RESEARCH CENTRE

Inria Center at the University of Bordeaux

IN PARTNERSHIP WITH:

Université de Bordeaux, CNRS, Institut Polytechnique de Bordeaux

2022 ACTIVITY REPORT

Project-Team HIEPACS

High-End Parallel Algorithms for Challenging Numerical Simulations

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

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed and High Performance Computing

Contents

Project-Team HIEPACS

Creation of the Project-Team: 2010 January 01

Keywords

Computer sciences and digital sciences

A1.1.4. – High performance computing

A1.1.5. – Exascale

A1.1.9. – Fault tolerant systems

A6.2.5. – Numerical Linear Algebra

A6.2.7. – High performance computing

A7.1. – Algorithms

A8.1. – Discrete mathematics, combinatorics

A8.2. – Optimization

A9.2. – Machine learning

A9.7. – AI algorithmics

Other research topics and application domains

B3.3.1. – Earth and subsoil

B4.2.2. – Fusion

B9.5.1. – Computer science

B9.5.2. – Mathematics

1 Team members, visitors, external collaborators

Research Scientists

- Olivier Beaumont [Team leader, INRIA, Senior Researcher, HDR]
- Lionel Eyraud Dubois [INRIA, Researcher]
- Yulia Gusak [INRIA, Starting Research Position, from Jul 2022]

Faculty Members

- Aurélien Esnard [UNIV BORDEAUX, Associate Professor]
- Mathieu Faverge [BORDEAUX INP, Associate Professor]
- Abdou Guermouche [UNIV BORDEAUX, Associate Professor, HDR]
- Pierre Ramet [UNIV BORDEAUX, Associate Professor, HDR]

Post-Doctoral Fellow

- Esragul Korkmaz [INRIA, from Oct 2022]

PhD Students

- Abel Calluaud [CEA, from Nov 2022]
- Jean Francois David [INRIA]
- Esragul Korkmaz [INRIA, until Aug 2022]
- Aboul-Karim Mohamed El Maarouf [IFPEN]
- Clément Richefort [CEA]
- Mathieu Verite [INRIA]
- Xunyi Zhao [INRIA]

Technical Staff

- Ahmed Abdourahman Mahamoud [Inria (plan de relance with Atos), Engineer, from Oct 2022]
- Marc Sergent [ATOS, Engineer, from Oct 2022]

Interns and Apprentices

- Alycia Lisito [INRIA, from Feb 2022]
- Brieuc Nicolas [INRIA, from Jun 2022 until Aug 2022]

Administrative Assistant

- Catherine Cattaert-Megrat [INRIA, from Jul 2022]

2 Overall objectives

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

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

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

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

The methodological part of HIEPACS covers several topics. First, we address generic studies concerning massively parallel computing, the design of high-end performance algorithms and software to be executed on future extreme scale platforms. Next, several research prospectives in scalable parallel linear algebra techniques are addressed, ranging from dense direct, sparse direct, iterative and hybrid approaches for large linear systems. We are also interested in the general problem of minimizing memory consumption and data movements, by changing algorithms and possibly performing extra computations, in particular in the context of Deep Neural Networks. Then we consider research on N-body interaction computations based on efficient parallel fast multipole methods and finally, we address research tracks related to the algorithmic challenges for complex code couplings in multiscale/multiphysic simulations.

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

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

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

3 Research program

3.1 Introduction

The methodological component of HIEPACS concerns the expertise for the design as well as the efficient and scalable implementation of highly parallel numerical algorithms to perform frontier simulations. In order to address these computational challenges a hierarchical organization of the research is considered. In this bottom-up approach, we first consider in Section 3.2 generic topics concerning high performance computational science. The activities described in this section are transversal to the overall project and their outcome will support all the other research activities at various levels in order to ensure the parallel scalability of the algorithms. The aim of this activity is not to study general purpose solution but rather to address these problems in close relation with specialists of the field in order to adapt and tune advanced approaches in our algorithmic designs. The next activity, described in Section 3.3, is related to the study of parallel linear algebra techniques that currently appear as promising approaches to tackle huge problems on extreme scale platforms. We highlight the linear problems (linear systems or eigenproblems) because they are in many large scale applications the main computational intensive numerical kernels and often the main performance bottleneck. 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 3.4.

3.2 High-performance computing on next generation architectures

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Mathieu Faverge, Abdou Guermouche, Pierre Ramet, Mathieu Vérite.

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 3.4).

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

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

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

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

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

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

3.3 High performance solvers for large linear algebra problems

Participants: Abel Calluaud, Mathieu Faverge, Abdou Guermouche, Esragul Korkmaz, Pierre Ramet, Clément Richefort.

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

3.3.1 Parallel sparse direct solvers

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

New supercomputers incorporate many microprocessors which are composed of one or many computational cores. These new architectures induce strongly hierarchical topologies. These are called NUMA architectures. In the context of distributed NUMA architectures, in collaboration with the Inria STORM team, we study optimization strategies to improve the scheduling of communications, threads and I/O. We have developed dynamic scheduling designed for NUMA architectures in the PaStiX solver. The data structures of the solver, as well as the patterns of communication have been modified to meet the needs of these architectures and dynamic scheduling. We are also interested in the dynamic adaptation of the computation 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 block low-rank compression techniques.

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

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

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

In that context, we continue our effort on the design of algebraic non-overlapping domain decomposition techniques that rely on the solution of a Schur complement system defined on the interface introduced by the partitioning of the adjacency graph of the sparse matrix associated with the linear system. Although it is better conditioned than the original system the Schur complement needs to be 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 library MaPHyS. This activity will be developed further developped in the H2020 EOCOE2 project. In addition to this numerical studies, advanced parallel implementation will be developed that will involve close collaborations between the hybrid and sparse direct activities.

3.4 Load balancing algorithms for complex simulations

Participants: Aurélien Esnard, Pierre Ramet.

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

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

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

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

There are several open algorithmic problems that we investigate in the HiePACS project-team. All these problems 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.

3.4.1 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.

3.4.2 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.

3.4.3 Load balancing strategies for hybrid sparse linear solvers

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

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

4 Application domains

4.1 High performance simulation for ITER tokamak

Participants: Pierre Ramet, Mathieu Faverge.

