IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Lyon

Université Claude Bernard (Lyon 1)

Activity Report 2015

Project-Team ROMA

Optimisation des ressources : modèles, algorithmes et ordonnancement

IN COLLABORATION WITH: Laboratoire de l'Informatique du Parallélisme (LIP)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed and High Performance Computing

Table of contents

Project-Team ROMA

Creation of the Team: 2012 February 01, updated into Project-Team: 2015 January 01

Keywords:

Computer Science and Digital Science:

1.1.1. - Multicore
1.1.3. - Memory models
1.1.4. - High performance computing
1.1.5. - Exascale
1.1.9. - Fault tolerant systems
1.6. - Green Computing
6.1. - Mathematical Modeling
6.2.5. - Numerical Linear Algebra
6.2.6. - Optimization
6.2.7. - High performance computing
6.3. - Computation-data interaction
7.1. - Parallel and distributed algorithms
7.11. - Performance evaluation
7.2. - Discrete mathematics, combinatorics
7.3. - Operations research, optimization, game theory
7.9. - Graph theory

Other Research Topics and Application Domains:

3.2. - Climate and meteorology
3.3. - Geosciences
4.1.1. - Oil, gas
4.1.2. - Nuclear energy
4.4.1. - Green computing
5.2.3. - Aviation
5.5. - Materials

1. Members

Research Scientists

Frédéric Vivien [Team leader, Inria, Senior Researcher, HdR]
Christophe Alias [Inria, Researcher, from Sept. 2015 (see Section 3.4)]
Jean-Yves L'Excellent [Inria, Researcher, HdR]
Loris Marchal [CNRS, Researcher]
Bora Uçar [CNRS, Researcher]

Faculty Members

Anne Benoit [ENS Lyon, Associate Professor, HdR]
Laure Gonnord [Univ. Lyon I, Associate Professor, from Sept. 2015 (see Section 3.4)]
Yves Robert [ENS Lyon, Professor, HdR]

Engineers

Marie Durand [Inria]
Guillaume Joslin [Inria]
Chiara Puglisi [Inria]

PhD Students

Guillaume Aupy [ENS Lyon, until Aug. 2015]
Aurélien Cavelan [Inria]
Julien Herrmann [ENS Lyon]
Oguz Kaya [Inria]
Maroua Maalej [Univ. Lyon]
Tatiana Martsinkevich [Inria, until Sept. 2015]
Gilles Moreau [Inria, from Dec. 2015]
Issam Rais [Inria, from Nov. 2015]
Loic Pottier [ENS Lyon, from Oct. 2015]
Bertrand Simon [ENS Lyon]

Post-Doctoral Fellows

Enver Kayaaslan [Inria, until Feb. 2015]
Hongyang Sun [Univ. Lyon]

Visiting Scientists

Jiafan Li [ECNU, from Nov. 2015]
Oliver Sinnen [ENS Lyon, Sept.-Nov. 2015]
Samuel Mccauley [Inria, from Oct. 2015]

Administrative Assistants

Virginie Bouyer [Inria, until Apr. 2015]
Laetitia Lecot [Inria, from May 2015]

Others

Patrick Amestoy [INP Toulouse, external collaborator, HdR]
Alfredo Buttari [CNRS, external collaborator]
Franck Cappello [Argonne National Laboratory – USA, external collaborator, HdR]

2. Overall Objectives

2.1. Overall Objectives

The ROMA project aims at designing models, algorithms, and scheduling strategies to optimize the execution of scientific applications.

Scientists now have access to tremendous computing power. For instance, the four most powerful computing platforms in the TOP 500 list [60] each includes more than 500,000 cores and deliver a sustained performance of more than 10 Peta FLOPS. The volunteer computing platform BOINC [56] is another example with more than 440,000 enlisted computers and, on average, an aggregate performance of more than 9 Peta FLOPS. Furthermore, it had never been so easy for scientists to have access to parallel computing resources, either through the multitude of local clusters or through distant cloud computing platforms.

Because parallel computing resources are ubiquitous, and because the available computing power is so huge, one could believe that scientists no longer need to worry about finding computing resources, even less to optimize their usage. Nothing is farther from the truth. Institutions and government agencies keep building larger and more powerful computing platforms with a clear goal. These platforms must allow to solve problems in reasonable timescales, which were so far out of reach. They must also allow to solve problems more precisely where the existing solutions are not deemed to be sufficiently accurate. For those platforms to fulfill their purposes, their computing power must therefore be carefully exploited and not be wasted. This often requires an efficient management of all types of platform resources: computation, communication, memory, storage, energy, etc. This is often hard to achieve because of the characteristics of new and emerging platforms. Moreover, because of technological evolutions, new problems arise, and fully tried and tested solutions need to be thoroughly overhauled or simply discarded and replaced. Here are some of the difficulties that have, or will have, to be overcome:

- computing platforms are hierarchical: a processor includes several cores, a node includes several processors, and the nodes themselves are gathered into clusters. Algorithms must take this hierarchical structure into account, in order to fully harness the available computing power;
- the probability for a platform to suffer from a hardware fault automatically increases with the number of its components. Fault-tolerance techniques become unavoidable for large-scale platforms;
- the ever increasing gap between the computing power of nodes and the bandwidths of memories and networks, in conjunction with the organization of memories in deep hierarchies, requires to take more and more care of the way algorithms use memory;
- energy considerations are unavoidable nowadays. Design specifications for new computing platforms always include a maximal energy consumption. The energy bill of a supercomputer may represent a significant share of its cost over its lifespan. These issues must be taken into account at the algorithm-design level.

We are convinced that dramatic breakthroughs in algorithms and scheduling strategies are required for the scientific computing community to overcome all the challenges posed by new and emerging computing platforms. This is required for applications to be successfully deployed at very large scale, and hence for enabling the scientific computing community to push the frontiers of knowledge as far as possible. The ROMA project-team aims at providing fundamental algorithms, scheduling strategies, protocols, and software packages to fulfill the needs encountered by a wide class of scientific computing applications, including domains as diverse as geophysics, structural mechanics, chemistry, electromagnetism, numerical optimization, or computational fluid dynamics, to quote a few. To fulfill this goal, the ROMA project-team takes a special interest in dense and sparse linear algebra.

The work in the ROMA team is organized along three research themes.

1. **Algorithms for probabilistic environments.** In this theme, we consider problems where some of the platform characteristics, or some of the application characteristics, are described by probability distributions. This is in particular the case when considering the resilience of applications in failure-prone environments: the possibility of faults is modeled by probability distributions.
2. **Platform-aware scheduling strategies.** In this theme, we focus on the design of scheduling strategies that finely take into account some platform characteristics beyond the most classical ones, namely the computing speed of processors and accelerators, and the communication bandwidth of network links. In the scope of this theme, when designing scheduling strategies, we focus either on the energy consumption or on the memory behavior. All optimization problems under study are multi-criteria.
3. **High-performance computing and linear algebra.** We work on algorithms and tools for both sparse and dense linear algebra. In sparse linear algebra, we work on most aspects of direct multifrontal solvers for linear systems. In dense linear algebra, we focus on the adaptation of factorization kernels to emerging and future platforms. In addition, we also work on combinatorial scientific computing, that is, on the design of combinatorial algorithms and tools to solve combinatorial problems, such as those encountered, for instance, in the preprocessing phases of solvers of sparse linear systems.

3. Research Program

3.1. Algorithms for probabilistic environments

There are two main research directions under this research theme. In the first one, we consider the problem of the efficient execution of applications in a failure-prone environment. Here, probability distributions are used to describe the potential behavior of computing platforms, namely when hardware components are subject to faults. In the second research direction, probability distributions are used to describe the characteristics and behavior of applications.

3.1.1. Application resilience

An application is resilient if it can successfully produce a correct result in spite of potential faults in the underlying system. Application resilience can involve a broad range of techniques, including fault prediction, error detection, error containment, error correction, checkpointing, replication, migration, recovery, etc. Faults are quite frequent in the most powerful existing supercomputers. The Jaguar platform, which ranked third in the TOP 500 list in November 2011 [59], had an average of 2.33 faults per day during the period from August 2008 to February 2010 [88]. The mean-time between faults of a platform is inversely proportional to its number of components. Progresses will certainly be made in the coming years with respect to the reliability of individual components. However, designing and building high-reliability hardware components is far more expensive than using lower reliability top-of-the-shelf components. Furthermore, low-power components may not be available with high-reliability. Therefore, it is feared that the progresses in reliability will far from compensate the steady projected increase of the number of components in the largest supercomputers. Already, application failures have a huge computational cost. In 2008, the DARPA white paper on “System resilience at extreme scale” [58] stated that high-end systems wasted 20% of their computing capacity on application failure and recovery.

In such a context, any application using a significant fraction of a supercomputer and running for a significant amount of time will have to use some fault-tolerance solution. It would indeed be unacceptable for an application failure to destroy centuries of CPU-time (some of the simulations run on the Blue Waters platform consumed more than 2,700 years of core computing time [54] and lasted over 60 hours; the most time-consuming simulations of the US Department of Energy (DoE) run for weeks to months on the most powerful existing platforms [57]).

Our research on resilience follows two different directions. On the one hand we design new resilience solutions, either generic fault-tolerance solutions or algorithm-based solutions. On the other hand we model and theoretically analyze the performance of existing and future solutions, in order to tune their usage and help determine which solution to use in which context.

3.1.2. Scheduling strategies for applications with a probabilistic behavior

Static scheduling algorithms are algorithms where all decisions are taken before the start of the application execution. On the contrary, in non-static algorithms, decisions may depend on events that happen during the execution. Static scheduling algorithms are known to be superior to dynamic and system-oriented approaches in stable frameworks [68], [74], [75], [87], that is, when all characteristics of platforms and applications are perfectly known, known a priori, and do not evolve during the application execution. In practice, the prediction of application characteristics may be approximative or completely infeasible. For instance, the amount of computations and of communications required to solve a given problem in parallel may strongly depend on some input data that are hard to analyze (this is for instance the case when solving linear systems using full pivoting).

We plan to consider applications whose characteristics change dynamically and are subject to uncertainties. In order to benefit nonetheless from the power of static approaches, we plan to model application uncertainties and variations through probabilistic models, and to design for these applications scheduling strategies that are either static, or partially static and partially dynamic.

3.2. Platform-aware scheduling strategies

In this theme, we study and design scheduling strategies, focusing either on energy consumption or on memory behavior. In other words, when designing and evaluating these strategies, we do not limit our view to the most classical platform characteristics, that is, the computing speed of cores and accelerators, and the bandwidth of communication links.

In most existing studies, a single optimization objective is considered, and the target is some sort of absolute performance. For instance, most optimization problems aim at the minimization of the overall execution time of the application considered. Such an approach can lead to a very significant waste of resources, because it does not take into account any notion of efficiency nor of yield. For instance, it may not be meaningful to use twice as many resources just to decrease by 10% the execution time. In all our work, we plan to look only for algorithmic solutions that make a "clever" usage of resources. However, looking for the solution that optimizes a metric such as the efficiency, the energy consumption, or the memory-peak minimization, is doomed for the type of applications we consider. Indeed, in most cases, any optimal solution for such a metric is a sequential solution, and sequential solutions have prohibitive execution times. Therefore, it becomes mandatory to consider multi-criteria approaches where one looks for trade-offs between some user-oriented metrics that are typically related to notions of Quality of Service—execution time, response time, stretch, throughput, latency, reliability, etc.—and some system-oriented metrics that guarantee that resources are not wasted. In general, we will not look for the Pareto curve, that is, the set of all dominating solutions for the considered metrics. Instead, we will rather look for solutions that minimize some given objective while satisfying some bounds, or "budgets", on all the other objectives.

3.2.1. Energy-aware algorithms

Energy-aware scheduling has proven an important issue in the past decade, both for economical and environmental reasons. Energy issues are obvious for battery-powered systems. They are now also important for traditional computer systems. Indeed, the design specifications of any new computing platform now always include an upper bound on energy consumption. Furthermore, the energy bill of a supercomputer may represent a significant share of its cost over its lifespan.

Technically, a processor running at speed s dissipates s^{α} watts per unit of time with $2 \leq \alpha \leq 3$ [66], [67], [72]; hence, it consumes $s^{\alpha} \times d$ joules when operated during d units of time. Therefore, energy consumption can be reduced by using speed scaling techniques. However it was shown in [89] that reducing the speed of a processor increases the rate of transient faults in the system. The probability of faults increases exponentially, and this probability cannot be neglected in large-scale computing [83]. In order to make up for the loss in *reliability* due to the energy efficiency, different models have been proposed for fault tolerance: (i) *re-execution* consists in re-executing a task that does not meet the reliability constraint [89]; (ii) *replication* consists in executing the same task on several processors simultaneously, in order to meet the reliability constraints [64]; and (iii) *checkpointing* consists in "saving" the work done at some certain instants, hence reducing the amount of work lost when a failure occurs [82].

Energy issues must be taken into account at all levels, including the algorithm-design level. We plan to both evaluate the energy consumption of existing algorithms and to design new algorithms that minimize energy consumption using tools such as resource selection, dynamic frequency and voltage scaling, or powering-down of hardware components.

3.2.2. Memory-aware algorithms

For many years, the bandwidth between memories and processors has increased more slowly than the computing power of processors, and the latency of memory accesses has been improved at an even slower pace. Therefore, in the time needed for a processor to perform a floating point operation, the amount of data transferred between the memory and the processor has been decreasing with each passing year. The risk is for an application to reach a point where the time needed to solve a problem is no longer dictated by the processor computing power but by the memory characteristics, comparable to the *memory wall* that limits CPU performance. In such a case, processors would be greatly under-utilized, and a large part of the computing power of the platform would be wasted. Moreover, with the advent of multicore processors, the amount of memory per core has started to stagnate, if not to decrease. This is especially harmful to memory intensive applications. The problems related to the sizes and the bandwidths of memories are further exacerbated on modern computing platforms because of their deep and highly heterogeneous hierarchies. Such a hierarchy can extend from core private caches to shared memory within a CPU, to disk storage and even tape-based storage systems, like in the Blue Waters supercomputer [55]. It may also be the case that heterogeneous cores are used (such as hybrid CPU and GPU computing), and that each of them has a limited memory.

Because of these trends, it is becoming more and more important to precisely take memory constraints into account when designing algorithms. One must not only take care of the amount of memory required to run an algorithm, but also of the way this memory is accessed. Indeed, in some cases, rather than to minimize the amount of memory required to solve the given problem, one will have to maximize data reuse and, especially, to minimize the amount of data transferred between the different levels of the memory hierarchy (minimization of the volume of memory inputs-outputs). This is, for instance, the case when a problem cannot be solved by just using the in-core memory and that any solution must be out-of-core, that is, must use disks as storage for temporary data.

