IN PARTNERSHIP WITH:
CNRS

Institut Polytechnique de Bordeaux

Université de Bordeaux

Activity Report 2018

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

Project-Team HIEPACS

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

Keywords:

Computer Science and Digital Science:

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

Other Research Topics and Application Domains:

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

1. Team, Visitors, External Collaborators

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, HDR]

Faculty Members

Aurélien Esnard [Univ de Bordeaux, Associate Professor]
Mathieu Faverge [Institut National Polytechnique de Bordeaux, Associate Professor]
Abdou Guermouche [Univ Bordeaux I, Associate Professor, HDR]
Jean Rene Poirier [INP Toulouse, Associate Professor, in delegation from Feb 2018 until July 2018, HDR]
Pierre Ramet [Univ de Bordeaux, Associate Professor, HDR]

Post-Doctoral Fellows

Oguz Kaya [Inria, until Aug 2018]
Ian Masliah [Inria, until Feb 2018]
Cristobal Samaniego Alvarado [Inria]
Nick Schenkels [Inria, from Dec 2018]

PhD Students

Arnaud Durocher [CEA, until Nov 2018]
Aurélien Falco [Inria]
Esragul Korkmaz [Inria, from Nov 2018]
Grégoire Pichon [Inria]
Louis Poirel [Inria, until Aug 2018]
Nicolas Venkovic [CERFACS, from Oct 2018]

Technical staff

Cyril Bordage [Inria, from May 2018]
Pierre Esterie [Inria, from Nov 2018]
Matthieu Kuhn [Inria, until May 2018]
Gilles Marait [Inria]

Interns
Tara Bahrami [Inria, from Jul 2018 until Aug 2018]
Ayoub Bellouch [Inria, from Jun 2018 until Aug 2018]
Vincent Bridonneau [Inria, from Jun 2018 until Aug 2018]
Hamza Hassoune [Inria, from Aug 2018 until Sep 2018]
Adnane Khattabi Riffi [Inria, from Mar 2018 until Aug 2018]
Claire Soyez Martin [Inria, from Jun 2018 until Aug 2018]

Administrative Assistant
Chrystel Plumejeau [Inria]

Visiting Scientist
Rocio Carratala Saez [Universidad Jaime I, from Jul 2018 until Sep 2018]

External Collaborator
Guillaume Sylvand [Airbus Central R & T]

2. Overall Objectives

2.1. Introduction

Over the last few decades, there have been innumerable science, engineering and societal breakthroughs enabled by the development of high performance computing (HPC) applications, algorithms and architectures. These powerful tools have provided researchers with the ability to computationally find efficient solutions for some of the most challenging scientific questions and problems in medicine and biology, climatology, nanotechnology, energy and environment. It is admitted today that *numerical simulation is the third pillar for the development of scientific discovery at the same level as theory and experimentation*. Numerous reports and papers also 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 conduced 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 prospectives in scalable parallel linear algebra techniques are addressed, ranging from dense direct, sparse direct, iterative and hybrid approaches for large linear systems. Then we consider research on N-body interaction computations based on efficient parallel fast multipole methods and finally, we adress 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 intiatives in various application domains in a co-design like framework. These research activities are conducted in a wider multi-disciplinary context with collegues 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/Conseil Régional Grande Aquitaine and with CEA, we have joint research efforts in a co-design framework enabling efficient and effective technological transfer towards industrial R&D. Furthermore, thanks to the past associate team FASTLA we contribute with world leading groups at Berkeley National Lab and Stanford University to the design of fast numerical solvers and their parallel implementations.

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

3. Research Program

3.1. Introduction

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

Currently, we have one major application (see Section 4.1) that is in material physics. We will collaborate to all steps of the design of the parallel simulation tool. More precisely, our applied mathematics skill will contribute to the modelling, our advanced numerical schemes will help in the design and efficient software implementation for very large parallel simulations. We also participate to a few co-design actions in close collaboration with some applicative groups. The objective of this activity is to instantiate our expertise in fields where they are critical for designing scalable simulation tools. We refer to Section 4.2 for a detailed description of these activities.

3.2. High-performance computing on next generation architectures

Participants: Emmanuel Agullo, Olivier Coulaud, Pierre Esterie, Mathieu Faverge, Luc Giraud, Abdou Guermouche, Gilles Marais, Ian Masliah, Grégoire Pichon, Louis Poirel, Pierre Ramet, Jean Roman, Nick Schenkels.

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

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

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

Another very important issue concerns high-performance computing using “heterogeneous” resources within a computational node. Indeed, with the deployment of the GPU and the use of more specific co-processors, it is important for our algorithms to efficiently exploit these new type of architectures. To adapt our algorithms and tools to these accelerators, we need to identify what can be done on the GPU for example and what cannot. Note that recent results in the field have shown the interest of using both regular cores and GPU to perform computations. Note also that in opposition to the case of the parallelism granularity needed by regular multicore architectures, GPU requires coarser grain parallelism. Thus, making both GPU and regular cores work all together will lead to two types of tasks in terms of granularity. This represents a challenging problem especially in terms of scheduling. From this perspective, we investigate new approaches for composing parallel applications within a runtime system for heterogeneous platforms.

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

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

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

3.3. High performance solvers for large linear algebra problems

Participants: Emmanuel Agullo, Olivier Coulaud, Mathieu Faverge, Aurélien Falco, Luc Giraud, Abdou Guermouche, Oguz Kaya, Esragul Korkmaz, Matthieu Kuhn, Gilles Marait, Grégoire Pichon, Louis Poirel, Jean Rene Poirier, Pierre Ramet, Jean Roman, Cristobal Samaniego Alvarado, Guillaume Sylvand, Nicolas Venkovic.

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

3.3.1. Parallel sparse direct solver

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

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

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

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

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

In that context, we continue our effort on the design of algebraic non-overlapping domain decomposition techniques that rely on the solution of a Schur complement system defined on the interface introduced by the partitioning of the adjacency graph of the sparse matrix associated with the linear system. Although it is better conditioned than the original system the Schur complement needs to be precondition 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 libray Fabulous originately 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 though 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, Pierre Esterie, Guillaume Sylvand.

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

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

3.4.1. Improvement of calculation efficiency

Nowadays, the high performance computing community is examining alternative architectures that address the limitations of modern cache-based designs. GPU (Graphics Processing Units) and the Cell processor have thus already been used in astrophysics and in molecular dynamics. The Fast 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 conduced 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 conduced on close connection with the activity described in Section 3.3. Following our earlier experience, we plan to first consider approximate inverse preconditionners that can efficiently exploit these data structures.

3.5. Load balancing algorithms for complex simulations

Participants: Cyril Bordage, Aurélien Esnard, 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 standalone application. There is typically one model per different scale or physics and each model is implemented by a parallel code.