Scientific simulation for ITER tokamak modeling provides a natural bridge between theory and experimentation and is also an essential tool for understanding and predicting plasma behavior. Recent progresses in numerical simulation of fine-scale turbulence and in large-scale dynamics of magnetically confined plasma have been enabled by access to petascale supercomputers. These progresses would have been unreachable without new computational methods and adapted reduced models. In particular, the plasma science community has developed codes for which computer runtime scales quite well with the number of processors up to thousands cores. The research activities of HIEPACS concerning the international ITER challenge have started in the Inria Project Lab C2S@EXA in collaboration with CEA-IRFM and were related to two complementary studies: a first one concerning the turbulence of plasma particles inside a tokamak (in the context of GYSELA code) and a second one concerning the MHD instability edge localized modes (in the context of JOREK code). The activity concerning GYSELA was completed at the end of 2018.

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

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

4.2 Optimization for Deep Convolutional Neural Networks

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Julia Gusak.

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

5 Social and environmental responsibility

5.1 Impact of research results

As part of the project, we propose strategies to optimize the use of computational resources and energy consumption (in particular through the collaboration with Qarnot Computing described in Section ref-sec.pulse. Furthermore, memory-saving strategies in the context of training can also have a positive impact by avoiding the renewal of accelerators because their memory capacities become insufficient for new models.

6 Highlights of the year

- The article "*Symmetric Block-Cyclic Distribution: Fewer Communications Leads to Faster Dense Cholesky Factorization*" [13] was the Best Paper Nominee for the Algorithms Track at SuperComputing 2022.

- We organized the international SBAC-PAD conference in Bordeaux. Olivier Beaumont was General Chair of the conference and several members of the team (Lionel Eyraud Dubois, Mathieu Faverge, Abdou Guermouche, Catherine Cattaert-Megrat) were involved in the local organizing committee. There were 80 international participants at the conference, which was held in-person.

7 New software and platforms

7.1 New software

7.1.1 Chameleon

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

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

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

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

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

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

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

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

Release Contributions: Chameleon includes the following features:

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

URL: https://gitlab.inria.fr/solverstack/chameleon

Contact: Emmanuel Agullo

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

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

7.1.2 MPICPL

Name: MPI CouPLing

Keywords: MPI, Coupling software

Functional Description: MPICPL is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive 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 inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the mpicplrun tool.

URL: https://gitlab.inria.fr/esnard/mpicpl

Contact: Aurélien Esnard

Participant: Aurélien Esnard

7.1.3 PaStiX

Name: Parallel Sparse matriX package

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

Scientific Description: PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles 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 a hybrid MPI-thread implementation.

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

URL: https://gitlab.inria.fr/solverstack/pastix

Contact: Pierre Ramet

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

Partners: INP Bordeaux, Université de Bordeaux

7.1.4 pmtool

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

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

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

URL: https://gitlab.inria.fr/eyrauddu/pmtool

Publications: hal-01386174, hal-01878606

Contact: Lionel Eyraud Dubois

Participant: Lionel Eyraud Dubois

7.1.5 rotor

Name: Re-materializing Optimally with pyTORch

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

Scientific Description: This software implements in PyTorch a new activation checkpointing method which allows to significantly decrease memory usage when training Deep Neural Networks with the back-propagation algorithm. Similarly to checkpointing techniques coming from the literature on Automatic Differentiation, it consists in dynamically selecting the forward activations that are saved during the training phase, and then automatically recomputing missing activations from those previously recorded. We propose an original computation model that combines two types of activation savings: either only storing the layer inputs, or recording the complete history of operations that produced the outputs (this uses more memory, but requires fewer recomputations in the backward phase), and we provide in https://hal.inria.fr/hal-02352969 an algorithm to compute the optimal computation sequence for this model.

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

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

URL: https://gitlab.inria.fr/hiepacs/rotor

Publication: hal-02352969

Contact: Lionel Eyraud Dubois

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

7.1.6 VITE

Name: Visual Trace Explorer

Keywords: Visualization, Execution trace

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

URL: https://solverstack.gitlabpages.inria.fr/vite/

Contact: Mathieu Faverge

Participant: Mathieu Faverge

7.1.7 StarPart

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

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

URL: https://gitlab.inria.fr/metapart/starpart

Contact: Aurélien Esnard

Participant: Aurélien Esnard

8 New results

8.1 High-performance computing on next generation architectures

Participants: Mathieu Faverge, Abdou Guermouche, Aboul-Karim Mohamed El Maarouf, Hayfa Tayeb.

8.1.1 Task-based randomized singular value decomposition and multidimensional scaling

In [23], The multidimensional scaling (MDS) is an important and robust algorithm for representing individual cases of a dataset out of their respective dissimilarities. However, heuristics, possibly trading-off with robustness, are often preferred in practice due to the potentially prohibitive memory and computational costs of the MDS. The recent introduction of random projection techniques within the MDS allowed it to be become competitive on larger test cases. The goal of this manuscript is to propose a high-performance distributed-memory MDS based on random projection for processing data sets of even larger size (up to one million items). We propose a task-based design of the whole algorithm and we implement it within an efficient software stack including state-of-the-art numerical solvers, runtime systems and communication layers. The outcome is the ability to efficiently apply robust MDS to large data sets on modern supercomputers. We assess the resulting algorithm and software stack to the point cloud visualization for analyzing distances between sequencesin metabarcoding.

8.1.2 Programming Heterogeneous Architectures Using Hierarchical Tasks

In [16, 26, 18], Task-based systems have gained popularity because of their promise of exploiting the computational power of complex heterogeneous systems. A common programming model is the so-called Sequential Task Flow (STF) model, which, unfortunately, has the intrinsic limitation of supporting static task graphs only. This leads to potential submission overhead and to a static task graph which is not necessarily adapted for execution on heterogeneous systems. A standard approach is to find a trade-off between the granularity needed by accelerator devices and the one required by CPU cores to achieve performance. To address these problems, we extend the STF model in the StarPU runtime system to enable tasks subgraphs at runtime. We refer to these tasks as hierarchical tasks. This approach allows for a more dynamic task graph. This extended model combined with an automatic data manager allows to dynamically adapt the granularity to meet the optimal size of the targeted computing resource. We show that the hierarchical task model is correct and we provide an early evaluation on shared memory heterogeneous systems, using the Chameleon dense linear algebra library.

8.1.3 MulTreePrio: Scheduling task-based applications for heterogeneous computing systems

In [19], Effective scheduling is crucial for task-based applications to achieve high performance in heterogeneous computing systems. These applications are usually represented by directed acyclic graphs (DAG). In this paper, we present a dynamic scheduling technique for DAGs intending to minimize the overall completion time of the parallelized applications. We introduce MulTreePrio, a novel scheduler based on a set of balanced trees data structure. The assignment of tasks to available resources is done according to priority scores per task for each type of processing unit. These scores are computed through heuristics built according to a set of rules that our scheduler should fulfil. We simulate the scheduling on three DAGs coming from numerical kernels with different configurations and we compare its behavior with both dynamic schedulers and static scheduling techniques based on the critical path. We show the efficiency of our scheduler with an average speedup of x2 with respect to the dynamic scheduler and x0,99 compared to the critical path-based scheduler. MulTreePrio is promising and in future works, it will be integrated into a task-based runtime system and tested in real-life scenarios.

