



IN PARTNERSHIP WITH:
CNRS

**Institut Polytechnique de
Bordeaux**

Université de Bordeaux

Activity Report 2016

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. Members	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	8
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	10
3.5.1. Dynamic load-balancing with variable number of processors	10
3.5.2. Load balancing of coupled codes	11
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
4.2.1. High performance simulation for ITER tokamak	12
4.2.2. SN Cartesian solver for nuclear core simulation	13
4.2.3. 3D aerodynamics for unsteady problems with bodies in relative motion	13
5. Highlights of the Year	13
6. New Software and Platforms	14
6.1. Spack-morse	14
6.2. Chameleon	14
6.3. HIPS	14
6.4. MaPHYS	15
6.5. MetaPart	15
6.6. PaStiX	16
6.7. QR_Mumps	16
6.8. ScalFMM	16
6.9. ViTE	17
6.10. Platforms	17
7. New Results	18
7.1. High-performance computing on next generation architectures	18
7.1.1. Numerical recovery strategies for parallel resilient Krylov linear solvers	18
7.1.2. Interpolation-restart strategies for resilient eigensolvers	18
7.2. High performance solvers for large linear algebra problems	19
7.2.1. Exploiting Kepler architecture in sparse direct solver with runtime systems	19
7.2.2. Blocking strategy optimizations for sparse direct linear solver on heterogeneous architectures	19
7.2.3. Sparse supernodal solver using hierarchical compression	19
7.2.4. Hierarchical hybrid sparse linear solver for multicore platforms	20
7.2.5. Task-based conjugate gradient: from multi-GPU towards heterogeneous architectures	20

7.2.6.	Analysis of rounding error accumulation in conjugate gradients to improve the maximal attainable accuracy of pipelined CG	20
7.2.7.	Nearly optimal fast preconditioning of symmetric positive definite matrices	21
7.2.8.	Robust coarse spaces for abstract Schwarz preconditioners via generalized eigenproblems	21
7.3.	High performance fast multipole method for N-body problems	22
7.3.1.	Task-based fast multipole method	22
7.3.2.	Task-based fast multipole method for clusters of multicore processors	22
7.4.	Efficient algorithmic for load balancing and code coupling in complex simulations	22
7.5.	Application Domains	23
7.5.1.	Material physics	23
7.5.1.1.	Molecular vibrational spectroscopy	23
7.5.1.2.	Dislocations	23
7.5.2.	Co-design for scalable numerical algorithms in scientific applications	24
7.5.2.1.	Interior penalty discontinuous Galerkin method for coupled elasto-acoustic media	24
7.5.2.2.	High performance simulation for ITER tokamak	24
7.5.2.3.	3D aerodynamics for unsteady problems with bodies in relative motion	24
8.	Bilateral Contracts and Grants with Industry	25
9.	Partnerships and Cooperations	25
9.1.	National Initiatives	25
9.1.1.	Inria Project Lab	25
9.1.2.	ANR	25
9.1.2.1.	SOLHAR: SOLvers for Heterogeneous Architectures over Runtime systems	25
9.1.2.2.	ANEMOS: Advanced Numeric for ELMs : Modeling and Optimized Schemes	26
9.1.2.3.	DEDALES: Algebraic and geometric domain decomposition for subsurface/groundwater flows	27
9.1.2.4.	TECSER: Novel high performance numerical solution techniques for RCS computations	28
9.2.	European Initiatives	28
9.2.1.1.	EoCoE	28
9.2.1.2.	HPC4E	29
9.2.1.3.	EXA2CT	30
9.3.	International Initiatives	30
9.3.1.1.	MORSE	30
9.3.1.2.	FASTLA	31
10.	Dissemination	31
10.1.	Promoting Scientific Activities	31
10.1.1.	Scientific Events Organisation	31
10.1.1.1.	General Chair, Scientific Chair	31
10.1.1.2.	Member of the Organizing Committees	31
10.1.2.	Scientific Events Selection	32
10.1.3.	Journal	32
10.1.3.1.	Member of the Editorial Boards	32
10.1.3.2.	Reviewer - Reviewing Activities	32
10.1.4.	Invited Talks	32
10.1.5.	Scientific Expertise	32
10.1.6.	Research Administration	32
10.2.	Teaching - Supervision - Juries	32
10.2.1.	Teaching	32
10.2.2.	Supervision	33
10.2.3.	Juries	34

11. Bibliography **34**

Project-Team HIEPACS

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

Keywords:

Computer Science and Digital Science:

- 1.1.4. - High performance computing
- 1.1.5. - Exascale
- 6.2.5. - Numerical Linear Algebra
- 6.2.7. - High performance computing
- 7.1. - Parallel and distributed algorithms
- 7.2. - Discrete mathematics, combinatorics

Other Research Topics and Application Domains:

- 3.3.1. - Earth and subsoil
- 3.4.2. - Industrial risks and waste
- 4.1. - Fossile energy production (oil, gas)
- 4.2.2. - Fusion
- 5.5. - Materials
- 9.4.1. - Computer science
- 9.4.2. - Mathematics
- 9.4.4. - Chemistry

1. Members

Research Scientists

- Emmanuel Agullo [Inria, Researcher]
- Olivier Coulaud [Inria, Senior Researcher, HDR]
- Luc Giraud [Team leader, Inria, Senior Researcher, HDR]
- Jean Roman [Inria, Senior Researcher, Direction for Science and half-time in **HIEPACS**, Professor at Bordeaux INP on secondment, HDR]

Faculty Members

- Aurélien Esnard [Univ. Bordeaux, Associate Professor]
- Mathieu Faverge [Bordeaux INP, Associate Professor]
- Abdou Guermouche [Univ. Bordeaux, Associate Professor]
- Pierre Ramet [Univ. Bordeaux, Associate Professor]

Engineers

- Patrick Dos Santos [Univ. Bordeaux, until Apr 2016]
- Matias Hastaran [Inria, granted by FP7 PRACE-4IP project]
- Quentin Khan [Inria]
- Florent Pruvost [Inria, SED]
- Julien Pedron [Inria, until Sep 2016, granted by Labex CPI]
- Cyrille Piacibello [Inria, granted by Hi-Box DGA project]
- Gilles Marait [Inria, granted by H2020 **EoCoE**, from Sep 2016]

PhD Students

- Pierre Blanchard [French Ministry through ENS Cachan]
- Nicolas Bouzat [Inria, co-supervised with **TONUS** project-team]

Astrid Casadei [Inria, until Mar 2016]
Jean-Marie Couteyen [Airbus Safran Launchers, granted by ANRT-CIFRE, until Jan 2016]
Arnaud Durocher [CEA, granted by ANRT-CIFRE]
Aurélien Falco [Inria, granted by Conseil Régional Aquitaine - Airbus Group Innovations]
Cyril Fournier [CERFACS]
Grégoire Pichon [Inria, granted by DGA-Inria]
Louis Poirel [Inria, granted by ANR **DEDALES** project]
Maria Predari [Inria, granted by Conseil Régional Aquitaine - Inria]

Post-Doctoral Fellows

Matthieu Kuhn [Inria, granted by **TECSER** project]
Ian Masliah [Inria, from Nov 2016]
Yuval Harness [Inria, granted by Conseil Régional Aquitaine - Inria]
Emrullah Fatih Yetkin [Inria, granted by FP7 **EXA2CT** project, until Aug 2016]

Administrative Assistant

Chrystel Plumejeau [Inria]

Others

Alban Bellot [Inria, Internship, from Jun 2016 until Jul 2016]
Pierre Brenner [Airbus Safran Launchers, Engineer Expert]
Mathieu Deschamps [Inria, Internship, from Jun 2016 until Sep 2016]
Martin Khannouz [Inria, Internship, from Feb 2016 until Aug 2016]
Guillaume Latu [CEA Cadarache, IRFM, Research Scientist]
Alexandre Moreau [Inria, Internship, Feb-Aug 2016]
Mathis Rade Morzadec [Inria, Internship, from May 2016 until Aug 2016]
Hugues Rauwel [Inria, Internship, from Jul 2016 until Aug 2016]
Guillaume Sylvand [Airbus Group Innovations, Engineer expert]
Nathan Thomasset [Inria, Internship, from May 2016 until Jul 2016]
Hugo Vikstrom [Inria, Internship, from Jun 2016 until Aug 2016]

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 plans for 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 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 two ongoing associated teams, namely **MORSE** and **FASTLA** we contribute with world leading groups 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, which are involved in the IPL **C2S@EXA**, 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, some of them being involved in the IPL **C2S@EXA**. 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, Guillaume Latu, Grégoire Pichon, Florent Pruvost, Pierre Ramet, Jean Roman, Emrullah Fatih Yetkin.

