direct k -way greedy graph growing partitioning that properly handles fixed vertices while the second one, introduced in kPaToH, uses RB and a post-processing technique to correct the obtained partition. Finally, experimental results on graphs that represent real-life numerical simulations show that both alternative methods provide improved partitions compared to RB. More details on this work can be found in [23].

7.5. Application Domains

7.5.1. Material physics

7.5.1.1. EigenSolver

The adaptive vibrational configuration interaction algorithm has been introduced as a new method to efficiently reduce the dimension of the set of basis functions used in a vibrational configuration interaction process. It is based on the construction of nested bases for the discretization of the Hamiltonian operator according to a theoretical criterion that ensures the convergence of the method. In the present work, the Hamiltonian is written as a sum of products of operators. The purpose of this paper is to study the properties and outline the performance details of the main steps of the algorithm. New parameters have been incorporated to increase flexibility, and their influence has been thoroughly investigated. The robustness and reliability of the method are demonstrated for the computation of the vibrational spectrum up to 3000 cm⁻¹ of a widely studied 6-atom molecule (acetonitrile). Our results are compared to the most accurate up to date computation; we also give a new reference calculation for future work on this system. The algorithm has also been applied to a more challenging 7-atom molecule (ethylene oxide). The computed spectrum up to 3200 cm⁻¹ is the most accurate computation that exists today on such systems. More details on this work can be found in [43], [21].

7.5.1.2. Dislocation

We have focused on the improvements of the parallel collision detection and of the accuracy in the force field computation in the **OPTIDIS** code.

- a new collision detection algorithm to reliably handle junction formation for Dislocation Dynamics using hybrid OpenMP + MPI parallelism has been developed. The enhanced precision and reliability of this new algorithm allows the use of larger time-steps for faster simulations. Hierarchical methods for collision detection, as well as hybrid parallelism are also used to improve performance;
- we observed that the force field computation depends on how the traversal of the segments list or boxes in the octree was done. New accurate formulas to remove this issue have been developed and we are implementing them in the code. They will be used in the Fast Multipole Method that we have developed previously.

Finally, a new distributed data structure has been developed to enhance the reliability and modularity of **OPTIDIS**. The new data structure provides an interface to modify safely and reliably the distributed dislocation mesh in order to enforce data consistency across all computation nodes. This interface also improves code modularity allowing the study of data layout performance without modifying the algorithms.

7.5.2. Co-design for scalable numerical algorithms in scientific applications

7.5.2.1. High performance simulation for ITER tokamak

Concerning the **GYSELA** global non-linear electrostatic code, the efforts during the period have concentrated on the design of a more efficient parallel gyro-average operator for the deployment of very large (future) **GYSELA** runs. The main unknown of the computation is a distribution function that represents either the density of the guiding centers, either the density of the particles in a tokamak. The switch between these two representations is done thanks to the gyro-average operator. In the previous version of **GYSELA**, the computation of this operator was achieved thanks to a Padé approximation. In order to improve the precision of the gyro-averaging, a new parallel version based on an Hermite interpolation has been done (in collaboration with the Inria **TONUS** project-team and IPP Garching). The integration of this new implementation of the gyro-average operator has been done in **GYSELA** and the parallel benchmarks have been successful. This work is carried on in the framework of the PhD of Nicolas Bouzat (funded by IPL **C2S@EXA**) co-advised with Michel Mehrenberger from **TONUS** project-team and in collaboration with Guillaume Latu from **CEA-IRFM**. The scientific objectives of this work is first to consolidate the parallel version of the gyro-average operator, in particular by designing a scalable MPI+OpenMP parallel version and using a new communication scheme, and second to design new numerical methods for the gyro-average, source and collision operators to deal with new physics in **GYSELA**. The objective is to tackle kinetic electron configurations for more realistic complex large simulations.

In the context of the EoCoE project, we have collaborations with **CEA-IRFM**. First, with G. Latu, we have investigated the potential of using the last release of the **PaStiX** solver (version 6.0) on Intel KNL architecture, and more especially on the MARCONI machine (one of the PRACE supercomputers at Cineca, Italia). The results obtained on this architecture are really promising since we are able to reach more than 1 Tflops using a single node. Secondly, we also have a collaboration with P. Tamain and G. Giorgani on the TOKAM3X code to analyze the performance of using **PaStiX** as a preconditioner. Since a distributed memory is required during the simulation, the previous release of **PaStiX** is then used. Some difficulties regarding the Fortran wrapper and some memory issues should be fixed when we will have reimplemented the MPI interface in the current release.

7.5.2.2. High performance simulation for 3D frequency-domain Maxwell's equations

We also recently developed a collaboration with **NACHOS** on the HORSE (High Order solver for Radar cross Section Evaluation) simulation code. The aim was to integrate the **PaStiX** solver, with low-rank compression technique, in a domain decomposition framework to solve 3D frequency-domain Maxwell's equations. The results are promising since we were able to reduce by two the factorization and the solve time for each subdomain. And we were also able to reduce by two the memory requirements thanks to our compression techniques. This would allow us to consider larger subdomains with the same memory constraints that currently limit the simulations.