8.1.4 Combining reduction with synchronization barrier on multi-core processor

In [12, 28], with the rise of multi-core processors with a large number of cores, the need for shared memory reduction that performs efficiently on a large number of cores is more pressing. Efficient shared memory reduction on these multi-core processors will help shared memory programs be more efficient. In this article, we propose a reduction method combined with a barrier method that uses SIMD read/write instructions to combine barrier signaling and reduction value to minimize memory/cache traffic between cores, thereby reducing barrier latency. We compare different barriers and reduction methods on three multi-core processors and show that the proposed combining barrier/reduction methods are 4 and 3.5 times faster than respectively GCC11.1 and Intel 21.2 OpenMP 4.5 reduction.

8.1.5 Task-based parallel programming for scalable matrix product algorithms

In [11, 22], task-based programming models have succeeded in gaining the interest of the high-performance mathematical software community because they relieve part of the burden of developing and implementing distributed-memory parallel algorithms in an efficient and portable way.In increasingly larger, more heterogeneous clusters of computers, these models appear as a way to maintain and enhance more complex algorithms. However, task-based programming models lack the flexibility and the features that are necessary to express in an elegant and compact way scalable algorithms that rely on advanced communication patterns. We show that the Sequential Task Flow paradigm can be extended to write compact yet efficient and scalable routines for linear algebra computations. Although, this work focuses on dense General Matrix Multiplication, the proposed features enable the implementation of more complex algorithms. We describe the implementation of these features and of the resulting GEMM operation. Finally, we present an experimental analysis on two homogeneous supercomputers showing that our approach is competitive up to 32,768 CPU cores with state-of-the-art libraries and may outperform them for some problem dimensions. Although our code can use GPUs straightforwardly, we do not deal with this case because it implies other issues which are out of the scope of this work.

8.2 High performance solvers for large linear algebra problems

Participants: Mathieu Faverge, Esragul Korkmaz, Pierre Ramet.

8.2.1 Reaching the Quality of SVD for Low-Rank Compression Through QR Variants

In [27], Solving linear equations for large sparse systems frequently emerges in science/engineering applications, which is the main bottleneck. In spite that the direct methods are costly in time and memory consumption, they are still the most robust way to solve these systems. Nowadays, increasing the amount of computational units for the supercomputers became trendy, while the memory available per core is reduced. Thus, when solving these linear equations, memory reduction becomes as important as time reduction. For this purpose, compression methods are introduced within sparse solvers to reduce both the memory and time consumption. In this respect, Singular Value Decomposition (SVD) is used to reach the smallest possible rank, but it is too costly in practice. Rank revealing QR decomposition variants are used as faster alternatives, which can introduce larger ranks. Among these variants, column pivoting or matrix rotation can be applied on the matrix, such that the most important information in the matrix is gathered to the leftmost columns and the remaining unnecessary information can be omitted. For reducing the communication cost of the QR decomposition with column pivoting, blocking versions with randomization are suggested as an alternative to find the pivots. In these randomized variants, the matrix is projected on a lower dimensional matrix by using an i.i.d. Gaussian matrix so that the pivoting/rotational matrix can be computed on the lower dimensional matrix. In addition, to avoid unnecessary updates of the trailing matrix at each iteration, a truncated randomized method is suggested to be more efficient for larger matrix sizes. Thanks to these methods, closer results to SVD are obtained with reduced compression cost. In this report, we compare all these methods in terms of complexity, numerical stability, obtained rank, performance and accuracy.

8.3 New Allocation Schemes for Linear Algebra Kernels

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Mathieu Vérité.

8.3.1 Symmetric Block-Cyclic Distribution: Fewer Communications Leads to Faster Dense Cholesky Factorization

In [13], in collaboration with Julien Langou (University of Colorado in Denver), we consider the distributed Cholesky factorization on homogeneous nodes. Inspired by recent progress on asymptotic lower bounds on the total communication volume required to perform Cholesky factorization, we present an original data distribution, Symmetric Block Cyclic (SBC), designed to take advantage of the symmetry of the matrix. We prove that SBC reduces the overall communication volume between nodes by a factor of square root of 2 compared to the standard 2D blockcyclic distribution. SBC can easily be implemented within the paradigm of task-based runtime systems. Experiments using the Chameleon library over the StarPU runtime system demonstrate that the SBC distribution reduces the communication volume as expected, and also achieves better performance and scalability than the classical 2D block-cyclic allocation scheme in all configurations. We also propose a 2.5D variant of SBC and prove that it further improves the communication and performance benefits.

8.3.2 I/O-Optimal Algorithms for Symmetric Linear Algebra Kernels

In [15], in collaboration with Julien Langou (University of Colorado in Denver), we consider two fundamental symmetric kernels in linear algebra: the Cholesky factorization and the symmetric rank-k update (SYRK), with the classical three nested loops algorithms for these kernels. In addition, we consider a machine model with a fast memory of size S and an unbounded slow memory. In this model, all computations must be performed on operands in fast memory, and the goal is to minimize the amount of communication between slow and fast memories. As the set of computations is fixed by the choice of the algorithm, only the ordering of the computations (the schedule) directly influences the volume of communications.

We prove lower bounds of $\frac{1}{3\sqrt{2}}\frac{N^3}{\sqrt{S}}$ for the communication volume of the Cholesky factorization of an $N \times N$ symmetric positive definite matrix, and of $\frac{1}{\sqrt{2}}\frac{N^2M}{\sqrt{S}}$ for the SYRK computation of $A \cdot A^T$, where **A** is an $N \times M$ matrix. Both bounds improve the best known lower bounds from the literature by a factor $\sqrt{2}$.

In addition, we present two out-of-core, sequential algorithms with matching communication volume: TBS for SYRK, with a volume of $\frac{1}{\sqrt{2}}\frac{N^2M}{\sqrt{S}} + O(NM\log N)$, and LBC for Cholesky, with a volume of $\frac{1}{3\sqrt{2}}\frac{N^3}{\sqrt{S}} + O(N^{5/2})$. Both algorithms improve over the best known algorithms from the literature by a factor $\sqrt{2}$, and prove that the leading terms in our lower bounds cannot be improved further. This work shows that the operational intensity of symmetric kernels like SYRK or Cholesky is intrinsically higher (by a factor $\sqrt{2}$) than that of corresponding non-symmetric kernels (GEMM and LU factorizations).