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

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

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

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

3.5.1. Dynamic load-balancing with variable number of processors

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

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

3.5.2. Load balancing of coupled codes

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

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

3.5.3. Load balancing strategies for hybrid sparse linear solvers

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

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

4. Application Domains

4.1. Material physics

Participants: Olivier Coulaud, Pierre Esterie, Arnaud Durocher, Oguz Kaya.

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 acheive 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: Emmanuel Agullo, Nicolas Bouzat, Mathieu Faverge, Luc Giraud, Matthieu Kuhn, Gilles Marait, Louis Poirel, Pierre Ramet, Jean Roman.

4.2.1. High performance simulation for ITER tokamak

Scientific simulation for ITER tokamak modeling provides a natural bridge between theory and experimentation and is also an essential tool for understanding and predicting plasma behavior. Recent progresses in numerical simulation of fine-scale turbulence and in large-scale dynamics of magnetically confined plasma have been enabled by access to petascale supercomputers. These progresses would have been unreachable without new computational methods and adapted reduced models. In particular, the plasma science community has developed codes for which computer runtime scales quite well with the number of processors up to thousands cores. The research activities of HIEPACS concerning the international ITER challenge were involved in the Inria Project Lab C2S@EXA in collaboration with CEA-IRFM and are related to two complementary studies: a first one concerning the turbulence of plasma particles inside a tokamak (in the context of GYSELA code) and a second one concerning the MHD instability edge localized modes (in the context of JOREK code).

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

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

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

Parallel and numerically scalable hybrid solvers based on a fully algebraic coarse space correction have been theoretically studied and various advanced parallel implementations have been designed. Their parallel scalability has been investigated on large scale problems within the EOCOE project thanks to a close collaboration with the BSC and the integration of MaPHyS within the Alya software. The performance has also been assessed on PRACE Tier-0 machine within a PRACE Project Access through a collaboration with CERFACS and Laboratoire de Physique des Plasmas at École Polytechnique for the calculation of plasma propulsion.

5. Highlights of the Year

5.1. Highlights of the Year

The year 2018 was rich in regional, national and European calls for projects. This year, our success rate was quite high for the proposals we submitted; four of them went through: one ANR, namely SASHIMI, a major regional project, namely hpc-ecosystem benefiting many EPIs in Inria Bordeaux Sud-Ouest, and two H2020 projects, namely EoCoE and PRACE-6IP. These projects will provide new applications and collaborative frameworks to support our ongoing and future research and transfert activities.

6. New Software and Platforms

6.1. Chameleon

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

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

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

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

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

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

RELEASE FUNCTIONAL DESCRIPTION: Chameleon includes the following features:

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

- Participants: Cédric Castagnede, Samuel Thibault, Emmanuel Agullo, Florent Pruvost and Mathieu Faverge
- Partners: Innovative Computing Laboratory (ICL) - King Abdullha University of Science and Technology - University of Colorado Denver
- Contact: Emmanuel Agullo
- URL: https://gitlab.inria.fr/solverstack/chameleon

6.2. Fabulous

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

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

6.3. HIPS

Hierarchical Iterative Parallel Solver
KEYWORDS: Simulation - HPC - Parallel calculation - Hybrid direct iterative method

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

HIPS can be used as a standalone program that reads a sparse linear system from a file , it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, HIPS is a software library that provides several methods to build an efficient preconditioner in almost all situations.
FUNCTIONAL DESCRIPTION: HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

- Participants: Jérémie Gaidamour, Pascal Hénon and Yousef Saad
- Contact: Pierre Ramet
- URL: https://gitlab.inria.fr/solverstack/hips

6.4. MAPHYS

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

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

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

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

6.5. MetaPart

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

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

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

6.6. MPICPL

MPI CouPLing

KEYWORDS: MPI - Coupling software

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

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

6.7. OptiDis

KEYWORDS: Dislocation dynamics simulation - Fast multipole method - Large scale - Collision

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

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

6.8. PaStiX

Parallel Sparse matriX package

KEYWORDS: Linear algebra - High-performance calculation - Factorisation - Sparse Matrices - Linear Systems Solver
SCIENTIFIC DESCRIPTION: PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handle a combination of 1D and 2D block distributions. A dynamic scheduling can also be applied to take care of NUMA architectures while taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations.
FUNCTIONAL DESCRIPTION: PaStiX is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) methods. It can handle low-rank compression techniques to reduce the computation and the memory complexity. Numerical algorithms are implemented in single or double precision (real or complex) for LLt, LDLt and LU factorization with static pivoting (for non symmetric matrices having a symmetric pattern). The PaStiX library uses the graph partitioning and sparse matrix block ordering packages Scotch or Metis.

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

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

- Participants: Grégoire Pichon, Mathieu Faverge and Pierre Ramet
- Partners: Université Bordeaux 1 - INP Bordeaux
- Contact: Pierre Ramet
- URL: https://gitlab.inria.fr/solverstack/pastix

6.9. ScalFMM

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

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

the management of the octree and the parallelization of the method the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization.
FUNCTIONAL DESCRIPTION: Compute N-body interactions using the Fast Multipole Method for large number of objects

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

6.10. VITE

Visual Trace Explorer

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

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

6.11. PlaFRIM

Plateforme Fédérative pour la Recherche en Informatique et Mathématiques
KEYWORDS: High-Performance Computing - Hardware platform
FUNCTIONAL DESCRIPTION: PlaFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine.

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

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

7. New Results

7.1. High-performance computing on next generation architectures

7.1.1. Evaluation of dataflow programming models for electronic structure theory

Dataflow programming models have been growing in popularity as a means to de- liver a good balance between performance and portability in the post-petascale era. In this paper we evaluate different dataflow programming models for electronic struc- ture methods and compare them in terms of programmability, resource utilization, and scalability. In particular, we evaluate two programming paradigms for expressing scientific applications in a dataflow form: (1) explicit dataflow, where the dataflow is specified explicitly by the developer, and (2) implicit dataflow, where a task schedul- ing runtime derives the dataflow using per-task data-access information embedded in a serial program. We discuss our findings and present a thorough experimental analysis using methods from the NWChem quantum chemistry application as our case study, and OpenMP, StarpU and ParSEC as the task-based runtimes that enable the different forms of dataflow execution. Furthermore, we derive an abstract model to explore the limits of the different dataflow programming paradigms.