It is worth noting that the cost of moving data has lead to the development of so called “communication-avoiding algorithms” [79]. Our approach is orthogonal to these efforts: in communication-avoiding algorithms, the application is modified, in particular some redundant work is done, in order to get rid of some communication operations, whereas in our approach, we do not modify the application, which is provided as a task graph, but we minimize the needed memory peak only by carefully scheduling tasks.

3.3. High-performance computing and linear algebra

Our work on high-performance computing and linear algebra is organized along three research directions. The first direction is devoted to direct solvers of sparse linear systems. The second direction is devoted to combinatorial scientific computing, that is, the design of combinatorial algorithms and tools that solve problems encountered in some of the other research themes, like the problems faced in the preprocessing phases of sparse direct solvers. The last direction deals with the adaptation of classical dense linear algebra kernels to the architecture of future computing platforms.

3.3.1. Direct solvers for sparse linear systems

The solution of sparse systems of linear equations (symmetric or unsymmetric, often with an irregular structure, from a few hundred thousand to a few hundred million equations) is at the heart of many scientific applications arising in domains such as geophysics, structural mechanics, chemistry, electromagnetism, numerical optimization, or computational fluid dynamics, to cite a few. The importance and diversity of applications are a main motivation to pursue research on sparse linear solvers. Because of this wide range of applications, any significant progress on solvers will have a significant impact in the world of simulation. Research on sparse direct solvers in general is very active for the following main reasons:

- many applications fields require large-scale simulations that are still too big or too complicated with respect to today’s solution methods;
- the current evolution of architectures with massive, hierarchical, multicore parallelism imposes to overhaul all existing solutions, which represents a major challenge for algorithm and software development;
- the evolution of numerical needs and types of simulations increase the importance, frequency, and size of certain classes of matrices, which may benefit from a specialized processing (rather than resort to a generic one).

Our research in the field is strongly related to the software package MUMPS (see Section 6.1). MUMPS is both an experimental platform for academics in the field of sparse linear algebra, and a software package that is widely used in both academia and industry. The software package MUMPS enables us to (i) confront our research to the real world, (ii) develop contacts and collaborations, and (iii) receive continuous feedback from real-life applications, which is extremely critical to validate our research work. The feedback from a large user community also enables us to direct our long-term objectives towards meaningful directions.

In this context, we aim at designing parallel sparse direct methods that will scale to large modern platforms, and that are able to answer new challenges arising from applications, both efficiently—from a resource consumption point of view—and accurately—from a numerical point of view. For that, and even with increasing parallelism, we do not want to sacrifice in any manner numerical stability, based on threshold partial pivoting, one of the main originalities of our approach (our "trademark") in the context of direct solvers for distributed-memory computers; although this makes the parallelization more complicated, applying the same pivoting strategy as in the serial case ensures numerical robustness of our approach, which we generally measure in terms of sparse backward error. In order to solve the hard problems resulting from the always-increasing demands in simulations, special attention must also necessarily be paid to memory usage (and not only execution time). This requires specific algorithmic choices and scheduling techniques. From a complementary point of view, it is also necessary to be aware of the functionality requirements from the applications and from the users, so that robust solutions can be proposed for a wide range of applications.

Among direct methods, we rely on the multifrontal method [76], [77], [81]. This method usually exhibits a good data locality and hence is efficient in cache-based systems. The task graph associated with the multifrontal method is in the form of a tree whose characteristics should be exploited in a parallel implementation.

Our work is organized along two main research directions. In the first one we aim at efficiently addressing new architectures that include massive, hierarchical parallelism. In the second one, we aim at reducing the running time complexity and the memory requirements of direct solvers, while controlling accuracy.

3.3.2. Combinatorial scientific computing

Combinatorial scientific computing (CSC) is a recently coined term (circa 2002) for interdisciplinary research at the intersection of discrete mathematics, computer science, and scientific computing. In particular, it refers to the development, application, and analysis of combinatorial algorithms to enable scientific computing applications. CSC's deepest roots are in the realm of direct methods for solving sparse linear systems of equations where graph theoretical models have been central to the exploitation of sparsity, since the 1960s. The general approach is to identify performance issues in a scientific computing problem, such as memory use, parallel speed up, and/or the rate of convergence of a method, and to develop combinatorial algorithms and models to tackle those issues.

Our target scientific computing applications are (i) the preprocessing phases of direct methods (in particular MUMPS), iterative methods, and hybrid methods for solving linear systems of equations, and tensor decomposition algorithms; and (ii) the mapping of tasks (mostly the sub-tasks of the mentioned solvers) onto modern computing platforms. We focus on the development and use of graph and hypergraph models, and related tools such as hypergraph partitioning algorithms, to solve problems of load balancing and task mapping. We also focus on bipartite graph matching and vertex ordering methods for reducing the memory overhead and computational requirements of solvers. Although we direct our attention on these models and algorithms through the lens of linear system solvers, our solutions are general enough to be applied to some other resource optimization problems.

3.3.3. Dense linear algebra on post-petascale multicore platforms

The quest for efficient, yet portable, implementations of dense linear algebra kernels (QR, LU, Cholesky) has never stopped, fueled in part by each new technological evolution. First, the LAPACK library [70] relied on BLAS level 3 kernels (Basic Linear Algebra Subroutines) that enable to fully harness the computing power of a single CPU. Then the SCALAPACK library [69] built upon LAPACK to provide a coarse-grain parallel version, where processors operate on large block-column panels. Inter-processor communications occur through highly tuned MPI send and receive primitives. The advent of multi-core processors has led to a major modification in these algorithms [71], [86], [80]. Each processor runs several threads in parallel to keep all cores within that processor busy. Tiled versions of the algorithms have thus been designed: dividing large block-column panels into several tiles allows for a decrease in the granularity down to a level where many smaller-size tasks are spawned. In the current panel, the diagonal tile is used to eliminate all the lower tiles in the panel. Because the factorization of the whole panel is now broken into the elimination of several tiles, the update operations can also be partitioned at the tile level, which generates many tasks to feed all cores.

The number of cores per processor will keep increasing in the following years. It is projected that high-end processors will include at least a few hundreds of cores. This evolution will require to design new versions of libraries. Indeed, existing libraries rely on a static distribution of the work: before the beginning of the execution of a kernel, the location and time of the execution of all of its component is decided. In theory, static solutions enable to precisely optimize executions, by taking parameters like data locality into account. At run time, these solutions proceed at the pace of the slowest of the cores, and they thus require a perfect load-balancing. With a few hundreds, if not a thousand, cores per processor, some tiny differences between the computing times on the different cores ("jitter") are unavoidable and irremediably condemn purely static solutions. Moreover, the increase in the number of cores per processor once again mandates to increase the number of tasks that can be executed in parallel.

We study solutions that are part-static part-dynamic, because such solutions have been shown to outperform purely dynamic ones [73]. On the one hand, the distribution of work among the different nodes will still be statically defined. On the other hand, the mapping and the scheduling of tasks inside a processor will be dynamically defined. The main difficulty when building such a solution will be to design lightweight dynamic schedulers that are able to guarantee both an excellent load-balancing and a very efficient use of data locality.

3.4. Compilers and code optimization

Christophe Alias and Laure Gonnord asked to join the ROMA team temporarily, starting from September 2015. This was accepted by the team and by Inria. The text below describes their research domain. The results that they have achieved in 2015 are included in this report.

The advent of parallelism in supercomputers, in embedded systems (smartphones, plane controllers), and in more classical end-user computers increases the need for high-level code optimization and improved compilers. Being able to deal with the complexity of the upcoming software and hardware is one of the main challenges cited in the Hipeac Roadmap which among others cites the two major issues :

- Enhance the efficiency of the design of embedded systems, and especially the design of optimized specialized hardware.
- Invent techniques to "expose data movement in applications and optimize them at runtime and compile time and to investigate communication-optimized algorithms".

In particular, the rise of embedded systems and high performance computers in the last decade has generated new problems in code optimization, with strong consequences on the research area. The main challenge is to take advantage of the characteristics of the specific hardware (generic hardware, or hardware accelerators such as GPUs and FPGAs). The long-term objective is to provide solutions for the end-user developers to use at their best the huge opportunities of these emerging platforms.

3.4.1. Compiler algorithms for irregular applications

In the last decades, several frameworks has emerged to design efficient compiler algorithms. The efficiency of all the optimizations performed in compilers strongly relies on performant *static analyses* and *intermediate representations*. Among these representations, the polyhedral model [78] focus on regular programs, whose execution trace is predictable statically. The program and the data accessed are represented with a single mathematical object endowed with powerful algorithmic techniques for reasoning about it. Unfortunately, most of the algorithms used in scientific computing do not fit totally in this category.

We plan to explore the extensions of these techniques to handle irregular programs with while loops and complex data structures (such as trees, and lists). This raises many issues. We cannot represent finitely all the possible executions traces. Which approximation/representation to choose? Then, how to adapt existing techniques on approximated traces while preserving the correctness? To address these issues, we plan to incorporate new ideas coming from the abstract interpretation community: control flow, approximations, and also shape analysis; and from the termination community: rewriting is one of the major techniques that are able to handle complex data structures and also recursive programs.

3.4.2. *High-level synthesis for FPGA*

The major challenge of high-performance computing (HPC) is to reach the exaflop at the horizon 2020 with a power consumption bounded to 20 megawatts. To reach that goal, the flop/W must be increased drastically, which is unlikely to be achieved with the mainstream HPC technologies. FPGAs (Field Programmable Gate Arrays) are arrays of programmable logic cells – almost look-up tables, arithmetic, registers and steering logic, allowing to "program" a computer architecture. The last FPGA chip from Altera shows a peak performance of 30 Gflop/W, which is 7 times better than the best architecture of the top-green 500 contest [61]. This makes FPGA a key technology to reach the exaflop. Unfortunately, programming an FPGA is still a big challenge: the application must be defined at circuit level and use properly the logic cells. Hence, there is a strong need for a compiler technology able to *map complex applications specified in a high-level language*. This compiler technology is usually refered as high-level synthesis (HLS).

We plan to investigate how to extend the models and the algorithms developed by the HPC community to map automatically a complex application to an FPGA. This raises many issues. How to schedule/allocate the computations and the data on the FPGA to reduce the data transfers while keeping a high throughput? How to use optimally the resources of the FPGA while keeping a low critical path? To address these issues, we plan to develop novel execution models based on process networks and to extend the algorithms coming from the HPC compiler community (such as affine scheduling and data allocation, I/O optimization or source-level code generation) and the high-level synthesis community (such as datapath generation or control factorization).

4. Application Domains

4.1. Applications of sparse direct solvers

Sparse direct (multifrontal) solvers have a wide range of applications as they are used at the heart of many numerical methods in computational science: whether a model uses finite elements or finite differences, or requires the optimization of a complex linear or nonlinear function, one often ends up solving a linear system of equations involving sparse matrices. There are therefore a number of application fields, among which some of the ones cited by the users of our sparse direct solver MUMPS (see Section 6.1) are: structural mechanics, biomechanics, medical image processing, tomography, geophysics, electromagnetism, fluid dynamics, econometric models, oil reservoir simulation, magneto-hydro-dynamics, chemistry, acoustics, glaciology, astrophysics, circuit simulation, and work on hybrid direct-iterative methods.

5. Highlights of the Year

5.1. Highlights of the Year

Yves Robert co-edited with Thomas Hérault (University of Tennessee, Knoxville) the book *Fault-Tolerance Techniques for High-Performance Computing* [38], which was published in May by Springer.

The version 5.0.0 of MUMPS was released in February 2015.

6. New Software and Platforms

6.1. MUMPS

A MUltifrontal Massively Parallel Solver
KEYWORDS: High-Performance Computing - Direct solvers - Finite element modelling
FUNCTIONAL DESCRIPTION

MUMPS is a software library to solve large sparse linear systems (AX=B) on sequential and parallel distributed memory computers. It implements a sparse direct method called the multifrontal method. It is used worldwide in academic and industrial codes, in the context numerical modeling of physical phenomena with finite elements. Its main characteristics are its numerical stability, its large number of features, its high performance and its constant evolution through research and feedback from its community of users. Examples of application fields include structural mechanics, electromagnetism, geophysics, acoustics, computational fluid dynamics. MUMPS has been developed by INPT(ENSEEIHT)-IRIT, Inria, CERFACS, University of Bordeaux, CNRS and ENS Lyon.

- Participants: Patrick Amestoy, Alfredo Buttari, Jean-Yves L'Excellent, Chiara Puglisi, Mohamed Sid-Lakhdar, Bora Uçar, Marie Durand, Abdou Guermouche, Maurice Bremond, Guillaume Joslin, Stéphane Pralet, Aurélia Fevre, Clément Weisbecker, Theo Mary, Emmanuel Agullo, Jacko Koster, Tzvetomila Slavova and François-Henry Rouet
- Partners: Université de Bordeaux - CNRS - CERFACS - ENS Lyon - INPT - IRIT - Université de Lyon - Université de Toulouse - LIP
- Contact: Jean-Yves L'Excellent
- Public releases in 2015: MUMPS 5.0.0 (February 2015), including major improvements in terms of performance and robustness, and MUMPS 5.0.1 (July 2015)
- URL: http://mumps-solver.org/

Following the creation in 2014 of a consortium for industrial users of MUMPS (http://mumps-consortium.org), some collaborations with industry (scientific exchanges, support, releases in advance) are mentioned in Section 8.1. We pursued our work on block low-rank solvers [2] (Section 7.13), which was extended and applied to 3D frequency domain seismic modeling [19], [18] (Section 7.15) in the context of an on-going collaboration with the Seiscope consortium (https://seiscope2.obs.ujf-grenoble.fr/?lang=en?). We also worked on the parallel computation of selected entries of the inverse of a sparse matrix [3] (Section 7.14).

6.2. DCC

DPN C Compiler
KEYWORDS: Polyhedral compilation - Automatic parallelization - High-level synthesis
FUNCTIONAL DESCRIPTION

Dcc (Data-aware process network C compiler) analyzes a sequential regular program written in C and generates an equivalent architecture of parallel computer as a communicating process network (Data-aware Process Network, DPN). Internal communications (channels) and external communications (external memory) are automatically handled while fitting optimally the characteristics of the global memory (latency and throughput). The parallelism can be tuned. Dcc has been registered at the APP ("Agence de protection des programmes") and transferred to the XtremLogic start-up under an Inria license.