8.4 High Performance Computing for Training

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Xunyi Zhao.

8.4.1 MadPipe: Memory Aware Dynamic Programming Algorithm for Pipelined Model Parallelism

In [14], we consider the use of Model Parallelism for training. The training phase in Deep Neural Networks (DNNs) is very computationally intensive and is nowadays often performed on parallel computing platforms, ranging from a few GPUs to several thousand GPUs. The strategy of choice for the parallelization of training is the so-called data parallel approach, based of the parallel training of the different inputs (typically images) and a the aggregation of network weights with collective communications (AllReduce). The scalability of this approach is limited both by the memory available on each node and the networking capacities for collective operations. Recently, a parallel model approach, in which the network weights are distributed and images are trained in a pipeline/stream manner over the computational nodes has been proposed (Pipedream, Gpipe). In this paper, we formalize in detail the optimization problem associated with the placement of DNN layers onto computation resources when using pipelined model parallelism, and we derive a dynamic programming based heuristic, MadPipe, that allows to significantly improve the performance of the parallel model approach compared to the literature.

8.4.2 An Integer Linear Programming Approach for Pipelined Model Parallelism

In [24], we propose a model for pipelined model parallelism. The training phase in Deep Neural Networks has become an important source of computing resource usage and because of the resulting volume of computation, it is crucial to perform it efficiently on parallel architectures. Even today, data parallelism is the most widely used method, but the associated requirement to replicate all the weights on the totality of computation resources poses problems of memory at the level of each node and of collective communications at the level of the platform. In this context, the model parallelism, which consists in distributing the different layers of the network over the computing nodes, is an attractive alternative. Indeed, it is expected to better distribute weights (to cope with memory problems) and it does not imply large collective communications since only forward activations are communicated. However, to be efficient, it must be combined with a pipelined/streaming approach, which leads in turn to new memory costs. The goal of this paper is to model these memory costs in detail and to show that it is possible to formalize this optimization problem as an Integer Linear Program (ILP).

8.4.3 Weight Offloading Strategies for Training Large DNN Models

In [25], we consider weight offloading strategies. The limited memory of GPUs induces serious problems in the training phase of deep neural networks (DNNs). Indeed, with the recent tremendous increase in the size of DNN models, which can now routinely include hundreds of billions or even trillions of parameters, it is impossible to store these models in the memory of a GPU and several strategies have been devised to solve this problem. In this paper, we analyze in detail the strategy that consists in offloading the weights of some model layers from the GPU to the CPU when they are not used. Since the PCI bus bandwidth between the GPU and the CPU is limited, it is crucial to know which layers should be transferred (offloaded and prefetched) and when. We prove that this problem is in general NP-Complete in the strong sense and we propose a lower bound formulation in the form of an Integer Linear Program (ILP). We propose heuristics to select the layers to offload and to build the schedule of data transfers. We show that this approach allows to build near-optimal weight offloading strategies on realistic size DNNs and architectures.

8.4.4 Survey on Large Scale Neural Network Training

Modern Deep Neural Networks (DNNs) require significant memory to store weight, activations, and other intermediate tensors during training. Hence, many models do not fit one GPU device or can be trained using only a small per-GPU batch size. In [17], we provide a systematic overview of the approaches that enable more efficient DNNs training. We analyze techniques that save memory and make good use of computation and communication resources on architectures with a single or several GPUs. We summarize the main categories of strategies and compare strategies within and across categories. Along with approaches proposed in the literature, we discuss available implementations.

In [10], we describe the Textarossa projetct, that covers issues related to high performance training and high performance linear algebra. In the near future, Exascale systems will need to bridge three technology gaps to achieve high performance while remaining under tight power constraints: energy efficiency and thermal control; extreme computation efficiency via HW acceleration and new arithmetic; methods and tools for seamless integration of reconfigurable accelerators in heterogeneous HPC multi-node platforms. TEXTAROSSA addresses these gaps through a co-design approach to heterogeneous HPC solutions, supported by the integration and extension of HW and SW IPs, programming models, and tools derived from European research.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Mathieu Faverge, Abdou Guermouche, Pierre Ramet.

Some on the ongoing PhD thesis are developed within bilateral contract with industry for PhD advisory:

- IFPEN for the PhD of Aboul-Karim Mohamed El Maarouf,
- CEA-Cesta for the PhD of Clément Richefort,
- CEA-Cesta for the PhD of Abel Calluaud.

We are also involved in a bilateral collaboration with Atos as part of the recovery plan, which has led in particular to the recruitment of Marc Sergent and Ahmed Abdourahmane. The collaboration focuses on the parallelisation of training strategies and the use of memory saving techniques.

10 Partnerships and cooperations

Participants: Olivier Beaumont, Lionel Eyraud-Dubois, Mathieu Faverge, Abdou Guermouche, Pierre Ramet.

10.1 European initiatives

10.1.1 H2020 projects

HPCQS HPCQS project on cordis.europa.eu

Title: High Performance Computer and Quantum Simulator hybrid

Duration: From December 1, 2021 to November 30, 2025

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (GENCI), France
- NATIONAL UNIVERSITY OF IRELAND GALWAY (NUI GALWAY), Ireland
- FORSCHUNGSZENTRUM JULICH GMBH (FZJ), Germany
- PARITY QUANTUM COMPUTING GMBH (ParityQC), Austria
- FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV (FHG), Germany
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), France
- EURICE EUROPEAN RESEARCH AND PROJECT OFFICE GMBH, Germany
- CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), Italy
- BULL SAS (BULL), France
- FLYSIGHT SRL, Italy
- PARTEC AG (PARTEC), Germany
- UNIVERSITAET INNSBRUCK (UIBK), Austria
- CINECA CONSORZIO INTERUNIVERSITARIO (CINECA), Italy
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (CNRS), France

- CENTRALESUPELEC (CentraleSupélec), France
- BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION (BSC CNS), Spain
- SORBONNE UNIVERSITE, France

Inria contact: Luc Giraud

Coordinator:

Summary: The aim of HPCQS is to prepare European research, industry and society for the use and federal operation of quantum computers and simulators. These are future computing technologies that are promising to overcome the most difficult computational challenges. HPCQS is developing the programming platform for the quantum simulator, which is based on the European ATOS Quantum Learning Machine (QLM), and the deep, low-latency integration into modular HPC systems based on ParTec's European modular supercomputing concept. A twin pilot system, developed as a prototype by the European company Pasqal, will be implemented and integrated at CEA/TGCC (France) and FZJ/JSC (Germany), both hosts of European Tier-0 HPC systems. The pre-exascale sites BSC (Spain) and CINECA (Italy) as well as ICECH (Ireland) will be connected to the TGCC and JSC via the European data infrastructure FENIX. It is planned to offer quantum HPC hybrid resources to the public via the access channels of PRACE. To achieve these goals, HPCQS brings together leading quantum and supercomputer experts from science and industry, thus creating an incubator for practical quantum HPC hybrid computing that is unique in the world. The HPC-QS technology will be developed in a co-design process together with selected exemplary use cases from chemistry, physics, optimization and machine learning suitable for quantum HPC hybrid calculations. HPCQS fits squarely to the challenges and scope of the call by acquiring a quantum device with two times 100+ neutral atoms. HPCQS develops the connection between the classical supercomputer and the quantum simulator by deep integration in the modular supercomputing architecture and will provide cloud access and middleware for programming and execution of applications on the quantum simulator through the QLM, as well as a Jupyter-Hub platform with safe access guarantee through the European UNICORE system to its ecosystem of quantum programming facilities and application libraries.

EUPEX EUPEX project on cordis.europa.eu

Title: EUROPEAN PILOT FOR EXASCALE

Duration: From January 1, 2022 to December 31, 2025

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (GENCI), France
- VSB - TECHNICAL UNIVERSITY OF OSTRAVA (VSB - TU Ostrava), Czechia
- FORSCHUNGSZENTRUM JULICH GMBH (FZJ), Germany
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), France
- IDRYMA TECHNOLOGIAS KAI EREVNAS (FOUNDATION FOR RESEARCH AND TECHNOLOGYHELLAS), Greece
- SVEUCILISTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RACUNARSTVA (UNIVERSITYOF ZAGREB FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING), Croatia
- UNIVERSITA DEGLI STUDI DI TORINO (UNITO), Italy
- CYBELETECH (Cybeletech), France
- UNIVERSITA DI PISA (UNIPI), Italy

- GRAN SASSO SCIENCE INSTITUTE (GSSI), Italy
- ISTITUTO NAZIONALE DI ASTROFISICA (INAF), Italy
- UNIVERSITA DEGLI STUDI DEL MOLISE, Italy
- E 4 COMPUTER ENGINEERING SPA (E4), Italy
- UNIVERSITA DEGLI STUDI DELL'AQUILA (UNIVAQ), Italy
- CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), Italy
- JOHANN WOLFGANG GOETHE-UNIVERSITAET FRANKFURT AM MAIN (GUF), Germany
- EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS (ECMWF), United Kingdom
- BULL SAS (BULL), France
- POLITECNICO DI MILANO (POLIMI), Italy
- EXASCALE PERFORMANCE SYSTEMS - EXAPSYS IKE, Greece
- ALMA MATER STUDIORUM - UNIVERSITA DI BOLOGNA (UNIBO), Italy
- PARTEC AG (PARTEC), Germany
- ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA, Italy
- CINECA CONSORZIO INTERUNIVERSITARIO (CINECA), Italy
- SECO SPA (SECO SRL), Italy
- CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA (CINI), Italy

Inria contact: Olivier Beaumont

Coordinator:

Summary: The EUPEX consortium aims to design, build, and validate the first EU platform for HPC, covering end-to-end the spectrum of required technologies with European assets: from the architecture, processor, system software, development tools to the applications. The EUPEX prototype will be designed to be open, scalable and flexible, including the modular OpenSequana-compliant platform and the corresponding HPC software ecosystem for the Modular Supercomputing Architecture. Scientifically, EUPEX is a vehicle to prepare HPC, AI, and Big Data processing communities for upcoming European Exascale systems and technologies. The hardware platform is sized to be large enough for relevant application preparation and scalability forecast, and a proof of concept for a modular architecture relying on European technologies in general and on European Processor Technology (EPI) in particular. In this context, a strong emphasis is put on the system software stack and the applications.

Being the first of its kind, EUPEX sets the ambitious challenge of gathering, distilling and integrating European technologies that the scientific and industrial partners use to build a production-grade prototype. EUPEX will lay the foundations for Europe's future digital sovereignty. It has the potential for the creation of a sustainable European scientific and industrial HPC ecosystem and should stimulate science and technology more than any national strategy (for numerical simulation, machine learning and AI, Big Data processing).

The EUPEX consortium – constituted of key actors on the European HPC scene – has the capacity and the will to provide a fundamental contribution to the consolidation of European supercomputing ecosystem. EUPEX aims to directly support an emerging and vibrant European entrepreneurial ecosystem in AI and Big Data processing that will leverage HPC as a main enabling technology.

TEXTAROSSA TEXTAROSSA project on cordis.europa.eu

Title: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

Duration: From April 1, 2021 to March 31, 2024

Partners:

- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- IN QUATTRO SRL (in quattro), Italy
- FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV (FHG), Germany
- UNIVERSITA DEGLI STUDI DI TORINO (UNITO), Italy
- INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK, Poland
- AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE (ENEA), Italy
- UNIVERSITA DI PISA (UNIPI), Italy
- E 4 COMPUTER ENGINEERING SPA (E4), Italy
- CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), Italy
- UNIVERSITE DE BORDEAUX (UBx), France
- BULL SAS (BULL), France
- POLITECNICO DI MILANO (POLIMI), Italy
- BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION (BSC CNS), Spain
- CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA (CINI), Italy
- ISTITUTO NAZIONALE DI FISICA NUCLEARE (INFN), Italy

Inria contact: Olivier BEAUMONT

Coordinator:

Summary: To achieve high performance and high energy efficiency on near-future exascale computing systems, a technology gap needs to be bridged: increase efficiency of computation with extreme efficiency in HW and new arithmetics, as well as providing methods and tools for seamless integration of reconfigurable accelerators in heterogeneous HPC multi-node platforms. TEXTAROSSA aims at tackling this gap through applying a co-design approach to heterogeneous HPC solutions, supported by the integration and extension of IPs, programming models and tools derived from European research projects, led by TEXTAROSSA partners. The main directions for innovation are towards: i) enabling mixed-precision computing, through the definition of IPs, libraries, and compilers supporting novel data types (including Posits), used also to boost the performance of AI accelerators; ii) implementing new multilevel thermal management and two-phase liquid cooling; iii) developing improved data movement and storage tools through compression; iv) ensure secure HPC operation through HW accelerated cryptography; v) providing RISC-V based IP for fast task scheduling and IPs for low-latency intra/inter-node communication. These technologies will be tested on the Integrated Development Vehicles mirroring and extending the European Processor Initiative ARM64-based architecture, and on an OpenSequana testbed. To drive the technology development and assess the impact of the proposed innovations TEXTAROSSA will use a selected but representative number of HPC, HPDA and AI demonstrators covering challenging HPC domains such as general-purpose numerical kernels, High Energy Physics (HEP), Oil & Gas, climate modelling, and emerging domains such as High Performance Data Analytics (HPDA) and High Performance Artificial Intelligence (HPC-AI).