More information on these results can be found in [8]

7.1.2. On soft errors in the Conjugate Gradient method: sensitivity and robust numerical detection

The conjugate gradient (CG) method is the most widely used iterative scheme for the solution of large sparse systems of linear equations when the matrix is symmetric positive definite. Although more than sixty year old, it is still a serious candidate for extreme-scale computation on large computing platforms. On the technological side, the continuous shrinking of transistor geometry and the increasing complexity of these devices affect dramatically their sensitivity to natural radiation, and thus diminish their reliability. One of the most common effects produced by natural radiation is the single event upset which consists in a bit-flip in a memory cell producing unexpected results at application level. Consequently, the future computing facilities at extreme scale might be more prone to errors of any kind including bit-flip during calculation. These numerical and technological observations are the main motivations for this work, where we first investigate through extensive numerical experiments the sensitivity of CG to bit-flips in its main computationally intensive kernels, namely the matrix-vector product and the preconditioner application. We further propose numerical criteria to detect the occurrence of such faults; we assess their robustness through extensive numerical experiments.

More information on these results can be found in [16].

7.1.3. Energy analysis of a solver stack for frequency-domain electromagnetics

High-performance computing (HPC) aims at developing models and simulations for applications in numerous scientific fields. Yet, the energy consumption of these HPC facilities currently limits their size and performance, and consequently the size of the tackled problems. The complexity of the HPC software stacks and their various optimizations makes it difficult to finely understand the energy consumption of scientific applications. To highlight this difficulty on a concrete use-case, we perform an energy and power analysis of a software stack for the simulation of frequency-domain electromagnetic wave propagation. This solver stack combines a high order finite element discretization framework of the system of three-dimensional frequency-domain Maxwell equations with an algebraic hybrid iterative-direct sparse linear solver. This analysis is conducted on the KNL-based PRACE-PCP system. Our results illustrate the difficulty in predicting how to trade energy and runtime.

More information on these results can be found in [18].

7.1.4. A compiler front-end for OpenMP's variants

OpenMP 5.0 introduced the concept of *variant*: a directive which can be used to indicate that a function is a variant of another existing *base function*, in a specific context (eg: `foo_gpu_nvidia` could be declared as a variant of `foo`, but only when executing on specific NVIDIA hardware).

In the context of PRACE-5IP, in collaboration with the Inria STORM team, we want to leverage this construct to be able to take advantage of the StarPU heterogeneous scheduler through the interoperability layer between OpenMP and StarPU. We started this work by implementing the necessary changes in the Clang front-end to support OpenMP's *variant*. We have assessed this support in the `Chameleon` dense linear algebra package. Indeed, `Chameleon` relies on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a runtime system. Additionally to the `quark`, `PaRSEC` and `StarPU` support, we have implemented an OpenMP support in `Chameleon`. The originality of the proposed approach is that this OpenMP support can either rely on a native OpenMP runtime system or indirectly use the above mentioned OpenMP-StarPU back-end. We are currently assessing the approach on multicore homogeneous machines, the next step being heterogeneous architectures.

7.2. High performance solvers for large linear algebra problems

7.2.1. Partitioning and communication strategies for sparse non-negative matrix factorization

Non-negative matrix factorization (NMF), the problem of finding two non-negative low-rank factors whose product approximates an input matrix, is a useful tool for many data mining and scientific applications such as topic modeling in text mining and blind source separation in microscopy. In this paper, we focus on scaling algorithms for NMF to very large sparse datasets and massively parallel machines by employing effective algorithms, communication patterns, and partitioning schemes that leverage the sparsity of the input matrix. In the case of machine learning workflow, the computations after SpMM must deal with dense matrices, as Sparse-Dense matrix multiplication will result in a dense matrix. Hence, the partitioning strategy considering only SpMM will result in a huge imbalance in the overall workflow especially on computations after SpMM and in this specific case of NMF on non-negative least squares computations. Towards this, we consider two previous works developed for related problems, one that uses a fine-grained partitioning strategy using a point-to-point communication pattern and on that uses a checkerboard partitioning strategy using a collective-based communication pattern. We show that a combination of the previous approaches balances the demands of the various computations within NMF algorithms and achieves high efficiency and scalability. From the experiments, we could see that our proposed algorithm communicates atleast 4x less than the collective and achieves upto 100x speed up over the baseline FAUN on real world datasets. Our algorithm was experimented in two different super computing platforms and we could scale up to 32000 processors on Bluegene/Q.

More information on these results can be found in [21].

7.2.2. Low-rank factorizations in data sparse hierarchical algorithms for preconditioning Symmetric positive definite matrices

We consider the problem of choosing low-rank factorizations in data sparse ma- trix approximations for preconditioning large scale symmetric positive definite matrices. These approximations are memory efficient schemes that rely on hierarchical matrix partitioning and compression of certain sub-blocks of the matrix. Typically, these matrix approximations can be constructed very fast, and their matrix product can be applied rapidly as well. The common prac- tice is to express the compressed sub-blocks by low-rank factorizations, and the main contribution of this work is the numerical and spectral analysis of SPD preconditioning schemes represented by 2× 2 block matrices, whose off-diagonal sub-blocks are low-rank approximations of the original matrix off-diagonal sub-blocks. We propose an optimal choice of low-rank approximations which minimizes the condition number of the preconditioned system, and demonstrate that the analysis can be applied to the class of hierarchically off-diagonal low-rank matrix approximations. Spec- tral estimates that take into account the error propagation through levels of the hierarchy which quantify the impact of the choice of low-rank compression on the global condition number are provided. The numerical results indicate that the properties of the preconditioning scheme using proper low-rank compression are superior to employing standard choices for low-rank compression. A major goal of this work is to provide an insight into how proper reweighted prior to low-rank compression influences the condition number for a simple case, which would lead to an extended analysis for more general and more efficient hierarchical matrix approximation techniques.

More information on these results can be found in [5].

7.2.3. Analyzing the effect of local rounding error propagation on the maximal attainable accuracy of the pipelined Conjugate Gradient method

Pipelined Krylov subspace methods typically offer improved strong scaling on par- allel HPC hardware compared to standard Krylov subspace methods for large and sparse linear systems. In pipelined methods the traditional synchronization bottleneck is mitigated by overlap- ping time-consuming global communications with useful computations. However, to achieve this communication-hiding strategy, pipelined methods introduce additional recurrence relations for a number of auxiliary variables that are required to update the approximate solution. This paper aims at studying the influence of local rounding errors that are introduced by the additional recurrences in the pipelined Conjugate Gradient (CG) method. Specifically, we analyze the impact of local round-off effects on the attainable accuracy of the pipelined CG algorithm and compare it to the traditional CG method. Furthermore, we estimate the gap between the true residual and the recursively computed residual used in the algorithm. Based on this estimate we suggest an automated residual replacement strategy to reduce the loss of attainable accuracy on the final iterative solution. The resulting pipelined CG method with residual replacement improves the maximal attainable accuracy of pipelined CG while maintaining the efficient parallel performance of the pipelined method. This conclusion is substantiated by numerical results for a variety of benchmark problems.