7.5.2.3. High performance simulation for atmospheric chemistry

We worked on the development and tests of the Adaptive Semi-Implicit Scheme (ASIS) solver for the simulation of atmospheric chemistry. To solve the Ordinary Differential Equation systems associated with the time evolution of the species concentrations, ASIS adopts a one step linearized implicit scheme with specific treatments of the Jacobian of the chemical fluxes. It conserves mass and has a time stepping module to control the accuracy of the numerical solution. In 5 idealized box model simulations ASIS gives results similar to the higher order implicit schemes derived from the Rosenbrock's and Gear's methods and requires less computation and run time at the moderate precision required for atmospheric applications. When implemented in the MOCAGE CTM and the LMD Mars GCM the ASIS solver performs well and reveals weaknesses and limitations of the original semi-implicit solvers used by these two models. ASIS can be easily adapted to various chemical schemes and further developments are foreseen to increase its computational efficiency, and to include the computation of the 10 concentrations of the species in aqueous phase in addition to gas phase chemistry.

More details on this work can be found in [19].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Grants with Industry

Airbus Group Innovations research and development contract:

- Design and implementation of linear algebra kernel for FEM-BEM coupling (A. Falco (PhD); Emmanuel Agullo, Luc Giraud, Guillaume Sylvand).
- Design and implementation of FMM and block Krylov solver for BEM applications. The HiBOX project is led by the SME IMACS and funded by the DGA Rapid programme (C. Piacibello (Engineer), Olivier Coulaud, Luc Giraud).

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR

9.1.1.1. SOLHAR: SOLvers for Heterogeneous Architectures over Runtime systems

Participants: Emmanuel Agullo, Mathieu Faverge, Abdou Guermouche, Pierre Ramet, Jean Roman, Guillaume Sylvand.

Grant: ANR-MONU

Dates: 2013 – 2017

Partners: Inria (**REALOPT**, **STORM** Bordeaux Sud-Ouest et **ROMA** Rhone-Alpes), IRIT/INPT, CEA-CESTA et Airbus Group Innovations.

Overview:

During the last five years, the interest of the scientific computing community towards accelerating devices has been rapidly growing. The reason for this interest lies in the massive computational power delivered by these devices. Several software libraries for dense linear algebra have been produced; the related algorithms are extremely rich in computation and exhibit a very regular pattern of access to data which makes them extremely good candidates for GPU execution. On the contrary, methods for the direct solution of sparse linear systems have irregular, indirect memory access patterns that adversely interact with typical GPU throughput optimizations.

This project aims at studying and designing algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computer equipped with accelerators. The ultimate aim of this project is to achieve the implementation of a software package providing a solver based on direct methods for sparse linear systems of equations. To date, the approaches proposed to achieve this objective are mostly based on a simple offloading of some computational tasks to the accelerators and rely on fine hand-tuning of the code and accurate performance modeling to achieve efficiency. This project proposes an innovative approach which relies on the efficiency and portability of runtime systems. The development of a production-quality, sparse direct solver requires a considerable research effort along three distinct axes:

- linear algebra: algorithms have to be adapted or redesigned in order to exhibit properties that make their implementation and execution on heterogeneous computing platforms efficient and reliable. This may require the development of novel methods for defining data access patterns that are more suitable for the dynamic scheduling of computational tasks on processing units with considerably different capabilities as well as techniques for guaranteeing a reliable and robust behavior and accurate solutions. In addition, it will be necessary to develop novel and efficient accelerator implementations of the specific dense linear algebra kernels that are used within sparse, direct solvers;
- runtime systems: tools such as the **StarPU** runtime system proved to be extremely efficient and robust for the implementation of dense linear algebra algorithms. Sparse linear algebra algorithms, however, are commonly characterized by complicated data access patterns, computational tasks with extremely variable granularity and complex dependencies. Therefore, a substantial research effort is necessary to design and implement features as well as interfaces to comply with the needs formalized by the research activity on direct methods;
- scheduling: executing a heterogeneous workload with complex dependencies on a heterogeneous architecture is a very challenging problem that demands the development of effective scheduling algorithms. These will be confronted with possibly limited views of dependencies among tasks and multiple, and potentially conflicting objectives, such as minimizing the makespan, maximizing the locality of data or, where it applies, minimizing the memory consumption.

Given the wide availability of computing platforms equipped with accelerators and the numerical robustness of direct solution methods for sparse linear systems, it is reasonable to expect that the outcome of this project will have a considerable impact on both academic and industrial scientific computing. This project will moreover provide a substantial contribution to the computational science and high-performance computing communities, as it will deliver an unprecedented example of a complex numerical code whose parallelization completely relies on runtime scheduling systems and which is, therefore, extremely portable, maintainable and evolvable towards future computing architectures.

9.1.1.2. *DEDALES: Algebraic and geometric domain decomposition for subsurface/groundwater flows*

Participants: Emmanuel Agullo, Mathieu Faverge, Luc Giraud, Louis Poiriel.

Grant: ANR-14-CE23-0005

Dates: 2014 – 2018

Partners: Inria EPI POMDAPI (leader); Université Paris 13 - Laboratoire Analyse, Géométrie et Applications; Maison de la Simulation; Andra.