- Participants: Christophe Alias and Alexandru Plesco
- Contact: Christophe Alias
- Software transferred by Inria under an exclusive license, no web page.

6.3. PoCo

Polyhedral Compilation library
KEYWORDS: Polyhedral compilation - Automatic parallelization
FUNCTIONAL DESCRIPTION

PoCo (Polyhedral Compilation library) is a compilation framework allowing to develop parallelizing compilers for regular programs. PoCo features many state-of-the-art polyhedral program analysis (dependences, affine scheduling, copde generation) and a symbolic calculator on execution traces (represented as convex polyhedra). PoCo has been registered at the APP ("agence de protection des programmes") and transferred to the XtremLogic start-up under an Inria licence.

- Participant: Christophe Alias
- Contact: Christophe Alias
- Software transferred by Inria under an exclusive license, no web page.

6.4. Aspic

Accelerated Symbolic Polyhedral Invariant Geneneration
KEYWORDS: Abstract Interpretation - Invariant Generation
FUNCTIONAL DESCRIPTION

Aspic is an invariant generator for general counter automata. Used with C2fsm (a tool developed by P. Feautrier in COMPSYS), it can be used to derivate invariants for numerical C programs, and also to prove safety. It is also part of the WTC toolsuite (see http://compsys-tools.ens-lyon.fr/wtc/index.html), a tool chain to compute worse-case time complexity of a given sequential program.

Aspic implements the theoretical results of Laure Gonnord's PhD thesis on acceleration techniques and has been maintained since 2007.

- Participant: Laure Gonnord
- Contact: Laure Gonnord
- URL: http://laure.gonnord.org/pro/aspic/aspic.html

6.5. Termite

Termination of C programs

KEYWORDS: Abstract Interpretation - Termination
FUNCTIONAL DESCRIPTION

TERMITE is the implementation of our new algorithm "Counter-example based generation of ranking functions" (see Section 7.29). Based on LLVM and Pagai (a tool that generates invariants), the tool automatically generates a ranking function for each *head of loop*.

TERMITE represents 3000 lines of OCaml and is now available via the opam installer.

- Participants: Laure Gonnord, Gabriel Radanne (PPS, Univ Paris 7), David Monniaux (CNRS/Verimag).
- Contact: Laure Gonnord
- URL: https://termite-analyser.github.io/

6.6. Vaphor

Validation of C programs with arrays with Horn Clauses

KEYWORDS: Abstract Interpretation - Safety - Array Programs
FUNCTIONAL DESCRIPTION

VAPHOR (Validation of Programs with Horn Clauses) is the implementation of our new algorithm "An encoding of array verification problems into array-free Horn clauses" (see Section 7.30). The tool implements a traduction from a C-like imperative language into Horn clauses in the SMT-lib Format.

VAPHOR represents 2000 lines of OCaml and its development is under consolidation.

- Participants: Laure Gonnord, David Monniaux (CNRS/Verimag).
- Contact: Laure Gonnord
- Software not yet published, under consolidation.

7. New Results

7.1. Scheduling computational workflows on failure-prone platforms

Participants: Guillaume Aupy, Anne Benoit, Henri Casanova [University of Hawaii], Yves Robert.

We study the scheduling of computational workflows on compute resources that experience exponentially distributed failures. When a failure occurs, rollback and recovery is used to resume the execution from the last checkpointed state. The scheduling problem is to minimize the expected execution time by deciding in which order to execute the tasks in the workflow and whether to checkpoint or not checkpoint a task after it completes. We give a polynomial-time algorithm for fork graphs and show that the problem is NP-complete with join graphs. Our main result is a polynomial-time algorithm to compute the execution time of a workflow with specified to-be-checkpointed tasks. Using this algorithm as a basis, we propose efficient heuristics for solving the scheduling problem. We evaluate these heuristics for representative workflow configurations.

This work has been published in the 17th Workshop on Advances in Parallel and Distributed Computational Models [20].

7.2. Efficient checkpoint/verification patterns

Participants: Anne Benoit, Saurabh K. Raina [Jaypee Institute of Information Technology], Yves Robert.

Errors have become a critical problem for high performance computing. Checkpointing protocols are often used for error recovery after fail-stop failures. However, silent errors cannot be ignored, and their peculiarity is that such errors are identified only when the corrupted data is activated. To cope with silent errors, we need a verification mechanism to check whether the application state is correct. Checkpoints should be supplemented with verifications to detect silent errors. When a verification is successful, only the last checkpoint needs to be kept in memory because it is known to be correct.

In this work, we analytically determine the best balance of verifications and checkpoints so as to optimize platform throughput. We introduce a balanced algorithm using a pattern with p checkpoints and q verifications, which regularly interleaves both checkpoints and verifications across same-size computational chunks. We show how to compute the waste of an arbitrary pattern, and we prove that the balanced algorithm is optimal when the platform MTBF (Mean Time Between Failures) is large in front of the other parameters (checkpointing, verification and recovery costs). We conduct several simulations to show the gain achieved by this balanced algorithm for well-chosen values of p and q, compared to the base algorithm that always perform a verification just before taking a checkpoint ($p = q = 1$), and we exhibit gains of up to 19%.

This work has been published in the International Journal of High Performance Computing Applications [8].

7.3. Assessing the impact of partial verifications against silent data corruptions

Participants: Aurélien Cavelan, Saurabh K. Raina [Jaypee Institute of Information Technology], Yves Robert, Hongyang Sun.

Silent errors, or silent data corruptions, constitute a major threat on very large scale platforms. When a silent error strikes, it is not detected immediately but only after some delay, which prevents the use of pure periodic checkpointing approaches devised for fail-stop errors. Instead, checkpointing must be coupled with some verification mechanism to guarantee that corrupted data will never be written into the checkpoint file. Such a guaranteed verification mechanism typically incurs a high cost. In this work, we assess the impact of using partial verification mechanisms in addition to a guaranteed verification. The main objective is to investigate to which extent it is worthwhile to use some light cost but less accurate verifications in the middle of a periodic computing pattern, which ends with a guaranteed verification right before each checkpoint. Introducing partial verifications dramatically complicates the analysis, but we are able to analytically determine the optimal computing pattern (up to the first-order approximation), including the optimal length of the pattern, the optimal number of partial verifications, as well as their optimal positions inside the pattern. Performance evaluations based on a wide range of parameters confirm the benefit of using partial verifications under certain scenarios, when compared to the baseline algorithm that uses only guaranteed verifications.

This work has been published in the proceedings of ICPP'15 [22].

7.4. Which Verification for Soft Error Detection?

Participants: Leonardo Bautista-Gomez [Argonne National Laboratory], Anne Benoit, Aurélien Cavelan, Saurabh K. Raina [Jaypee Institute of Information Technology], Yves Robert, Hongyang Sun.

This work is an extension of the work described in Section 7.4 to cope with imperfect verifications. Many methods are available to detect silent errors in high-performance computing (HPC) applications. Each comes with a given cost and recall (fraction of all errors that are actually detected). The main contribution of this work is to characterize the optimal computational pattern for an application: which detector(s) to use, how many detectors of each type to use, together with the length of the work segment that precedes each of them. We conduct a comprehensive complexity analysis of this optimization problem, showing NP-completeness and designing an FPTAS (Fully Polynomial-Time Approximation Scheme). On the practical side, we provide a greedy algorithm whose performance is shown to be close to the optimal for a realistic set of evaluation scenarios.

This work has been published in the proceedings of HiPC'15 [21].

7.5. Composing resilience techniques: ABFT, periodic and incremental checkpointing

Participants: George Bosilca [University of Tennessee, Knoxville], Aurélien Bouteiller [University of Tennessee, Knoxville], Thomas Hérault [University of Tennessee, Knoxville], Yves Robert, Jack Dongarra [University of Tennessee, Knoxville].

Algorithm Based Fault Tolerant (ABFT) approaches promise unparalleled scalability and performance in failure-prone environments. Thanks to recent advances in the understanding of the involved mechanisms, a growing number of important algorithms (including all widely used factorizations) have been proven ABFT-capable. In the context of larger applications, these algorithms provide a temporal section of the execution, where the data is protected by its own intrinsic properties, and can therefore be algorithmically recomputed without the need of checkpoints. However, while typical scientific applications spend a significant fraction of their execution time in library calls that can be ABFT-protected, they interleave sections that are difficult or even impossible to protect with ABFT. As a consequence, the only practical fault-tolerance approach for these applications is checkpoint/restart. In this work, we propose a model to investigate the efficiency of a composite protocol, that alternates between ABFT and checkpoint/restart for the effective protection of an iterative application composed of ABFT-aware and ABFT-unaware sections. We also consider an incremental checkpointing composite approach in which the algorithmic knowledge is leveraged by a novel optimal dynamic programming to compute checkpoint dates. We validate these models using a simulator. The model and simulator show that the composite approach drastically increases the performance delivered by an execution platform, especially at scale, by providing the means to increase the interval between checkpoints while simultaneously decreasing the volume of each checkpoint.

This work has been published in the International Journal of Networking and Computing [9].

7.6. Voltage Overscaling Algorithms for Energy-Efficient Workflow Computations With Timing Errors

Participants: Aurélien Cavelan, Yves Robert, Hongyang Sun, Frédéric Vivien.

We proposed a software-based approach using dynamic voltage overscaling to reduce the energy consumption of HPC applications. This technique aggressively lowers the supply voltage below nominal voltage, which introduces timing errors, and we used Algorithm-Based Fault-Tolerance (ABFT) to provide fault tolerance for matrix operations. We introduced a formal model, and we designed optimal polynomial-time solutions, to execute a linear chain of tasks. Evaluation results obtained for matrix multiplication demonstrated that our approach indeed leads to significant energy savings, compared to the standard algorithm that always operates at nominal voltage.

This work has been published in the proceedings of the 5th Workshop on Fault Tolerance for HPC at eXtreme Scale [23].

7.7. Approximation algorithms for energy, reliability and makespan optimization problems

Participants: Guillaume Aupy, Anne Benoit.

We consider the problem of scheduling an application on a parallel computational platform. The application is a particular task graph, either a linear chain of tasks, or a set of independent tasks. The platform is made of identical processors, whose speed can be dynamically modified. It is also subject to failures: if a processor is slowed down to decrease the energy consumption, it has a higher chance to fail. Therefore, the scheduling problem requires us to re-execute or replicate tasks (i.e., execute twice the same task, either on the same processor, or on two distinct processors), in order to increase the reliability. It is a tri-criteria problem: the goal is to minimize the energy consumption, while enforcing a bound on the total execution time (the makespan), and a constraint on the reliability of each task.

Our main contribution is to propose approximation algorithms for linear chains of tasks and independent tasks. For linear chains, we design a fully polynomial-time approximation scheme. However, we show that there exists no constant factor approximation algorithm for independent tasks, unless P=NP, and we propose in this case an approximation algorithm with a relaxation on the makespan constraint.

This work has been published in the Parallel Processing Letters [4].

7.8. Co-scheduling algorithms for high-throughput workload execution

Participants: Guillaume Aupy, Manu Shantharam [San Diego Supercomputer Center], Anne Benoit, Yves Robert, Padma Raghavan [Penn State University].

This work investigates co-scheduling algorithms for processing a set of parallel applications. Instead of executing each application one by one, using a maximum degree of parallelism for each of them, we aim at scheduling several applications concurrently. We partition the original application set into a series of packs, which are executed one by one. A pack comprises several applications, each of them with an assigned number of processors, with the constraint that the total number of processors assigned within a pack does not exceed the maximum number of available processors. The objective is to determine a partition into packs, and an assignment of processors to applications, that minimize the sum of the execution times of the packs.

We thoroughly study the complexity of this optimization problem, and propose several heuristics that exhibit very good performance on a variety of workloads, whose application execution times model profiles of parallel scientific codes. We show that co-scheduling leads to to faster workload completion time and to faster response times on average (hence increasing system throughput and saving energy), for significant benefits over traditional scheduling from both the user and system perspectives.

This work has been published in the Journal of Scheduling [6].

7.9. Scheduling the I/O of HPC Applications Under Congestion

Participants: Ana Gainaru [University of Illinois at Urbana Champaign], Guillaume Aupy, Anne Benoit, Franck Cappello, Yves Robert.

A significant percentage of the computing capacity of large-scale platforms is wasted due to interferences incurred by multiple applications that access a shared parallel file system concurrently. One solution to handling I/O bursts in large-scale HPC systems is to absorb them at an intermediate storage layer consisting of burst buffers. However, our analysis of the Argonne's Mira system shows that burst buffers cannot prevent congestion at all times. As a consequence, I/O performance is dramatically degraded, showing in some cases a decrease in I/O throughput of 67%.

In this work, we analyze the effects of interference on application I/O bandwidth, and propose several scheduling techniques to mitigate congestion. We focus on typical HPC applications, which have a periodic pattern consisting of some amount of computation followed by some volume of I/O to be transferred. We show through extensive experiments that our global I/O scheduler is able to reduce the effects of congestion, even on systems where burst buffers are used, and can increase the overall system throughput up to 56%. We also show that it outperforms current Mira I/O schedulers, even for non-periodic applications.

This work has been published in IPDPS'15 [26].

7.10. Scheduling trees of malleable tasks for sparse linear algebra

Participants: Abdou Guermouche [Univ. Bordeaux/Inria Bordeaux Sud-Ouest], Loris Marchal, Bertrand Simon, Oliver Sinnen [Univ. Auckland/New Zealand], Frédéric Vivien.

Scientific workloads are often described by directed acyclic task graphs. This is in particular the case for multifrontal factorization of sparse matrices —the focus of this work— whose task graph is structured as a tree of parallel tasks. Prasanna and Musicus [84], [85] advocated using the concept of *malleable* tasks to model parallel tasks involved in matrix computations. In this powerful model each task is processed on a time-varying number of processors. Following Prasanna and Musicus, we consider malleable tasks whose speedup is p^{α}, where p is the fractional share of processors on which a task executes, and α $(0 < \alpha \leq 1)$ is a task-independent parameter. Firstly, we use actual experiments on multicore platforms to motivate the relevance of this model for our application. Then, we study the optimal time-minimizing allocation proposed by Prasanna and Musicus using optimal control theory. We greatly simplify their proofs by resorting only to pure scheduling arguments. Building on the insight gained thanks to these new proofs, we extend the study to distributed (homogeneous or heterogeneous) multicore platforms. We prove the NP-completeness of the corresponding scheduling problem, and we then propose some approximation algorithms [28].