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

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

Considering the complexity of modern architectures like massively parallel architectures or new generation heterogeneous multicore architectures, task scheduling becomes a challenging problem which is central to obtain a high efficiency. 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. The framework that hosts all these research activities is the associate team **MORSE**. 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. In that respect, we are involved in the **EXA2CT** FP7 project.

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, Julien Pedron, Cyrille Piacibello, Grégoire Pichon, Louis Poirel, Pierre Ramet, Jean Roman.

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 associate team **MORSE** and 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 intend to 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 will be 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, Olivier Coulaud, Quentin Khan, Cyrille Piacibello, Guillaume Sylvand.

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

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

3.4.1. Improvement of calculation efficiency

Nowadays, the high performance computing community is examining alternative architectures that address the limitations of modern cache-based designs. GPU (Graphics Processing Units) and the Cell processor have thus already been used in astrophysics and in molecular dynamics. The Fast Mutipole 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: Astrid Casadei, Olivier Coulaud, 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 use a similar methodology based upon the graph model and are expressed as variants of the classic graph partitioning problem, using additional constraints or different objectives.

3.5.1. Dynamic load-balancing with variable number of processors

As a preliminary step related to the dynamic load balancing of coupled codes, we focus on the problem of dynamic load balancing of a single parallel code, with variable number of processors. Indeed, if the workload varies drastically during the simulation, the load must be redistributed regularly among the processors. Dynamic load balancing is a well studied subject but most studies are limited to an initially fixed number of processors. Adjusting the number of processors at runtime allows one to preserve the parallel code efficiency or keep running the simulation when the current memory resources are exceeded. We call this problem, *MxN graph repartitioning*.

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

3.5.2. Load balancing of coupled codes

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

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

3.5.3. Load balancing strategies for hybrid sparse linear solvers

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

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, Pierre Brenner, Jean-Marie Couteyen, Mathieu Faverge, Guillaume Latu, Pierre Ramet, Jean Roman.

The research activities concerning the ITER challenge are involved in the Inria Project Lab (IPL) **C2S@EXA**.

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 **JOEK** 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 (**JOEK** 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.

4.2.2. SN Cartesian solver for nuclear core simulation

As part of its activity, EDF R&D is developing a new nuclear core simulation code named COCAGNE that relies on a Simplified PN (SPN) method to compute the neutron flux inside the core for eigenvalue calculations. In order to assess the accuracy of SPN results, a 3D Cartesian model of PWR nuclear cores has been designed and a reference neutron flux inside this core has been computed with a Monte Carlo transport code from Oak Ridge National Lab. This kind of 3D whole core probabilistic evaluation of the flux is computationally very demanding. An efficient deterministic approach is therefore required to reduce the computation effort dedicated to reference simulations.

In this collaboration, we work on the parallelization (for shared and distributed memories) of the DOMINO code, a parallel 3D Cartesian SN solver specialized for PWR core reactivity computations which is fully integrated in the COCAGNE system.

4.2.3. 3D aerodynamics for unsteady problems with bodies in relative motion

Airbus Defence and Space has developed for 20 years the FLUSEPA code which focuses on unsteady phenomenon with changing topology like stage separation or rocket launch. The code is based on a finite volume formulation with temporal adaptive time integration and supports bodies in relative motion. The temporal adaptive integration classifies cells in several temporal levels and this repartition can evolve during the computation, leading to load-balancing issues in a parallel computation context. Bodies in relative motion are managed through a CHIMERA-like technique which allows building a composite mesh by merging multiple meshes. The meshes with the highest priorities recover the least ones, and at the boundaries of the covered mesh, an intersection is computed. Unlike classical CHIMERA technique, no interpolation is performed, allowing a conservative flow integration. The main objective of this collaboration is to design a new scalable version of FLUSEPA from a task-based parallelization over a runtime system (**StarPU**) in order to run efficiently on modern heterogeneous multicore parallel architectures very large 3D simulations (for example ARIANE 5 and 6 booster separation).

5. Highlights of the Year

5.1. Highlights of the Year

5.1.1. Conference organization

We organized the 9th International workshop on Parallel Matrix Algorithms and Application (PMAA'16 - July 6-8) in collaboration with Bordeaux INP, CNRS and Université de Bordeaux. The conference that was composed of 4 invited plenary presentations and 76 regular talks. Around 120 people attended the conference, 70 % were from Europe, 20 % from North America, 7 % from Asia; among them more than 25 % were students. We succeeded to offer free registration to the students thanks to the sponsorship we arose from Airbus DS, CEA, CERFACS, Clustervision, Labex CPU, DELL, EDF, IBM and Total that contributed to balance our budget.

More details can be found on <http://pmaa16.inria.fr>

6. New Software and Platforms

6.1. Spack-morse

The radical change we have adopted in terms of methodology (task-based programming strongly) changes the software design. In particular, our codes become more and more modular and the complexity of their inter-dependencies is subsequently very high.

In order to address this complexity we have chosen to rely on the **Spack** flexible package manager designed to support multiple versions, configurations, platforms, and compilers (<http://software.llnl.gov/spack> developed and maintained at LLNL). We have integrated all our libraries above this package manager in the **Spack-Morse** extension that we maintain in **HIEPACS**.

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-3 (original software reusing known ideas and introducing new ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web). source distribution or a commercially-distributed product).
- Contact: Florent Pruvost
- URL: <http://morse.gforge.inria.fr/spack/spack.html>

6.2. Chameleon

Chameleon is a dense linear algebra software relying on the STF sequential task-based programming paradigm. It implements the tile algorithms originally designed for multicore architectures in the **PLASMA** package and extends them so that they can be processed on by a runtime system to exploit any type of hardware architecture (multicore, GPU, heterogeneous, supercomputer). This software is central for the team as it allows to investigate in a relatively simple context (regular dense linear algebra algorithms) new types of designs before implementing them for the more irregular algorithms implemented in the software packages described below.

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web). source distribution or a commercially-distributed product).
- Contact: Emmanuel Agullo
- URL: <https://project.inria.fr/chameleon>

6.3. HIPS

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

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 sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being).

Thus, **HIPS** is a software library that provides several methods to build an efficient preconditioner in almost all situations.

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-2 (basic maintenance to keep the software alive).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Pierre Ramet
- URL: <http://hips.gforge.inria.fr>

6.4. MaPHYs

MaPHYs (Massively Parallel Hybrid Solver) is an hybrid iterative/direct parallel (MPI-threads) sparse linear solver based on algebraic domain decomposition technique for real/complex symmetric positive definite/unsym/-metric matrices. For a given number of MPI processes/domains, **MaPHYs** solves the Schur complement computed from the adjacency graph of the sparse matrix using a preconditioned Krylov subspace method (CG or GMRES). The provided preconditioners are variants of an algebraic Additive Schwarz methods. A prototype version of a two level preconditionner using an algebraic coarse space is available but not yet publicly distributed (provided upon request for beta testers).

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-4 (well-defined and implemented plans for future maintenance and evolution).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Emmanuel Agullo
- URL: <https://project.inria.fr/maphys/fr>

6.5. MetaPart

MetaPart is a library that addresses the challenge of (dynamic) load balancing for emerging complex parallel simulations, such as multi-physics or multi-scale coupling applications. First, it offers a uniform API over state-of-the-art (hyper-) graph partitioning & ordering software packages such as **Scotch**, PaToH, METIS, Zoltan, Mondriaan, etc. Based upon this API, it provides a framework that facilitates the development and the evaluation of high-level partitioning methods, such as MxN repartitioning or coupling-aware partitioning (co-partitioning).