More information on these results can be found in [7].

7.2.4. *Sparse supernodal solver using block low-rank compression: Design, performance and analysis*

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

These contributions have been published in International Journal of Computational Science and Engineering (JoCS) [9].

7.2.5. *Supernodes ordering to enhance Block Low-Rank compression in a sparse direct solver*

Solving sparse linear systems appears in many scientific applications, and sparse direct linear solvers are widely used for their robustness. Still, both time and memory complexities limit the use of direct methods to solve larger problems. In order to tackle this problem, low-rank compression techniques have been introduced in direct solvers to compress large dense blocks appearing in the symbolic factorization. In this paper, we consider the Block Low-Rank compression (BLR) format and adress the problem of clustering unknowns that come from separators issued from the nested dissection process. We show that methods considering only intra-separators connectivity (i.e., k-way or recursive bisection) as well as methods managing only interaction between separators have some limitations. We propose a new strategy that considers interactions between a separator and its children to pre-select some interactions while reducing the number of off-diagonal blocks in the symbolic structure. We demonstrate how this new method enhances the BLR strategies in the sparse direct supernodal solver PaStiX.

These contributions have been submitted in SIAM Journal on Matrix Analysis and Applications (SIMAX) [22].

7.3. Parallel Low-Rank Linear System and Eigenvalue Solvers Using Tensor Decompositions

At the core of numerical simulations for scientific computing applications, one typically needs to solve an equation either in the form of a linear system (Ax = b) or an eigenvalue problem (Ax = x) to determine the course of the simulation. A major breakthrough in this solution step is exploiting the inherent low-rank structure in the problem; an idea stemming from the observation that particles in the same spatial locality exhibit similar interactions with others in a distant cluster/region. This property has been exploited in many contexts such as fast multipole methods (FMM) and hierarchical matrices (H-matrices) in applications ranging from n-body simulations to electromagnetics, which amount to numerically compressing the matrix in order to reduce computational and memory costs. Recent theory along this direction involves employing tensor decomposition to quantize the matrix in the form of a tensor (through logical restructuring/reshaping) and use tensor decomposition to approximate it with a controllable global error. Once the matrix and vectors are compressed this way, one can similarly use the compressed tensor to carry out matrix-vector operations with significantly better compression rate than the H-matrix approach.

Despite these major recent breakthroughs in the theory and application of tensor-based methods, addressing large-scale real-world problems with these methods requires immense computational power, which necessitates highly optimized parallel algorithms and implementations. To this end, we have initiated the development of a tensor-based linear system and eigenvalue solver library called Celeste++(C++ library for Efficient low-rank Linear and Eigenvalue Solvers using Tensor decomposition) providing a complete framework for expressing a problem in tensor form, then effectuating all matrix-vector operations under this compressed form with tremendous computational and memory efficiency. The fruits of our preliminary studies led two project submissions at the national scale (ANR JCJC and CNRS PEPS JCJC, currently under evaluation) and one Severo Ochoa Mobility Grant for a collaboration visit to Barcelona Supercomputing Center (BSC). We also supervised an internship on the application of tensor solvers in the context of electromagnetic applications with very promising results for future work.

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

7.4.1. StarPart Redesign

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

This work was initially integrated into the StarPart platform. The necessary extension of our algorithms to parallel & distributed (increasingly dynamic) versions has led to a complete redesign of StarPart, which has been the focus of our efforts this year. The StarPart framework provides the necessary building blocks to develop new graph algorithms in the context of HPC, such as those we are targeting. The strength of StarPart lies in the fact that it is a light runtime system applied to the issue of "graph computing". It provides a unified data model and a uniform programming interface that allows easy access to a dozen partitioning libraries, including Metis, Scotch, Zoltan, etc. Thus, it is possible, for example, to load a mesh from an industrial test case provided by our partners (or an academic graph collection as DIMACS'10) and to easily compare the results for the different partitioners integrated in StarPart.

7.5. Application Domains

7.5.1. Material physics

7.5.1.1. EigenSolver

The adaptive vibrational configuration interaction algorithm has been introduced as a new eigennvalues method for large dimension problem. It is based on the construction of nested bases for the discretization of the Hamiltonian operator according to a theoretical criterion that ensures the convergence of the method. It efficiently reduce the dimension of the set of basis functions used and then we are able solve vibrationnal eigenvalue problem up to the dimension 15 (7 atoms). This year we have worked on three main areas. First, we extend our shared memory parallelization to distributed memory using the message exchange paradigm. This new version should allow us to process larger systems quickly. To target the eigenvalues relevant for chemists, i. eigenvalues with an intensity. This requires calculating the scalar product between the smallest eigenvalues and the dipole moment applied to an eigenvector to evaluate its intensity. In addition, to get closer to the experimental values, we introduced the Coriolis operator into the Hamiltonian. A paper is being written on these last two points.

7.5.1.2. Dislocation

We have focused on the improvements of the parallel collision detection and the data structure in the OPTIDIS code [11].

- The new collision detection algorithm to reliably handle junction formation for Dislocation Dynamics using hybrid OpenMP + MPI parallelism has been developed. The enhanced precision and reliability of this new algorithm allows the use of larger time-steps for faster simulations. Hierarchical methods for collision detection, as well as hybrid parallelism are also used to improve performance;
- A new distributed data structure has been developed to enhance the reliability and modularity of OPTIDIS. The new data structure provides an interface to modify safely and reliably the distributed dislocation mesh in order to enforce data consistency across all computation nodes. This interface also improves code modularity allowing the study of data layout performance without modifying the algorithms.

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

7.5.2.1. A geometric view of biodiversity: scaling to metagenomics

We have designed a new efficient dimensionality reduction algorithm in order to investigate new ways of accurately characterizing the biodiversity, namely from a geometric point of view, scaling with large environmental sets produced by NGS (∼105 sequences). The approach is based on Multidimensional Scaling (MDS) that allows for mapping items on a set of n points into a low dimensional euclidean space given the set of pairwise distances. We compute all pairwise distances between reads in a given sample, run MDS on the distance matrix, and analyze the projection on first axis, by visualization tools. We have circumvented the quadratic complexity of computing pairwise distances by implementing it on a hyperparallel computer (Turing, a Blue Gene Q), and the cubic complexity of the spectral decomposition by implementing a dense random projection based algorithm. We have applied this data analysis scheme on a set of 10^5 reads, which are amplicons of a diatom environmental sample from Lake Geneva. Analyzing the shape of the point cloud paves the way for a geometric analysis of biodiversity, and for accurately building OTUs (Operational Taxonomic Units), when the data set is too large for implementing unsupervised, hierarchical, high-dimensional clustering.