Overview: Project **DEDALES** aims at developing high performance software for the simulation of two phase flow in porous media. The project will specifically target parallel computers where each node is itself composed of a large number of processing cores, such as are found in new generation many-core architectures. The project will be driven by an application to radioactive waste deep geological disposal. Its main feature is phenomenological complexity: water-gas flow in highly heterogeneous medium, with widely varying space and time scales. The assessment of large scale model is of major importance and issue for this application, and realistic geological models have several million grid cells. Few, if at all, software codes provide the necessary physical features with massively parallel simulation capabilities. The aim of the **DEDALES** project is to study, and experiment with, new approaches to develop effective simulation tools with the capability to take advantage of modern computer architectures and their hierarchical structure. To achieve this goal, we will explore two complementary software approaches that both match the hierarchical hardware architecture: on the one hand, we will integrate a hybrid parallel linear solver into an existing flow and transport code, and on the other hand, we will explore a two level approach with the outer level using (space time) domain decomposition, parallelized with a distributed memory approach, and the inner level as a subdomain solver that will exploit thread level parallelism. Linear solvers have always been, and will continue to be, at the center of simulation codes. However, parallelizing implicit methods on unstructured meshes, such as are required to accurately represent the fine geological details of the heterogeneous media considered, is notoriously difficult. It has also been suggested that time level parallelism could be a useful avenue to provide an extra degree of parallelism, so as to exploit the very large number of computing elements that will be part of these next generation computers. Project **DEDALES** will show that space-time DD methods can provide this extra level, and can usefully be combined with parallel linear solvers at the subdomain level. For all tasks, realistic test cases will be used to show the validity and the parallel scalability of the chosen approach. The most demanding models will be at the frontier of what is currently feasible for the size of models.

9.1.1.3. *TECSER: Novel high performance numerical solution techniques for RCS computations*

Participants: Emmanuel Agullo, Luc Giraud, Matthieu Kuhn.

Grant: ANR-14-ASTRID

Dates: 2014 – 2017

Partners: Inria EPI NACHOS (leader), Corida, HiePACS; Airbus Group Innovations, Nucléotides.

Overview: the objective of the **TECSER** projet is to develop an innovative high performance numerical methodology for frequency-domain electromagnetics with applications to RCS (Radar Cross Section) calculation of complicated structures. This numerical methodology combines a high order hybridized DG method for the discretization of the frequency-domain Maxwell in heterogeneous media with a BEM (Boundary Element Method) discretization of an integral representation of Maxwell's equations in order to obtain the most accurate treatment of boundary truncation in the case of theoretically unbounded propagation domain. Beside, scalable hybrid iterative/direct domain decomposition based algorithms are used for the solution of the resulting algebraic system of equations.

9.2. European Initiatives

9.2.1. FP7 & H2020 Projects

9.2.1.1. *EoCoE*

Title: Energy oriented Centre of Excellence for computer applications

Programm: H2020

Duration: October 2015 - October 2018

Coordinator: CEA

Partners:

Barcelona Supercomputing Center - Centro Nacional de Supercomputacion (Spain)
 Commissariat A L Energie Atomique et Aux Energies Alternatives (France)
 Centre Europeen de Recherche et de Formation Avancee 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)
 Universite Libre de Bruxelles (Belgium)
 Universita Degli Studi di Trento (Italy)

Inria contact: Michel Kern

The aim of the present proposal is to establish an Energy Oriented Centre of Excellence for computing applications, (EoCoE). EoCoE (pronounce "Echo") will use the prodigious potential offered by the ever-growing computing infrastructure to foster and accelerate the European transition to a reliable and low carbon energy supply. To achieve this goal, we believe that the present revolution in hardware technology calls for a similar paradigm change in the way application codes are designed. EoCoE will assist the energy transition via targeted support to four renewable energy pillars: Meteo, Materials, Water and Fusion, each with a heavy reliance on numerical modelling. These four pillars will be anchored within a strong transversal multidisciplinary basis providing high-end expertise in applied mathematics and HPC. EoCoE is structured around a central Franco-German hub coordinating a pan-European network, gathering a total of 8 countries and 23 teams. Its partners are strongly engaged in both the HPC and energy fields; a prerequisite for the long-term sustainability of EoCoE and also ensuring that it is deeply integrated in the overall European strategy for HPC. The primary goal of EoCoE is to create a new, long lasting and sustainable community around computational energy science. At the same time, EoCoE is committed to deliver high-impact results within the first three years. It will resolve current bottlenecks in application codes, leading to new modelling capabilities and scientific advances among the four user communities; it will develop cutting-edge mathematical and numerical methods, and tools to foster the usage of Exascale computing. Dedicated services for laboratories and industries will be established to leverage this expertise and to foster an ecosystem around HPC for energy. EoCoE will give birth to new collaborations and working methods and will encourage widely spread best practices.