- Audience: A-1 (internal prototype).
- Software originality: SO-3 (original software reusing known ideas and introducing new ideas).
- Software maturity: SM-2 (basic usage works, terse documentation).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Aurélien Esnard
- URL: <http://metapart.gforge.inria.fr>

6.6. PaStiX

PaStiX (Parallel Sparse matriX package) 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) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The **PaStiX** solver is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer 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.

- Audience: A-5 (wide audience, large user's community).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-4 (major software project, strong software engineering).
- Evolution and maintenance: EM-4 (well-defined and implemented plans for future maintenance and evolution).
- Software distribution and licensing: SDL-5 (external packaging and distribution, as part of a popular open source distribution or a commercially-distributed product).
- Contact: Pierre Ramet
- URL: <http://pastix.gforge.inria.fr>

6.7. QR_Mumps

qr_mumps is a software package for the solution of sparse, linear systems on multicore computers. It implements a direct solution method based on the QR factorization of the input matrix. Therefore, it is suited to solving sparse least-squares problems and to computing the minimum-norm solution of sparse, under-determined problems. It can obviously be used for solving square problems in which case the stability provided by the use of orthogonal transformations comes at the cost of a higher operation count with respect to solvers based on, e.g., the LU factorization. **qr_mumps** supports real and complex, single or double precision arithmetic.

qr_mumps is mainly developed and maintained by the APO team of the IRIT laboratory of Toulouse. **HIEPACS** is an active contributor to this project.

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Emmanuel Agullo
- URL: http://buttari.perso.enseeiht.fr/qr_mumps/

6.8. ScalFMM

ScalFMM is a library to compute N-body interactions using the Fast Multipole Method. This is a parallel kernel independent fast multipole method based on interpolation (Chebychev or equispaced grid points).

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 (fork-join and tasks models), MPI and runtime system (**StarPU**). 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 tree structure (hierarchical octree and Group-Tree) and the parallel algorithms ;
- the kernels (scalar, tensorial and multi-rhs). Classical kernels are available (Coulombic, Leonard-Jones, Gaussian, Stokes, ...)

This modular architecture allows us to easily add new FMM algorithms or kernels and new paradigm of parallelization. Today, we also proposed the FMM based on spherical harmonic expansion with Blas or rotation optimization for Coulombic potential) and all algorithms are designed to treat more complex kernels by adding multiple right-hand sides, tensorial structures, ...

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Olivier Coulaud
- URL: <http://scalymm-public.gforge.inria.fr/doc/>

6.9. ViTE

ViTE is a trace explorer. It is a tool to visualize execution traces in Pajé or OTF format for debugging and profiling parallel or distributed applications. It is developed with C++ programming language with OpenGL and Qt technologies.

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-3 (original software reusing known ideas and introducing new ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-2 (basic maintenance to keep the software alive).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).
- Contact: Mathieu Faverge
- URL: <http://vite.gforge.inria.fr>

6.10. Platforms

6.10.1. PlaFRIM: Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

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. Numerical recovery strategies for parallel resilient Krylov linear solvers

As the computational power of high performance computing (HPC) systems continues to increase by using a huge number of cores or specialized processing units, HPC applications are increasingly prone to faults. In this paper, we present a new class of numerical fault tolerance algorithms to cope with node crashes in parallel distributed environments. This new resilient scheme is designed at application level and does not require extra resources, i.e., computational unit or computing time, when no fault occurs. In the framework of iterative methods for the solution of sparse linear systems, we present numerical algorithms to extract relevant information from available data after a fault, assuming a separate mechanism ensures the fault detection. After data extraction, a well chosen part of missing data is regenerated through interpolation strategies to constitute meaningful inputs to restart the iterative scheme. We have developed these methods, referred to as Interpolation-Restart techniques, for Krylov subspace linear solvers. After a fault, lost entries of the current iterate computed by the solver are interpolated to define a new initial guess to restart the Krylov method. A well suited initial guess is computed by using the entries of the faulty iterate available on surviving nodes. We present two interpolation policies that preserve key numerical properties of well-known linear solvers, namely the monotonic decrease of the A-norm of the error of the conjugate gradient or the residual norm decrease of GMRES. The qualitative numerical behavior of the resulting scheme have been validated with sequential simulations, when the number of faults and the amount of data losses are varied. Finally, the computational costs associated with the recovery mechanism have been evaluated through parallel experiments.

More details on this work can be found in [7].

7.1.2. Interpolation-restart strategies for resilient eigensolvers

The solution of large eigenproblems is involved in many scientific and engineering applications when for instance, stability analysis is a concern. For large simulation in material physics or thermo-acoustics, the calculation can last for many hours on large parallel platforms. On future large-scale systems, the mean time between failures (MTBF) of the system is expected to decrease so that many faults could occur during the solution of large eigenproblems. Consequently, it becomes critical to design parallel eigensolvers that can survive faults. In that framework, we investigate the relevance of approaches relying on numerical techniques, which might be combined with more classical techniques for real large-scale parallel implementations. Because we focus on numerical remedies we do not consider parallel implementations nor parallel experiments but only numerical experiments. We assume that a separate mechanism ensures the fault detection and that a system layer provides support for setting back the environment (processes, . . .) in a running state. Once the system is in a running state, after a fault, our main objective is to provide robust resilient schemes so that the eigensolver may keep converging in the presence of the fault without restarting the calculation from scratch. For this purpose, we extend the interpolation-restart (IR) strategies initially introduced for the solution of linear systems in a previous work to the solution of eigenproblems in this paper. For a given numerical scheme, the IR strategies consist of extracting relevant spectral information from available data after a fault. After data extraction, a well-selected part of the missing data is regenerated through interpolation strategies

to constitute a meaningful input to restart the numerical algorithm. One of the main features of this numerical remedy is that it does not require extra resources, i.e., computational unit or computing time, when no fault occurs. In this paper, we revisit a few state-of-the-art methods for solving large sparse eigenvalue problems namely the Arnoldi methods, subspace iteration methods and the Jacobi-Davidson method, in the light of our IR strategies. For each considered eigensolver, we adapt the IR strategies to regenerate as much spectral information as possible. Through extensive numerical experiments, we study the respective robustness of the resulting resilient schemes with respect to the MTBF and to the amount of data loss via qualitative and quantitative illustrations.

More details on this work can be found in [8].

7.2. High performance solvers for large linear algebra problems

7.2.1. Exploiting Kepler architecture in sparse direct solver with runtime systems

Many works have addressed heterogeneous architectures to exploit accelerators such as GPUs or Intel Xeon Phi with interesting speedup. Despite researches towards generic solutions to efficiently exploit those accelerators, their hardware evolution requires continual adaptation of the kernels running on those architectures. The recent Nvidia architectures, as Kepler, present a larger number of parallel units thus requiring more data to feed every computational units. A solution considered to supply enough computation has been to study problems with large number of small computations. The batched BLAS libraries proposed by Intel, Nvidia, or the University of Tennessee are examples of this solution. We have investigated the use of the variable size batched matrix-matrix multiply to improve the performance of a the PaStiX sparse direct solver. Indeed, this kernel suits the super-nodal method of the solver, and the multiple updates of variable sizes that occur during the numerical factorization.

These contributions have been presented at the PMAA'16 conference [28].

7.2.2. 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 computations 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 10% on CPUs and up to 40% on GPUs.

These contributions have been presented at the SIAM PP'16 conference [35] and an extended paper has been submitted to SIAM Journal on Matrix Analysis and Applications [49].

7.2.3. Sparse supernodal solver using hierarchical compression

In the context of FASTLA associate team, during the last 3 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.

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 the first step, we are targeting an $O(N^{4/3})$ solver. Preliminary benchmarks indicate that a speed up of 2x to 10x is possible (on the largest test cases).