In a second step, we studied a simplified speed-up function for malleable tasks, corresponding to perfect parallelism for a number of processors below a given threshold. The threshold depends on the task. We proved that scheduling independent chains of malleable tasks under this model is NP-complete. We study the performance of a classical allocation policy which is agnostic of the threshold and a simple greedy heuristic, and proved that both are 2-approximation algorithms, even if in practice, the latter often outperforms the former.

7.11. Parallel scheduling of task trees with limited memory

Participants: Clément Brasseur [ENS Lyon], Guillaume Aupy, Loris Marchal.

Scientific workloads are often described by directed acyclic task graphs. This is in particular the case for multifrontal factorization of sparse matrices —the focus of this work— whose task graph is structured as a tree of parallel tasks. When processing this tree on a multicore machine, we have to find a tradeoff between task parallelism and memory usage. In this context, Agullo et al. [62] proposed an activation scheme which follows a postorder traversal and books the memory needed for the task. This strategy has a low complexity and thus has been implemented in the lightweight runtime system StarPU [65], but may lead to excessive memory booking, which limits the task parallelism. In this work, we proposed a new booking strategy that books exactly what is necessary for a task, given what is already booked by its predecessors in the tree. We have shown by extensive simulations on realistic trees that this leads to better task parallelism and reduces the overall processing time.

7.12. Locality of Map tasks in MapReduce computations

Participants: Olivier Beaumont [Inria Bordeaux Sud-Ouest], Loris Marchal.

In data parallel system such as MapReduce, large data files are distributed among the storage attached to computing nodes, and the computation is afterwards allocated close to the data whenever it is possible. Several parameters may affect the locality of the data, and thus the amount of data that needs to be communicated during the computation: the possible replication of the data when it is distributed on the platform, and the load-balancing mechanism that transmits new data to node which have exhausted their own data. In this work, we have proposed a simple analytical model to estimate the amount of data transfer of various scenarios for the Map phase of MapReduce computations and we have validated this model using simulations.

7.13. Improving multifrontal methods by means of block low-rank representations

Participants: Patrick Amestoy [INPT-IRIT, Université of Toulouse], Cleve Ashcraft [LSTC], Olivier Boiteau [EDF], Alfredo Buttari [CNRS-IRIT, Université of Toulouse], Jean-Yves L'Excellent, Clément Weisbecker [INPT-IRIT, now at LSTC].

Matrices coming from elliptic Partial Differential Equations (PDEs) have been shown to have a low-rank property: well defined off-diagonal blocks of their Schur complements can be approximated by low-rank products. Given a suitable ordering of the matrix which gives the blocks a geometrical meaning, such approximations can be computed using an SVD or a rank-revealing QR factorization. The resulting representation offers a substantial reduction of the memory requirement and gives efficient ways to perform many of the basic dense linear algebra operations.

Several strategies, mostly based on hierarchical formats, have been proposed to exploit this property. We study a simple, non-hierarchical, low-rank format called Block Low-Rank (BLR), and explain how it can be used to reduce the memory footprint and the complexity of sparse direct solvers based on the multifrontal method. We present experimental results on matrices coming from elliptic PDEs and from various other applications. We show that even if BLR based factorizations are asymptotically less efficient than hierarchical approaches, they still deliver considerable gains. The BLR format is compatible with numerical pivoting, and its simplicity and flexibility make it easy to use in the context of a general purpose, algebraic solver. This work has been published in the SIAM Journal on Scientific Computing [2].

7.14. Parallel Computation of a subset of entries of the inverse

Participants: Patrick Amestoy [INPT-IRIT, Université of Toulouse], Iain Duff [RAL and CERFACS], Jean-Yves L'Excellent, François-Henry Rouet.

We consider the computation in parallel of several entries of the inverse of a large sparse matrix. We assume that the matrix has already been factorized by a direct method and that the factors are distributed. Entries are efficiently computed by exploiting sparsity of the right-hand sides and the solution vectors in the triangular solution phase. We demonstrate that in this setting, parallelism and computational efficiency are two contrasting objectives. We develop an efficient approach and show its efficiency on a general purpose parallel multifrontal solver. This work has been published in the SIAM Journal on Scientific Computing [3].

7.15. Efficient 3D frequency-domain seismic modeling with a parallel block low-rank (BLR) direct solver

Participants: Patrick Amestoy [INPT-IRIT, University of Toulouse], Romain Brossier [ISTerre, University of Grenoble-Alpes], Alfredo Buttari [CNRS-IRIT, University of Toulouse], Jean-Yves L'Excellent, Théo Mary [UPS-IRIT, University of Toulouse], Ludovic Métivier [ISTerre-JK-CNRS], Alain Miniussi [Geoazur-CNRS-UNSA], Stéphane Operto [Geoazur-CNRS-UNSA], Alessandra Ribodetti [Geoazur-CNRS-UNSA], Jean Virieux [ISTerre-UJF, University of Grenoble-Alpes], Clément Weisbecker [INPT-IRIT, now at LSTC].

Three-dimensional frequency-domain full waveform inversion (FWI) of fixed-spread data can be efficiently performed in the visco-acoustic approximation when seismic modeling is based on a sparse direct solver. Based on the work in [3] and its extension to a parallel environment, we studied the application of a parallel algebraic Block Low-Rank (BLR) multifrontal solver providing an approximate solution of the time-harmonic wave equation with a reduced operation count, memory demand, and volume of communication relative to the full-rank solver. We analyzed the parallel efficiency and the accuracy of the solver with a realistic FWI case [19]. The application of this parallel BLR solver to a real data case from the North Sea for full waveform inversion of ocean-bottom cable data was also presented in [18], where a multiscale frequency-domain FWI is applied by successive inversions of 11 discrete frequencies in the 3.5Hz-10Hz frequency band. The velocity model built by FWI reveals short-scale features such as channels, scrapes left by drifting icebergs, fractures and deep reflectors below the reservoir level, alhough the presence of gas in the overburden. The quality of the FWI results is controlled by time-domain modeling and source wavelet estimation. This work was done in the context of an on-going collaboration with the Seiscope consortium (https://seiscope2.obs.ujf-grenoble.fr/?lang=en?).

7.16. Approximation algorithms for bipartite matching on multicore architectures

Participants: Fanny Dufossé [DOLPHIN/Inria Lille - Nord Europe], Kamer Kaya [BMI, The Ohio State Univ., USA], Bora Uçar.

We proposed [13] two heuristics for the bipartite matching problem that are amenable to shared-memory parallelization. The first heuristic is very intriguing from a parallelization perspective. It has no significant algorithmic synchronization overhead and no conflict resolution is needed across threads. We showed that this heuristic has an approximation ratio of around 0.632 under some common conditions. The second heuristic was designed to obtain a larger matching by employing the well-known Karp-Sipser heuristic on a judiciously chosen subgraph of the original graph. We showed that the Karp-Sipser heuristic always finds a maximum cardinality matching in the chosen subgraph. Although the Karp-Sipser heuristic is hard to parallelize for general graphs, we exploited the structure of the selected subgraphs to propose a specialized implementation which demonstrates very good scalability. We proved that this second heuristic has an approximation guarantee of around 0.866 under the same conditions as in the first algorithm. We discussed parallel implementations of the proposed heuristics on a multicore architecture. Experimental results, for demonstrating speed-ups and verifying the theoretical results in practice, were also provided.

7.17. Hypergraph partitioning for multiple communication cost metrics

Participants: Mehmet Deveci [BMI, The Ohio State Univ., USA], Kamer Kaya [BMI, The Ohio State Univ., USA], Umit V. Çatalyürek [BMI, The Ohio State Univ., USA], Bora Uçar.

We investigated [12] hypergraph partitioning-based methods for efficient parallelization of communicating tasks. A good partitioning method should divide the load among the processors as evenly as possible and minimize the inter-processor communication overhead. The total communication volume is the most popular communication overhead metric which is reduced by the existing state-of-the-art hypergraph partitioners. However, other metrics such as the total number of messages, the maximum amount of data transferred by a processor, or a combination of them are equally, if not more, important. Existing hypergraph-based solutions use a two phase approach to minimize such metrics where in each phase, they minimize a different metric, sometimes at the expense of others. We proposed a one-phase approach where all the communication cost metrics can be effectively minimized in a multi-objective setting and reductions can be achieved for all metrics together. For an accurate modeling of the maximum volume and the number of messages sent and received by a processor, we proposed the use of directed hypergraphs. The directions on hyperedges necessitate revisiting the standard partitioning heuristics. We did so and proposed a multi-objective, multi-level hypergraph partitioner. The partitioner takes various prioritized communication metrics into account, and optimizes all of them together in the same phase. Compared to the state-of-the-art methods which only minimize the total communication volume, we showed on a large number of problem instances that the new method produced better partitions in terms of several communication metrics.

7.18. Comments on the hierarchically structured bin packing problem

Participants: Thomas Lambert [Inria Bordeaux Sud-Ouest], Loris Marchal, Bora Uçar.

We studied [16] the hierarchically structured bin packing problem. In this problem, the items to be packed into bins are at the leaves of a tree. The objective of the packing is to minimize the total number of bins into which the descendants of an internal node are packed, summed over all internal nodes. We investigated an existing algorithm and made a correction to the analysis of its approximation ratio. Further results regarding the structure of an optimal solution and a strengthened inapproximability result were given.

7.19. Semi-two-dimensional partitioning for parallel sparse matrix-vector multiplication

Participants: Enver Kayaaslan, Cevdet Aykanat [Bilkent Univ., Turkey], Bora Uçar.

We proposed [31] a novel sparse matrix partitioning scheme, called semi-two-dimensional (s2D), for efficient parallelization of sparse matrix-vector multiply (SpMV) operations on distributed memory systems. In s2D, matrix nonzeros are more flexibly distributed among processors than one dimensional (rowwise or columnwise) partitioning schemes. Yet, there is a constraint which renders s2D less flexible than two-dimensional (nonzero based) partitioning schemes. The constraint is enforced to confine all communication operations in a single phase, as in 1D partition, in a parallel SpMV operation. In a positive view, s2D thus can be seen as being close to 2D partitions in terms of flexibility, and being close to 1D partitions in terms of computation/communication organization. We described two methods that take partitions on the input and output vectors of SpMV and produce s2D partitions while reducing the total communication volume. The first method obtains optimal total communication volume, while the second one heuristically reduces this quantity and takes computational load balance into account. We demonstrated that the proposed partitioning method improves the performance of parallel SpMV operations both in theory and practice with respect to 1D and 2D partitionings.

7.20. Combining backward and forward recovery to cope with silent errors in iterative solvers

Participants: Massimiliano Fasi [Univ Manchester, UK], Julien Langou [Univ. Colorado Denver, USA], Yves Robert, Bora Uçar.

We proposed combining checkpointing and verification for coping with silent errors in iterative solvers. We used algorithm based fault tolerance for error detection and error correction, allowing a forward recovery (and no rollback nor re-execution) when a single error is detected. We introduced an abstract performance model to compute the performance of all schemes, and we instantiated it using the Conjugate Gradient (CG) algorithm. Finally, we validate our new approach through a set of simulations both in normal and preconditioned CG [48], [25], [47].

7.21. Load-balanced local time stepping for large-scale wave propagation

Participants: Max Rietmann [Univ. Lugano, CH], Daniel Peter [Univ. Lugano, CH], Olaf Schenk [Univ. Lugano, CH], Bora Uçar, Marcus J. Grote [Univ. Basel, CH].

In complex acoustic or elastic media, finite element meshes often require regions of refinement to honor external or internal topography, or small-scale features. These localized smaller elements create a bottleneck for explicit time-stepping schemes due to the Courant-Friedrichs-Lewy stability condition. Recently developed local time stepping (LTS) algorithms reduce the impact of these small elements by locally adapting the time-step size to the size of the element. The recursive, multi-level nature of our LTS scheme introduces an additional challenge, as standard partitioning schemes create a strong load imbalance across processors. We examined [33] the use of multi-constraint graph and hypergraph partitioning tools to achieve effective, load-balanced parallelization. We implemented LTS-Newmark in the seismology code SPECFEM3D and compared performance and scalability between different partitioning tools on CPU and GPU clusters using examples from computational seismology.

7.22. Fast and high quality topology-aware task mapping

Participants: Mehmet Deveci [BMI, The Ohio State Univ., USA], Kamer Kaya [BMI, The Ohio State Univ., USA], Umit V. Çatalyürek [BMI, The Ohio State Univ., USA], Bora Uçar.

Considering the large number of processors and the size of the interconnection networks on exascale-capable supercomputers, mapping concurrently executable and communicating tasks of an application is a complex problem that needs to be dealt with care. For parallel applications, the communication overhead can be a significant bottleneck on scalability. Topology-aware task-mapping methods that map the tasks to the processors (i.e., cores) by exploiting the underlying network information are very effective to avoid, or at worst bend, this limitation. We proposed [24] novel, efficient, and effective task mapping algorithms employing a graph model. The experiments showed that the methods are faster than the existing approaches proposed for the same task, and on 4096 processors, the algorithms improved the communication hops and link contentions by 16% and 32%, respectively, on the average. In addition, they improved the average execution time of a parallel SpMV kernel and a communication-only application by 9% and 14%, respectively.

7.23. Distributed memory tensor computations

Participants: Oguz Kaya, Bora Uçar.

There are two prominent tensor decomposition formulations. CANDECOMP/PARAFAC (CP) formulation approximates a tensor as a sum of rank-one tensors. *Tucker* formulation approximates a tensor with a core tensor multiplied by a matrix along each mode. Both of these formulations have uses in applications. The most common algorithms for both decompositions are based on the alternating least squares method. The algorithms of this type are iterative, where the computational core of an iteration is a special operation operation between an N-mode tensor and N matrices. These key operations are called the matricized tensor times Khatri-Rao product (MTTKRP) in the CP-ALS case, and the n-mode product in the Tucker decomposition case. We have investigated efficient parallelizations of full fledged algorithms for obtaining these two decompositions in distributed memory systems [30], [51] with a special focus on the mentioned key operations. In both studies, hypergraphs are used for computational load balancing and communication cost reduction. We are currently finalizing our last touches on the Tucker decomposition algorithms [51] to submit it to a conference. We are also working towards a unified view of the parallelization of the two algorithms. This work with its whole extend is carried out in the context of the thesis of Oguz Kaya.