10.2 National initiatives

10.2.1 ANR

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

Duration: 2018 – 2022

Coordinator: Mathieu Faverge

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

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

Duration: 2018 – 2022

Coordinator: Alfredo Buttari (IRIT)

HIEPACS **contact:** Abdou Guermouche

Partners:

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

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

10.2.2 FUI

ICARUS: Intensive Calculation for AeRo and automotive engines Unsteady Simulations

Duration: 2018 – 2022

Coordinator: SAFRAN

Inria contact: Aurélien Esnard

Partners:

- CENAERO

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

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

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

10.2.3 Inria Project Labs

Challenge HPC BigData

Duration: 2018 – 2022

Coordinator: Bruno Raffin

HIEPACS **contact:** Olivier Beaumont

Inria teams:

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

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

Challenge PULSE: Pushing low-carbon services towards the Edge

Duration: 2022 –

Coordinator: Romain Rouvoy

HIEPACS **contact:** Olivier Beaumont & Lionel Eyraud Dubois

Partners: Qarnot Computing, ADEME

Inria teams:

- Avalon
- Ctrl-A
- Spirals
- Stack
- Storm
- Topal

Summary: The challenge aims to develop and promote best practices in geo-repaired hardware and software infrastructures for more environmentally friendly intensive computing. The idea is to analyze which solutions are the most relevant, and which levers need to be focused on, to reduce the impact of infrastructures while maximizing the usefulness of their emissions. To this end, the challenge is structured around two complementary research axes to address this technological and environmental issue: holistic analysis of the environmental impact of intensive computing, and implementing more virtuous edge services.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

General chair, scientific chair In 2022, we organized the international SBAC-PAD conference in Bordeaux. Olivier Beaumont was General Chair of the conference and several members of the team (Lionel Eyraud Dubois, Mathieu Faverge, Abdou Guermouche) were involved in the local organizing committee. There were 80 international participants at the conference, which was held in-person.

Member of the conference program committees

- Olivier Beaumont was involved in the following programm committees: SC 22, ICPP 22, IPDPS 22 and PPAM 22
- Lionel Eyraud Dubois was part of the program committee of ICPP 22, SC 22
- Mathieu Faverge was part of the program committees of Cluster 22, ICPP 22, IPDPS 22
- Abdou Guermouche was part of the program commitees of ICPP 22, IPDPS 22
- Pierre Ramet was part of the program committee of HiPC 22

Reviewer The members of the HiePACS project have performed reviewing for the following list of conferences: IPDPS 22, HiPC 22, SPAC-PAD 22, SC 22, ICPP 22, Cluster 22, PPAM 22.

11.1.2 Journal

Member of the editorial boards

- Olivier Beaumont is Associate Editor in Chief of the Journal of Parallel and Distributed Computing (JPDC, Elsevier)

Reviewer - reviewing activities The members of the HiePACS project have performed reviewing for the following list of journals:

- Journal of Parallel and Distributed Computing (Mathieu Faverge, Lionel Eyraud Dubois, Abdou Guermouche)
- Journal of Computational and Applied Mathematics (Pierre Ramet),
- Journal of Scheduling (Lionel Eyraud Dubois),
- Computers and Fluids (Pierre Ramet),
- International Journal of High Performance Computing Applications (Pierre Ramet),
- ACM Transactions on Mathematical Software (Pierre Ramet, Mathieu Faverge),
- ACM Transactions on Computers (Lionel Eyraud-Dubois),
- Concurrency and Computation: Practice and Experience (Lionel Eyraud-Dubois),
- Parallel Computing (Abdou Guermouche).

11.1.3 Scientific expertise

- Pierre Ramet is Scientific Advisor at the CEA-DAM CESTA.

11.1.4 Research administration

- Aurélien Esnard is responsible for the second year of the computer science degree (L2 Informatique), which involves managing about 200 students each year.
- Aurélien Esnard is in charge of the *Numerical Transformation* mission at the College ST (Science & Technology) of the University of Bordeaux. In this context, he assists the management of the College in its decisions, informs the College's advisors about the current issues in various projects, and leads a working group to propose improvements to business software for education and its administration.
- Abdou Guermouche is member of Scientific comittee of the LaBRI.

- Mathieu Faverge is member of the Study committee of Bordeaux INP.
- Pierre Ramet is member of the CDT (Technological Development Commission) at Inria Bordeaux.
- Pierre Ramet is the head of the CNRS Satanas department.
- Pierre Ramet is member of Scientific comittee of the LaBRI.

11.2 Teaching - Supervision - Juries

11.2.1 Teaching

- Undergraduate level/Licence:
  - Aurélien Esnard: Network (54h), Software technologies (80h) at Bordeaux University.
  - Lionel Eyraud Dubois: Graphs and Algorithms (18h).
  - Mathieu Faverge: Programming environment 26h, Numerical algorithmic 40h, C projects 25h at Bordeaux INP (ENSEIRB-MatMeca).
  - Abdou Guermouche: System programming 36h at Bordeaux University.
  - Pierre Ramet: System programming 24h, Databases 32h, Object programming 48h, Distributed programming 16h, Cryptography 16h, Introduction to AI Deep Learning and Data Analytics 16h at Bordeaux University.
- Post graduate level/Master:
  - Aurélien Esnard: Network management (24h), Network security (24h) at Bordeaux University.
  - Lionel Eyraud Dubois and Olivier Beaumont: Approximation and BigData 24h at Bordeaux University.
  - Olivier Beaumont: Parallel Algorithms 24h at Bordeaux INP (Enseirb).
  - Mathieu Faverge: System programming 72h, Linear Algebra for high Performance Computing 13h at Bordeaux INP (ENSEIRB-MatMeca). He is also in charge of the master 2 internship for the Computer Science department at Bordeaux INP (ENSEIRB-MatMeca) and he is in charge, with Raymond Namyst, of the High Performance Computing - High Performance Data Analytics specialty at ENSEIRB-MatMeca. This is a common training curriculum between the Computer Science and the MatMeca departments at Bordeaux INP and with the Bordeaux University in the context of the Computer Science Research Master.
  - Olivier Beaumont: Sketching and Streaming Algorithms, ENS Lyon, 8h.
  - Abdou Guermouche: Network management 92h, Network security 64h, Operating system 24h at Bordeaux University.
  - Pierre Ramet: Cryptography 20h and Numerical algorithms 40h at Bordeaux INP (ENSEIRB-Matmeca).
  - Yulia Gusak: Deep Learning Frameworks, at Bordeaux INP (ENSEIRB-MatMeca), 20h.