In the context of the **FASTLA** team, 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 a new implementation of the **PaStiX** solver using hierarchical compression to reduce the burden on large blocks appearing during the nested dissection process. To improve the efficiency of our sparse update kernel for both BLR (block low-rank) and HODLR (hierarchically off-diagonal low-rank), we are now investigating to BDLR (boundary distance low-rank) approximation scheme to preselect rows and columns in the low-rank approximation algorithm. We also have to improve our ordering strategies to enhance data locality and compressibility. The implementation is based on runtime systems to exploit parallelism.

Some contributions have already been presented at the workshops on Fast Solvers [32], [31], [30]. This work is a joint effort between Professor Darve's group at Stanford and the Inria HiePACS team within **FASTLA**.

7.2.4. Hierarchical hybrid sparse linear solver for multicore platforms

The solution of large sparse linear systems is a critical operation for many numerical simulations. To cope with the hierarchical design of modern supercomputers, hybrid solvers based on Domain Decomposition Methods (DDM) have been proposed. Among them, approaches consisting of solving the problem on the interior of the domains with a sparse direct method and the problem on their interface with a preconditioned iterative method applied to the related Schur Complement have shown an attractive potential as they can combine the robustness of direct methods and the low memory footprint of iterative methods. In this report, we consider an additive Schwarz preconditioner for the Schur Complement, which represents a scalable candidate but whose numerical robustness may decrease when the number of domains becomes too large. We thus propose a two-level MPI/thread parallel approach to control the number of domains and hence the numerical behaviour. We illustrate our discussion with large-scale matrices arising from real-life applications and processed on both a modern cluster and a supercomputer. We show that the resulting method can process matrices such as tdr455k for which we previously either ran out of memory on few nodes or failed to converge on a larger number of nodes. Matrices such as Nachos_4M that could not be correctly processed in the past can now be efficiently processed up to a very large number of CPU cores (24 576 cores). The corresponding code has been incorporated into the **MaPHyS** package.

More details on this work can be found in [44]

7.2.5. Task-based conjugate gradient: from multi-GPU towards heterogeneous architectures

Whereas most parallel High Performance Computing (HPC) numerical libraries have been written as highly tuned and mostly monolithic codes, the increased complexity of modern architectures led the computational science and engineering community to consider more modular programming paradigms such as task-based paradigms to design new generation of parallel simulation code; this enables to delegate part of the work to a third party software such as a runtime system. That latter approach has been shown to be very productive and efficient with compute-intensive algorithms, such as dense linear algebra and sparse direct solvers. In this study, we consider a much more irregular, and synchronizing algorithm, namely the Conjugate Gradient (CG) algorithm. We propose a task-based formulation of the algorithm together with a very fine instrumentation of the runtime system. We show that almost optimum speed up may be reached on a multi-GPU platform (relatively to the mono-GPU case) and, as a very preliminary but promising result, that the approach can be effectively used to handle heterogeneous architectures composed of a multicore chip and multiple GPUs. We expect that these results will pave the way for investigating the design of new advanced, irregular numerical algorithms on top of runtime systems.

More details on this work can be found in [42]

7.2.6. Analysis of rounding error accumulation in conjugate gradients to improve the maximal attainable accuracy of pipelined CG

Pipelined Krylov solvers typically offer better scalability in the strong scaling limit compared to standard Krylov methods. The synchronization bottleneck is mitigated by overlapping time-consuming global communications with useful computations in the algorithm. However, to achieve this communication hiding strategy, pipelined methods feature multiple recurrence relations on additional auxiliary variables to update the guess for the solution. This paper aims at studying the influence of rounding errors on the convergence of the pipelined Conjugate Gradient method. It is analyzed why rounding effects have a significantly larger impact on the maximal attainable accuracy of the pipelined CG algorithm compared to the traditional CG method. Furthermore, an algebraic model for the accumulation of rounding errors throughout the (pipelined) CG algorithm is derived. Based on this rounding error model, we then propose an automated residual replacement strategy to reduce the effect of rounding errors on the final iterative solution. The resulting pipelined CG method with automated residual replacement improves the maximal attainable accuracy of pipelined CG to a precision comparable to that of standard CG, while maintaining the efficient parallel performance of the pipelined method.

More details on this work can be found in [46].

7.2.7. *Nearly optimal fast preconditioning of symmetric positive definite matrices*

We consider the hierarchical off-diagonal low-rank preconditioning of symmetric positive definite matrices arising from second order elliptic boundary value problems. When the scale of such problems becomes large combined with possibly complex geometry or unstable of boundary conditions, the representing matrix is large and typically ill-conditioned. Multilevel methods such as the hierarchical matrix approximation are often a necessity to obtain an efficient solution. We propose a novel hierarchical preconditioner that attempts to minimize the condition number of the preconditioned system. The method is based on approximating the low-rank off-diagonal blocks in a norm adapted to the hierarchical structure. Our analysis shows that the new preconditioner effectively maps both small and large eigenvalues of the system approximately to 1. Finally through numerical experiments, we illustrate the effectiveness of the new designed scheme which outperforms more classical techniques based on regular SVD to approximate the off-diagonal blocks and SVD with filtering.

This work is a joint effort between Professor Darve's group at Stanford and the Inria HiePACS team within **FASTLA**. More details on this work can be found in [41].

7.2.8. *Robust coarse spaces for abstract Schwarz preconditioners via generalized eigenproblems*

The solution of large sparse linear systems is one of the most important kernels in many numerical simulations. The domain decomposition methods (DDM) community has developed many efficient and robust solvers in the last decades. While many of these solvers fall in Abstract Schwarz (AS) framework, their robustness has often been demonstrated on a case-by-case basis. In this paper, we propose a bound for the condition number of all deflated AS methods provided that the coarse grid consists of the assembly of local components that contain the kernel of some local operators. We show that classical results from the literature on particular instances of AS methods can be retrieved from this bound. We then show that such a coarse grid correction can be explicitly obtained algebraically via generalized eigenproblems, leading to a condition number independent of the number of domains. This result can be readily applied to retrieve the bounds previously obtained via generalized eigenproblems in the particular cases of Neumann-Neumann (NN), additive Schwarz (aS) and optimized Robin but also generalizes them when applied with approximate local solvers. Interestingly, the proposed methodology turns out to be a comparison of the considered particular AS method with generalized versions of both NN and aS for tackling the lower and upper part of the spectrum, respectively. We furthermore show that the application of the considered grid corrections in an additive fashion is robust in the aS case although it is not robust for AS methods in general. In particular, the proposed framework allows for ensuring the robustness of the aS method applied on the Schur complement (aS/S), either with deflation or additively, and with the freedom of relying on an approximate local Schur complement, leading to a new powerful and versatile substructuring method. Numerical experiments illustrate these statements.

More details on this work can be found in [45]

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

7.3.1. Task-based fast multipole method

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. The successful ports of fully-featured numerical libraries on several recent runtime system proposals have shown, indeed, the benefit of task-based parallelism models in terms of performance portability on complex platforms. 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 a 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 as part of 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. We illustrate our discussion with the *ScalFMM* library, which implements state-of-the-art fast multipole method (FMM) algorithms, that we have deeply re-designed 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. We furthermore propose extensions to the OPENMP 4 standard and show how they can enhance FMM performance. To assess our statement, we have implemented this support within the KLANG-OMP source-to-source compiler that translates OPENMP directives into calls to the *StarPU* task-based runtime system. This study, [38] shows that we can take advantage of the advanced capabilities of a fully-featured runtime system without resorting to a specific, native runtime port, hence bridging the gap between the OPENMP standard and the very high performance that was so far reserved to expert-only runtime system APIs.

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. In [40] we illustrate our discussion with the *ScalFMM* FMM library and the *StarPU* runtime system.