9.2.1.2. HPC4E

Title: HPC for Energy

Programm: H2020

Duration: December 2015 - November 2017

Coordinator: Barcelona Supercomputing Center

Partners:

Centro de Investigaciones Energeticas, Medioambientales Y Tecnologicas-Ciemat (Spain)
 Iberdrola Renovables Energia (Spain)
 Repsol (Spain)
 Total S.A. (France)
 Lancaster University (United Kingdom)

Inria contact: Stéphane Lanteri

This project aims to apply the new exascale HPC techniques to energy industry simulations, customizing them, and going beyond the state-of-the-art in the required HPC exascale simulations for different energy sources: wind energy production and design, efficient combustion systems for biomass-derived fuels (biogas), and exploration geophysics for hydrocarbon reservoirs. For wind energy industry HPC is a must. The competitiveness of wind farms can be guaranteed only with accurate wind resource assessment, farm design and short-term micro-scale wind simulations to forecast the daily power production. The use of CFD LES models to analyse atmospheric flow in a wind farm capturing turbine wakes and array effects requires exascale HPC systems. Biogas, i.e. biomass-derived fuels by anaerobic digestion of organic wastes, is attractive because of its wide availability, renewability and reduction of CO₂ emissions, contribution to diversification of energy supply, rural development, and it does not compete with feed and food feedstock. However, its use in practical systems is still limited since the complex fuel composition might lead to unpredictable combustion performance and instabilities in industrial combustors. The next generation of exascale HPC systems will be able to run combustion simulations in parameter regimes relevant to industrial applications using alternative fuels, which is required to design efficient furnaces, engines, clean burning vehicles and power plants. One of the main HPC consumers is the oil & gas (O&G) industry. The computational requirements arising from full wave-form modelling and inversion of seismic and electromagnetic data is ensuring that the O&G industry will be an early adopter of exascale computing technologies. By taking into account the complete physics of waves in the subsurface, imaging tools are able to reveal information about the Earth's interior with unprecedented quality.

9.3. International Initiatives

9.3.1. Inria Associate Teams Not Involved in an Inria International Labs

9.3.1.1. FASTLA

Title: Fast and Scalable Hierarchical Algorithms for Computational Linear Algebra

International Partner (Institution - Laboratory - Researcher):

Stanford (United States) - Institute for Computational and Mathematical Engineering)

ICME - Eric Darve

Start year: 2012

See also: http://people.bordeaux.inria.fr/coulaud/projets/FastLA_Website/

In this project, we propose to study fast and scalable hierarchical numerical kernels and their implementations on heterogeneous manycore platforms for two major computational kernels in intensive challenging applications. Namely, fast multipole methods (FMM) and sparse linear solvers that appear in many intensive numerical simulations in computational sciences. For the solution of large linear systems, the ultimate goal is to design parallel scalable methods that rely on efficient sparse and dense direct methods using H-matrix arithmetic. Finally, the innovative algorithmic design will be essentially focused on heterogeneous manycore platforms by using task based runtime systems. The partners, Inria HiePACS, Lawrence Berkeley Nat. Lab and Stanford University, have strong, complementary and recognized experiences and backgrounds in these fields

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Member of the Conference Program Committees

SC'17 (E. Agullo, A. Guermouche), ICCP'17 (A. Guermouche), HiPC'17 (A. Guermouche), IEEE PDP'17 (J. Roman), PDSEC'17 (O. Coulaud, L. Giraud).

10.1.2. Journal

10.1.2.1. Member of the Editorial Boards

L. Giraud is member of the editorial board of the SIAM Journal on Matrix Analysis and Applications (**SIMAX**).

10.1.2.2. Reviewer - Reviewing Activities

ACM Trans. on Mathematical Software, Advances in Computational Mathematics, Computing and Fluid, IEEE Trans. on Parallel and Distributed Systems, International Journal of High Performance Computing Applications, Journal of Computational Physics, Journal of Scientific Computing, Linear algebra with applications, Mathematics and Computers in Simulation, Parallel Computing, SIAM J. Matrix Analysis and Applications, SIAM J. Scientific Computing, Theory of Computing Systems.

10.1.3. Scientific Expertise

- Emmanuel Agullo: US Department of Energy's (DOE's) Exascale Computing Project (ECP) reviewing for research and development in Software Technology, specifically in the area of Math Libraries.
- Luc Giraud is member of the board on Modelisation, Simulation and data analysis of the **Competitiveness Cluster for Aeronautics, Space and Embedded Systems**.
- Pierre Ramet is "Scientific Expert" at the CEA-DAM CESTA since Oct. 2015.
- Jean Roman is member of the "Scientific Board" of the CEA-DAM. As representative of Inria, he is member of the board of ETP4HPC (European Technology Platform for High Performance Computing), of the French Information Group for PRACE, of the Technical Group of GENCI and of the Scientific Advisory Board of the Maison de la Simulation.

10.1.4. Research Administration

- Emmanuel Agullo and Luc Giraud are the scientific correspondents of the European and International partnership for Inria Bordeaux Sud-Ouest.
- Olivier Coulaud is the coordinator of the **PLAFRIM** platform for Inria Bordeaux Sud-Ouest.
- Jean Roman is a member of the Direction for Science at Inria : he is the Deputy Scientific Director of the Inria research domain entitled *Applied Mathematics, Computation and Simulation* and is in charge at the national level of the Inria activities concerning High Performance Computing.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

We indicate below the number of hours spent in teaching activities on a yearly basis for each scientific staff member involved.