11.2.2 Supervision

- Defended PhD: Esragul Korkmaz; Sparse linear solver and hierachical matrices; defended: Sep. 2022; M. Faverge, P. Ramet.
- Defended PhD: Mathieu Verite; Static allocation algorithms for scheduling High-Performance applications; defended: Dec. 202; L. Eyraud-Dubois, O. Beaumont.
- PhD in progress: Aboul-Karim Mohamed El Maarouf; Parallel fine grain imcomplete LU factorization for the solution of sparse linear systems; started: Dec. 2019; L. Giraud, A. Guermouche.

- PhD in progress: Clément Richefort; Multigrid methods applied to electromagnetism problems; started Nov. 2021; P. Ramet, M. Lecouvez (CEA Cesta).
- PhD in progress: Abel Calluaud; Combined compiler and runtime approach for a direct hierarchical solver; started Nov. 2022; P. Ramet, M. Faverge, D. Lugato (CEA Cesta).
- PhD in progress: Xunyi Zhao; Memory optimization for Deep Learning Applications; started Sept. 2020; L. Eyraud Dubois, O. Beaumont.
- PhD in progress: Jean Francois David; Task-based inference for heterogeneous architectures; started Sept. 2020; L. Eyraud Dubois, O. Beaumont.

11.3 Popularization

11.3.1 Education

- as part of the COP'27, Olivier Beaumont made an intervention in a high school Jean Monnet of Libourne on the mitigation of digital impacts (in the form of an open discussion with the students).
- as part of the science festival and the Bordeaux scientific circuit, Olivier Beaumont gave a talk (at the junior high school level) on on-line algorithms and the cost of uncertainty
- within the framework of Maths en Jeans, Olivier Beaumont work this year with a group of students in the Montessori high school of Mios (Gironde) on some combinatorial problems.
- in the framework of the "Chiche" program, Olivier Beaumont intervened at the Grand Air high school in Arcachon on the problems of dynamic routing of vehicles.

12 Scientific production

12.1 Major publications

[1] E. Agullo, A. Buttari, A. Guermouche and F. Lopez. 'Implementing multifrontal sparse solvers for multicore architectures with Sequential Task Flow runtime systems'. In: *ACM Transactions on Mathematical Software* (July 2016). DOI: 10.1145/2898348. URL: https://hal.inria.fr/hal-01333645.

[2] O. Beaumont, L.-C. Canon, L. Eyraud-Dubois, G. Lucarelli, L. Marchal, C. Mommessin, B. Simon and D. Trystram. 'Scheduling on Two Types of Resources: a Survey'. In: *ACM Computing Surveys* 53.3 (May 2020). DOI: 10.1145/3387110. URL: https://hal.inria.fr/hal-02432381.

[3] O. Beaumont, L. Eyraud-Dubois and S. Kumar. 'Fast Approximation Algorithms for Task-Based Runtime Systems'. In: *Concurrency and Computation: Practice and Experience* 30.17 (Sept. 2018). DOI: 10.1002/cpe.4502. URL: https://hal.inria.fr/hal-01878606.

[4] O. Beaumont, L. Eyraud-Dubois and A. Shilova. 'Optimal GPU-CPU Offloading Strategies for Deep Neural Network Training'. In: *Euro-Par 2020 - 26th International Conference on Parallel and Distributed Computing*. Ed. by M. Malawski and K. Rzadca. Euro-Par 2020: Parallel Processing 12247. Warsaw / Virtual, Poland: Springer, Aug. 2020, pp. 151–166. DOI: 10.1007/978-3-030-57675-2_10. URL: https://hal.inria.fr/hal-02316266.

[5] O. Beaumont, T. Lambert, L. Marchal and B. Thomas. 'Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs'. In: *Future Generation Computer Systems* 111 (Oct. 2020), pp. 582–598. DOI: 10.1016/j.future.2019.08.024. URL: https://hal.inria.fr/hal-02275473.

[6] A. Hugo, A. Guermouche, P.-A. Wacrenier and R. Namyst. 'Composing multiple StarPU applications over heterogeneous machines: A supervised approach'. In: *International Journal of High Performance Computing Applications* 28 (Feb. 2014), pp. 285–300. DOI: 10.1177/1094342014527575. URL: https://hal.inria.fr/hal-01101045.

[7] G. Pichon, E. Darve, M. Faverge, P. Ramet and J. Roman. 'Sparse supernodal solver using block low-rank compression: Design, performance and analysis'. In: *International Journal of Computational Science and Engineering* 27 (July 2018), pp. 255–270. DOI: 10.1016/J.JOCS.2018.06.007. URL: https://hal.inria.fr/hal-01824275.

[8] G. Pichon, M. Faverge, P. Ramet and J. Roman. 'Reordering Strategy for Blocking Optimization in Sparse Linear Solvers'. In: *SIAM Journal on Matrix Analysis and Applications*. SIAM Journal on Matrix Analysis and Applications 38.1 (2017), pp. 226–248. DOI: 10.1137/16M1062454. URL: https://hal.inria.fr/hal-01485507.

[9] M. Predari, A. Esnard and J. Roman. 'Comparison of initial partitioning methods for multilevel direct k-way graph partitioning with fixed vertices'. In: *Parallel Computing* (2017). DOI: 10.1016/j.parco.2017.05.002. URL: https://hal.inria.fr/hal-01538600.

12.2 Publications of the year

International journals

[10] G. Agosta, M. Aldinucci, C. Alvarez, R. Ammendola, Y. Arfat, O. Beaumont, M. Bernaschi, A. Biagioni, T. Boccali, B. Bramas et al. 'Towards EXtreme scale technologies and accelerators for euROhpc hw/Sw supercomputing applications for exascale: The TEXTAROSSA approach'. In: *Microprocessors and Microsystems: Embedded Hardware Design* 95 (Nov. 2022), p. 104679. DOI: 10.1016/j.micpro.2022.104679. URL: https://hal.inria.fr/hal-03936864.