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

7.4.1. Load Balancing for Coupled Simulations

In the field of scientific computing, the load balancing is an important step conditioning the performance of parallel programs. The goal is to distribute the computational load across multiple processors in order to minimize the execution time. This is a well-known problem that is unfortunately NP-hard. The most common approach to solve it is based on graph or hypergraph partitioning method, using mature and efficient software tools such as Metis, Zoltan or Scotch. Nowadays, numerical simulation are becoming more and more complex, mixing several models and codes to represent different physics or scales. Here, the key idea

is to reuse available legacy codes through a coupling framework instead of merging them into a standalone application. For instance, the simulation of the earth's climate system typically involves at least 4 codes for atmosphere, ocean, land surface and sea-ice . Combining such different codes are still a challenge to reach high performance and scalability. 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 codes and not only for each standalone code, as it is 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, that can dramatically decrease the overall performance. Therefore, one argues that it is required to model the coupling itself in order to ensure a good scalability, especially when running on tens of thousands of processors. In this work, we develop new algorithms to perform a coupling-aware partitioning of the whole application.

Surprisingly, we observe in our experiments that our proposed algorithms do not highly degrade the global edgcut for either component and thus the internal communication among processors of the same component is still minimized. This is not the case for the *Multiconst* method especially as the number of processors increases. Regarding the coupled simulation for the real application AVTP-AVBP (provided by Cerfacs), we noticed that one may carefully decide the parameters of the co-partitioning algorithms in order not to increase the global edgcut. More precisely, the number of processors assigned in the coupling interface is an important factor that needs to be determined based on the geometry of the problem and the ratio of the coupling interface compared to the entire domain. Again, we remark that our work on co-partitioning is still theoretical and further investigation should be conducted with different geometries and more coupled simulations that are more or less coupling-intensive.

This work corresponds to the PhD of Maria Predari, defended on December 9th 2016.

7.5. Application Domains

7.5.1. Material physics

7.5.1.1. Molecular vibrational spectroscopy

Quantum chemistry eigenvalue problem is a big challenge in recent research. Here we are interested in solving eigenvalue problems coming from the molecular vibrational analysis. These problems are challenging because the size of the vibrational Hamiltonian matrix to be diagonalized is exponentially increasing with the size of the molecule we are studying. So, for molecules bigger than 10 atoms the actual existent algorithms suffer from a curse of dimensionality or computational time.

A new variational algorithm called adaptive vibrational configuration interaction (A-VCI) intended for the resolution of the vibrational Schrödinger equation was developed. The main advantage of this approach is to efficiently reduce the dimension of the active space generated into the configuration interaction (CI) process. Here, we assume that the Hamiltonian writes as a sum of products of operators. This adaptive algorithm was developed with the use of three correlated conditions i.e. a suitable starting space ; a criterion for convergence, and a procedure to expand the approximate space. The velocity of the algorithm was increased with the use of a posteriori error estimator (residue) to select the most relevant direction to increase the space. Two examples have been selected for benchmark. In the case of H₂CO, we mainly study the performance of A-VCI algorithm: comparison with the variation-perturbation method, choice of the initial space, residual contributions. For CH₃CN, we compare the A-VCI results with a computed reference spectrum using the same potential energy surface and for an active space reduced by about 90 %. This work was published in [9].

7.5.1.2. Dislocations

We have focused on the improvements in collision detection in the Optidis Code. Junction formation mechanisms are essential to characterize material behavior such as strain hardening and irradiation effects. Dislocations junctions appear when dislocation segments collide with each other, therefore, reliable collision detection algorithms must be used to detect and handle junction formations. Collision detection is also a very costly operation in dislocation dynamics simulations, and performance must be carefully optimized to allow massive simulations.

During the first year of this PhD thesis, new collision algorithms have been implemented for the Dislocation Dynamics code OptiDis. The aim was to allow fast and accurate collision detection between dislocation segments using hierarchical methods. The complexity to solve the N-body collision problem can be reduced to $O(N)$ using spatial partitioning; computation can be accelerated using fast-reject techniques, and OpenMP parallelism. Finally, new collision handling algorithms for dislocations have been implemented to increase the reliability of the simulation.

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

7.5.2.1. Interior penalty discontinuous Galerkin method for coupled elasto-acoustic media

We introduce a high order interior penalty discontinuous Galerkin scheme for the numerical solution of wave propagation in coupled elasto-acoustic media. A displacement formulation is used, which allows for the solution of the acoustic and elastic wave equations within the same framework. Weakly imposing the correct transmission condition is achieved by the derivation of adapted numerical fluxes. This generalization does not weaken the discontinuous Galerkin method, thus *hp*-non-conforming meshes are supported. Interior penalty discontinuous Galerkin methods were originally developed for scalar equations. Therefore, we propose an optimized formulation for vectorial equations more suited than the straightforward standard transposition. We prove consistency and stability of the proposed schemes. To study the numerical accuracy and convergence, we achieve a classic plane wave analysis. Finally, we show the relevance of our method on numerical experiments.

More details on this work can be found in [47].

7.5.2.2. 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 had been carried on in the framework of Fabien Rozar's PhD in collaboration with **CEA-IRFM** (defended in November 2015) and is continued in the PhD of Nicolas Bouzat funded by IPL **C2S@Exa**. The scientific objectives of this new work will be 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.

7.5.2.3. 3D aerodynamics for unsteady problems with bodies in relative motion

The first part of our research work concerning the parallel aerodynamic code FLUSEPA has been to design an operational MPI+OpenMP version based on a domain decomposition. We achieved an efficient parallel version up to 400 cores and the temporal adaptive method used without bodies in relative motion has been tested successfully for complex 3D take-off blast wave computations. Moreover, an asynchronous strategy for computing bodies in relative motion and mesh intersections has been developed and has been used for 3D stage separation cases. This first version is the current industrial production version of FLUSEPA for Airbus Safran Launchers.

However, this intermediate version shows synchronization problems for the aerodynamic solver due to the time integration used. To tackle this issue, a task-based version over the runtime system **StarPU** has been developed and evaluated. Task generation functions have been designed in order to maximize asynchronism during execution while respecting the data pattern access of the code. This led to the re-factorization of the FLUSEPA computation kernels. It's clearly a successful proof of concept as a task-based version is now

available for the aerodynamic solver and for both shared and distributed memory. It uses three parallelism levels : MPI processes between sub-domains, **StarPU** workers in shared memory (for each sub-domain) themselves running OpenMP parallel tasks. This version has been validated for large 3D take-off blast wave computations (80 millions of cells) and is much more efficient than the previous MPI+OpenMP version: we achieve a gain in computation time equal to 70 % for 320 cores and to 50 % for 560 cores. The next step will consist in extending the task-based version to the motion and intersection operations. This work has been carried on in the framework of Jean-Marie Couteyen's PhD (defended in September 2016) in collaboration with Airbus Safran Launchers ([2], [17]).

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Grants with Industry

Airbus Safran Launchers research and development contract:

- Design of a parallel version of the FLUSEPA software (Jean-Marie Couteyen (PhD); Pierre Brenner, Jean Roman).

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. Inria Project Lab

9.1.1.1. C2S@Exa - Computer and Computational Sciences at Exascale

Since January 2013, the team is participating to the **C2S@EXA** Inria Project Lab (IPL). This national initiative aims at the development of numerical modeling methodologies that fully exploit the processing capabilities of modern massively parallel architectures in the context of a number of selected applications related to important scientific and technological challenges for the quality and the security of life in our society. At the current state of the art in technologies and methodologies, a multidisciplinary approach is required to overcome the challenges raised by the development of highly scalable numerical simulation software that can exploit computing platforms offering several hundreds of thousands of cores. Hence, the main objective of **C2S@EXA** is the establishment of a continuum of expertise in the computer science and numerical mathematics domains, by gathering researchers from Inria project-teams whose research and development activities are tightly linked to high performance computing issues in these domains. More precisely, this collaborative effort involves computer scientists that are experts of programming models, environments and tools for harnessing massively parallel systems, algorithmists that propose algorithms and contribute to generic libraries and core solvers in order to take benefit from all the parallelism levels with the main goal of optimal scaling on very large numbers of computing entities and, numerical mathematicians that are studying numerical schemes and scalable solvers for systems of partial differential equations in view of the simulation of very large-scale problems.