Undergraduate level/Licence

- A. Esnard: System programming 36h, Computer architecture 40h, Network 23h at Bordeaux University.
- M. Faverge: Programming environment 26h, Numerical algorithmic 30h, C projects 24h at Bordeaux INP (ENSEIRB-MatMeca).
- P. Ramet: System programming 24h, Databases 32h, Object programming 48h, Distributed programming 32h, Cryptography 32h 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 24h, Hierarchical methods 8h 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).
- M. Faverge: System programming 74h, Load balancing and scheduling 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).

- L. Giraud: Introduction to intensive computing and related programming tools 20h, INSA Toulouse; Introduction to high performance computing and applications 20h, ISAE; On mathematical tools for numerical simulations 10h, ENSEEIHT Toulouse; Parallel sparse linear algebra 11h at Bordeaux INP (ENSEIRB-MatMeca).
- A. Guermouche: Network management 92h, Network security 64h, Operating system 24h at Bordeaux University.
- P. Ramet: Load balancing and scheduling 13h, Numerical algorithms 30h at Bordeaux INP (ENSEIRB-MatMeca). He also gives classes on Cryptography 30h, Ho Chi Minh City in Vietnam.
- J. Roman: Parallel sparse linear algebra 10h, Algorithmic and parallel algorithms 22h at Bordeaux INP (ENSEIRB-MatMeca).

He is also in charge of the last year “Parallel and Distributed Computing” option at ENSEIRB-MatMeca which is specialized in HPC (methodologies and applications). This is a common training curriculum between Computer Science and MatMeca departments at Bordeaux INP and with Bordeaux University in the context of Computer Science Research Master. It provides a lot of well-trained internship students for Inria projects working on HPC and simulation.

Summer School: on an annual basis, we run a three day advanced training (lecture and hands on) on parallel linear algebra in the framework of the European PRACE PATC (PRACE Advanced Training Centres) initiative. This training has been organized in many places in France and Europe.

10.2.2. Supervision

HdR : Abdou Guermouche, Towards efficient sparse direct solvers for modern high-performance architectures, Université de Bordeaux, 27 novembre 2017.

HdR : Pierre Ramet, Hierarchical matrices, Hybrid methods, Heterogeneous architectures in sparse linear solvers, Université de Bordeaux, 27 novembre 2017.

PhD in progress : Nicolas Bouzat; Fine grain algorithms and deployment methods for exascale plasma physics applications ; M.Mehrenberger (TONUS project-team), J.Roman, G. Latu (CEA-IRFM).

PhD in progress : Arnaud Durocher; High performance Dislocation Dynamics simulations on heterogeneous computing platforms for the study of creep deformation mechanisms for nuclear applications; O. Coulaud, L. Dupuy (CEA).

PhD in progress : Aurélien Falco; Data sparse calculation in FEM/BEM solution; E. Agullo, L. Giraud, G. Sylvand.

PhD in progress : Grégoire Pichon; Utilisation de techniques de compression \mathcal{H} -matrices pour solveur direct creux parallèle dans le cadre des applications FEM; M. Faverge, P. Ramet.

PhD in progress : Louis Poirel; Algebraic coarse space correction for parallel hybrid solvers; E. Agullo, L. Giraud.

10.2.3. Juries

- Vanessa C. Vargas Vallejo, “Approche logicielle pour améliorer la fiabilité d’applications parallèles implémentées sur des processeurs multi-cœur et many-cœur”, referee: L. Giraud, O. Romain, Université Joseph-Fourier de Grenoble, spécialité: nano électronique et nano technologie, 28 April 2018.
- Roberto Molina, “Hybridation of FETI methods”, referee: L. Giraud, D. J. Rixen; Université Pierre et Marie Curie, spécialité: mathématiques appliquées, 19 Décembre 2017.

- Krishna Kant Singh, "Algorithms for Adaptively Restrained Molecular Dynamics", referee: O. Coulaud, K. Hinsén. Université Joseph-Fourier de Grenoble, spécialité: informatique , 8 Novembre 2017.