[11] E. Agullo, A. Buttari, A. Guermouche, J. Herrmann and A. Jego. 'Task-based parallel programming for scalable matrix product algorithms'. In: *ACM Transactions on Mathematical Software* (2023). URL: https://hal.science/hal-03936659.

[12] A.-k. Mohamed El Maarouf, L. Giraud, A. Guermouche and T. Guignon. 'Combining reduction with synchronization barrier on multi-core processors'. In: *Concurrency and Computation: Practice and Experience* 35.1 (1st Dec. 2022). DOI: 10.1002/cpe.7402. URL: https://hal.inria.fr/hal-03948901.

International peer-reviewed conferences

[13] O. Beaumont, P. Duchon, L. Eyraud-Dubois, J. Langou and M. Vérité. 'Symmetric Block-Cyclic Distribution: Fewer Communications Leads to Faster Dense Cholesky Factorization'. In: SC 2022 - Supercomputing. Dallas, Texas, United States, 13th Nov. 2022. URL: https://hal.inria.fr/hal-03768910.

[14] O. Beaumont, L. Eyraud-Dubois and A. Shilova. 'MadPipe: Memory Aware Dynamic Programming Algorithm for Pipelined Model Parallelism'. In: ScaDL 2022 - Scalable Deep Learning over Parallel and Distributed Infrastructure - An IPDPS 2022 Workshop. Proceedings of IPDPS W'22. Lyon / Virtual, France, 2022. URL: https://hal.science/hal-03025305.

[15] O. Beaumont, L. Eyraud-Dubois, M. Vérité and J. Langou. 'I/O-Optimal Algorithms for Symmetric Linear Algebra Kernels'. In: ACM Symposium on Parallelism in Algorithms and Architectures. Philadelphie, United States, 11th July 2022. URL: https://hal.inria.fr/hal-03580531.

[16] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, R. Namyst, S. Thibault and P.-A. Wacrenier. 'Programming Heterogeneous Architectures Using Hierarchical Tasks'. In: HeteroPar 2022. Glasgow, United Kingdom, 23rd Aug. 2022, p. 12. URL: https://hal.science/hal-03789625.

[17] J. Gusak, D. Cherniuk, A. Shilova, A. Katrutsa, D. Bershatsky, X. Zhao, L. Eyraud-Dubois, O. Shliazhko, D. Dimitrov, I. Oseledets and O. Beaumont. 'Survey on Large Scale Neural Network Training'. In: IJCAI-ECAI 2022 - 31st International Joint Conference on Artificial Intelligence. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence Survey Track. Vienna, Austria: International Joint Conferences on Artificial Intelligence Organization, 23rd July 2022, pp. 5494–5501. DOI: 10.24963/ijcai.2022/769. URL: https://hal.inria.fr/hal-03952171.

National peer-reviewed Conferences

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

[19] H. Tayeb, B. Bramas, A. Guermouche and M. Faverge. 'MulTreePrio: Scheduling task-based applications for heterogeneous computing systems'. In: COMPAS 2022 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Amiens, France, 5th July 2022. URL: https://hal.inria.fr/hal-03763824.

Doctoral dissertations and habilitation theses

[20] E. Korkmaz. 'Improving the memory and time overhead of low-rank parallel linear sparse direct solvers'. Université de Bordeaux, 21st Sept. 2022. URL: https://theses.hal.science/tel-03875858.

[21] M. Verite. 'Static Allocation Algorithms for Scheduling High-Performance Applications'. Université de Bordeaux, 7th Dec. 2022. URL: https://theses.hal.science/tel-03956040.

Reports & preprints

[22] E. Agullo, A. Buttari, A. Guermouche, J. Herrmann and A. Jego. *Task-Based Parallel Programming for Scalable Algorithms: application to Matrix Multiplication*. RR-9461. Inria Bordeaux - Sud-Ouest, Feb. 2022, p. 29. URL: https://hal.inria.fr/hal-03588491.

[23] E. Agullo, O. Coulaud, A. Denis, M. Faverge, A. Franc, J.-M. Frigerio, N. Furmento, A. Guilbaud, E. Jeannot, R. Peressoni, F. Pruvost and S. Thibault. *Task-based randomized singular value decomposition and multidimensional scaling*. RR-9482. Inria Bordeaux - Sud Ouest; Inrae - BioGeCo, 9th Sept. 2022, p. 37. URL: https://hal.inria.fr/hal-03773985.

[24] O. Beaumont, L. Eyraud-Dubois and A. Shilova. *An Integer Linear Programming Approach for Pipelined Model Parallelism*. RR-9452. Inria, Jan. 2022. URL: https://hal.inria.fr/hal-03549009.

[25] O. Beaumont, L. Eyraud-Dubois, A. Shilova and X. Zhao. *Weight Offloading Strategies for Training Large DNN Models*. 18th Feb. 2022. URL: https://hal.inria.fr/hal-03580767.

[26] M. Faverge, N. Furmento, A. Guermouche, G. Lucas, R. Namyst, S. Thibault and P.-A. Wacrenier. *Programming Heterogeneous Architectures Using Hierarchical Tasks*. RR-9466. Inria Bordeaux Sud-Ouest, 15th Mar. 2022, p. 21. URL: https://hal.inria.fr/hal-03609275.

[27] E. Korkmaz, M. Faverge, G. Pichon and P. Ramet. *Reaching the Quality of SVD for Low-Rank Compression Through QR Variants*. RR-9476. Inria Bordeaux - Sud Ouest, July 2022, p. 43. URL: https://hal.inria.fr/hal-03718312.

[28] A.-K. Mohamed El Maarouf, L. Giraud, A. Guermouche and T. Guignon. *Combining reduction with synchronization barrier on multi-core processors*. 16th Feb. 2022. URL: https://hal.archives-ouvertes.fr/hal-03577306.

12.3 Other

Softwares

[29] [SW] O. Beaumont, P. Duchon, J. Langou, L. Eyraud-Dubois and M. Verite, *Experimental code and results for the paper "Symmetric Block-Cyclic Distribution: Fewer Communications leads to Faster Dense Cholesky Factorization"*, 10th June 2022. LIC: CeCILL Free Software License Agreement v2.0. HAL: ⟨hal-03643569⟩, URL: https://hal.inria.fr/hal-03643569, SWHID: ⟨swh:1:dir:ec92870dfc2a7e7c8a6e87914823e25bc651a5b7;origin=https://hal.archives-ouvertes.fr/hal-03643569;visit=swh:1:snp:6b8402b24d2133d01fc48d0f01d83f605cc77898;anchor=swh:1:rel:3068c5a3c30d3a673a553d033029929a03ddb4cf;path=/⟩.