More information on these results can be found in [19].

7.5.2.2. High performance simulation for ITER tokamak

Concerning the GYSELA global non-linear electrostatic code, a critical problem is the design of a more efficient parallel gyro-average operator for the deployment of very large (future) GYSELA runs. The main unknown of the computation is a distribution function that represents either the density of the guiding centers, either the density of the particles in a tokamak. The switch between these two representations is done thanks to the gyro-average operator. In the previous version of GYSELA, the computation of this operator was achieved thanks to a Padé approximation. In order to improve the precision of the gyro-averaging, a new parallel version based on an Hermite interpolation has been done (in collaboration with the Inria TONUS project-team and IPP Garching). The integration of this new implementation of the gyro-average operator has been done in GYSELA and the parallel benchmarks have been successful. This work is carried on in the framework of the PhD of Nicolas Bouzat (funded by IPL C2S@Exa) co-advised with Michel Mehrenberger from TONUS project-team and in collaboration with Guillaume Latu from CEA-IRFM. The scientific objectives of this work are first to consolidate the parallel version of this gyro-average operator, in particular by designing a scalable MPI+OpenMP parallel version and by 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. This has been done by using a new data distribution for a new irregular mesh in order to take into account the complex geometries of modern tokamak reactors. All these contributions have been validated on a new object-oriented proptotype of GYSELA which uses a task based programming model. The PhD thesis of Nicolas Bouzat has been defended on December 17, 2018.

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

7.5.2.3. Numerical and parallel scalable hybrid solvers in large scale calculations

Numerically scalable hybrid solvers based on a fully algebraic coarse space correction have been theoretically studied within the PhD thesis of Louis Poirel defended on November 28, 2018. Some of the proposed numerical schemes have been integrated in the MaPHyS parallel package. In particular, multiple parallel strategies have been designed and their parallel efficiencies were assessed in two large application codes. The first one is Alya developed at BSC, that is a high performance computational mechanics code to solve coupled multi-physics / multi-scale problems, which are mostly coming from engineering applications. This activity was carried out in the framework of the EoCoE project. Thye second large code is AVIP jointly developed by CERFACS and Laboratoire de Physique des Plasmas at E´cole Polytechnique for the calculation of plasma propulsion. For this latter code, part of the parallel experiments were conduced on a PRACE Tier-0 machine within a PRACE Project Access.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. HPC Scalable Ecosystem

Participants: Emmanuel Agullo, Olivier Coulaud, Aurélien Esnard, Mathieu Faverge, Luc Giraud, Abdou Guermouche, Pierre Ramet.

Grant: Regional council
Dates: 2018 – 2020
Partners: EPIs REALOPT, STORM from Inria Bordeaux Sud-Ouest, CEA-CESTA and l'Institut pluridisciplinaire de recherche sur l'environnement et les matériaux (IPREM) .
Overview:

Numerical simulation is today integrated in all cycles of scientific design and studies, whether academic or industrial, to predict or understand the behavior of complex phenomena often coupled or multi-physical. The quality of the prediction requires having precise and adapted models, but also to have computation algorithms efficiently implemented on computers with architectures in permanent evolution. Given the ever increasing size and sophistication of simulations implemented, the use of parallel computing on computers with up to several hundred thousand computing cores and consuming / generating massive volumes of data becomes unavoidable; this domain corresponds to what is now called High Performance Computing (HPC). On the other hand, the digitization of many processes and the proliferation of connected objects of all kinds generate ever-increasing volumes of data that contain multiple valuable information; these can only be highlighted through sophisticated treatments; we are talking about Big Data. The intrinsic complexity of these digital treatments requires a holistic approach with collaborations of multidisciplinary teams capable of mastering all the scientific skills required for each component of this chain of expertise.

To have a real impact on scientific progress and advances, these skills must include the efficient management of the massive number of compute nodes using programming paradigms with a high level of expressiveness, exploiting high-performance communications layers, effective management for intensive I / O, efficient scheduling mechanisms on platforms with a large number of computing units and massive I / O volumes, innovative and powerful numerical methods for analyzing volumes of data produced and efficient algorithms that can be integrated into applications representing recognized scientific challenges with high societal and economic impacts. The project we propose aims to consider each of these links in a consistent, coherent and consolidated way.

For this purpose, we propose to develop a unified Execution Support (SE) for large-scale numerical simulation and the processing of large volumes of data. We identified four Application Challenges (DA) identified by the Nouvelle-Aquitaine region that we propose to carry over this unified support. We will finally develop four Methodological Challenges (CM) to evaluate the impact of the project. This project will make a significant contribution to the emerging synergy on the convergence between two yet relatively distinct domains, namely High Performance Computing (HPC) and the processing, management of large masses of data (Big Data); this project is therefore clearly part of the emerging field of High Performance Data Analytics (HPDA).

8.2. National Initiatives

8.2.1. ANR

8.2.1.1. SASHIMI: Sparse Direct Solver using Hierarchical Matrices

Participants: Mathieu Faverge, Pierre Ramet, Aurelien Esnard.

Grant: ANR-18-CE46-0006
Dates: 2018 – 2022
Overview: Nowadays, the number of computational cores in supercomputers has grown largely to a few millions. However, the amount of memory available has not followed this trend, and the memory per core ratio is decreasing quickly with the advent of accelerators. To face this problem, the SaSHiMi project wants to tackle the memory consumption of linear solver libraries used by many major simulation applications by using low-rank compression techniques. In particular, the direct solvers which offer the most robust solution to strategy but suffer from their memory cost. The project will especially investigate the super-nodal approaches for which low-rank compression techniques have been less studied despite the attraction of their large parallelism and their lower memory cost than for the multi-frontal approaches. The results will be integrated in the PaStiX solver that supports distributed and heterogeneous architectures.

8.2.1.2. 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.

8.2.2. FUI

8.2.2.1. ICARUS: Intensive Calculation for AeRo and automotive engines Unsteady Simulations

Participants: Cyril Bordage, Aurelien Esnard.

Grant: FUI-22
Dates: 2016-2019
Partners: SAFRAN, SIEMENS, IFPEN, ONERA, DISTENE, CENAERO, GDTECH, Inria, CORIA, CERFACS.
Overview: Large Eddy Simulation (LES) is an increasingly attractive unsteady modelling approach for modelling reactive turbulent flows due to the constant development of massively parallel supercomputers. It can provide open and robust design tools that allow access to new concepts (technological breakthroughs) or a global consideration of a structure (currently processed locally). The mastery of this method is therefore a major competitive lever for industry. However, it is currently constrained by its access and implementation costs in an industrial context. The ICARUS project aims to significantly reduce them (costs and deadlines) by bringing together major industrial and research players to work on the entire high-fidelity LES computing process by:

- increasing the performance of existing reference tools (for 3D codes: AVBP, Yales2, ARGO) both in the field of code coupling and code/machine matching;
- developing methodologies and networking tools for the LES;
- adapting the ergonomics of these tools to the industrial world: interfaces, data management, code interoperability and integrated chains;
- validating this work on existing demonstrators, representative of the aeronautics and automotive industries.