11. Bibliography

Major publications by the team in recent years

- [1] E. AGULLO, O. AUMAGE, M. FAVERGE, N. FURMENTO, F. PRUVOST, M. SERGENT, S. THIBAUT. *Achieving High Performance on Supercomputers with a Sequential Task-based Programming Model*, in "IEEE Transactions on Parallel and Distributed Systems", 2017 [DOI : 10.1109/TPDS.2017.2766064], <https://hal.inria.fr/hal-01618526>
- [2] E. AGULLO, B. BRAMAS, O. COULAUD, E. DARVE, M. MESSNER, T. TAKAHASHI. *Task-Based FMM for Multicore Architectures*, in "SIAM Journal on Scientific Computing", 2014, vol. 36, n^o 1, pp. 66-93 [DOI : 10.1137/130915662], <https://hal.inria.fr/hal-00911856>
- [3] E. AGULLO, A. BUTTARI, A. GUERMOUCHE, F. LOPEZ. *Implementing multifrontal sparse solvers for multicore architectures with Sequential Task Flow runtime systems*, in "ACM Transactions on Mathematical Software", July 2016 [DOI : 10.1145/0000000.0000000], <https://hal.inria.fr/hal-01333645>
- [4] E. AGULLO, L. GIRAUD, Y.-F. JING. *Block GMRES method with inexact breakdowns and deflated restarting*, in "SIAM Journal on Matrix Analysis and Applications", November 2014, vol. 35, n^o 4, pp. 1625-1651, <https://hal.inria.fr/hal-01067159>
- [5] E. AGULLO, L. GIRAUD, P. SALAS, M. ZOUNON. *Interpolation-restart strategies for resilient eigensolvers*, in "SIAM Journal on Scientific Computing", 2016, vol. 38, n^o 5, pp. C560-C583 [DOI : 10.1137/15M1042115], <https://hal.inria.fr/hal-01347793>
- [6] A. CASADEI, P. RAMET, J. ROMAN. *An improved recursive graph bipartitioning algorithm for well balanced domain decomposition*, in "21st annual IEEE International Conference on High Performance Computing (HiPC 2014)", Goa, India, December 2014, <https://hal.inria.fr/hal-01100962>
- [7] J. DONGARRA, M. FAVERGE, T. HÉRAULT, M. JACQUELIN, J. LANGOU, Y. ROBERT. *Hierarchical QR factorization algorithms for multi-core clusters*, in "Parallel Computing", 2013, vol. 39, n^o 4-5, pp. 212-232 [DOI : 10.1016/J.PARCO.2013.01.003], <http://hal.inria.fr/hal-00809770>
- [8] X. LACOSTE, M. FAVERGE, P. RAMET, S. THIBAUT, G. BOSILCA. *Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes*, in "HCW'2014 workshop of IPDPS", Phoenix, United States, IEEE, May 2014, <https://hal.inria.fr/hal-00987094>
- [9] S. MOUSTAFA, M. FAVERGE, L. PLAGNE, P. RAMET. *3D Cartesian Transport Sweep for Massively Parallel Architectures with PARSEC*, in "29th IEEE International Parallel & Distributed Processing Symposium", Hyderabad, India, May 2015, pp. 581-590 [DOI : 10.1109/IPDPS.2015.75], <https://hal.inria.fr/hal-01078362>
- [10] M. ODUNLAMI, V. LE BRIS, D. BÉGUÉ, I. BARAILLE, O. COULAUD. *A-VCI: A flexible method to efficiently compute vibrational spectra*, in "Journal of Chemical Physics", June 2017, vol. 146, n^o 21 [DOI : 10.1063/1.4984266], <https://hal.inria.fr/hal-01534134>

- [11] G. PICHON, M. FAVERGE, P. RAMET, J. ROMAN. *Reordering Strategy for Blocking Optimization in Sparse Linear Solvers*, in "SIAM Journal on Matrix Analysis and Applications", 2017, vol. 38, n^o 1, pp. 226 - 248 [DOI : 10.1137/16M1062454], <https://hal.inria.fr/hal-01485507>
- [12] M. PREDARI, A. ESNARD, 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://hal.inria.fr/hal-01538600>
- [13] F. ROZAR, G. LATU, J. ROMAN, V. GRANDGIRARD. *Toward memory scalability of GYSELA code for extreme scale computers*, in "Concurrency and Computation: Practice and Experience", November 2014, pp. 1-16 [DOI : 10.1002/CPE.3429], <https://hal.inria.fr/hal-01111720>

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [14] B. ALZAIX. *Mathematical and numerical analysis of the Herberthson integral equation dedicated to electromagnetic plane wave scattering*, Université de Bordeaux, April 2017, <https://tel.archives-ouvertes.fr/tel-01558135>
- [15] A. GUERMOUCHE. *Towards Sparse Direct Solvers for Modern High-Performance Architectures*, Université de Bordeaux, November 2017, Habilitation à diriger des recherches
- [16] P. RAMET. *Heterogeneous architectures, Hybrid methods, Hierarchical matrices for Sparse Linear Solvers*, Université de Bordeaux, November 2017, Habilitation à diriger des recherches, <https://hal.inria.fr/tel-01668740>