7.24. Bridging the gap between performance and bounds of Cholesky factorization on heterogeneous platforms

Participants: Emmanuel Agullo [Inria Bordeaux Sud-Ouest], Olivier Beaumont [Inria Bordeaux Sud-Ouest], Lionel Eyraud-Dubois [Inria Bordeaux Sud-Ouest], Julien Herrmann, Suraj Kumar [Inria Bordeaux Sud-Ouest], Loris Marchal, Samuel Thibault [Inria Bordeaux Sud-Ouest].

In this work, we consider the problem of allocating and scheduling dense linear application on fully heterogeneous platforms made of CPUs and GPUs. More specifically, we focus on the Cholesky factorization since it exhibits the main features of such problems. Indeed, the relative performance of CPU and GPU highly depends on the sub-routine: GPUs are for instance much more efficient to process regular kernels such as matrix-matrix multiplications rather than more irregular kernels such as matrix factorization. In this context, one solution consists in relying on dynamic scheduling and resource allocation mechanisms such as the ones provided by PaRSEC or StarPU. We analyze the performance of dynamic schedulers based on both actual executions and simulations, and we investigate how adding static rules based on an offline analysis of the problem to their decision process can indeed improve their performance, up to reaching some improved theoretical performance bounds which we introduce [17].

7.25. Assessing the cost of redistribution followed by a computational kernel: Complexity and performance results

Participants: Julien Herrmann, George Bosilca [University of Tennessee, Knoxville], Thomas Hérault [University of Tennessee, Knoxville], Loris Marchal, Yves Robert, Jack Dongarra [University of Tennessee, Knoxville].

The classical redistribution problem aims at optimally scheduling communications when reshuffling from an initial data distribution to a target data distribution. This target data distribution is usually chosen to optimize some objective for the algorithmic kernel under study (good computational balance or low communication volume or cost), and therefore to provide high efficiency for that kernel. However, the choice of a distribution minimizing the target objective is not unique. This leads to generalizing the redistribution problem as follows: find a re-mapping of data items onto processors such that the data redistribution cost is minimal, and the operation remains as efficient. This work studies the complexity of this generalized problem. We compute optimal solutions and evaluate, through simulations, their gain over classical redistribution. We also show the NP-hardness of the problem to find the optimal data partition and processor permutation (defined by new subsets) that minimize the cost of redistribution followed by a simple computational kernel. Finally, experimental validation of the new redistribution algorithms are conducted on a multicore cluster, for both a 1D-stencil kernel and a more compute-intensive dense linear algebra routine.

This work has been published in the Parallel Computing journal [15].

7.26. STS-k: A Multi-level Sparse Triangular Solution Scheme for NUMA Multicores

Participants: Humayun Kabir [Penn State University], Joshua Booth [Sandia National Laboratories], Guillaume Aupy, Anne Benoit, Yves Robert, Padma Raghavan [Penn State University].

We consider techniques to improve the performance of parallel sparse triangular solution on non-uniform memory architecture multicores by extending earlier coloring and level set schemes for single-core multiprocessors. We develop STS-k, where k represents a small number of transformations for latency reduction from increased spatial and temporal locality of data accesses. We propose a graph model of data reuse to inform the development of STS-k and to prove that computing an optimal cost schedule is NP-complete. We observe significant speed-ups with STS-3 on 32-core Intel Westmere-EX and 24-core AMD 'MagnyCours' processors. Incremental gains solely from the 3-level transformations in STS-3 for a fixed ordering, correspond to reductions in execution times by factors of 1.4 (Intel) and 1.5 (AMD) for level sets and 2 (Intel) and 2.2 (AMD) for coloring. On average, execution times are reduced by a factor of 6 (Intel) and 4 (AMD) for STS-3 with coloring compared to a reference implementation using level sets.

This work has been published in SC'15 [29].

7.27. Mono-parametric Tiling

Participants: Guillaume Iooss [Inria/ENS-Lyon/UCBL/CNRS], Sanjay Rajopadhye [Colorado State University], Christophe Alias, Yun Zou [Colorado State University].

Tiling is a crucial program transformation with many benefits: it improves locality, exposes parallelism, allows for adjusting the ops-to-bytes balance of codes, and can be applied at multiple levels. Allowing tile sizes to be symbolic parameters at compile time has many benefits, including efficient autotuning, and run-time adaptability to system variations. For polyhedral programs, parametric tiling in its full generality is known to be non-linear, breaking the mathematical closure properties of the polyhedral model. Most compilation tools therefore either avoid it by only performing fixed size tiling, or apply it in only the final, code generation step. Both strategies have limitations.

We first introduce mono-parametric partitioning, a restricted parametric, tiling-like transformation which can be used to express a tiling. We show that, despite being parametric, it is a polyhedral transformation. We first prove that applying mono-parametric partitioning (i) to a polyhedron yields a union of polyhedra, and (ii) to an affine function produces a piecewise-affine function. We then use these properties to show how to partition an entire polyhedral program, including one with reductions. Next, we generalize this transformation to tiles with arbitrary tile shapes that can tesselate the iteration space (e.g., hexagonal, trapezoidal, etc). We show how mono-parametric tiling can be applied at multiple levels, and enables a wide range of polyhedral analyses and transformations to be applied.

This work has been published as an Inria research report [49] and will be submitted to a journal.

7.28. Data-aware Process Networks

Participants: Christophe Alias, Alexandru Plesco [XtremLogic SAS].

High-level circuit synthesis (HLS, high-level synthesis) consists in compiling a C-like high-level program to a circuit. The circuit must be as efficient as possible while using properly the resources (energy, memory, FPGA building blocks, etc). Thought many progresses were achieved on the low aspects of circuit generation (pipeline, place/route), the front-end aspects (parallelism, communications) are still rudimentary compared to the state-of-the-art techniques in the HPC community.

We introduce the Data-aware Process Networks (DPN), a new parallel execution model adapted to the hardware constraints of high-level synthesis, where the data transferts are made explicit. We show that the DPN model is consistant in the meaning where any translation of a sequential program produces an equivalent DPN without deadlocks. Finally, we show how to compile a sequential program to a DPN and how to optimize the input/output and the parallelism.

This work was published as an Inria research report [63] and will be submitted to a journal.

7.29. Termination of C programs

Participants: Laure Gonnord, David Monniaux [CNRS/VERIMAG], Gabriel Radanne [Univ Paris 7/ PPS].

We designed a complete method for synthesizing lexicographic linear ranking functions (and thus proving termination), supported by inductive invariants, in the case where the transition relation of the program includes disjunctions and existentials (large block encoding of control flow).

Previous work would either synthesize a ranking function at every basic block head, not just loop headers, which reduces the scope of programs that may be proved to be terminating, or expand large block transitions including tests into (exponentially many) elementary transitions, prior to computing the ranking function, resulting in a very large global constraint system. In contrast, our algorithm incrementally refines a global linear constraint system according to extremal counterexamples: only constraints that exclude spurious solutions are included.

Experiments with our tool Termite 6.5 show marked performance and scalability improvements compared to other systems.

This work has been published in the proceedings of PLDI'15 [27].

7.30. Analysing C programs with arrays

Participants: Laure Gonnord, David Monniaux [CNRS/VERIMAG].

Automatically verifying safety properties of programs is hard, and it is even harder if the program acts upon arrays or other forms of maps. Many approaches exist for verifying programs operating upon Boolean and integer values (e.g. abstract interpretation, counterexample-guided abstraction refinement using interpolants), but transposing them to array properties has been fraught with difficulties.

In contrast to most preceding approaches, we do not introduce a new abstract domain or a new interpolation procedure for arrays. Instead, we generate an abstraction as a scalar problem and feed it to a preexisting solver. The intuition is that if there is a proof of safety of the program, it is likely that it can be expressed by elementary steps between properties involving only a small (tunable) number N of cells from the array.

Our transformed problem is expressed using Horn clauses over scalar variables, a common format with clear and unambiguous logical semantics, for which there exist several solvers. In contrast, solvers directly operating over Horn clauses with arrays are still very immature.

An important characteristic of our encoding is that it creates a nonlinear Horn problem, with tree unfoldings, contrary to the linear problems obtained by flatly encoding the control-graph structure. Our encoding thus cannot be expressed by encoding into another control-flow graph problem, and truly leverages the Horn clause format.

Experiments with our prototype VAPHOR show that this approach can prove automatically the functional correctness of several classical examples of the literature, including *selection sort*, *bubble sort*, *insertion sort*, as well as examples from previous articles on array analysis.

This work has been published as a research report [53] and is currently under submission.

7.31. Symbolic Range Analysis of Pointers in C programs

Participants: Maroua Maalej, Vitor Paisante [Univ. Mineas Gerais, Brasil], Laure Gonnord, Fernando Pereira [Univ. Mineas Gerais, Brasil], Vitor Paisante [Univ. Mineas Gerais, Brasil].

Alias analysis is one of the most fundamental techniques that compilers use to optimize languages with pointers. However, in spite of all the attention that this topic has received, the current state-of-the-art approaches inside compilers still face challenges regarding precision and speed. In particular, pointer arithmetic, a key feature in C and C++, is yet to be handled satisfactorily. We designed a new alias analysis algorithm to solve this problem. The key insight of our approach is to combine alias analysis with symbolic range analysis. This combination lets us disambiguate fields within arrays and structs, effectively achieving more precision than traditional algorithms. To validate our technique, we have implemented it on top of the LLVM compiler. Tests on a vast suite of benchmarks show that we can disambiguate several kinds of C idioms that current state-of-the-art analyses cannot deal with. In particular, we can disambiguate 1.35x more queries than the alias analysis currently available in LLVM. Furthermore, our analysis is very fast: we can go over one million assembly instructions in 10 seconds.

This work has been published at CGO'16 [32].

An extended version of the related work has also been published as an Inria research report [52] and will be the basis of a journal submission.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

8.1.1. Mumps Consortium (2014-2019)

In the context of the MUMPS consortium (http://mumps-consortium.org):

- We have signed three new membership agreements, with ESI-Group, Siemens SISW (Belgium) and TOTAL in 2015, on top of the on-going agreements signed in 2014 with Altair, EDF, LSTC, Michelin.

- We have organized point-to-point meetings with several members.
- We have provided technical support and scientific advice to members.
- We have provided non-public releases in advance to members, with a specific licence.
- We have organized the first consortium committee meeting, at EDF (Clamart).
- Two engineers have been funded by the membership fees, for software engineering and software development, comparison with other solvers, business development and management of the consortium.

8.1.2. Contract with EMGS (Norway)

Following a strong interest from EMGS (Norway) in the latest evolutions of MUMPS (see Section 6.1) we worked on the third and final phase of a contract related to low-rank compression for electromagnetics applications in geophysics; the contract was managed by INP Toulouse.

8.2. Technological Transfer: XtremLogic Start-Up

The XTREMLOGIC start-up (former Zettice project) was initiated 4 years ago by Alexandru Plesco and Christophe Alias. The goal of XTREMLOGIC is to build on the state-of-the-art research results from the polyhedral community to provide the HPC market with efficient and communication-optimal circuit blocks (IP) for FPGA. The compiler technology transferred to XTREMLOGIC is the result of a tight collaboration between Christophe Alias and Alexandru Plesco.

XTREMLOGIC won several awards and grants: Rhône Développement Initiative 2015 (loan), “concours émergence OSEO 2013” at Banque Publique d’Investissement (grant), “most promising start-up award” at SAME 2013 (award), “lean Startup award” at Startup Weekend Lyon 2012 (award), “excel&rate award 2012” from Crealys incubation center (award).

9. Partnerships and Cooperations

9.1. Regional Initiatives

9.1.1. PhD grant laboratoire d’excellence MILYON-Mumps consortium

Thanks to the doctoral program from the MILYON labex dedicated to applied research in collaboration with industrial partners, we obtained 50% of a PhD grant, the other 50% being funded by the MUMPS consortium. The PhD student will focus on improvements of the solution phase of the MUMPS solver, in accordance to requirements from industrial members of the consortium.

9.1.2. Cooperation with ECNU

ENS Lyon has launched a partnership with ECNU, the East China Normal University in Shanghai, China. This partnership includes both teaching and research cooperation.

As for teaching, the PROSFER program includes a joint Master of Computer Science between ENS Rennes, ENS Lyon and ECNU. In addition, PhD students from ECNU are selected to conduct a PhD in one of these ENS. Yves Robert is responsible for this cooperation. He has already given two classes at ECNU, on Algorithm Design and Complexity, and on Parallel Algorithms, together with Patrice Quinton (from ENS Rennes).

As for research, the JORISS program funds collaborative research projects between ENS Lyon and ECNU. Yves Robert and Changbo Wang (ECNU) are leading a JORISS project on resilience in cloud and HPC computing.

9.2. National Initiatives

9.2.1. ANR

ANR White Project RESCUE (2010-2015), 4 years. The ANR White Project RESCUE was launched in November 2010, for a duration of 48 months (and was later extended for 6 additional months, up to June 2015). It gathers three Inria partners (ROMA, Grand-Large and Hiepacs) and is led by ROMA. The main objective of the project is to develop new algorithmic techniques and software tools to solve the *exascale resilience problem*. Solving this problem implies a departure from current approaches, and calls for yet-to-be-discovered algorithms, protocols and software tools.

This proposed research follows three main research thrusts. The first thrust deals with novel *checkpoint protocols*. The second thrust entails the development of novel *execution models*, i.e., accurate stochastic models to predict (and, in turn, optimize) the expected performance (execution time or throughput) of large-scale parallel scientific applications. In the third thrust, we will develop novel *parallel algorithms* for scientific numerical kernels.

ANR Project SOLHAR (2013-2017), 4 years. The ANR Project SOLHAR was launched in November 2013, for a duration of 48 months. It gathers five academic partners (the HiePACS, Cepage, ROMA and Runtime Inria project-teams, and CNRS-IRIT) and two industrial partners (CEA/CESTA and EADS-IW). This project aims at studying and designing algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computers equipped with accelerators.

The proposed research is organized along three distinct research thrusts. The first objective deals with linear algebra kernels suitable for heterogeneous computing platforms. The second one focuses on runtime systems to provide efficient and robust implementation of dense linear algebra algorithms. The third one is concerned with scheduling this particular application on a heterogeneous and dynamic environment.