9.1.2. ANR

9.1.2.1. 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.

9.1.2.2. ANEMOS: Advanced Numeric for ELMs : Modeling and Optimized Schemes

Participants: Guillaume Latu, Pierre Ramet.

Grant: ANR-MN

Dates: 2012 – 2016

Partners: Univ. Nice, CEA/IRFM, CNRS/MDS.

Overview: The main goal of the project is to make a significant progress in understanding of active control methods of plasma edge MHD instabilities Edge Localized Modes (ELMs) which represent particular danger with respect to heat and particle loads for Plasma Facing Components (PFC) in ITER. The project is focused in particular on the numerical modelling study of such ELM control methods as Resonant Magnetic Perturbations (RMPs) and pellet ELM pacing both foreseen in ITER. The goals of the project are to improve understanding of the related physics and propose possible new strategies to improve effectiveness of ELM control techniques. The tool for the non-linear MHD modeling is the **JOREK** code which was essentially developed within previous ANR **ASTER**. **JOREK** will be largely developed within the present project to include corresponding new physical models in conjunction with new developments in mathematics and computer science strategy. The present project will put the non-linear MHD modeling of ELMs and ELM control on the solid ground theoretically, computationally, and applications-wise in order to progress in urgently needed solutions for ITER.

Regarding our contributions, the **JOREK** code is mainly composed of numerical computations on 3D data. The toroidal dimension of the tokamak is treated in Fourier space, while the poloidal plane is decomposed in Bezier patches. The numerical scheme used involves a direct solver on a large sparse matrix as a main computation of one time step. Two main costs are clearly identified: the assembly of the sparse matrix, and the direct factorization and solve of the system that includes communications between all processors. The efficient parallelization of **JOREK** is one of our main goals, to do so we will reconsider: data distribution, computation distribution or GMRES implementation. The quality of the sparse solver is also crucial, both in term of performance and accuracy. In the current release of **JOREK**, the memory scaling is not satisfactory to solve problems listed above, since at present as one increases the number of processes for a given problem size, the memory footprint on each process does not reduce as much as one can expect. In order to access finer meshes on available supercomputers, memory savings have to be done in the whole code. Another key point for improving parallelization is to carefully profile the application to understand the regions of the code that do not scale well. Depending on the timings obtained, strategies to diminish communication overheads will be evaluated and schemes that improve load balancing will be initiated. **JOREK** uses **PaStiX** sparse matrix library for matrix inversion. However, large number of toroidal harmonics and particular thin structures to resolve for realistic plasma parameters and ITER machine size still require more aggressive optimisation in numeric dealing with numerical stability, adaptive meshes etc. However many possible applications of **JOREK** code we proposed here which represent urgent ITER relevant issues related to ELM control by RMPs and pellets remain to be solved.

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

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

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.

9.1.2.4. *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, Nucléutudes.

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.

9.2.1.3. EXA2CT

Title: EXascale Algorithms and Advanced Computational Techniques

Programm: FP7

Duration: September 2013 - August 2016

Coordinator: IMEC

Partners:

Fraunhofer-Gesellschaft Zur Foerderung Der Angewandten Forschung E.V (Germany)

Interuniversitair Micro-Electronica Centrum Vzw (Belgium)

Intel Corporations (France)

Numerical Algorithms Group Ltd (United Kingdom)

T-Systems Solutions for Research (Germany)

Universiteit Antwerpen (Belgium)

Universita della Svizzera italiana (Switzerland)

Universite de Versaillesint-Quentin-En-Yvelines. (France)

Vysoka Skola Banska - Technicka Univerzita Ostrava (Czech Republic)

Inria contact: Luc Giraud

Numerical simulation is a crucial part of science and industry in Europe. The advancement of simulation as a discipline relies on increasingly compute intensive models that require more computational resources to run. This is the driver for the evolution to exascale. Due to limits in the increase in single processor performance, exascale machines will rely on massive parallelism on and off chip, with a complex hierarchy of resources. The large number of components and the machine complexity introduce severe problems for reliability and programmability. The former of these will require novel fault-aware algorithms and support software. In addition, the scale of the numerical models exacerbates the difficulties by making the use of more complex simulation algorithms necessary, for numerical stability reasons. A key example of this is increased reliance on solvers. Such solvers require global communication, which impacts scalability, and are often used with preconditioners, increasing complexity again. Unless there is a major rethink of the design of solver algorithms, their components and software structure, a large class of important numerical simulations will not scale beyond petascale. This in turn will hold back the development of European science and industry which will fail to reap the benefits from exascale. The EXA2CT project brings together experts at the cutting edge of the development of solvers, related algorithmic techniques, and HPC software architects for programming models and communication. It will take a revolutionary approach to exascale solvers and programming models, rather than the incremental approach of other projects. We will produce modular open source proto-applications that demonstrate the algorithms and programming techniques developed in the project, to help boot-strap the creation of genuine exascale codes.

9.3. International Initiatives

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

9.3.1.1. MORSE

Title: Matrices Over Runtime Systems @ Exascale

International Partner (Institution - Laboratory - Researcher):

KAUST Supercomputing Laboratory (United States) - KSL - Hatem Ltaief

Start year: 2011

See also: <http://icl.cs.utk.edu/morse/index.html>

The goal of Matrices Over Runtime Systems at Exascale (MORSE) project is to design dense and sparse linear algebra methods that achieve the fastest possible time to an accurate solution on large-scale multicore systems with GPU accelerators, using all the processing power that future high end systems can make available. To develop software that will perform well on petascale and exascale systems with thousands of nodes and millions of cores, several daunting challenges have to be overcome, both by the numerical linear algebra and the runtime system communities. By designing a research framework for describing linear algebra algorithms at a high level of abstraction, the MORSE team will enable the strong collaboration between research groups in linear algebra, runtime systems and scheduling needed to develop methods and libraries that fully benefit from the potential of future large-scale machines. Our project will take a pioneering step in the effort to bridge the immense software gap that has opened up in front of the High-Performance Computing (HPC) community.

9.3.1.2. FASTLA

Title: Fast and Scalable Hierarchical Algorithms for Computational Linear Algebra

International Partner (Institution - Laboratory - Researcher):

Stanford University (USA) - Institute for Computational and Mathematical Engineering -
Eric Darve

Start year: 2015

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. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

PMAA'16 The 9th International Workshop on Parallel Matrix Algorithms and Applications (PMAA16) was held in Bordeaux (France) and run from July 6 to 8, 2016.

10.1.1.2. Member of the Organizing Committees

The members of **HIEPACS** were involved in the **PMAA'16** organizing committees.

10.1.2. Scientific Events Selection

10.1.2.1. Member of the Conference Program Committees

IEEE PDP'16 (J. Roman), IPDPS'16 (L. Giraud), HiPC'16 (A. Guermouche), PDCN'16 (L. Giraud), PDSEC'16 (O. Coulaud, L. Giraud), SC'16 (E. Agullo, L. Giraud).

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

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

10.1.3.2. 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.

10.1.4. Invited Talks

L. Giraud, "Hard faults and soft-errors: possible numerical remedies in linear algebra solvers", VecPar'16, Porto.

10.1.5. Scientific Expertise

- E. 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.
- P. 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.

10.1.6. Research Administration

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 20h 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 second year of Embedded Electronic Systems option 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 will be held next year in Ostrava - Czech Republic.

10.2.2. Supervision

PhD in progress : Pierre Blanchard; Fast hierarchical algorithms for the low-rank approximation of dense matrices and applications ; O. Coulaud, E. Darve.

PhD in progress : Nicolas Bouzat; Fine grain algorithms and deployment methods for exascale plasma physic applications ; M.Mehrenberger, J.Roman, G. Latu (CEA Cadarache).

PhD : Jean-Marie Couteyen Carpaye; Contributions to the parallelization and the scalability of the FLUSEPA code; defended on September 19th; P. Brenner, J. Roman.