Articles in International Peer-Reviewed Journals

- [17] E. AGULLO, O. AUMAGE, B. BRAMAS, O. COULAUD, S. PITOISET. *Bridging the gap between OpenMP and task-based runtime systems for the fast multipole method*, in "IEEE Transactions on Parallel and Distributed Systems", April 2017, 14 p. [DOI : 10.1109/TPDS.2017.2697857], <https://hal.inria.fr/hal-01517153>
- [18] E. AGULLO, O. AUMAGE, M. FAVERGE, N. FURMENTO, F. PRUVOST, M. SERGENT, S. THIBAUT. *Achieving High Performance on Supercomputers with a Sequential Task-based Programming Model*, in "IEEE Transactions on Parallel and Distributed Systems", 2017, forthcoming [DOI : 10.1109/TPDS.2017.2766064], <https://hal.inria.fr/hal-01618526>
- [19] D. CARIOLLE, P. MOINAT, H. TEYSSÈDRE, L. GIRAUD, B. JOSSE, F. LEFÈVRE. *ASIS v1.0: an adaptive solver for the simulation of atmospheric chemistry*, in "Geoscientific Model Development", 2017, vol. 10, pp. 1467 - 1485 [DOI : 10.5194/GMD-10-1467-2017], <https://hal.inria.fr/hal-01507392>
- [20] J. M. COUTEYEN CARPAYE, J. ROMAN, P. BRENNER. *Design and Analysis of a Task-based Parallelization over a Runtime System of an Explicit Finite-Volume CFD Code with Adaptive Time Stepping*, in "International Journal of Computational Science and Engineering", 2017, pp. 1 - 22, <https://arxiv.org/abs/1704.01144> [DOI : 10.1016/J.JOCS.2017.03.008], <https://hal.inria.fr/hal-01507613>
- [21] M. ODUNLAMI, V. LE BRIS, D. BÉGUÉ, I. BARAILLE, O. COULAUD. *A-VCI: A flexible method to efficiently compute vibrational spectra*, in "Journal of Chemical Physics", June 2017, vol. 146, n^o 21 [DOI : 10.1063/1.4984266], <https://hal.inria.fr/hal-01534134>

- [22] G. PICHON, M. FAVERGE, P. RAMET, J. ROMAN. *Reordering Strategy for Blocking Optimization in Sparse Linear Solvers*, in "SIAM Journal on Matrix Analysis and Applications", 2017, vol. 38, n^o 1, pp. 226 - 248 [DOI : 10.1137/16M1062454], <https://hal.inria.fr/hal-01485507>
- [23] M. PREDARI, A. ESNARD, 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://hal.inria.fr/hal-01538600>
- [24] D. SUKKARI, H. LTAIEF, M. FAVERGE, D. KEYES. *Asynchronous Task-Based Polar Decomposition on Single Node Manycore Architectures*, in "IEEE Transactions on Parallel and Distributed Systems", August 2017, vol. XX [DOI : 10.1109/TPDS.2017.2755655], <https://hal.inria.fr/hal-01585079>

International Conferences with Proceedings

- [25] E. AGULLO, S. COOLS, L. GIRAUD, W. VANROOSE, E. F. YETKIN. *Soft Error in Classical PCG and its Variants: Sensitivity, Numerical Detections and Possible Recovery Policies*, in "SIAM Annual meeting 2017, AN'17", Pittsburgh, United States, July 2017, <https://hal.inria.fr/hal-01670198>
- [26] O. BEAUMONT, L. EYRAUD-DUBOIS, S. KUMAR. *Approximation Proofs of a Fast and Efficient List Scheduling Algorithm for Task-Based Runtime Systems on Multicores and GPUs*, in "IEEE International Parallel & Distributed Processing Symposium (IPDPS)", Orlando, United States, May 2017, <https://hal.inria.fr/hal-01386174>
- [27] N. BOUZAT, F. ROZAR, G. LATU, J. ROMAN. *A New Parallelization Scheme for the Hermite Interpolation Based Gyroaverage Operator*, in "ISPDC 2017 - 16th International Symposium on Parallel and Distributed Computing", Innsbruck, Austria, IEEE, July 2017, pp. 1-8 [DOI : 10.1109/ISPDC.2017.12], <https://hal.inria.fr/hal-01687727>
- [28] P. CLAUSS, E. ALTINTAS, M. KUHN. *Automatic Collapsing of Non-Rectangular Loops*, in "Parallel and Distributed Processing Symposium (IPDPS), 2017", Orlando, United States, IEEE International, May 2017, pp. 778 - 787 [DOI : 10.1109/IPDPS.2017.34], <https://hal.inria.fr/hal-01581081>
- [29] M. FAVERGE, J. LANGOU, Y. ROBERT, J. DONGARRA. *Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-Memory Implementation*, in "IPDPS'17 - 31st IEEE International Parallel and Distributed Processing Symposium", Orlando, United States, May 2017, <https://hal.inria.fr/hal-01484113>
- [30] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver Using Block Low-Rank Compression*, in "18th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2017)", Orlando, United States, June 2017, <https://hal.inria.fr/hal-01502215>

National Conferences with Proceedings

- [31] E. AGULLO, A. FALCO, L. GIRAUD, G. SYLVAND. *Vers une factorisation symbolique hiérarchique de rang faible pour des matrices creuses*, in "Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS'17)", Sophia Antipolis, France, June 2017, <https://hal.inria.fr/hal-01597072>
- [32] G. PICHON. *Utilisation de la compression Block Low-Rank pour accélérer un solveur direct creux supernodal*, in "Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS'17)", Sophia Antipolis, France, June 2017, <https://hal.inria.fr/hal-01585660>