8.3. European Initiatives

8.3.1. FP7 & H2020 Projects

8.3.1.1. 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.

8.3.1.2. PRACE 5IP

Title: PRACE Fifth Implementation Phase (PRACE-5IP) project

Duration: January 2017 - April 2019

Partners: see the following url

Inria contact: Stéphane Lanteri

The mission of PRACE (Partnership for Advanced Computing in Europe) is to enable high-impact scientific discovery and engineering research and development across all disciplines to enhance European competitiveness for the benefit of society. PRACE seeks to realise this mission by offering world class computing and data management resources and services through a peer review process. PRACE also seeks to strengthen the European users of HPC in industry through various initiatives. PRACE has a strong interest in improving energy efficiency of computing systems and reducing their environmental impact.

The objectives of PRACE-5IP are to build on and seamlessly continue the successes of PRACE and start new innovative and collaborative activities proposed by the consortium. These include:

- assisting the transition to PRACE2 including ananalysis of TransNational Access;
- strengthening the internationally recognised PRACE brand;
- continuing and extend advanced training which so far provided more than 18 800 person-training days;
- preparing strategies and best practices towards Exascale computing;
- coordinating and enhancing the operation of the multi-tier HPC systems and services;
- supporting users to exploit massively parallel systems and novel architectures.

8.4. International Initiatives

8.4.1. Inria International Labs

In the framework of the Joint Laboratory for Extreme Scale Computing (JLESC) within a collaboration between Inria and Argonne national laboratory an new joint project studies how lossy compression can be monitored by Krylov solvers to significantly reduce the memory footprint when solving very-large sparse linear systems. The resulting solvers will alleviate the I/O penalty paid when running large calculations using either check-point mechanisms to address resiliency or out-of-core techniques to solve huge problems. For the solution of large linear systems of the form $Ax = b$ where $A \in \mathbb{R}^{n \times n}$, x and $b \in \mathbb{R}^n$, Krylov subspace methods are among the most commonly used iterative solvers; they are further extended to cope with extreme scale computing as one can integrate features such as communication hidden in their variants referred to as pipelined Krylov solvers [7]. On the one hand, the Krylov subspace methods such as GMRES allow some inexactness when computing the orthonormal search basis; more precisely theoretical results [24], [25] show that the matrix-vector product involved in the construction of the new search directions can be more and more inexact when the convergence towards the solution takes place. An inexact scheme of that form writes into a generalized Arnoldi equality

$$[(A + E_1)v_1, \cdots, (A + E_k)v_k] = [v_1, \cdots, v_k, v_{k+1}]\overline{H}_k, \qquad (1)$$

where the theory gives a bound on $\|E_k\|$ that depends on the residual norm $\|b - Ax_k\|$ at step k, where x_k is the k^{th} iterate. Such a result has a major interest in applications where the matrix is not formed explicitly, e.g., in the fast mutipole (FMM) or domain decomposition (DDM) methods context, where this allows one to drastically reduce the computational effort.

One the other hand, novel agnostic lossy data compression techniques are studied to reduce the I/O footprint of large applications that have to store snapshots of the calculation, for a posteriori analysis, because they implement out-of-core calculation or for checkpointing data for resilience. Those lossy compression techniques allow for precise control on the error introduced by the compressor to ensure that the stored data are still meaningful for the considered application. In the context of the Krylov method, the basis $V_{k+1} = [v_1, \cdots, v_k, v_{k+1}]$ represents the most demanding data in terms of memory footprint, so that, in a fault-tolerant or out-of-core context, storing it in a lossy form would allow for a tremendous saving.

The objective of this work, developped within the post-doc of N. Schenkels, is to dynamically control the compression error of V_{k+1} to comply with the inexact Krylov theory. The main difficulty is to translate the known theoretical inexactness on E_k into a suited lossy compression mechanism for v_k with loss $\|\delta v_k\|$.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. Member of the Organizing Committees

L. Giraud was member of the SIAM Conference of parallel processing in scientific computing, March 2018, Tokyo, Japan and the 10th International Workshop on Parallel Matrix Algorithms and Applications PMAA'18, June 2018, Zurich, Switzerland.

9.1.1.2. Chair of Conference Program Committees

A. Guermouche was the vice-chair of the Architechture track for the 25th IEEE International conference on High Performance Computing, Data, and Analytics HIPC'18, December 2018, Bengaluru, India.

9.1.1.3. Member of the Conference Program Committees

HiPC'18 (A. Guermouche, L. Giraud), ICPP'18 (E. Agullo), IEEE PDP'18 (J. Roman), IPDPS'18 (O. Coulaud), PDSEC'18 (O. Coulaud, M. Faverge, L. Giraud), COMPAS'18 (P. Ramet), SBAC-PAD'18 (M. Faverge).

9.1.2. Journal

9.1.2.1. Member of the Editorial Boards

- E. Agullo and L. Giraud were guest editor of the special issue of Parallel Computing dedicated to PMAA'16 [4].
- L. Giraud is member of the editorial board of the SIAM Journal on Scientific Computing (SISC) and SIAM Journal on Matrix Analysis and Applications (SIMAX).

9.1.2.2. Reviewer - Reviewing Activities

The members of the HIEPACS project have performed reviewing for the following list of journals: Computing and Fluid, International Journal of Antennas and Propagation, Parallel Computing, SIAM J. Matrix Analysis and Applications, SIAM J. Scientific Computing, Journal of Parallel and Distributed Computing, IEEE Transactions on Parallel and Distributed Systems, ACM Transactions on Mathematical Software, ACM Computational and Mathematical Methods, International Journal of High Performance Computing Applications, Journal Of Computational Science.

The members of the HIEPACS project have performed reviewing for the following list of conferences (additionally to PC): Europar'18, IPDPS'19, SC'18.

9.1.3. Scientific Expertise

- Luc Giraud is member of the board on Modelisation, Simulation and data analysis of the Competitiveness Cluster for Aeronautics, Space and Embedded Systems. He also acted as an expert for The Israel Science Foundation, on the Individual Research Grants and for the Czech Science Foundation, the main public funding agency in the Czech Republic supporting all areas of basic research.
- Pierre Ramet is "Scientific Expert" at the CEA-DAM CESTA since Oct. 2015.
- Jean Roman is member of the “Scientific Board” of the CEA-DAM. As representative of Inria, he is member of the board of ETP4HPC (European Technology Platform for High Performance Computing), of the French Information Group for PRACE, of the French Working Group for EuroHPC, of the Technical Group of GENCI and of the Scientific Advisory Board of the Maison de la Simulation.