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 : Cyril Fournier; Task based programming for unstructured mesh calculations; L. Giraud, G. Stafelbach.

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

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

PhD : Maria Predari; Load balancing for parallel coupled simulations; defended on December 9th; A. Esnard, J. Roman.

10.2.3. Juries

- Okba Hamitou, "Efficient preconditioning method for the CARP-CG iterative solver for the solution of the frequency- domain visco-elastic wave equation", referee: Jan S. Hesthaven, Luc Giraud; Université de Grenoble, spécialité: mathématiques appliquées, 22 Décembre 2016.
- Jean-Charles Papin, "A Scheduling and Partitioning Model for Stencil-based Applications on Many-Core Devices", referee Jean-François Méhaut, Olivier Coulaud; Université Paris-Saclay préparée à l'École Normale Supérieure de Cachan spécialité: mathématiques appliquées, 8 Septembre 2016.

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [1] B. BRAMAS. *Optimization and parallelization of the boundary element method for the wave equation in time domain*, Université de Bordeaux, February 2016, <https://tel.archives-ouvertes.fr/tel-01306571>
- [2] J. M. COUTEYEN CARPAYE. *Contributions to the parallelization and the scalability of the FLUSEPA code*, Université de Bordeaux, September 2016, <https://tel.archives-ouvertes.fr/tel-01399952>
- [3] P. MARIA. *Load Balancing for Parallel Coupled Simulations*, Université de Bordeaux, 2016

Articles in International Peer-Reviewed Journals

- [4] E. AGULLO, P. R. AMESTOY, A. BUTTARI, A. GUERMOUCHE, J.-Y. L'EXCELLENT, F.-H. ROUET. *Robust memory-aware mappings for parallel multifrontal factorizations*, in "SIAM Journal on Scientific Computing", July 2016, vol. 38, n^o 3, 23 p. , <https://hal.inria.fr/hal-01334113>
- [5] E. AGULLO, B. BRAMAS, O. COULAUD, E. DARVE, M. MESSNER, T. TAKAHASHI. *Task-based FMM for heterogeneous architectures*, in "Concurrency and Computation: Practice and Experience", June 2016, vol. 28, n^o 9 [DOI : 10.1002/CPE.3723], <https://hal.inria.fr/hal-01359458>
- [6] 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>
- [7] E. AGULLO, L. GIRAUD, A. GUERMOUCHE, J. ROMAN, M. ZOUNON. *Numerical recovery strategies for parallel resilient Krylov linear solvers*, in "Numerical Linear Algebra with Applications", May 2016, <https://hal.inria.fr/hal-01323192>
- [8] 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>

- [9] R. GARNIER, M. ODUNLAMI, V. LE BRIS, D. BÉGUÉ, I. BARAILLE, O. COULAUD. *Adaptive vibrational configuration interaction (A-VCI): a posteriori error estimation to efficiently compute anharmonic IR spectra*, in "Journal of Chemical Physics", May 2016, vol. 144, n^o 20, <https://hal.inria.fr/hal-01310708>

Invited Conferences

- [10] E. AGULLO, S. COOLS, L. GIRAUD, A. MOREAU, P. SALAS, W. VANROOSE, E. F. YETKIN, M. ZOUNON. *Hard faults and soft errors: possible numerical remedies in linear algebra solvers*, in "VecPar - International meeting on High Performance Computing for Computational science", Porto, Portugal, June 2016, <https://hal.inria.fr/hal-01334675>

International Conferences with Proceedings

- [11] E. AGULLO, O. BEAUMONT, L. EYRAUD-DUBOIS, S. KUMAR. *Are Static Schedules so Bad ? A Case Study on Cholesky Factorization*, in "IEEE International Parallel & Distributed Processing Symposium (IPDPS 2016)", Chicago, IL, United States, IEEE, May 2016, <https://hal.inria.fr/hal-01223573>
- [12] E. AGULLO, G. BOSILCA, A. BUTTARI, A. GUERMOUCHE, F. LOPEZ. *Exploiting a Parametrized Task Graph model for the parallelization of a sparse direct multifrontal solver*, in "Euro-Par 2016: Parallel Processing Workshops", Grenoble, France, August 2016, <https://hal.archives-ouvertes.fr/hal-01337748>
- [13] E. AGULLO, L. GIRAUD, A. GUERMOUCHE, S. NAKOV, J. ROMAN. *Task-based Conjugate Gradient: from multi-GPU towards heterogeneous architectures*, in "HeteroPar'2016 workshop of Euro-Par", Grenoble, France, August 2016, <https://hal.inria.fr/hal-01334734>
- [14] E. AGULLO, L. GIRAUD, S. NAKOV. *Task-based sparse hybrid linear solver for distributed memory heterogeneous architectures*, in "HeteroPar'2016 workshop of Euro-Par", Grenoble, France, August 2016, <https://hal.inria.fr/hal-01334738>
- [15] T. COJEAN, A. GUERMOUCHE, A. HUGO, R. NAMYST, P.-A. WACRENIER. *Resource aggregation for task-based Cholesky Factorization on top of heterogeneous machines*, in "HeteroPar'2016 workshop of Euro-Par", Grenoble, France, August 2016, <https://hal.inria.fr/hal-01181135>
- [16] T. COJEAN, A. GUERMOUCHE, A.-E. HUGO, R. NAMYST, P.-A. WACRENIER. *Resource aggregation in task-based applications over accelerator-based multicore machines*, in "HeteroPar'2016 workshop of Euro-Par", Grenoble, France, August 2016, <https://hal.inria.fr/hal-01355385>
- [17] J. M. COUTEYEN CARPAYE, J. ROMAN, P. BRENNER. *Towards an efficient Task-based Parallelization over a Runtime System of an Explicit Finite-Volume CFD Code with Adaptive Time Stepping*, in "International Parallel and Distributed Processing Symposium", Chicago, IL, United States, PDSEC'2016 workshop of IPDPS, May 2016, 10 p. [DOI : 10.1109/IPDPSW.2016.125], <https://hal.inria.fr/hal-01324331>
- [18] V. GARCIA PINTO, L. STANISIC, A. LEGRAND, L. MELLO SCHNORR, S. THIBAUT, V. DANJEAN. *Analyzing Dynamic Task-Based Applications on Hybrid Platforms: An Agile Scripting Approach*, in "3rd Workshop on Visual Performance Analysis (VPA)", Salt Lake City, United States, November 2016, Held in conjunction with SC16, <https://hal.inria.fr/hal-01353962>
- [19] M. PREDARI, A. ESNARD. *A k-way Greedy Graph Partitioning with Initial Fixed Vertices for Parallel Applications*, in "24th Euromicro International Conference on Parallel, Distributed, and Network-Based

Processing", heraklion, Greece, Parallel, Distributed, and Network-Based Processing (PDP 2016), February 2016, 8 p. , <https://hal.inria.fr/hal-01277392>