Conferences without Proceedings

- [33] E. AGULLO, L. GIRAUD, E. DARVE, Y. HARNESS. *Soft Error in Classical PCG and its Variants: Sensitivity, Numerical Detections and Possible Recovery Policies*, in "SIAM Annual meeting 2017, AN'17", Pittsburgh, United States, July 2017, <https://hal.inria.fr/hal-01670160>
- [34] E. AGULLO, L. GIRAUD, L. POIREL. *Robust coarse spaces for abstract Schwarz preconditioners via generalized eigenproblems*, in "International conference on domain decomposition methods, DD24", Svalbard, Norway, February 2017, <https://hal.inria.fr/hal-01670178>
- [35] E. AGULLO, L. GIRAUD, E. F. YETKIN. *Soft Error in PCG: Sensitivity, Numerical Detections and Possible Recoveries*, in "SIAM Conference on Computational Science and Engineering, CSE'17", Atlanta, United States, February 2017, <https://hal.inria.fr/hal-01670189>
- [36] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver exploiting Low-Rankness Property*, in "Sparse Days 2017", Toulouse, France, September 2017, <https://hal.inria.fr/hal-01585622>
- [37] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver Using Hierarchical Compression over Runtime System*, in "SIAM Conference on Computation Science and Engineering (CSE'17)", Atlanta, United States, February 2017, <https://hal.inria.fr/hal-01421379>
- [38] G. PICHON, M. FAVERGE, P. RAMET. *Exploiting Modern Manycore Architecture in Sparse Direct Solver with Runtime Systems*, in "SIAM Conference on Computation Science and Engineering (CSE'17)", Atlanta, United States, February 2017, <https://hal.inria.fr/hal-01421383>
- [39] G. PICHON, M. FAVERGE, P. RAMET, J. ROMAN. *Impact of Blocking Strategies for Sparse Direct Solvers on Top of Generic Runtimes*, in "SIAM Conference on Computation Science and Engineering (CSE'17)", Atlanta, United States, February 2017, <https://hal.inria.fr/hal-01421384>

Research Reports

- [40] E. AGULLO, B. BRAMAS, O. COULAUD, M. KHANNOUZ, L. STANISIC. *Task-based fast multipole method for clusters of multicore processors*, Inria Bordeaux Sud-Ouest, March 2017, n^o RR-8970, 15 p., <https://hal.inria.fr/hal-01387482>
- [41] E. AGULLO, B. BRAMAS, O. COULAUD, L. STANISIC, S. THIBAUT. *Modeling Irregular Kernels of Task-based codes: Illustration with the Fast Multipole Method*, Inria Bordeaux, February 2017, n^o RR-9036, 35 p., <https://hal.inria.fr/hal-01474556>
- [42] E. AGULLO, A. BUTTARI, M. BYCKLING, A. GUERMOUCHE, I. MASLIAH. *Achieving high-performance with a sparse direct solver on Intel KNL*, Inria Bordeaux Sud-Ouest ; CNRS-IRIT ; Intel corporation ; Université Bordeaux, February 2017, n^o RR-9035, 15 p., <https://hal.inria.fr/hal-01473475>
- [43] I. BARAILLE, D. BÉGUÉ, O. COULAUD, V. LE BRIS, M. ODUNLAMI. *A-VCI: a flexible method to efficiently compute vibrational spectra*, Inria, March 2017, n^o RR-9043, 35 p., <https://hal.inria.fr/hal-01485877>

- [44] P. BLANCHARD, P. P. CHAUMEIL, J.-M. FRIGERIO, F. RIMET, F. SALIN, S. THÉRON, O. COULAUD, A. FRANC. *A geometric view of Biodiversity: scaling to metagenomics*, Inria ; INRA, January 2018, n^o RR-9144, pp. 1-16, <https://hal.inria.fr/hal-01685711>
- [45] N. BOUZAT, F. ROZAR, G. LATU, J. ROMAN. *A new parallelization scheme for the Hermite interpolation based gyroaverage operator*, Inria, April 2017, n^o RR-9054, 22 p. , <https://hal.inria.fr/hal-01502513>
- [46] M. FAVERGE, S. MOUSTAFA, F. FÉVOTTE, L. PLAGNE, P. RAMET. *Efficient Parallel Solution of the 3D Stationary Boltzmann Transport Equation for Diffusive Problems*, Inria ; EDF Lab, September 2017, n^o RR-9116, 22 p. , <https://hal.inria.fr/hal-01630208>
- [47] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver Using Block Low-Rank Compression*, Inria Bordeaux Sud-Ouest, January 2017, n^o RR-9022, 24 p. , <https://hal.inria.fr/hal-01450732>
- [48] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver Using Block Low-Rank Compression: design, performance and analysis*, Inria Bordeaux Sud-Ouest, December 2017, n^o RR-9130, pp. 1-32, <https://hal.inria.fr/hal-01660665>

Other Publications

- [49] G. PICHON, E. DARVE, M. FAVERGE, S. LANTERI, P. RAMET, J. ROMAN. *Sparse supernodal solver with low-rank compression for solving the frequency-domain Maxwell equations discretized by a high order HDG method*, November 2017, pp. 1-55, Journées jeunes chercheur-e-s - Résolution de problèmes d'ondes harmoniques de grande taille, <https://hal.inria.fr/hal-01660653>