9.2.2. Inria Project Lab C2S@Exa - Computer and Computational Scienecs at Exascale

Participants: Olivier Aumage [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Jocelyne Erhel [SAGE project-team, Inria Rennes - Bretagne Atlantique], Philippe Helluy [TONUS project-team, Inria Nancy - Grand-Est], Laura Grigori [ALPINE project-team, Inria Saclay - Île-de-France], Jean-Yves L'excellent [ROMA project-team, Inria Grenoble - Rhône-Alpes], Thierry Gautier [MOAIS project-team, Inria Grenoble - Rhône-Alpes], Luc Giraud [HIEPACS project-team, Inria Bordeaux - Sud-Ouest], Michel Kern [POMDAPI project-team, Inria Paris - Rocquencourt], Stéphane Lanteri [Coordinator of the project], François Pellegrini [BACCHUS project-team, Inria Bordeaux - Sud-Ouest], Christian Perez [AVALON project-team, Inria Grenoble - Rhône-Alpes], Frédéric Vivien [ROMA project-team, Inria Grenoble - Rhône-Alpes].

Since January 2013, the team is participating to the C2S@Exa http://www-sop.inria.fr/c2s_at_exa Inria Project Lab (IPL). This national initiative aims at the development of numerical modeling methodologies that fully exploit the processing capabilities of modern massively parallel architectures in the context of a number of selected applications related to important scientific and technological challenges for the quality and the security of life in our society. At the current state of the art in technologies and methodologies, a multidisciplinary approach is required to overcome the challenges raised by the development of highly scalable numerical simulation software that can exploit computing platforms offering several hundreds of thousands of cores. Hence, the main objective of C2S@Exa is the establishment of a continuum of expertise in the computer science and numerical mathematics domains, by gathering researchers from Inria project-teams whose research and development activities are tightly linked to high performance computing issues in these domains. More precisely, this collaborative effort involves computer scientists that are experts of programming models, environments and tools for harnessing massively parallel systems, algorithmists that propose algorithms and contribute to generic libraries and core solvers in order to take benefit from all the parallelism levels with the main goal of optimal scaling on very large numbers of computing entities and, numerical mathematicians that are studying numerical schemes and scalable solvers for systems of partial differential equations in view of the simulation of very large-scale problems.

9.3. European Initiatives

9.3.1. FP7 & H2020 Projects

9.3.1.1. SCORPIO

Title: Significance-Based Computing for Reliability and Power Optimization

Programm: FP7

Duration: June 2013 - May 2016

Coordinator: Nikolaos Bellas

Partners: CERTH, Greece; EPFL, Switzerland; RWTH Aachen University, Germany; The Queen's University of Belfast, UK; IMEC, Belgium

Inria contact: Frédéric Vivien

Manufacturing process variability at low geometries and power dissipation are the most challenging problems in the design of future computing systems. Currently manufacturers go to great lengths to guarantee fault-free operation of their products by introducing redundancy in voltage margins, conservative layout rules, and extra protection circuitry. However, such design redundancy may result into energy overheads. Energy overheads cannot be alleviated by lowering supply voltage below a nominal value without hardware components experiencing faulty operation due to timing errors. On the other hand, many modern workloads, such as multimedia, machine learning, visualization, etc. are designed to tolerate a degree of imprecision in computations and data.SCoRPiO seeks to exploit this observation and to relax reliability requirements for the hardware layer by allowing a controlled degree of imprecision to be introduced to computations and data. It proposes to introduce methodologies that allow the system- and application-software layers to synergistically characterize the significance of various parts of the program for the quality of the end result, and their tolerance to faults. Based on this information, extracted automatically or semi-automatically, the system software will steer computations and data to either low-power, yet unreliable or higher-power and reliable functional and storage units. In addition, the system will be able to aggressively reduce its power footprint by opportunistically powering hardware modules below nominal values.Significance-based computing lays the foundations for not only approaching the theoretical limits of energy reduction of CMOS technology, but moving beyond those limits by accepting hardware faults in a controlled manner. Significance-based computing promises to be a preferred alternative to dark silicon, which requires that large portions of a chip be powered-off in every cycle to avoid excessive power dissipation.

9.4. International Initiatives

9.4.1. Inria International Labs

The University of Illinois at Urbana-Champaign, Inria, the French national computer science institute, Argonne National Laboratory, Barcelona Supercomputing Center, Jülich Supercomputing Centre and the Riken Advanced Institute for Computational Science formed the Joint Laboratory on Extreme Scale Computing, a follow-up of the Inria-Illinois Joint Laboratory for Petascale Computing. The Joint Laboratory is based at Illinois and includes researchers from Inria, and the National Center for Supercomputing Applications, ANL, BSC and JSC. It focuses on software challenges found in extreme scale high-performance computers.

Research areas include:

- Scientific applications (big compute and big data) that are the drivers of the research in the other topics of the joint-laboratory.
- Modeling and optimizing numerical libraries, which are at the heart of many scientific applications.
- Novel programming models and runtime systems, which allow scientific applications to be updated or reimagined to take full advantage of extreme-scale supercomputers.

- Resilience and Fault-tolerance research, which reduces the negative impact when processors, disk drives, or memory fail in supercomputers that have tens or hundreds of thousands of those components.
- I/O and visualization, which are important part of parallel execution for numerical silulations and data analytics
- HPC Clouds, that may execute a portion of the HPC workload in the near future.

Several members of the ROMA team are involved in the JLESC joint lab through their research on resilience. Yves Robert is the Inria executive director of JLESC.

9.4.2. Inria Associate Teams not involved in an Inria International Labs

- Laure Gonnord and Maroua Maalej are involved in the PROSPIEL Associate Team (Inria/ Brasil, https://team.inria.fr/alf/prospiel/). The PROSPIEL project aims at optimizing parallel applications for high performance on new throughput-oriented architectures: GPUs and many-core processors. Specifically, Laure Gonnord and Maroua Maalej are in charge of designing static analyses for GPUs. In Feb.-Apr. 2016, ROMA will host one student coming from the Brasilian team.

9.4.3. Inria International Partners

9.4.3.1. Declared Inria International Partners

- Christophe Alias has a regular collaboration with Sanjay Rajopadhye from Colorado State University (USA) through the advising of the PhD thesis of Guillaume Iooss. Since September 2015, this collaboration led to one publication, see Section 7.27.
- Anne Benoit and Yves Robert have a regular collaboration with Padma Raghavan from Penn State University (USA). They have achieved several publications in 2015, see Sections 7.8 and 7.26.
- Anne Benoit, Frédéric Vivien and Yves Robert have a regular collaboration with Henri Casanova from Hawaii University (USA). This is a follow-on of the Inria Associate team that ended in 2014. They have achieved one publication in 2015, see Section 7.1.

9.5. International Research Visitors

9.5.1. Visits of International Scientists

- Fernando M. Pereira was invited in Jan. 2015 to work with Maroua Maalej and Laure Gonnord on static analyses for pointers.
- Oliver Sinnen was invited for two months (Sept./Oct. 2015) to work with Loris Marchal, Bertrand Simon and Frédéric Vivien on scheduling malleable task trees.
- Samuel McCauley visited the team for four months (Oct. 2015 - Feb. 2016) to work with Loris Marchal, Bertrand Simon and Frédéric Vivien on the minimization of I/Os during the out-of-core execution of task trees.

9.5.1.1. Internships

- Anne Benoit and Yves Robert advised the M2 internship of Loic Pottier on resilient application co-scheduling with processor redistribution.
- Christophe Alias advised the M2 internship of Adilla Susungi on the compilation of pipelined parallelism on multi-GPU.
- Guillaume Aupy and Loris Marchal advised the L3 internship of Clément Brasseur on memory minimization for the parallel processing of task trees.
- Julien Herrmann and Yves Robert advised the L3 internship of Nicolas Vidal on the evaluation of the makespan of stochastic computational workflows.

9.5.2. Visits to International Teams

9.5.2.1. Research stays abroad

- Yves Robert has been appointed as a visiting scientist by the ICL laboratory (headed by Jack Dongarra) at the University of Tennessee Knoxville. He collaborates with several ICL researchers on high-performance linear algebra and resilience methods at scale.
- Bertrand Simon spent six months (Feb.-Jul. 2015) at Stony Brooks University (USA) to work with Michael Bender.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific events organisation

10.1.1.1. General chair, scientific chair

Laure Gonnord is co-chair of the "Compilation French community", with Florian Brandner (ENSTA) and Fabrice Rastello (Inria Corse).

10.1.2. Scientific events selection

10.1.2.1. Steering committees

Yves Robert is a member of the steering committee of HCW, Heteropar and IPDPS. He is the chair of the steering committee of Euro-EduPar.

10.1.2.2. Chair of conference program committees

Anne Benoit was program vice-chair for the Applications and Algorithms track of SBAC-PAD'2015, program vice-chair for the Algorithms track of HiPC'2015, and workshops co-chair of ICPP'2015.

Bora Uçar was IPDPS 2015 Workshops vice-chair, and was a co-chair of PCO2015 (a workshop of IPDPS).

10.1.2.3. Member of the conference program committees

Christophe Alias was a member of the program committee of IMPACT'16.

Anne Benoit was a member of the program committee of SC, CCGrid, HCW, Ena-HPC, and FEEDBACK.

Jean-Yves L'Excellent was a member of the program committee of Compas.

Loris Marchal was a member of the program committee of IPDPS. and HiPeR.

Yves Robert was a member of the program committee of FTXS, ICCS, IPDPS, and SC.

Bora Uçar was in the PC of IPDPS, HiPC, PPAM, ICCS, HPC4BD, and IEEE CSE.

Frédéric Vivien was a member of the program committee of IPDPS, SC, HiPC, PDP, ComPAS, EduHPC, EduPar, ROADEF, and WAPCO.

10.1.2.4. Reviewer

Christophe Alias reviewed papers for DATE'15.

Laure Gonnord reviewed papers for VMCAI'15, CGO'15.

Bora Uçar reviewed papers for SC15 and MFCS 2015.

Jean-Yves L'Excellent reviewed papers for Compas'15 and Europar 2015.

10.1.3. Journal

10.1.3.1. Member of the editorial boards

Anne Benoit is Associate Editor of TPDS (IEEE Transactions on Parallel and Distributed Systems) since October 2015, and also of JPDC (Elsevier Journal of Parallel and Distributed Computing) and SUSCOM (Elsevier Journal of Sustainable Computing).

Yves Robert is Associate Editor of JPDC (Elsevier Journal of Parallel and Distributed Computing), IJHPCA (Sage International Journal of High Performance Computing Applications), and JOCS (Elsevier Journal of Computational Science).

Anne Benoit is Associate Editor of Parallel Computing (Elsevier) and of JPDC (Elsevier Journal of Parallel and Distributed Computing).

10.1.3.2. Reviewer - Reviewing activities

Anne Benoit reviewed papers for TOPC and TCS.

Christophe Alias reviewed papers for PARCO and IEEE TVLSI.

Laure Gonnord reviewed papers for PARCO.

Bora Uçar reviewed papers for SIAM SISC, ACM ToPC, IEEE TPDS, JPDC, and NLAA.

Loris Marchal reviewed papers for IEEE TPDS, SIAM TOPC, JPDC and ADHOC.

Jean-Yves L'Excellent reviewed a paper for ACM TOMS.

Frédéric Vivien reviewed papers for Journal of Grid Computing, Cluster Computing, and the Journal of Supercomputing.

10.1.4. Invited talks

In June 2015, Laure Gonnord was invited at Google, Mountain View and SRI, to give talks about her research about static analyses for compilers.

10.1.5. Leadership within the scientific community

Laure Gonnord, together with Fabrice Rastello (CORSE) and Florian Brandner (Telecom Paris Tech) animate since 2010 the French Compilation Community (http://compilfr.ens-lyon.fr).

Yves Robert participated to the following selection committees:

- IEEE Fellows: vice-president and attended the physical selection meeting in Los Alamitos in May 2015.
- IEEE TCSC Award for Excellence in Scalable Computing: member
- IEEE TCSC Mid-Career Award: member

10.1.6. Scientific expertise

In 2015, Maroua Maalej has produced 7 Research Tax Credit documents for Accenture group France as a scientific consultant. The goal is to expertise research done by Accenture project-teams and suggest further ideas by evaluating the state of the art.

10.1.7. Research administration

Anne Benoit is a member of the executive committee of the Labex MI-LYON.

Bora Uçar served as the Secretary of the SIAM activity group on Supercomputing, (1 January 2014–12 December 2015).

Loris Marchal is a member of the scientific council of the "Ecole Nationale Supe´rieure de Me´canique et des Microtechniques" (ENSMM, Besançon).

Jean-Yves L'Excellent evaluated a CIFRE PhD proposal for ANRT. He is a member of the direction board of the LIP laboratory (since November 2015).

Frédéric Vivien is a member of the scientific council of the École normale supérieure de Lyon and of the academic council of the University of Lyon.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Licence:

- Anne Benoit : Algorithmique avancée (CM 32h), L3, Ecole Normale Supérieure de Lyon.
- Laure Gonnord : Architecture des ordinateurs (TD+TP=40h), L2, Université Lyon 1 Claude Bernard : Spring 2015.
- Maroua Maalej : Algorithmique et Programmation Fonctionnelle et Récursive (TP=28h), L1, Université Lyon 1 Claude Bernard : Autumn 2015.
- Christophe Alias : Introduction à la compilation (CM+TD=24h), L3, INSA Centre-Val-de-Loire : Spring 2015.
- Christophe Alias : Concours E3A – épreuve informatique MPSI (correcteur) : Spring 2015.
- Yves Robert : Algorithmique (CM 32h), L3, Ecole Normale Supérieure de Lyon

Master:

- Anne Benoit, Resilient and energy-aware scheduling algorithms (CM 24h), M2, Ecole Normale Supérieure de Lyon.
- Laure Gonnord, Program Analysis and Verification (CM 24h), M2, Ecole Normale Supérieure de Lyon. With David Monniaux
- Laure Gonnord, Compilation (TP 28h), M1, Ecole Normale Supérieure de Lyon.
- Laure Gonnord, Introduction aux systèmes et réseaux (CM/TP 52h), M2 Pro, Université Lyon 1.
- Laure Gonnord, Compilation (TD/TP 24h), M1, Université Lyon 1 Claude Bernard.
- Laure Gonnord, Complexité (TD 15h), M1, Université Lyon 1 Claude Bernard.
- Laure Gonnord, Temps Réel (TP 24h), M1, Université Lyon 1 Claude Bernard.
- Laure Gonnord organised (Jan. 2015) a research school entitled “Static analyses in the state-of-the-art compilers” (invited speaker : Fernando Pereira).
- Christophe Alias, Optimisation d’applications embarquées (CM+TD=18h), M1, INSA Centre-Val-de-Loire.
- Christophe Alias, Advanced Compilers: Loop Transformations and High-Level Synthesis (CM 8h), M2, Ecole Normale Supérieure de Lyon.
- Christophe Alias, Compilation (CM 16h), M1, Ecole Normale Supérieure de Lyon.
- Frédéric Vivien, Algorithmique et Programmation Parallèles et Distribuées (CM 36 h), M1, École normale supérieure de Lyon, France.