9.1.4. Research Administration

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

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Undergraduate level/Licence

- A. Esnard: System programming 36h, Computer architecture 40h, Network 23h at Bordeaux University.
- M. Faverge: Programming environment 26h, Numerical algorithmic 40h, C projects 25h at Bordeaux INP (ENSEIRB-MatMeca).
- A. Guermouche: System programming 36h at Bordeaux University.
- P. Ramet: System programming 24h, Databases 32h, Object programming 48h, Distributed programming 32h, Cryptography 32h at Bordeaux University, and Numerical algorithmic 40h at Bordeaux INP (ENSEIRB-Matmeca).

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 72h, Load balancing and scheduling 13h at Bordeaux INP (ENSEIRB-MatMeca).
  He is also in charge of the master 2 internship for the Computer Science department at Bordeaux INP (ENSEIRB-MatMeca).
- L. Giraud: Introduction to intensive computing and related programming tools 20h, INSA Toulouse; Introduction to high performance computing and applications 20h, ISAE; On mathematical tools for numerical simulations 10h, ENSEEIHT Toulouse; Parallel sparse linear algebra 11h at Bordeaux INP (ENSEIRB-MatMeca).
- A. Guermouche: Network management 92h, Network security 64h, Operating system 24h at Bordeaux University.
- P. Ramet: Load balancing and scheduling 13h at Bordeaux INP (ENSEIRB-MatMeca).
- 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.

9.2.2. Supervision

PhD: Nicolas Bouzat; Fine grain algorithms and numerical schemes for exascale simulations of turbulent plasmas ; M.Mehrenberger (TONUS project-team), J. Roman, G. Latu (CEA-IRFM); defended on December 17, 2018; jury members: N. Crouseilles (referee, Inria Rennes Bretagne Atlantique), Ph. Helluy (Université Strasbourg), G. Latu (CEA Cadarache), M. Mehrenberger (Université Aix-Marseille), R. Namyst (referee, Université Bordeaux), S. Salmon (Université Reims).

PhD: 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); defended on December 19, 2018; jury members: D. Barthou (Bordeaux INP), M. Blétry (Université Paris XII), L. Dupuy (CEA, Saclay), M. Fivel (referee, CNRS, Grenoble), J.F. Méhaud (referee, Université de Grenoble).

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

PhD in progress: Esragul Korkmaz; Solveurs creux direct et matrices hierarchiques; M. Faverge, P. Ramet.

PhD: Grégoire Pichon; Utilisation de techniques de compression $\mathcal{H}$-matrices pour solveur direct creux parallèle dans le cadre des applications FEM; M. Faverge, P. Ramet; defended on November 29, 2018; jury members: A. Buttari (CNRS, Toulouse), D. Goudin (CEA-CESTA, Le Barp), G. Kubické (DGA, Rennes), S. Lanteri (Inria, Sophia Antipolis), E. Ng (Lawrence Berkeley Nat. Lab., Berkeley), F. Pellegrini (Université de Bordeaux).

PhD in progress: Louis Poirel; Algebraic coarse space correction for parallel hybrid solvers; E. Agullo, L. Giraud; defended on November 28, 2018; jury members: B. Cuenot (CERFACS, Toulouse), M. Gander (referee, Université de Genève). M. Heroux (referee, Sandia Nat. Lab.), A. Legrand (CNRS, Grenoble), F.X. Roux (ONERA, UPMC Paris), P. Tallec (referee, Ecole Polytechnique).

9.2.3. Juries

- Aloïs Bissuel, “Résolution des équations de Navier-Stokes linéarisées pour l’aéroélasticité, l’optimisation de forme et l’aéroacoustique", referees: V. Dolean, R. Abgrall, president: L. Giraud, Université Paris-Saclay à l’Ecole polytechnique, spécialité: mathématiques appliquées, 22 Janvier 2018.
- Eemeho Edorh, “Incremental algorithms for long range interactions", referees: M. Bolden, O. Coulaud, Université Grenoble Alpes, spécialité: mathématiques et informatique, 2 Octobre 2018.
- Vinicius Garcia Pinto, “Performance Analysis Strategies for Task-based Applications on Hybrid Platforms", referees: G. Cavalheiro, B. Mohr, P. Navaux, N. Maillard, A. Legrand reviewers: A. Goldman, G. Thomas Université Grenoble Alpes, spécialité: mathématiques et informatique, et Universidade Federal do Rio Grande do Sul, 30 Octobre 2018.
- Guillaume Latu, HDR, “Contribution à la simulation haute-performance et aux méthodes de calcul très extensibles", referees: R. Abgrall, F. Desprez, R. Namyst, reviewers: S. Genaud, J. Roman, E. Sonnendrücker, Université Strasbourg, spécialité: informatique et calcul scientifique, 18 Mai 2018.
- Gilles Moreau, “On the solution phase of direct methods for sparse linear systems with multiple right-hand sides", referees: P. Amestoy, J. Erhel, L. Grigori, J.-Y. L’Excellent, reviewers: J. Gilbert, P. Ramet, ENS Lyon, spécialité: informatique, 10 Decembre 2018.

9.3. Popularization

9.3.1. Interventions

During the 10th anniversary of the Inria Bordeaux Sud-Ouest centre and the open day, scientific popularisation materials of the HiePACS team’s research work were presented to the attendees.

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] N. BOUZAT. *Fine grain algorithms and numerical schemes for exascale simulations of turbulent plasmas*, Université de strasbourg, December 2018, https://tel.archives-ouvertes.fr/tel-01975275

[2] A. DUROCHER. *Large scale dislocation dynamics simulation : performance and reliability on parallel and distributed architectures.*, Université de Bordeaux, 2018

[3] L. POIREL. *Algebraic domain decomposition methods for hybrid (direct/iterative) solvers*, Université de Bordeaux, November 2018

Articles in International Peer-Reviewed Journals

[4] E. AGULLO, P. ARBENZ, L. GIRAUD, O. SCHENK. *Guest editorial: Special issue on parallel matrix algorithms and applications (PMAA'16)*, in "Parallel Computing", May 2018, vol. 74, pp. 1 - 2 [*DOI :* 10.1016/J.PARCO.2018.01.003], https://hal.inria.fr/hal-01927721