Conferences without Proceedings

- [20] E. AGULLO, S. COOLS, L. GIRAUD, W. VANROOSE, E. F. YETKIN. *Soft errors in PCG: detection and correction*, in "SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP 2016)", Paris, France, April 2016, <https://hal.inria.fr/hal-01301240>
- [21] E. AGULLO, L. GIRAUD, S. NAKOV. *Combining Software Pipelining with Numerical Pipelining in the Conjugate Gradient Algorithm*, in "SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP 2016)", Paris, France, April 2016, <https://hal.inria.fr/hal-01301237>
- [22] E. AGULLO, L. GIRAUD, P. SALAS, M. ZOUNON. *Numerical fault tolerant strategies for resilient parallel eigensolvers*, in "IMA Conference on Numerical Linear Algebra and Optimization", Birmingham, United Kingdom, September 2016, <https://hal.inria.fr/hal-01334631>
- [23] E. AGULLO, M. KUHN, S. LANTERI, L. MOYA. *High order scalable HDG method fro frequency-domain electromagnetics*, in "Icosahom 2016 - International Conference on Spectral and High Order Methods", Rio de Janeiro, Brazil, June 2016, <https://hal.inria.fr/hal-01404669>
- [24] O. BEAUMONT, T. COJEAN, L. EYRAUD-DUBOIS, A. GUERMOUCHE, S. KUMAR. *Scheduling of Linear Algebra Kernels on Multiple Heterogeneous Resources*, in "International Conference on High Performance Computing, Data, and Analytics (HiPC 2016)", Hyderabad, India, December 2016, <https://hal.inria.fr/hal-01361992>
- [25] P. BLANCHARD, O. COULAUD, A. ETCHEVERRY, L. DUPUY, E. DARVE. *An Efficient Interpolation Based FMM for Dislocation Dynamics Simulations: Based on uniform grids and FFT acceleration*, in "Platform for Advanced Scientific Computing", Lausanne, Switzerland, USI and CSCS and EPFL, June 2016, <https://hal.archives-ouvertes.fr/hal-01334842>
- [26] L. BOILLOT, C. ROSSIGNON, G. BOSILCA, E. AGULLO, H. CALANDRA, H. BARUCQ, J. DIAZ. *Handling clusters with a task-based runtime system: application to Geophysics*, in "Rice - Oil & Gas HPC Workshop", HOUSTON, United States, March 2016, <https://hal.inria.fr/hal-01303373>
- [27] L. BOILLOT, C. ROSSIGNON, G. BOSILCA, E. AGULLO, H. CALANDRA. *Optimizing numerical simulations of elastodynamic wave propagation thanks to task-based parallel programming*, in "SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP 2016)", Paris, France, April 2016, <https://hal.inria.fr/hal-01303379>
- [28] M. FAVERGE, G. PICHON, P. RAMET. *Exploiting Kepler architecture in sparse direct solver with runtime systems*, in "9th International Workshop on Parallel Matrix Algorithms and Applications (PMAA'2016)", Bordeaux, France, July 2016, <https://hal.inria.fr/hal-01421372>
- [29] Y.-F. JING, E. AGULLO, B. CARPENTIERI, L. GIRAUD, T.-Z. HUANG. *Two New Block Krylov Methods for Linear Systems with Multiple Right-hand Sides*, in "IMA Conference on Numerical Linear Algebra and Optimization", Birmingham, United Kingdom, September 2016, <https://hal.inria.fr/hal-01334648>

- [30] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Exploiting H-Matrices in Sparse Direct Solvers*, in "SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP 2016)", Paris, France, April 2016, <https://hal.inria.fr/hal-01251812>
- [31] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *On the use of low rank approximations for sparse direct solvers*, in "SIAM Annual Meeting (AN'16)", Boston, United States, July 2016, <https://hal.inria.fr/hal-01421376>
- [32] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse Supernodal Solver Using Hierarchical Compression*, in "Workshop on Fast Direct Solvers", Purdue, United States, November 2016, <https://hal.inria.fr/hal-01421368>
- [33] 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>
- [34] 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>
- [35] 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 Parallel Processing for Scientific Computing (SIAM PP 2016)", Paris, France, April 2016, <https://hal.inria.fr/hal-01251808>
- [36] 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>
- [37] L. POIREL, E. AGULLO, L. GIRAUD. *Coarse Grid Correction for Algebraic Domain Decomposition Solvers*, in "ECCOMAS Congress 2016", Hersonissos, Greece, June 2016, <https://hal.inria.fr/hal-01355534>

Research Reports

- [38] E. AGULLO, O. AUMAGE, B. BRAMAS, O. COULAUD, S. PITOISET. *Bridging the gap between OpenMP 4.0 and native runtime systems for the fast multipole method*, Inria, March 2016, n^o RR-8953, 49 p. , <https://hal.inria.fr/hal-01372022>
- [39] 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*, Inria Bordeaux Sud-Ouest ; Bordeaux INP ; CNRS ; Université de Bordeaux ; CEA, June 2016, n^o RR-8927, 27 p. , <https://hal.inria.fr/hal-01332774>
- [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, October 2016, n^o RR-8970, 15 p. , <https://hal.inria.fr/hal-01387482>
- [41] E. AGULLO, E. DARVE, L. GIRAUD, Y. HARNESS. *Nearly optimal fast preconditioning of symmetric positive definite matrices*, Inria Bordeaux Sud-Ouest, November 2016, n^o RR-8984, 34 p. , <https://hal.inria.fr/hal-01403480>

- [42] E. AGULLO, L. GIRAUD, A. GUERMOUCHE, S. NAKOV, J. ROMAN. *Task-based Conjugate Gradient: from multi-GPU towards heterogeneous architectures*, Inria, May 2016, n° RR-8912, <https://hal.inria.fr/hal-01316982>
- [43] E. AGULLO, L. GIRAUD, S. NAKOV. *Task-based hybrid linear solver for distributed memory heterogeneous architectures*, Inria Bordeaux Sud-Ouest, May 2016, n° RR-8913, <https://hal.inria.fr/hal-01316783>
- [44] E. AGULLO, L. GIRAUD, S. NAKOV, J. ROMAN. *Hierarchical hybrid sparse linear solver for multicore platforms*, Inria Bordeaux, October 2016, n° RR-8960, 25 p. , <https://hal.inria.fr/hal-01379227>
- [45] E. AGULLO, L. GIRAUD, L. POIREL. *Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems*, Inria Bordeaux, November 2016, n° RR-8978, <https://hal.inria.fr/hal-01399203>
- [46] S. COOLS, E. F. YETKIN, E. AGULLO, L. GIRAUD, W. VANROOSE. *Analysis of rounding error accumulation in Conjugate Gradients to improve the maximal attainable accuracy of pipelined CG*, Inria Bordeaux Sud-Ouest, January 2016, n° RR-8849, <https://hal.inria.fr/hal-01262716>
- [47] Y. DUDOIT, L. GIRAUD, F. MILLOT, S. PERNET. *Interior penalty discontinuous Galerkin method for coupled elasto-acoustic media*, Inria Bordeaux Sud-Ouest, December 2016, n° RR-8986, <https://hal.inria.fr/hal-01406158>
- [48] M. FAVERGE, J. LANGOU, Y. ROBERT, J. DONGARRA. *Bidiagonalization with Parallel Tiled Algorithms*, Inria, October 2016, n° RR-8969, <https://hal.inria.fr/hal-01389232>
- [49] G. PICHON, M. FAVERGE, P. RAMET, J. ROMAN. *Reordering strategy for blocking optimization in sparse linear solvers*, Inria Bordeaux Sud-Ouest ; LaBRI - Laboratoire Bordelais de Recherche en Informatique ; Bordeaux INP ; Université de Bordeaux, February 2016, n° RR-8860, 26 p. , <https://hal.inria.fr/hal-01276746>
- [50] D. SUKKARI, H. LTAIEF, M. FAVERGE, D. KEYES. *Asynchronous Task-Based Polar Decomposition on Manycore Architectures*, KAUST, October 2016, <https://hal.inria.fr/hal-01387575>

Other Publications

- [51] P. BLANCHARD, O. COULAUD, E. DARVE, A. FRANC. *FMR: Fast randomized algorithms for covariance matrix computations*, June 2016, Platform for Advanced Scientific Computing (PASC), Poster, <https://hal.archives-ouvertes.fr/hal-01334747>
- [52] 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*, October 2016, working paper or preprint, <https://hal.inria.fr/hal-01386174>
- [53] T. COJEAN, A. GUERMOUCHE, A. A. HUGO, R. A. NAMYST, P.-A. WACRENIER. *Resource aggregation for task-based Cholesky Factorization on top of modern architectures*, November 2016, This paper is submitted for review to the Parallel Computing special issue for HCW and HeteroPar 16 workshops, <https://hal.inria.fr/hal-01409965>
- [54] M. PREDARI, A. ESNARD. *Graph partitioning techniques for load balancing of coupled simulations*, October 2016, SIAM Workshop on Combinatorial Scientific Computing , Poster, <https://hal.archives-ouvertes.fr/hal-01399392>