10.2.2. Supervision

PhD in progress: Aurélien Cavelan, "Resilient and energy-aware scheduling algorithms for large-scale distributed systems", started in September 2014, advisors: Anne Benoit and Yves Robert.

PhD in progress: Guillaume Iooss, “Semantic tiling”, started in September 2011, joint PhD ENS-Lyon/Colorado State University, advisors: Christophe Alias and Alain Darte (ENS-Lyon) / Sanjay Rajopadhye (Colorado State University).

PhD in progress: Oguz Kaya, “High performance parallel tensor computations”, started in September 2014, funding: Inria, advisors: Bora Uçar and Yves Robert.

PhD in progress: Maroua Maalej, “Low cost static analyses for compilers”, started in October 2014, advisors : Laure Gonnord and Frédéric Vivien.

PhD in progress: Gilles Moreau, “High-performance multifrontal solution of sparse linear systems with multiple right-hand sides, application to the MUMPS solver”, started in December 2015, funding: MUMPS consortium and Labex MILYON, advisor: Jean-Yves L’Excellent.

PhD in progress: Loic Pottier, "Scheduling concurrent applications in the presence of failures", started in September 2015, advisors: Anne Benoit and Yves Robert.

PhD in progress: Issam Rais, "Multi-criteria scheduling for high-performance computing", started in November 2015, advisors: Anne Benoit, Laurent Lefèvre (LIP, ENS Lyon, Avalon team), and Anne-Cécile Orgerie (IRISA, Myriads team).

PhD in progress: Bertrand Simon, “Task-graph scheduling and memory optimization”, started in September 2015, funding: ENS Lyon, advisors: Loris Marchal and Frédéric Vivien.

PhD defended on November 25: Julien Herrmann, “Memory-aware algorithms and scheduling techniques for matrix computations”, started in September 2012, funding: ENS Lyon, advisors: Loris Marchal and Yves Robert.

10.2.3. Juries

Laure Gonnord was a member of the doctoral committee for the evaluation of the first year of Phd of F. Maurica, Université de la Réunion, December 2015.

Jean-Yves L’Excellent was a reviewer for the PhD thesis of Corentin Rossignon, University of Bordeaux, 2015.

Yves Robert chaired the HDR defense committee of Arnaud Legrand (University of Grenoble) in November 2015. He was a member of the PhD defense committee of Mathias Coqblin (University of Bsançon) in January 2015. He was a reviewer for the PhD defense committee of Pooja Aggarwal (Indian Institute of Technology Delhi) in October 2015.

Frédéric Vivien was a member of the PhD defense committee of Mawussi Zounon (University of Bordeaux).

10.3. Popularization

Loris Marchal led a “Math en Jeans” workshop at “Lycée Lacassagne” in Lyon.

Frédéric Vivien gave two lectures about “Scheduling” at the CIRM “Algorithmique et programmation” workshop for Maths teachers in *Classes préparatoires aux grandes écoles*. The two lectures were recorded and are available online (http://library.cirm-math.fr/Record.htm?Record=19278406157910966889).

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] J. HERRMANN. *Memory-aware Algorithms and Scheduling Techniques for Matrix Computattions*, Ecole normale supérieure de lyon - ENS LYON, November 2015, https://tel.archives-ouvertes.fr/tel-01241485

Articles in International Peer-Reviewed Journals

[2] P. R. AMESTOY, C. ASHCRAFT, O. BOITEAU, A. BUTTARI, J.-Y. L’EXCELLENT, C. WEISBECKER. *Improving Multifrontal Methods by Means of Block Low-Rank Representations*, in "SIAM Journal on Scientific Computing", 2015, vol. 37, n^{o} 3, pp. A1451-A1474 [*DOI :* 10.1137/120903476], https://hal.inria.fr/hal-01237169

[3] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, F.-H. ROUET. *Parallel computation of entries in A-1*, in "SIAM Journal on Scientific Computing", 2015, vol. 37, n^{o} 2, pp. C268–C284 [*DOI :* 10.1137/120902616], https://hal.inria.fr/hal-01237166

[4] G. AUPY, A. BENOIT. *Approximation Algorithms for Energy, Reliability, and Makespan Optimization Problems*, in "Parallel Processing Letters", 2015, https://hal.inria.fr/hal-01252333

[5] G. AUPY, A. BENOIT, M. JOURNAULT, Y. ROBERT. *Power-aware replica placement in tree networks with multiple servers per client*, in "Sustainable Computing", March 2015, vol. 5, pp. 41-53 [*DOI :* 10.1016/J.SUSCOM.2014.08.013], https://hal.inria.fr/hal-01059364

[6] G. AUPY, M. SHANTHARAM, A. BENOIT, Y. ROBERT, P. RAGHAVAN. *Co-scheduling algorithms for high-throughput workload execution*, in "Journal of Scheduling", 2015 [*DOI :* 10.1007/S10951-015-0445-X], https://hal.inria.fr/hal-01252366

[7] A. BENOIT, L.-C. CANON, L. MARCHAL. *Non-clairvoyant reduction algorithms for heterogeneous platforms*, in "Concurrency and Computation: Practice and Experience", April 2015, vol. 27, n^{o} 6, pp. 1612-1624 [*DOI :* 10.1002/CPE.3347], https://hal.inria.fr/hal-01090232

[8] A. BENOIT, S. K. RAINA, Y. ROBERT. *Efficient checkpoint/verification patterns*, in "International Journal of High Performance Computing Applications", July 2015 [*DOI :* 10.1177/1094342015594531], https://hal-ens-lyon.archives-ouvertes.fr/ensl-01252342

[9] G. BOSILCA, A. BOUTEILLER, J. DONGARRA, T. HÉRAULT, Y. ROBERT. *Composing resilience techniques: ABFT, periodic and incremental checkpointing*, in "The International Journal of Networking and Computing", March 2015, 18 p. , https://hal.inria.fr/hal-01091930

[10] H. CASANOVA, F. DUFOSSÉ, Y. ROBERT, F. VIVIEN. *Mapping Applications on Volatile Resources*, in "International Journal of High Performance Computing Applications", February 2015, vol. 29, n^{o} 1, 19 p. [*DOI :* 10.1177/1094342013518806], https://hal.inria.fr/hal-00923948

[11] H. CASANOVA, Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *On the impact of process replication on executions of large-scale parallel applications with coordinated checkpointing*, in "Future Generation Computer Systems", October 2015, vol. 51, 13 p. [*DOI :* 10.1016/J.FUTURE.2015.04.003], https://hal.inria.fr/hal-01199752

[12] M. DEVECI, K. KAYA, B. UÇAR, U. V. CATALYUREK. *Hypergraph partitioning for multiple communication cost metrics: Model and methods*, in "Journal of Parallel and Distributed Computing", 2015, vol. 77, pp. 69–83 [*DOI :* 10.1016/J.JPDC.2014.12.002], https://hal.inria.fr/hal-01159676

[13] F. DUFOSSÉ, K. KAYA, B. UÇAR. *Two approximation algorithms for bipartite matching on multicore architectures*, in "Journal of Parallel and Distributed Computing", 2015, vol. 85, pp. 62-78 [*DOI :* 10.1016/J.JPDC.2015.06.009], https://hal.inria.fr/hal-01242516

[14] L. EYRAUD-DUBOIS, L. MARCHAL, O. SINNEN, F. VIVIEN. *Parallel scheduling of task trees with limited memory*, in "ACM Transactions on Parallel Computing", July 2015, vol. 2, n^{o} 2, 36 p. [*DOI :* 10.1145/2779052], https://hal.inria.fr/hal-01160118

[15] J. HERRMANN, G. BOSILCA, T. HÉRAULT, L. MARCHAL, Y. ROBERT, J. DONGARRA. *Assessing the cost of redistribution followed by a computational kernel: Complexity and performance results*, in "Parallel Computing", 2016, vol. 52, 20 p. [*DOI :* 10.1016/J.PARCO.2015.09.005], https://hal.inria.fr/hal-01254167

[16] T. LAMBERT, L. MARCHAL, B. UÇAR. *Comments on the hierarchically structured bin packing problem*, in "Information Processing Letters", 2015, vol. 115, n^{o} 2, pp. 306–309 [*DOI :* 10.1016/J.IPL.2014.10.001], https://hal.inria.fr/hal-01071414

International Conferences with Proceedings

[17] E. AGULLO, O. BEAUMONT, L. EYRAUD-DUBOIS, J. HERRMANN, S. KUMAR, L. MARCHAL, S. THIBAULT. *Bridging the Gap between Performance and Bounds of Cholesky Factorization on Heterogeneous Platforms*, in "Heterogeneity in Computing Workshop 2015", Hyderabad, India, May 2015, https://hal.inria.fr/hal-01120507

[18] P. R. AMESTOY, R. BROSSIER, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY, L. MÉTIVIER, A. MINIUSSI, S. OPERTO, A. RIBODETTI, J. VIRIEUX, C. WEISBECKER. *Efficient 3D frequency-domain full-waveform inversion of ocean-bottom cable data with sparse block low-rank direct solver: a real data case study from the North Sea*, in "SEG Annual meeting", New Orleans, United States, SEG Technical Program Expanded Abstracts 2015, 2015, n^{o} 251, pp. 1303-1308 [*DOI :* 10.1190/SEGAM2015-5713962.1], https://hal.inria.fr/hal-01239896

[19] P. R. AMESTOY, R. BROSSIER, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY, L. MÉTIVIER, A. MINIUSSI, S. OPERTO, J. VIRIEUX, C. WEISBECKER. *3D frequency-domain seismic modeling with a Parallel BLR multifrontal direct solver*, in "SEG Annual meeting", New Orleans, United States, SEG Technical Program Expanded Abstracts 2015, 2015, n^{o} 692, pp. 3606-3611, https://hal.inria.fr/hal-01237869

[20] G. AUPY, A. BENOIT, H. CASANOVA, Y. ROBERT. *Scheduling Computational Workflows on Failure-Prone Platforms*, in "17th Workshop on Advances in Parallel and Distributed Computational Models", Hyderabad, India, 2015 IEEE International Parallel and Distributed Processing Symposium Workshop (APDCM), May 2015 [*DOI :* 10.1109/IPDPSW.2015.33], https://hal.inria.fr/hal-01251939

[21] L. BAUTISTA-GOMEZ, A. BENOIT, A. CAVELAN, S. K. RAINA, Y. ROBERT, H. SUN. *Which Verification for Soft Error Detection?*, in "High Performance Computing 2015", Bangalore, India, December 2015, https://hal.inria.fr/hal-01252382

[22] A. CAVELAN, S. K. RAINA, H. SUN, Y. ROBERT. *Assessing the impact of partial verifications against silent data corruptions*, in "ICPP'2015, The Int. Conference on Parallel Processing", Beijing, China, ICPP'2015, The Int. Conference on Parallel Processing, IEEE, 2015, https://hal.inria.fr/hal-01253493

[23] A. CAVELAN, Y. ROBERT, H. SUN, F. VIVIEN. *Voltage Overscaling Algorithms for Energy-Efficient Workflow Computations With Timing Errors*, in "FTXS '15: 5th Workshop on Fault Tolerance for HPC at eXtreme Scale", Portland, United States, ACM, June 2015, 8 p. [*DOI :* 10.1145/2751504.2751508], https://hal.inria.fr/hal-01199250

[24] M. DEVECI, K. KAYA, B. UÇAR, U. V. CATALYUREK. *Fast and high quality topology-aware task mapping*, in "29th IEEE International Parallel & Distributed Processing Symposium", Hyderabad, India, IEEE CPS, May 2015, https://hal.inria.fr/hal-01159677

[25] M. FASI, Y. ROBERT, B. UÇAR. *Combining backward and forward recovery to cope with silent errors in iterative solvers*, in "PDSEC2015", Hyderabad, India, IPDPSW 2015: 29th IEEE International Parallel & Distributed Processing Symposium Workshops 2015, IEEE CPS, May 2015, pp. 980–989, https://hal.inria.fr/hal-01159679

[26] A. GAINARU, G. AUPY, A. BENOIT, F. CAPPELLO, Y. ROBERT, M. SNIR. *Scheduling the I/O of HPC Applications Under Congestion*, in "IEEE International Parallel and Distributed Processing Symposium, IPDPS 2015, Hyderabad, India, May 25-29, 2015", Hyderabad, India, May 2015 [*DOI :* 10.1109/IPDPS.2015.116], https://hal.inria.fr/hal-01251938

[27] L. GONNORD, D. MONNIAUX, G. RADANNE. *Synthesis of ranking functions using extremal counterexamples*, in "Programming Languages, Design and Implementation", Portland, Oregon, United States, June 2015 [*DOI :* 10.1145/2737924.2737976], https://hal.archives-ouvertes.fr/hal-01144622

[28] A. GUERMOUCHE, L. MARCHAL, B. SIMON, F. VIVIEN. *Scheduling Trees of Malleable Tasks for Sparse Linear Algebra*, in "European Conference on Parallel Processing (Euro-Par)", Vienna, Austria, 2015, https://hal.inria.fr/hal-01160104

[29] H. KABIR, J. BOOTH, G. AUPY, A. BENOIT, Y. ROBERT, P. RAGHAVAN. *STS-k: A Multilevel Sparse Triangular Solution Scheme for NUMA Multicores*, in "Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, Austin, TX, USA, November 15-20, 2015", Austin, United States, November 2015, 11 p. [*DOI :* 10.1145/2807591.2807667], https://hal.inria.fr/hal-01251937

[30] O. KAYA, B. UÇAR. *Scalable sparse tensor decompositions in distributed memory systems*, in "International Conference for High Performance Computing, Networking, Storage and Analysis (SC15)", Austin, TX, United States, November 2015 [*DOI :* 10.1145/2807591.2807624], https://hal.inria.fr/hal-01148202

[31] E. KAYAASLAN, B. UÇAR, C. AYKANAT. *Semi-two-dimensional partitioning for parallel sparse matrix-vector multiplication*, in "PCO2015 (IPDPSW)", Hyderabad, India, IEEE CPS, May 2015, pp. 1125–1134, https://hal.inria.fr/hal-01159692