[5] E. AGULLO, E. DARVE, L. GIRAUD, Y. HARNESS. *Low-Rank Factorizations in Data Sparse Hierarchical Algorithms for Preconditioning Symmetric Positive Definite Matrices*, in "SIAM Journal on Matrix Analysis and Applications", October 2018, vol. 39, n^o 4, pp. 1701-1725, https://hal.inria.fr/hal-01940053

[6] T. COJEAN, A. GUERMOUCHE, A. HUGO, R. NAMYST, P.-A. WACRENIER. *Resource aggregation for task-based Cholesky Factorization on top of modern architectures*, in "Parallel Computing", October 2018, This paper is submitted for review to the Parallel Computing special issue for HCW and HeteroPar 16 workshops, https://hal.inria.fr/hal-01957086

[7] S. COOLS, E. F. YETKIN, E. AGULLO, L. GIRAUD, W. VANROOSE. *Analyzing the Effect of Local Rounding Error Propagation on the Maximal Attainable Accuracy of the Pipelined Conjugate Gradient Method*, in "SIAM Journal on Matrix Analysis and Applications", March 2018, vol. 39, n^o 1, pp. 426 - 450 [*DOI :* 10.1137/17M1117872], https://hal.inria.fr/hal-01753411

[8] H. JAGODE, A. DANALIS, R. HOQUE, M. FAVERGE, J. DONGARRA. *Evaluation of Dataflow Programming Models for Electronic Structure Theory*, in "Concurrency and Computation: Practice and Experience", 2018, 23 p. , https://hal.inria.fr/hal-01725804

[9] G. PICHON, E. DARVE, M. FAVERGE, P. RAMET, J. ROMAN. *Sparse supernodal solver using block low-rank compression: Design, performance and analysis*, in "International Journal of Computational Science and Engineering", July 2018, vol. 27, pp. 255 - 270 [*DOI :* 10.1016/J.JOCS.2018.06.007], https://hal.inria.fr/hal-01824275

International Conferences with Proceedings

[10] A. FALCO, E. AGULLO, L. GIRAUD, G. SYLVAND. *Hierarchical Symbolic Factorization for Sparse Matrices*, in "Sparse Days 2018", Toulouse, France, September 2018, https://hal.inria.fr/hal-01999895

Conferences without Proceedings

[11] A. Durocher, L. Dupuy, O. Coulaud. *Dislocation Dynamics Simulation : Large scale simulations with Numodis*, in "Journées scientifiques de la DANS", Saclay, France, May 2018, https://hal.archives-ouvertes.fr/hal-01893794

[12] M. Faverge, G. Pichon, P. Ramet. *Exploiting Parameterized Task-graph in Sparse Direct Solvers*, in "SIAM Conference on Computational Science and Engineering (CSE19)", Spokane, United States, February 2019, https://hal.inria.fr/hal-01956963

[13] G. Pichon, E. Darve, M. Faverge, P. Ramet, J. Roman. *Supernodes ordering to enhance Block Low-Rank compression in sparse direct solvers*, in "PMAA 2018 - 10th International Workshop on Parallel Matrix Algorithms and Applications", Zurich, Switzerland, June 2018, https://hal.inria.fr/hal-01956960

[14] G. Pichon, E. Darve, M. Faverge, P. Ramet, J. Roman. *Utilisation de la compression Block Low-Rank pour accélérer un solveur direct creux supernodal*, in "COMPAS 2018 - Conférence d'informatique en Parallélisme, Architecture et Système", Toulouse, France, July 2018, https://hal.inria.fr/hal-01956959

[15] G. Pichon, E. Darve, M. Faverge, P. Ramet, J. Roman. *Block Low-rank Algebraic Clustering for Sparse Direct Solvers*, in "SIAM Conference on Computational Science and Engineering (CSE19)", Spokane, United States, February 2019, https://hal.inria.fr/hal-01956962

Research Reports

[16] E. Agullo, S. Cools, E. Fatih-Yetkin, L. Giraud, W. Vanroose. *On soft errors in the Conjugate Gradient method: sensitivity and robust numerical detection*, Inria Bordeaux Sud-Ouest, November 2018, n^{o} RR-9226, https://hal.inria.fr/hal-01929738

[17] E. Agullo, E. Darve, L. Giraud, Y. Harness. *Low-rank Factorizations in Data Sparse Hierarchical Algorithms for Preconditioning Symmetric Positive Definite Matrices*, Inria Bordeaux Sud-Ouest, August 2018, n^{o} RR-9200, https://hal.inria.fr/hal-01856399

[18] E. Agullo, L. Giraud, S. Lanteri, G. Marait, A.-C. Orgerie, L. Poirel. *Energy analysis of a solver stack for frequency-domain electromagnetics*, Inria Bordeaux Sud-Ouest, December 2018, n^{o} RR-9240, https://hal.inria.fr/hal-01962629

[19] P. Blanchard, P. Chaumeil, J.-M. Frigerio, F. Rimet, F. Salin, S. Thérond, O. Coulaud, A. Franc. *A geometric view of Biodiversity: scaling to metagenomics*, Inria ; INRA, January 2018, n^{o} RR-9144, pp. 1-16, https://arxiv.org/abs/1803.02272 , https://hal.inria.fr/hal-01685711

[20] A. Franc, P. Blanchard, O. Coulaud. *Nonlinear Mapping and Distance Geometry*, Inria Bordeaux Sud-Ouest, September 2018, n^{o} RR-9210, 14 p. , https://arxiv.org/abs/1810.08661 , https://hal.inria.fr/hal-01897104

[21] O. Kaya, R. Kannan, G. Ballard. *Partitioning and Communication Strategies for Sparse Non-negative Matrix Factorization*, Inria Bordeaux Sud-Ouest, July 2018, n^{o} RR-9198, https://hal.inria.fr/hal-01849084

[22] G. Pichon, E. Darve, M. Faverge, P. Ramet, J. Roman. *Supernodes ordering to enhance Block Low-Rank compression in sparse direct solvers*, Inria Bordeaux Sud-Ouest, December 2018, n$^{\text{o}}$ RR-9238, pp. 1-31, https://hal.inria.fr/hal-01961675

Other Publications

[23] G. Pichon, M. Faverge, P. Ramet, J. Roman. *Utilisation de la compression low-rank pour réduire la complexité du solveur PaStiX*, October 2018, JCAD 2018 - Journées Calcul et Données, https://hal.inria.fr/hal-01956928

References in notes

[24] L. Giraud, S. Gratton, J. Langou. *Convergence in backward error of relaxed GMRES*, in "SIAM J. Scientific Computing", 2007, vol. 29, n$^{\text{o}}$ 2, pp. 710-728

[25] V. Simoncini, D. B. Szyld. *Theory of Inexact Krylov Subspace Methods and Applications to Scientific Computing*, in "SIAM J. Scientific Computing", 2003, vol. 25, pp. 454-477