[32] V. PAISANTE, M. MAALEJ, L. BARBOSA, L. GONNORD, F. M. Q. PEREIRA. *Symbolic Range Analysis of Pointers*, in "International Symposium of Code Generation and Optmization", Barcelon, Spain, March 2016, pp. 791-809, https://hal.inria.fr/hal-01228928

[33] M. RIETMANN, D. PETER, O. SCHENK, B. UÇAR, M. J. GROTE. *Load-Balanced Local Time Stepping for Large-Scale Wave Propagation*, in "29th IEEE International Parallel & Distributed Processing Symposium", Hyderabad, India, IEEE CPS, May 2015, pp. 925–935, https://hal.inria.fr/hal-01159687

Conferences without Proceedings

[34] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Two-level checkpointing and partial verifications for linear task graphs*, in "6th International Workshop in Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS15)", Austin, TX, United States, November 2015, https://hal.inria.fr/hal-01252400

Scientific Books (or Scientific Book chapters)

[35] G. AUPY, A. BENOIT, M. E. M. DIOURI, O. GLÜCK, L. LEFÈVRE. *Energy-aware checkpointing strategies*, in "Fault-Tolerance Techniques for High-Performance Computing", T. HÉRAULT, Y. ROBERT (editors), Springer, May 2015, pp. 279-317, https://hal.inria.fr/hal-01205153

[36] H. CASANOVA, D. ZAIDOUNI, F. VIVIEN. *Using replication for resilience on exascale systems*, in "Fault-Tolerance Techniques for High-Performance Computing", T. HÉRAULT, Y. ROBERT (editors), Springer, May 2015, 50 p. , https://hal.inria.fr/hal-01200486

[37] J. DONGARRA, T. HÉRAULT, Y. ROBERT. *Fault Tolerance Techniques for High-Performance Computing*, in "Fault-Tolerance Techniques for High-Performance Computing", T. HÉRAULT, Y. ROBERT (editors), Springer, May 2015, 83 p. , https://hal.inria.fr/hal-01200488

[38] T. HÉRAULT, Y. ROBERT. *Fault-Tolerance Techniques for High-Performance Computing*, Springer, May 2015, 320 p. [*DOI :* 10.1007/978-3-319-20943-2], https://hal.inria.fr/hal-01200479

Research Reports

[39] G. AUPY, A. BENOIT, M. FASI, Y. ROBERT, H. SUN, B. UÇAR. *Coping with silent errors in HPC applications*, CNRS, ENS Lyon & Inria, December 2015, n^{o} RR-8825, https://hal.inria.fr/hal-01242369

[40] G. AUPY, J. HERRMANN. *Periodicity is Optimal for Offline and Online Multi-Stage Adjoint Computations*, Inria Grenoble - Rhône-Alpes, December 2015, n^{o} RR-8822, https://hal.inria.fr/hal-01244584

[41] L. BAUTISTA-GOMEZ, A. BENOIT, A. CAVELAN, S. K. RAINA, Y. ROBERT, H. SUN. *Coping with Recall and Precision of Soft Error Detectors*, ENS Lyon, CNRS & Inria, December 2015, n^{o} RR-8832, 30 p. , https://hal.inria.fr/hal-01246639

[42] L. BAUTISTA-GOMEZ, A. BENOIT, A. CAVELAN, S. K. RAINA, Y. ROBERT, H. SUN. *Which Verification for Soft Error Detection?*, Inria Grenoble ; ENS Lyon ; Jaypee Institute of Information Technology, India ; Argonne National Laboratory ; University of Tennessee Knoxville, USA ; Inria, June 2015, n^{o} RR-8741, 20 p. , https://hal.inria.fr/hal-01164445

[43] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Optimal resilience patterns to cope with fail-stop and silent errors*, LIP - ENS Lyon, October 2015, n^{o} RR-8786, https://hal.inria.fr/hal-01215857

[44] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Two-level checkpointing and partial verifications for linear task graphs*, ENS Lyon ; Inria Grenoble Rhône-Alpes, Université de Grenoble, October 2015, n^{o} RR-8794, https://hal.inria.fr/hal-01216850

[45] A. CAVELAN, S. K. RAINA, Y. ROBERT, H. SUN. *Assessing the Impact of Partial Verifications Against Silent Data Corruptions*, Inria Grenoble - Rhône-Alpes ; ENS Lyon ; Université Lyon 1 ; Jaypee Institute of Information Technology, India ; CNRS - Lyon (69) ; University of Tennessee Knoxville, USA ; Inria, April 2015, n^{o} RR-8711, https://hal.inria.fr/hal-01143832

[46] A. CAVELAN, Y. ROBERT, H. SUN, F. VIVIEN. *Voltage Overscaling Algorithms for Energy-Efficient Workflow Computations With Timing Errors*, Inria, February 2015, n^{o} RR-8682, https://hal.inria.fr/hal-01121065

[47] M. FASI, J. LANGOU, Y. ROBERT, B. UÇAR. *A Backward/Forward Recovery Approach for the Preconditioned Conjugate Gradient Algorithm*, ENS Lyon, CNRS & Inria, December 2015, n^{o} RR-8826, https://hal.inria.fr/hal-01242327

[48] M. FASI, Y. ROBERT, B. UÇAR. *Combining algorithm-based fault tolerance and checkpointing for iterative solvers*, Inria Grenoble - Rhône-Alpes ; Inria, January 2015, n^{o} RR-8675, https://hal.inria.fr/hal-01111707

[49] G. IOOSS, S. RAJOPADHYE, C. ALIAS, Y. ZOU. *Mono-parametric Tiling is a Polyhedral Transformation*, Inria Grenoble - Rhône-Alpes ; CNRS, October 2015, n^{o} RR-8802, 40 p. , https://hal.inria.fr/hal-01219452

[50] H. KABIR, J. BOOTH, G. AUPY, A. BENOIT, Y. ROBERT, P. RAGHAVAN. *STS-k: A Multilevel Sparse Triangular Solution Scheme for NUMA Multicores*, Penn State University ; ENS Lyon ; Inria, August 2015, n^{o} RR-8763, https://hal.inria.fr/hal-01183904

[51] O. KAYA, B. UÇAR. *High-performance parallel algorithms for the Tucker decomposition of higher order sparse tensors*, Inria - Research Centre Grenoble – Rhône-Alpes, October 2015, n^{o} RR-8801, https://hal.inria.fr/hal-01219316

[52] M. MAALEJ, L. GONNORD. *Do we still need new Alias Analyses?*, Université Lyon Claude Bernard / Laboratoire d'Informatique du Parallélisme, November 2015, n^{o} RR-8812, https://hal.inria.fr/hal-01228581

Other Publications

[53] D. MONNIAUX, L. GONNORD. *An encoding of array verification problems into array-free Horn clauses*, July 2015, working paper or preprint, https://hal.archives-ouvertes.fr/hal-01206882

References in notes

[54] *Blue Waters Newsletter*, dec 2012, http://cgi.ncsa.illinois.edu/BlueWaters/pdfs/bw-newsletter-1212.pdf

[55] *Blue Waters Resources*, 2013, https://bluewaters.ncsa.illinois.edu/data

[56] *The BOINC project*, 2013, http://boinc.berkeley.edu/

[57] *Final report of the Department of Energy Fault Management Workshop*, December 2012, http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/FaultManagement-wrkshpRpt-v4-final.pdf

[58] *System Resilience at Extreme Scale: white paper*, 2008, DARPA, http://institute.lanl.gov/resilience/docs/IBM%20Mootaz%20White%20Paper%20System%20Resilience.pdf

[59] *Top500 List - November*, 2011, http://www.top500.org/list/2011/11/

[60] *Top500 List - November*, 2012, http://www.top500.org/list/2012/11/

[61] *The Green500 List - November*, 2015, http://www.green500.org/lists/green201511

[62] E. AGULLO, A. BUTTARI, A. GUERMOUCHE, F. LOPEZ. *Multifrontal QR Factorization for Multicore Architectures over Runtime Systems*, in "Euro-Par 2013 Parallel Processing", F. WOLF, B. MOHR, D. MEY (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, vol. 8097, pp. 521–532

[63] C. ALIAS, A. PLESCO. *Data-aware Process Networks*, Inria - Research Centre Grenoble – Rhône-Alpes, June 2015, n^o RR-8735, 32 p. , https://hal.inria.fr/hal-01158726

[64] I. ASSAYAD, A. GIRAULT, H. KALLA. *Tradeoff exploration between reliability power consumption and execution time*, in "Proceedings of SAFECOMP, the Conf. on Computer Safety, Reliability and Security", Washington, DC, USA, 2011

[65] C. AUGONNET, S. THIBAULT, R. NAMYST, P.-A. WACRENIER. *StarPU: A unified platform for task scheduling on heterogeneous multicore architectures*, in "Concurrency and Computation: Practice and Experience, Special Issue: Euro-Par 2009", February 2011, vol. 23, n^o 2, pp. 187–198 [*DOI :* 10.1002/CPE.1631], http://hal.inria.fr/inria-00550877

[66] H. AYDIN, Q. YANG. *Energy-aware partitioning for multiprocessor real-time systems*, in "IPDPS'03, the IEEE Int. Parallel and Distributed Processing Symposium", 2003, pp. 113–121

[67] N. BANSAL, T. KIMBREL, K. PRUHS. *Speed Scaling to Manage Energy and Temperature*, in "Journal of the ACM", 2007, vol. 54, n^o 1, pp. 1 – 39, http://doi.acm.org/10.1145/1206035.1206038

[68] A. BENOIT, L. MARCHAL, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Scheduling concurrent bag-of-tasks applications on heterogeneous platforms*, in "IEEE Transactions on Computers", 2010, vol. 59, n^o 2, pp. 202-217

[69] S. BLACKFORD, J. CHOI, A. CLEARY, E. D'AZEVEDO, J. DEMMEL, I. DHILLON, J. DONGARRA, S. HAMMARLING, G. HENRY, A. PETITET, K. STANLEY, D. WALKER, R. C. WHALEY. *ScaLAPACK Users' Guide*, SIAM, 1997

[70] S. BLACKFORD, J. DONGARRA. *Installation Guide for LAPACK*, LAPACK Working Note, June 1999, n^o 41, originally released March 1992

[71] A. BUTTARI, J. LANGOU, J. KURZAK, J. DONGARRA. *Parallel tiled QR factorization for multicore architectures*, in "Concurrency: Practice and Experience", 2008, vol. 20, n^o 13, pp. 1573-1590

[72] J.-J. CHEN, T.-W. KUO. *Multiprocessor energy-efficient scheduling for real-time tasks*, in "ICPP'05, the Int. Conference on Parallel Processing", 2005, pp. 13–20

[73] S. DONFACK, L. GRIGORI, W. GROPP, L. V. KALE. *Hybrid Static/dynamic Scheduling for Already Optimized Dense Matrix Factorization*, in "Parallel Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International", 2012, pp. 496-507, http://dx.doi.org/10.1109/IPDPS.2012.53

[74] J. DONGARRA, J.-F. PINEAU, Y. ROBERT, Z. SHI, F. VIVIEN. *Revisiting Matrix Product on Master-Worker Platforms*, in "International Journal of Foundations of Computer Science", 2008, vol. 19, n^o 6, pp. 1317-1336

[75] J. Dongarra, J.-F. Pineau, Y. Robert, F. Vivien. *Matrix Product on Heterogeneous Master-Worker Platforms*, in "13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming", Salt Lake City, Utah, February 2008, pp. 53–62

[76] I. S. Duff, J. K. Reid. *The multifrontal solution of indefinite sparse symmetric linear systems*, in ""ACM Transactions on Mathematical Software"", 1983, vol. 9, pp. 302-325

[77] I. S. Duff, J. K. Reid. *The multifrontal solution of unsymmetric sets of linear systems*, in "SIAM Journal on Scientific and Statistical Computing", 1984, vol. 5, pp. 633-641

[78] P. Feautrier, C. Lengauer. *The Polyhedron Model*, in "Encyclopedia of Parallel Programming", 2011

[79] L. Grigori, J. W. Demmel, H. Xiang. *Communication avoiding Gaussian elimination*, in "Proceedings of the 2008 ACM/IEEE conference on Supercomputing", Piscataway, NJ, USA, SC '08, IEEE Press, 2008, 29:1 p. , http://dl.acm.org/citation.cfm?id=1413370.1413400

[80] B. Hadri, H. Ltaief, E. Agullo, J. Dongarra. *Tile QR Factorization with Parallel Panel Processing for Multicore Architectures*, in "IPDPS'10, the 24st IEEE Int. Parallel and Distributed Processing Symposium", 2010

[81] J. W. H. Liu. *The multifrontal method for sparse matrix solution: Theory and Practice*, in "SIAM Review", 1992, vol. 34, pp. 82–109

[82] R. Melhem, D. Mossé, E. Elnozahy. *The Interplay of Power Management and Fault Recovery in Real-Time Systems*, in "IEEE Transactions on Computers", 2004, vol. 53, n^o 2, pp. 217-231

[83] A. J. Oliner, R. K. Sahoo, J. E. Moreira, M. Gupta, A. Sivasubramaniam. *Fault-aware job scheduling for bluegene/l systems*, in "IPDPS'04, the IEEE Int. Parallel and Distributed Processing Symposium", 2004, pp. 64–73

[84] G. N. S. Prasanna, B. R. Musicus. *Generalized Multiprocessor Scheduling and Applications to Matrix Computations*, in "IEEE Trans. Parallel Distrib. Syst.", 1996, vol. 7, n^o 6, pp. 650-664

[85] G. N. S. Prasanna, B. R. Musicus. *The Optimal Control Approach to Generalized Multiprocessor Scheduling*, in "Algorithmica", 1996, vol. 15, n^o 1, pp. 17-49

[86] G. Quintana-Ortí, E. Quintana-Ortí, R. A. van de Geijn, F. G. V. Zee, E. Chan. *Programming Matrix Algorithms-by-Blocks for Thread-Level Parallelism*, in "ACM Transactions on Mathematical Software", 2009, vol. 36, n^o 3

[87] Y. Robert, F. Vivien. *Algorithmic Issues in Grid Computing*, in "Algorithms and Theory of Computation Handbook", Chapman and Hall/CRC Press, 2009

[88] G. Zheng, X. Ni, L. V. Kale. *A scalable double in-memory checkpoint and restart scheme towards exascale*, in "Dependable Systems and Networks Workshops (DSN-W)", 2012, http://dx.doi.org/10.1109/DSNW.2012.6264677

[89] D. ZHU, R. MELHEM, D. MOSSÉ. *The effects of energy management on reliability in real-time embedded systems*, in "Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)", 2004, pp. 35–40